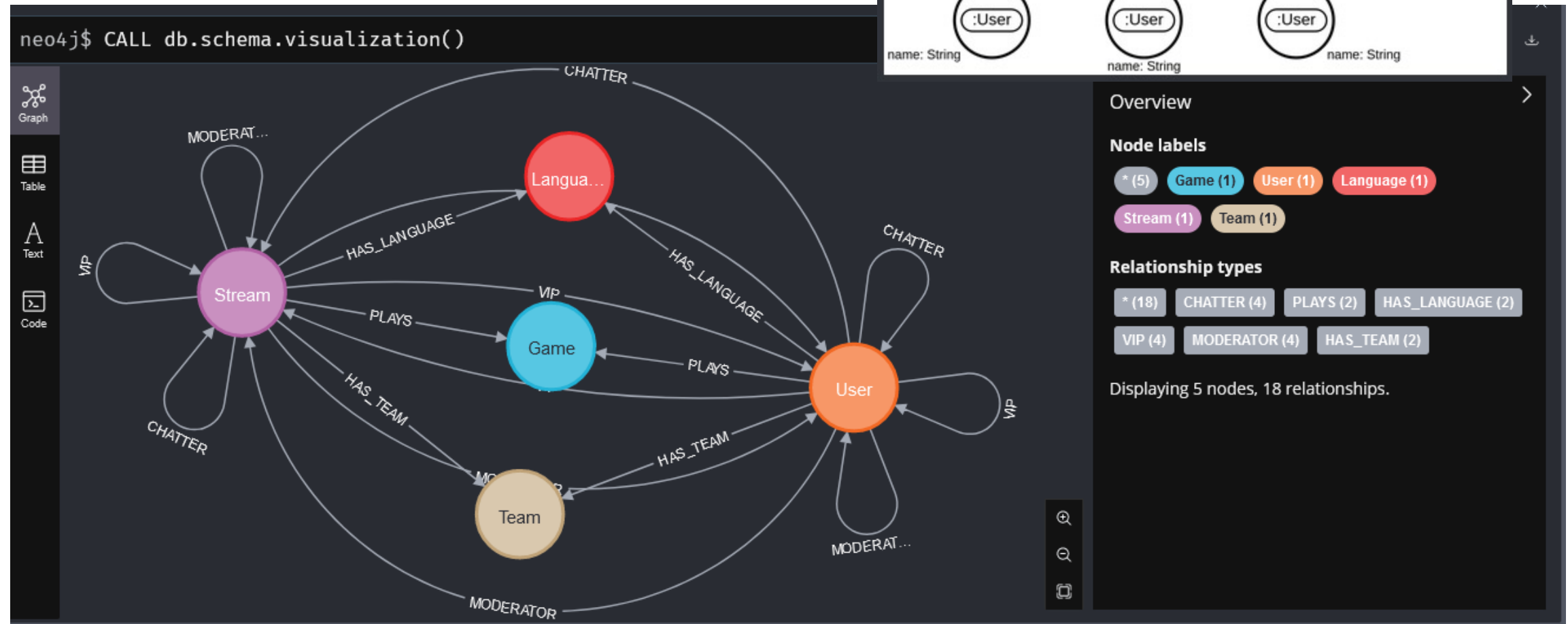


Base de datos

twitch



Parte 1: Escribir consultas

1. Escribe una consulta en Neo4j para obtener las 30 personas **con mayor cantidad de vistas totales en el grafo de twitch**. El resultado debe incluir: el nombre de la persona, la cantidad de vistas y el ID del nodo correspondiente. Asegúrate de descargar los resultados en formato CSV.

Primeros resultados de la lista de 30:

```
1 MATCH (s:Stream)
2 WHERE s.total_view_count IS NOT NULL
3 RETURN s.name as Nombre,
4        s.total_view_count as Num_vistas
5        ,s.id as ID_del_Nodo
6 ORDER BY s.total_view_count DESC
7 LIMIT 30;
```

	Nombre	Num_vistas	ID_del_Nodo
1	"fextralife"	1451487256	"156037856"
2	"riotgames"	1272988816	"36029255"
3	"esl_csgo"	610467739	"31239503"
4	"shroud"	470283753	"37402112"
5	"beyondthesummit"	469790640	"29578325"
6	"summit1g"	449304545	"26490481"
7			

Started streaming 30 records after 2 ms and completed after 12 ms.

```
//1) Query usado:
MATCH (s:Stream)
WHERE s.total_view_count IS NOT NULL
RETURN s.name as Nombre,
       s.total_view_count as Num_vistas
       ,s.id as ID_del_Nodo
ORDER BY s.total_view_count DESC
LIMIT 30
```

Últimos resultados de la lista:

	Nombre	Num_vistas	ID_del_Nodo
25	"smitegame"	187157548	"31500812"
26	"dreadztv"	186308314	"31089858"
27	"izakooo"	176823989	"36717908"
28	"pokimane"	174477299	"44445592"
29	"alanzoka"	166758215	"38244180"
30	"driupo"	161656718	"29829912"

Started streaming 30 records after 2 ms and completed after 12 ms.

Parte 1: Escribir consultas

2. Escribe una consulta en Neo4j para obtener las 30 personas **con mayor cantidad de aristas entrantes en el grafo de twitch** (mayor indegree). El resultado debe incluir: nombre, cantidad de aristas, ID del nodo. Asegúrate de descargar los resultados en formato CSV.

```
//2) Query usado:  
MATCH (u:User)<-[r]-()  
RETURN u.name as Nombre,  
COUNT(r) AS Cantidad_de_Aristas,  
u.id as ID_del_Nodo  
ORDER BY Cantidad_de_Aristas DESC LIMIT 30
```

Primeros resultados de la lista de 30:

```
1 MATCH (u:User)<-[r]-()  
2 RETURN u.name as Nombre,  
3 COUNT(r) AS Cantidad_de_Aristas,  
4 u.id as ID_del_Nodo  
5 ORDER BY Cantidad_de_Aristas DESC LIMIT 30
```

	Nombre	Cantidad_de_Aristas	ID_del_Nodo
1	"riotgames"	194559	"36029255"
2	"xqcow"	171950	"71092938"
3	"itsbigchase"	120736	"129004176"
4	"esl_csgo"	111493	"31239503"
5	"enardo"	109137	"78131255"
6	"csgomic_ru"	91283	"213748641"
7			

Started streaming 30 records after 6 ms and completed after 6356 ms.

últimos resultados de la lista de 30:

```
1 MATCH (u:User)<-[r]-()  
2 RETURN u.name as Nombre,  
3 COUNT(r) AS Cantidad_de_Aristas,  
4 u.id as ID_del_Nodo  
5 ORDER BY Cantidad_de_Aristas DESC LIMIT 30
```

	Nombre	Cantidad_de_Aristas	ID_del_Nodo
25	"thisisnotgeorgenotfound"	39321	"654556126"
26	"clix"	39062	"233300375"
27	"fundy"	38565	"93028922"
28	"elraenn"	38370	"165080419"
29	"roshtein"	37824	"72550899"
30	"cohhcarnage"	37679	"26610234"

Parte 1: Escribir consultas

3. Escribe una consulta en Neo4j para obtener las 30 personas **con mayor pagerank en el grafo de twitch**(mayor indegree). El resultado debe incluir: nombre, pagerank, id del nodo. Asegúrate de descargar los resultados en formato CSV.

```
CALL gds.graph.project('twitch',
  ['User', 'Stream'], ['CHATTER', 'VIP', 'MODERATOR'])
```

Primeros resultados de la lista de 30:

```
1 CALL gds.pageRank.stream('twitch')
2 YIELD nodeId, score
3 WITH nodeId, score
4 ORDER BY score DESC LIMIT 30
5 RETURN gds.util.asNode(nodeId).name as Nombre,
6 score as pagerank, gds.util.asNode(nodeId).id as ID_Del_Nodo
```

	Nombre	pagerank	ID_Del_Nodo
1	"yassuo"	27115.19300072124	"121203480"
2	"trainwreckstv"	23731.962030380615	"71190292"
3	"riotgames"	16071.769903521406	"36029255"
4	"loltyler1"	12968.519773729013	"51496027"
5	"xqcw"	12166.968110834692	"71092938"
6	"enardo"	8897.815030948148	"78131255"
7			

Started streaming 30 records after 6 ms and completed after 3659 ms.

```
CALL gds.pageRank.stream('twitch')
YIELD nodeId, score
WITH nodeId, score
ORDER BY score DESC LIMIT 30
RETURN gds.util.asNode(nodeId).name as Nombre,
score as pagerank, gds.util.asNode(nodeId).id as ID_Del_Nodo
```

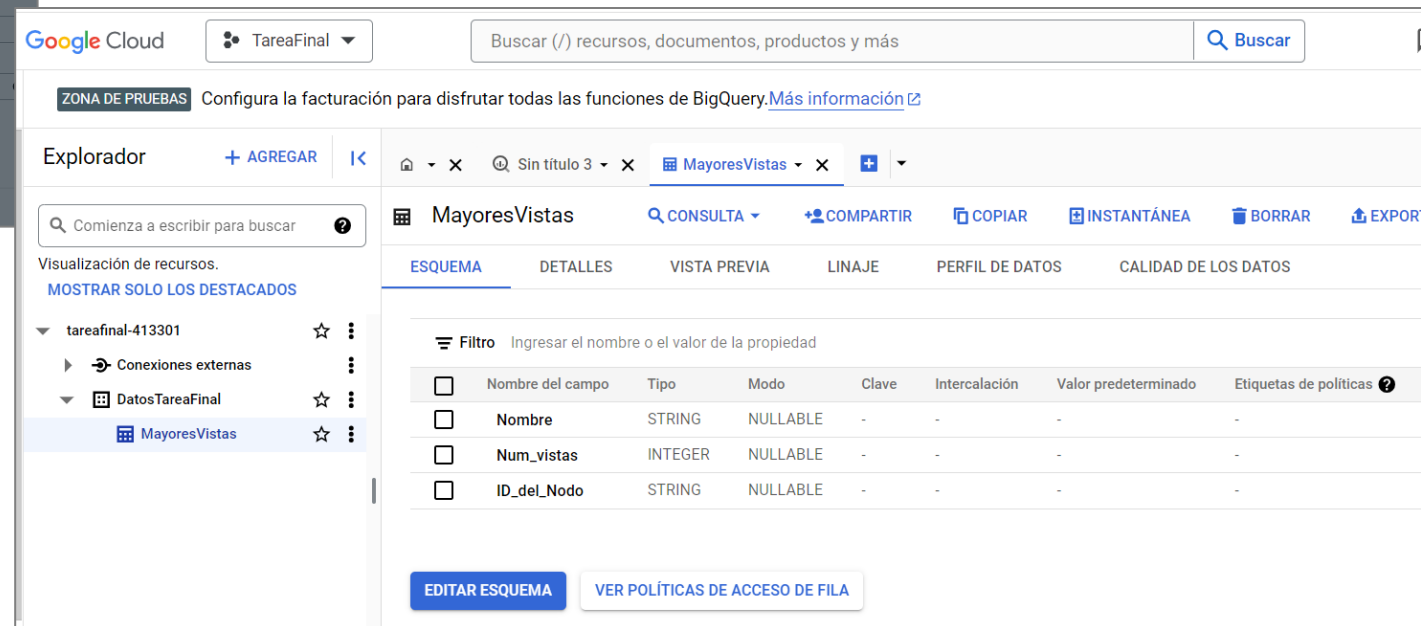
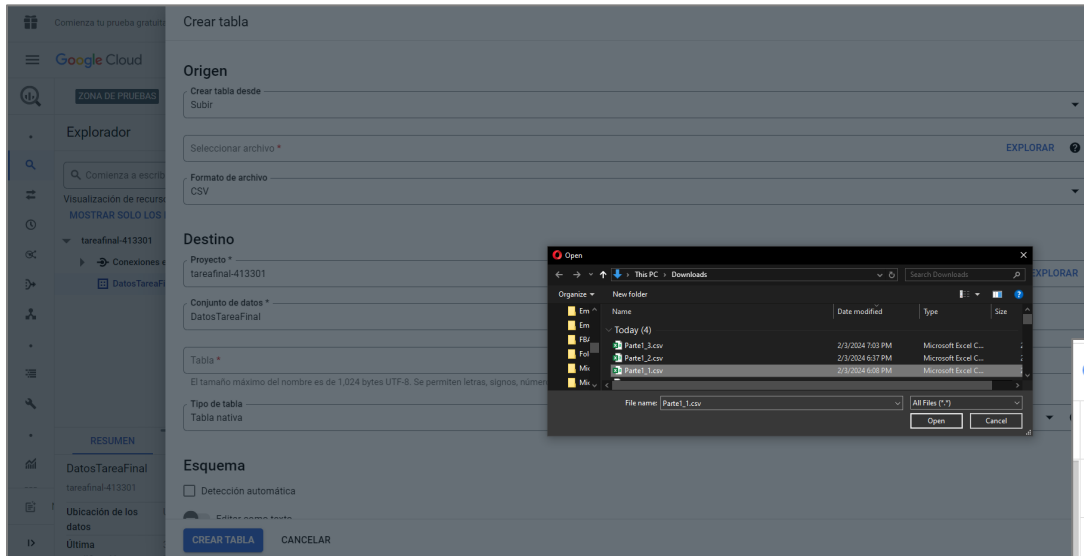
últimos resultados de la lista de 30:

```
1 CALL gds.pageRank.stream('twitch')
2 YIELD nodeId, score
3 WITH nodeId, score
4 ORDER BY score DESC LIMIT 30
5 RETURN gds.util.asNode(nodeId).name as Nombre,
6 score as pagerank, gds.util.asNode(nodeId).id as ID_Del_Nodo
```

	Nombre	pagerank	ID_Del_Nodo
25	"dota2mc_ru"	3708.071786496547	"213749122"
26	"raenee"	3671.0567949714705	"145202260"
27	"vondice"	3666.127416223407	"638511570"
28	"xhanky"	3626.9632436513452	"48870826"
29	"fps_shaka"	3599.5952578131964	"49207184"
30	"xchocobars"	3581.72902184833	"42583390"

Parte 2: Importar los archivos CSV a BigQuery (1-3)

1. Importar los archivos CSV de las tres preguntas anteriores en un entorno de BigQuery. Asegúrate de tomar una captura de pantalla que muestre dicho entorno.



Parte 2: Importar los archivos CSV a BigQuery (2-3)

Comienza tu prueba gratuita

Google Cloud

ZONA DE PRUEBAS

Explorador

Comienza a escribir

Visualización de recursos

MOSTRAR SOLO LOS DESTACADOS

tareafinal-413301

Conexiones e

DatosTareaFi

MayoresV

RESUMEN

MayoresVistas

tareafinal-413301.DatosT

Última modificación

Ubicación de los

Crear tabla

Origen

Crear tabla desde

Subir

Seleccionar archivo *

Parte1_2.csv

Formato de archivo

CSV

Destino

Proyecto *

tareafinal-413301

Conjunto de datos *

DatosTareaFinal

Tabla *

MayorIndegree

El tamaño máximo del nombre es de 1,024 bytes UTF-8. Se permiten letras, signos, núme

Tipo de tabla

Tabla nativa

Esquema

☒ Detección automática

CREAR TABLA

CANCELAR

Explorador

+ AGREGAR

<

Comienza a escribir para buscar

Visualización de recursos.

MOSTRAR SOLO LOS DESTACADOS

tareafinal-413301

Conexiones externas

DatosTareaFinal

MayorIndegree

MayoresVistas

Sin título 3

MayorIndegree

CONSULTA

COMPARTIR

ESQUEMA

DETALLES

VISTA PREVIA

LINAJE

Filtro

Ingresa el nombre o el valor de la propiedad

<input type="checkbox"/>	Nombre del campo	Tipo	Modo	Clave
<input type="checkbox"/>	Nombre	STRING	NULLABLE	-
<input type="checkbox"/>	Cantidad_de_Aristas	INTEGER	NULLABLE	-
<input type="checkbox"/>	ID_del_Nodo	STRING	NULLABLE	-

Parte 2: Importar los archivos CSV a BigQuery (3-3)

Comienza tu prueba gratuita

Google Cloud

ZONA DE PRUEBAS

Explorador

Comienza a escribir

Visualización de recursos

MOSTRAR SOLO LOS DESTACADOS

tareafinal-413301

Conexiones externas

DatosTareaFinal

MayorIndegree

MayoresVistas

Crear tabla

Subir

Seleccionar archivo *

Parte1_3.csv

Formato de archivo

CSV

Destino

Proyecto *

tareafinal-413301

Conjunto de datos *

DatosTareaFinal

Tabla *

MayorPagerank

El tamaño máximo del nombre es de 1,024 bytes UTF-8

Tipo de tabla

Tabla nativa

Esquema

☒ Detección automática

El esquema se generará automáticamente

CREAR TABLA

CANCELAR

Google Cloud

TareaFinal

Buscar (/) recursos, documentos, productos

ZONA DE PRUEBAS

Configura la facturación para disfrutar todas las funciones de BigQuery. [Más info](#)

Explorador

+ AGREGAR

Comienza a escribir para buscar

Visualización de recursos.

MOSTRAR SOLO LOS DESTACADOS

tareafinal-413301

Conexiones externas

DatosTareaFinal

MayorIndegree

MayorPagerank

MayoresVistas

MayorPagerank

CONSULTA

ESQUEMA

DETALLES

VISTA PREVIA

Filtro

Ingresar el nombre o el valor de la propiedad

<input type="checkbox"/>	Nombre del campo	Tipo	Modo
<input type="checkbox"/>	Nombre	STRING	NULLABLE
<input type="checkbox"/>	pagerank	FLOAT	NULLABLE
<input type="checkbox"/>	ID_Del_Nodo	STRING	NULLABLE

EDITAR ESQUEMA

VER POLÍTICAS DE ACCESO DE FIL

Parte 2: Importar los archivos CSV a BigQuery

2. Escribe una consulta en Bigquery que retorne los nombres de los usuarios que se encuentren en las tres tablas.

```
SELECT t1.Nombre
FROM `tareafinal-413301.DatosTareaFinal.MayoresVistas` AS t1
JOIN `tareafinal-413301.DatosTareaFinal.MayorIndegree` AS t2 ON t1.ID_del_Nodo = t2.ID_del_Nodo
JOIN `tareafinal-413301.DatosTareaFinal.MayorPagerank` AS t3 ON t2.ID_del_Nodo = t3.ID_del_Nodo
```

ZONA DE PRUEBAS Configura la facturación para disfrutar todas las funciones de BigQuery. [Más información](#)

Explorador + AGREGAR <

Comienza a escribir para buscar ?

Visualización de recursos. [MOSTRAR SOLO LOS DESTACADOS](#)

- tareafinal-413301
 - Conexiones externas
 - DatosTareaFinal
 - MayorIndegree
 - MayorPagerank
 - MayoresVistas

Sin título 3

```
1 SELECT t1.Nombre
2 FROM `tareafinal-413301.DatosTareaFinal.MayoresVistas` AS t1
3 JOIN `tareafinal-413301.DatosTareaFinal.MayorIndegree` AS t2 ON t1.ID_del_Nodo = t2.ID_del_Nodo
4 JOIN `tareafinal-413301.DatosTareaFinal.MayorPagerank` AS t3 ON t2.ID_del_Nodo = t3.ID_del_Nodo
5
```

Resultados de la consulta

GUARDAR LOS RESULTADOS

INFORMACIÓN DEL TRABAJO RESULTADOS GRÁFICO VISTA PREVIA JSON DETALLES DE LA EJECUCIÓN GRÁF

Fila	Nombre
1	"riotgames"
2	"xqcow"
3	"esl_csgo"
4	"cblol"
5	"lck_korea"

Usuarios resultantes:

	Nombre
1	"riotgames"
2	"xqcow"
3	"esl_csgo"
4	"cblol"
5	"lck_korea"

Parte 3: Responder las preguntas

1. ¿Cuántos usuarios retornó tu consulta?

R: La consulta retornó 5 usuarios, los cuales se pueden ver en la diapositiva anterior.

2. ¿Cuál es tu perspectiva sobre la relación entre el Pagerank y la cantidad de aristas entrantes en un grafo?

R: El Pagerank permite entregar más información de tipo semántica para poder analizar un problema de la vida real modelado en un grafo. En este sentido, no solo nos basta tomar en cuenta la cantidad de aristas entrantes, sino en la importancia de las aristas entrantes. De esta manera, no necesariamente existe una relación directamente proporcional entre importancia de un nodo vs cantidad de aristas entrantes en el caso de Pagerank. Podríamos decir que dicho algoritmo es más propio de minería de datos en el que se ofrece interpretabilidad a un grafo, versus una métrica como lo es la cantidad de aristas entrantes, la cual es útil, pero no suficiente para poder analizar un problema real y con cierta complejidad.

Parte 3: Responder las preguntas

3. ¿Cuál es tu opinión acerca de la relación entre la cantidad total de vistas y el PageRank en un grafo? Para responder de manera precisa, puedes utilizar las consultas proporcionadas en el documento "Guía para la tarea Final.docx" ahí puedes observar los datos de un nodo en particular.

R: Observando las queries ya ejecutadas respecto a cantidad total de vistas y Pagerank, vemos que efectivamente hay varios nodos que se repiten en dichos rankings, pero al mismo tiempo hay otros nodos que no coinciden en ranking de Pagerank vs ranking total de vistas. Dado la construcción del algoritmo y del grafo de Twitch, podemos afirmar empíricamente que efectivamente puede existir una alta influencia en la cantidad total de vistas de un sitio para aumentar el Pagerank, que recordemos estima el grado de importancia de un sitio. Sin embargo, también existe la posibilidad de que se haya caído en los casos de que el sitio no fue accedido por clicks, por lo que la cantidad de aristas entrantes es baja y por ende el Pagerank puede ser bajo o hasta nulo, pero con alta cantidad de vistas. El *damping factor* podría ser el factor que explica lo anterior. En este caso, quizás se podría concluir que el PageRank podría no ser tan alto para sitios bien específicos que los usuarios acceden no por medio de clicks, sino que directamente, más asociado a un nicho específico. En este caso, se podría incluso realizar clustering y quizás ahí podríamos determinar otra relación considerable, en el que aquellos sitios con alta cantidad de vistas y PageRank bajo, dadas ciertas condiciones, podrían formar clusters. Es importante entonces analizar el problema bajo varias perspectivas, complementando PageRank con métricas y clustering, para luego llegar a conclusiones más efectivas sobre los datos.