



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
BASES DE DATOS NOSQL



VER LA VERSION QUE
SE MANDO POR
CORREO, YA QUE ESTA
NO DISPONDRA DE LOS
DRIVERS.

Fase 2 - Casa Hightower

Proyecto: Los Siete Reinos

Versión: 1.2.0

Elaborado por:

Rosmeli Quintero
Jean Carlos Gomes Pinto

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Historial de Revisiones

Versión	Fecha	Autor	Descripción
1.2.0	02/03/2016	Rosmeli Quintero	Completar el árbol genealógico en MongoDB
1.2.0	04/03/2016	Jean Carlos Gomes	Creación de los ETL personas, eventos, familia, pertenece
1.2.0	04/03/2016	Rosmeli Quintero	Creación de los ETL casado_con, hijo_de, padre_de, mato_a
1.2.0	07/03/2016	Rosmeli Quintero	Consultas 1, 2, 5, 6, 9
1.2.0	07/03/2016	Jean Carlos Gomes	Consultas 3, 4, 6, 7, 8

Tabla de contenido

Modelo de Dato:.....	3
Instalación para correr este proyecto:.....	6
NOTA: ETL_SECUENCIA	6
Consultas:	7
Consulta 1: Cantidad de miembros por cada Familia.....	7
Consulta 2: Cantidad de hombres con dos o más hijos varones por Familia	8
Consulta 3:Cuál es el miembro de cada Familia que ha participado en más eventos	9
Consulta 4:Cuál es la familia con más facciones aliadas.	10
Consulta 5:Cuál es el miembro (por familia) que ha matado la mayor cantidad de enemigos.	11
Consulta 6: Cuántas bodas entre miembros de distintas familias ha habido	12
Consulta 7: Cantidad de miembros por familia que han quedado huérfanos de padre.	13
Consulta 8: Extraída del enunciado del proyecto:.....	14
“El señor Baelish necesita determinar qué tan probable es que un miembro de una familia A mate (o mande a asesinar) a otro B”	14
Consulta 9: Muestra todas las relaciones que posee el personaje Leyton con los otros personajes.	15

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Modelo de Dato:

Territorio: Cuenta con la información detallada del territorio. Nuestra casa por ser una casa menor, se encuentra en el territorio gobernado por una casa mayor llamada “House of Tyrell”.

Territorio	
_id	int
reino	string
region	string
continente	string
castillos	ArrayString
gorbenante	string
ciudad	id
nombreCiud	string
poblacionEstimada	int
gobernanteCiud	string
religion	string

Familia: cuenta con información de la familia y las referencias a los id de jefe de familia, heredero, aliados y enemigos.

Familia	
_id	int
familia	string
lema	string
asentamiento	string
fundador	ArrayString
lealtad_overlord	string
vasallos	id
jefe_de_familia	ArrayString
heredero_a_jefe	ArrayString
aliados	ArrayInt
enemigos	ArrayInt

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Evento: id y nombre de los eventos.

Evento	
_id	int
nombre	string

Personaje: cuenta con:

- Nombre, apellido y apodo (si lo tiene) del personaje.
- La familia a donde pertenece el personaje: si hay una boda entre los personajes, sólo uno de los dos adquiere el id de la familia de su esposo/esposa.
- Los id de los hijos que tiene el personaje junto con su género para facilitar las consultas.
- El id de la persona con la que está casado/a el personaje.
- Los nombres de los eventos en los que ha participado.
- Los id de las personas a las que ha matado.
- El estado de “vivo” o “muerto” del personaje.

Personaje	
_id	int
nombre	string
apellido	string
titulo	ArrayString
pertenece	ArrayString
apodo	ArrayString
padre_de	ArrayInt
hijo_De	ArrayInt
casado_con	ArrayInt
eventos	ArrayInt
mato_a	ArrayInt
estado	string

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Para entender más el modelo de dato, pero ahora por el lado de Neo4j, basta por ejecutar esta consulta extra que nos mostrara los nodos y como están relacionados

```
1 MATCH (a)-[r]->(b)
2 WHERE labels(a) <> [] AND labels(b) <> []
3 RETURN DISTINCT head(labels(a)) AS Nodo_A, type(r) as Relacionado, head(labels(b)) AS Nodo_B
```

\$ MATCH (a)-[r]->(b) WHERE labels(a) <> [] AND labels(b) <> [] RETURN DISTINCT head(labels(a)) AS Nodo_A, type(r) as Relacionado, head(labels(b))...			
	Nodo_A	Relacionado	Nodo_B
Rows	Personaje	CASADO_CON	Personaje
	Personaje	HIJO_DE	Personaje
	Personaje	PADRE_DE	Personaje
	Personaje	MATO_A	Personaje
	Personaje	PERTENECE	Familia
	Familia	ALIADO_CON	Familia
	Personaje	PARTICIPO_EN	Evento
Code	Returned 7 rows in 37 ms.		

Tenemos los nodos: Personaje, Familia y Evento

Tenemos las aristas: Casado_Con, Hijo_de, Padre_de, Mato_a que poseen un atributo adicional llamado “Rating” que representa el Rating de Amistad de una persona a otra.

Tenemos otras aristas como Aliado, Participo_en, pertenece.

La arista Padre_De tiene además de rating, un atributo adicional llamado género.

Otra consulta extra para entender mejor el modelo y poder ver el grafo completo seria:

```
1 MATCH (n)
2 RETURN n
```

La imagen del grafo generado, la puede ver mediante el siguiente link:

<https://drive.google.com/file/d/0B0DZHqKK5I9FS1IPMTVTcDIteXM/view?usp=sharing>

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Instalación para correr este proyecto:

Para correr este proyecto tiene que tener instalado MongoDB, Neo4j y Node.JS con sus modulos. Omitire la instalación de los dos primeros ya que se encuentra en el material en el portal web.

Para instalar Node.JS en Ubuntu, una de las formas seria conectarse en <https://nodejs.org/en/> y escoger una versión, la usada en este proyecto fue la versión v5.7.0 Stable pero cualquier versión reciente sea estable o su versión madura sirve. Después hay que descomprimir NodeJS, y guardarla en un directorio fijo, luego se debe editar las variables de entornos con la consola

```
sudo gedit ~/.bashrc
export NODE_HOME=/opt/node-v5.7.0-linux-x64
export PATH=/opt/node-v5.7.0-linux-x64/bin:$PATH
```

Para la instalación de los módulos en Node.JS, es mucho más fácil. Solo basta con abrir un terminal (Ctrl+Alt+T) y escribir estos dos comandos. De igual forma la instalación de los módulos no será necesaria, ya que en la carpeta donde se subio el proyecto se encuentra los modulos ya bajados.

```
npm install require
npm install mongodb
npm install assert
```

NOTA: ETL_SECUENCIA

Dado que Node.JS es asíncrono y que no logramos manejar bien los callback, para hacer que se ejecute de forma secuencial, en las tres máquinas que se probó, y se corrió este proyecto, se presentó el problema solo en una máquina Linux-64bit, el cual se llamaba primero a las función de crear las aristas sin antes haber creado los vértices primero, ocasionando que no todos los datos se suban de forma correcta. Otra forma aparte de ver la consola es ver si los números coinciden en neo4j. Evento(5), PARTICIPO_EN(15).

Para Solucionar esto, basta con abrir en el navegador e ir localhost:7474, borrar todos los datos en neo4j con el comando: “MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r” y volver a ejecutar el proyecto con el comando “node proyecto.js”

De igual forma, Este error solo se presentó en una máquina. Si se ejecuta “MATCH (n) RETURN n”. Tiene que devolver la siguiente imagen haciendo excepción en los colores

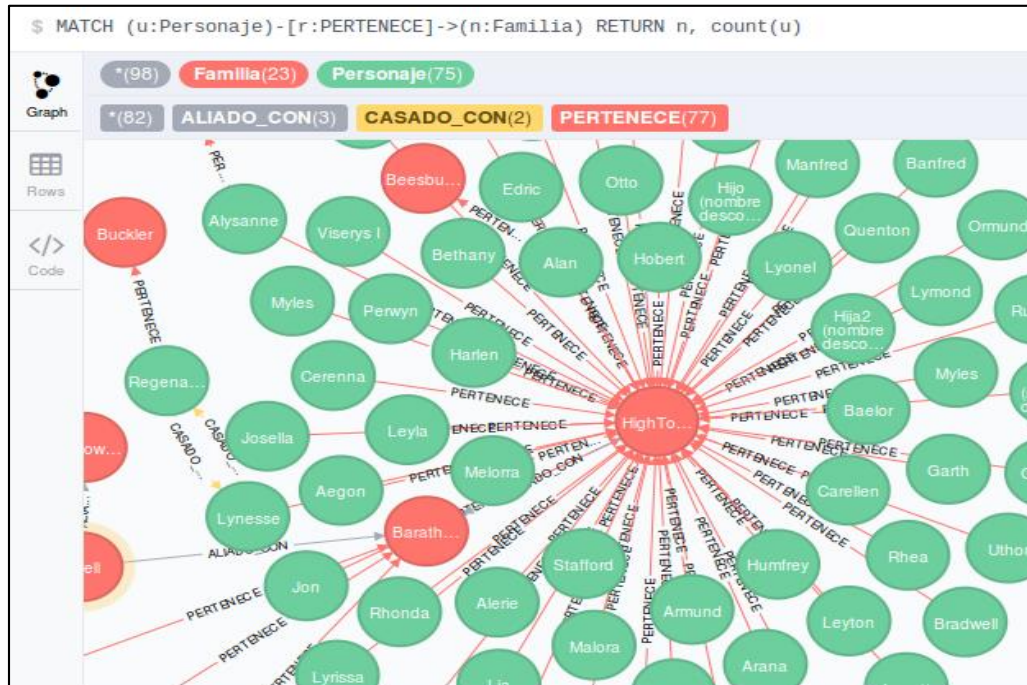


Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Consultas:

Consulta 1: Cantidad de miembros por cada Familia.

```
1 MATCH (u:Personaje)-[r:PERTENECE]->(n:Familia)
2 RETURN n, count(u)
```



\$ MATCH (u:Personaje)-[r:PERTENECE]->(n:Familia) RETURN n, count(u)

nombreFamilia	count(u)
Florent	1
Gardener	4
Lydden	1
Cupps	2
Caswell	7

Returned 23 rows in 21 ms.

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Consulta 2: Cantidad de hombres con dos o más hijos varones por Familia

```

1 MATCH (u:Personaje)-[r:PADRE_DE]->(m:Personaje)
2 WITH count(m) as Canthijos, u, r.genero as genero
3 WHERE Canthijos > 1
4 AND genero="Hombre"
5 RETURN u, Canthijos

```

\$ MATCH (u:Personaje)-[r:PADRE_DE]->(m:Personaje) WITH count(m) as Canthijos, u, r.genero as genero WHE...		Download	Pin
Graph	u	Canthijos	
Rows	<div> <div>estado</div> <div>vivo</div> </div> <div> <div>apellido</div> <div>Hightower</div> </div> <div> <div>id</div> <div>5000</div> </div> <div> <div>nombre</div> <div>Leyton</div> </div>	4	
Code	<div> <div>estado</div> <div>muerto</div> </div> <div> <div>apellido</div> <div>Caswell</div> </div> <div> <div>id</div> <div>5143</div> </div> <div> <div>nombre</div> <div>Masson</div> </div>	3	
	<div> <div>estado</div> <div>vivo</div> </div> <div> <div>apellido</div> <div>Tyrell</div> </div> <div> <div>id</div> <div>5056</div> </div>	3	
Returned 10 rows in 15 ms.			

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Consulta 3:Cuál es el miembro de cada Familia que ha participado en más eventos

```

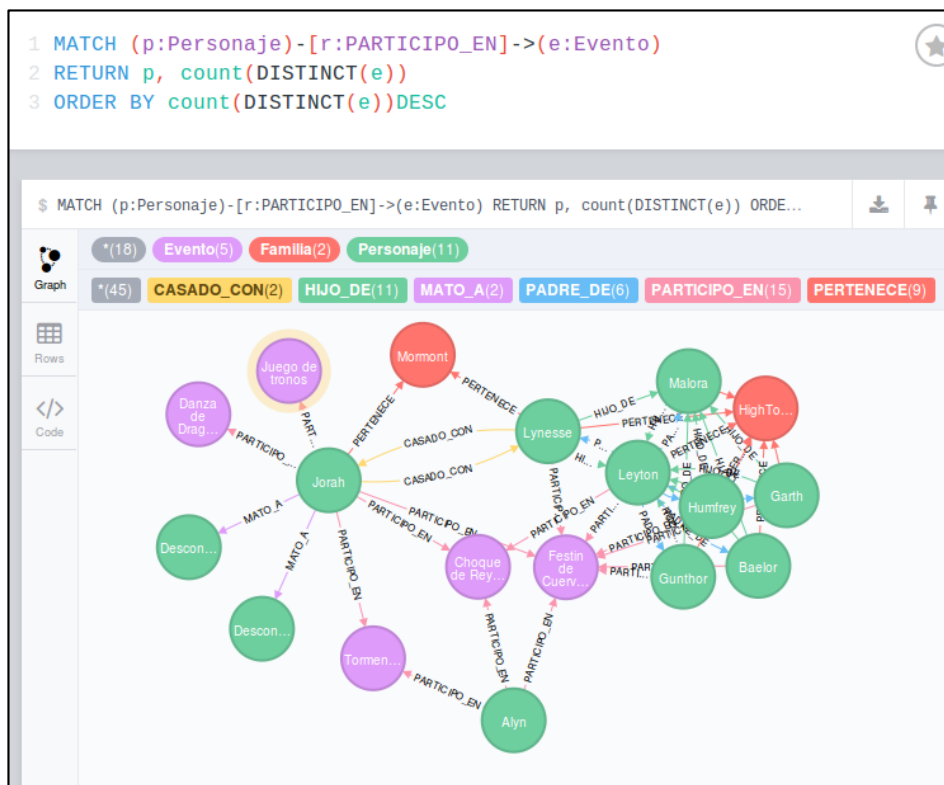
1 MATCH (p:Personaje)-[r:PARTICIO_EN]->(e:Evento)
2 RETURN p.apellido, count(DISTINCT(e))
3 ORDER BY count(DISTINCT(e))DESC

```

\$ MATCH (p:Personaje)-[r:PARTICIO_EN]->(e:Evento) RETURN p.apellido, count(DISTINCT(e)) ORDER BY cou...

	p.apellido	count(DISTINCT(e))
Rows	Mormon	5
	Ambrose	3
Code	Hightower	2

Ahora bien, Si queremos ver el grafo, si cambiamos la consulta



Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Consulta 4:Cuál es la familia con más facciones aliadas.

```

1 MATCH (f:Familia)-[a:ALIADO_CON]->(n:Familia)
2 RETURN f, count(n)
3 ORDER BY count(n) DESC
4 LIMIT 1

```

\$ MATCH (f:Familia)-[a:ALIADO_CON]->(n:Familia) RETURN f, count(n) ORDER BY count(n) DESC LIMIT 1		
Graph	f	count(n)
Rows	nombreFamilia Tyrell	4
Code	Id 2010	

Si borramos **LIMIT 1**, se podrá ver por familia la cantidad de fracciones aliadas.



Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Consulta 5:Cuál es el miembro (por familia) que ha matado la mayor cantidad de enemigos.

```
1 MATCH (p:Personaje)-[m:MATO_A]->(u:Personaje)
2 RETURN count(DISTINCT(p.apellido))
```

\$	MATCH (p:Personaje)-[m:MATO_A]->(u:Personaje) RETURN count(DISTINCT(p.apellido))
	count(DISTINCT(p.apellido))
Rows	3
	
Code	

```
1 MATCH (p:Personaje)-[m:MATO_A]->(u:Personaje)
2 RETURN p.nombre, p.apellido, count(DISTINCT(u))
3 ORDER BY count(DISTINCT(u)) DESC
4 LIMIT 3
```

\$	MATCH (p:Personaje)-[m:MATO_A]->(u:Personaje) RETURN p.nombre, p.apellido, count(DISTINCT(u)) ORDER...		
	p.nombre	p.apellido	count(DISTINCT(u))
Rows	Gerold	Hightower	8
	Mace	Tyrell	4
Code	Jorah	Mormon	2
Returned 3 rows in 26 ms.			

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016



Consulta 6: Cuántas bodas entre miembros de distintas familias ha habido

```
1 MATCH (p:Personaje)-[c:CASADO_CON]->(u:Personaje)
2 RETURN count(DISTINCT(p.apellido))
```

\$ MATCH (p:Personaje)-[c:CASADO_CON]->(u:Personaje) RETURN count(DISTINCT(p.apellido))	
	count(DISTINCT(p.apellido))
Rows	23
	
Code	

Si queremos ver la lista de familias que se han casado con un Hightower, basta con ejecutar la siguiente consulta:

```
1 MATCH (p:Personaje)-[c:CASADO_CON]->(u:Personaje)
2 RETURN p.apellido, count(DISTINCT(p.apellido))
```

\$ MATCH (p:Personaje)-[c:CASADO_CON]->(u:Personaje) RETURN p.apellido, count(DISTINCT(p.apellido))		
	p.apellido	count(DISTINCT(p.apellido))
Rows	Connington	1
	Florent	1
Code	Baratheon	1
	Brax	1
	Fossoway	1
	Mormon	1
	Ambrose	1
	Reyne	1
	Caron	1
	Hightower	1
	Lydden	1
	Cordwayner	1
	Beesbury	1
	Tyrell	1
	Rowan	1
	Targaryen	1
Returned 23 rows in 43 ms		


Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Consulta 7: Cantidad de miembros por familia que han quedado huérfanos de padre.

```

1 MATCH (p:Personaje)-[h:HIJO_DE]->(u:Personaje)
2 WHERE p.estado="vivo"
3 AND u.estado="muerto"
4 return count(DISTINCT(p.nombre)) as Cantidad_Hijos_Huerfanos

```

\$ MATCH (p:Personaje)-[h:HIJO_DE]->(u:Personaje) WHERE	
	Cantidad_Hijos_Huerfanos
Rows	1
	
Code	

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

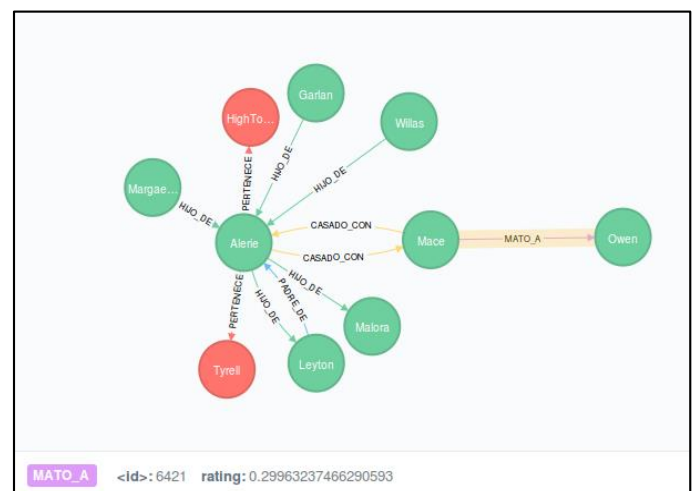
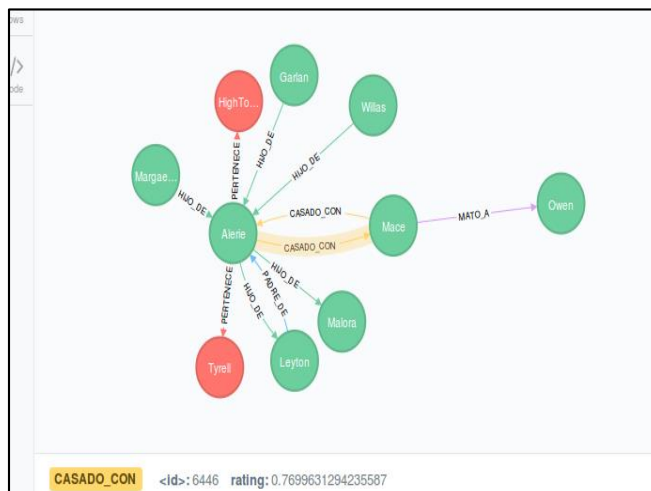
Consulta 8: Extraída del enunciado del proyecto:

“El señor Baelish necesita determinar qué tan probable es que un miembro de una familia A mate (o mande a asesinar) a otro B”.

```

1 MATCH (p:Personaje)-[c:CASADO_CON]->(u:Personaje)-[m:MATO_A]->(a:Personaje
  {nombre: "Owen"})
2 RETURN p.nombre, SUM(1-(m.rating * c.rating)) AS Probabilidad_Matar_A,
  a.nombre

```



(Aleire)-->[Casada_Con(Rating: 0.76996312...)]-->(Mace)-->[Mato_a(Rating: 0.2996323...)]-->Owen

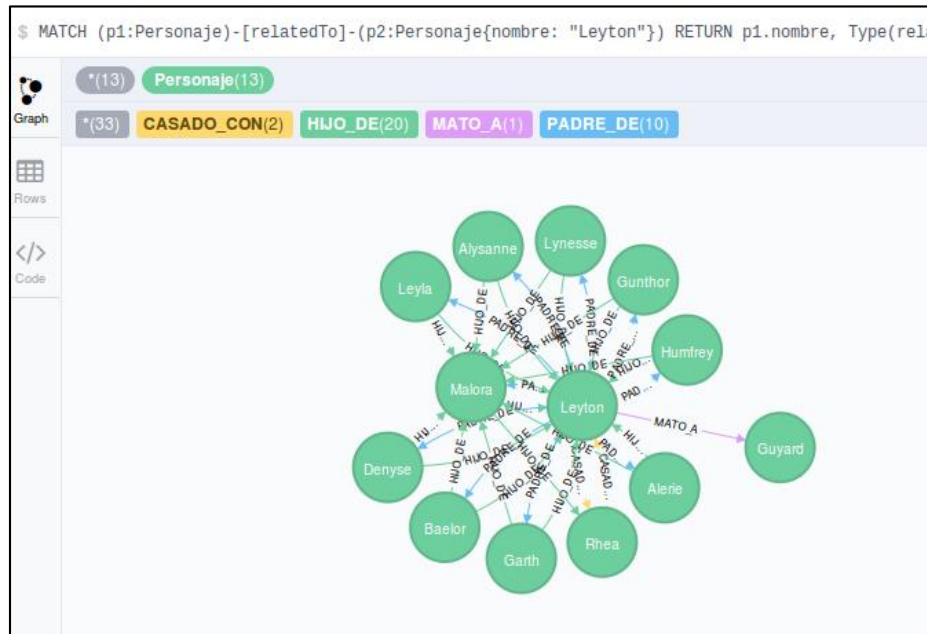
Si multiplicamos nos da: 0.23070582, Este porcentaje es muy alto, por lo tanto Aleire odia a Owen y por lo tanto la probabilidad de que lo mande a matar es alta dado que seria $1-(\text{Rating}_1 * \text{Rating}_2)$

Nota: Se ejecutó una función Math.random(), es decir que a la hora que se corrija, arrojará este resultado. Para verificar que la operación se hizo bien, solo basta con seguir el orden de las flechas y buscar la relación Casado_con y Mato_a

Casa Hightower – Neo4j usando NodeJS	Versión 1.2.0
Proyecto: Los Siete Reinos	9 de Marzo de 2016

Consulta 9: Muestra todas las relaciones que posee el personaje Leyton con los otros personajes.

```
1 MATCH (p1:Personaje)-[relatedTo]-(p2:Personaje{nombre: "Leyton"})
2 RETURN p1.nombre, Type(relatedTo), relatedTo
```



Si queremos ver los rating de Amistad de Leyton con las otras personas

\$ MATCH (p1:Personaje)-[relatedTo]-(p2:Personaje{nombre: "Leyton"}) RETURN p1.nombre...		
p1.nombre	Type(relatedTo)	relatedTo
Gunthor	HIJO_DE	rating 0.3982933384831995
Baelor	HIJO_DE	rating 0.41796120698563755
Malora	HIJO_DE	rating 0.04476303933188319
Leyla	HIJO_DE	rating 0.11384296300821006
Denyse	HIJO_DE	rating 0.1996127979364246
Alysanne	HIJO_DE	rating 0.5664692788850516

Returned 23 rows in 65 ms.