

## A. OBJETIVO

Este apunte tiene como objeto mostrar una implementación de grafos en el Motor de Base de Datos orientada a Grafos Neo4J.

La primera parte del apunte detalla los pasos necesarios para realizar el Download del motor de BD e instalarlo.

La segunda parte consta de un ejemplo de creación de nodos y relaciones de una base de datos que contendrá el Organigrama de una empresa.

Cada nodo representa a un empleado y contendrá los siguientes atributos:

### **Empleado**

**ID**

**Nombre**

**Apellido**

**Cargo**

Cada relación entre dos empleados representa a la supervisión de un empleado con otro, a continuación mostramos sus atributos.

### **Relación SupervisaA.**

**Fecha**

*(Fecha a partir de la cuál el jefe supervisa al empleado)*

En la parte final se muestran describen diferentes consultas de información de la Base de Datos generada, relacionando concepto de Grafos con ejemplos reales, mostrando las pantallas de la consola con los resultados obtenidos.

Ejemplificamos algunas consuntas con su relación con conceptos de Grafos.

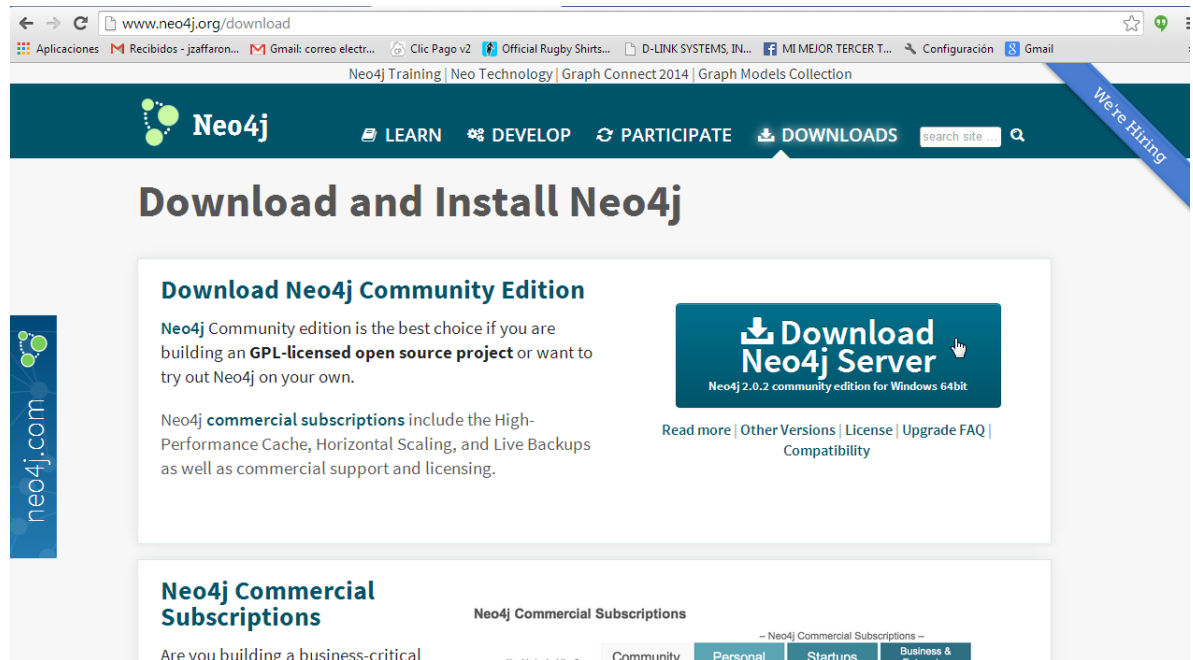
**RIGHT (empleadoA)** = Conjunto de empleados a los que el empleadoA Supervisa en forma directa.

**IDEAL DERECHO(empleadoA)** = Conjunto de empleados a los que el empleadoA supervisa en forma directa o indirecta. Por ejemplo: En el caso del presidente el resultado contendrá a todos los empleados de la empresa.

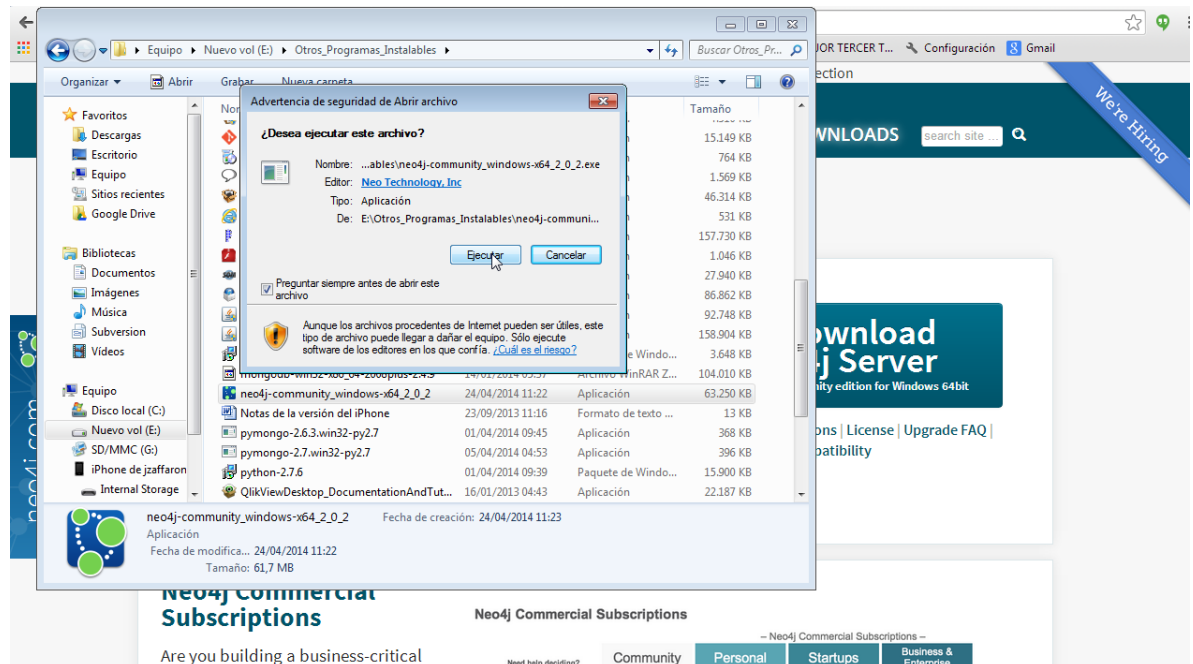
## B. INSTALACIÓN DE NEO4J

### 1. Realizar el download

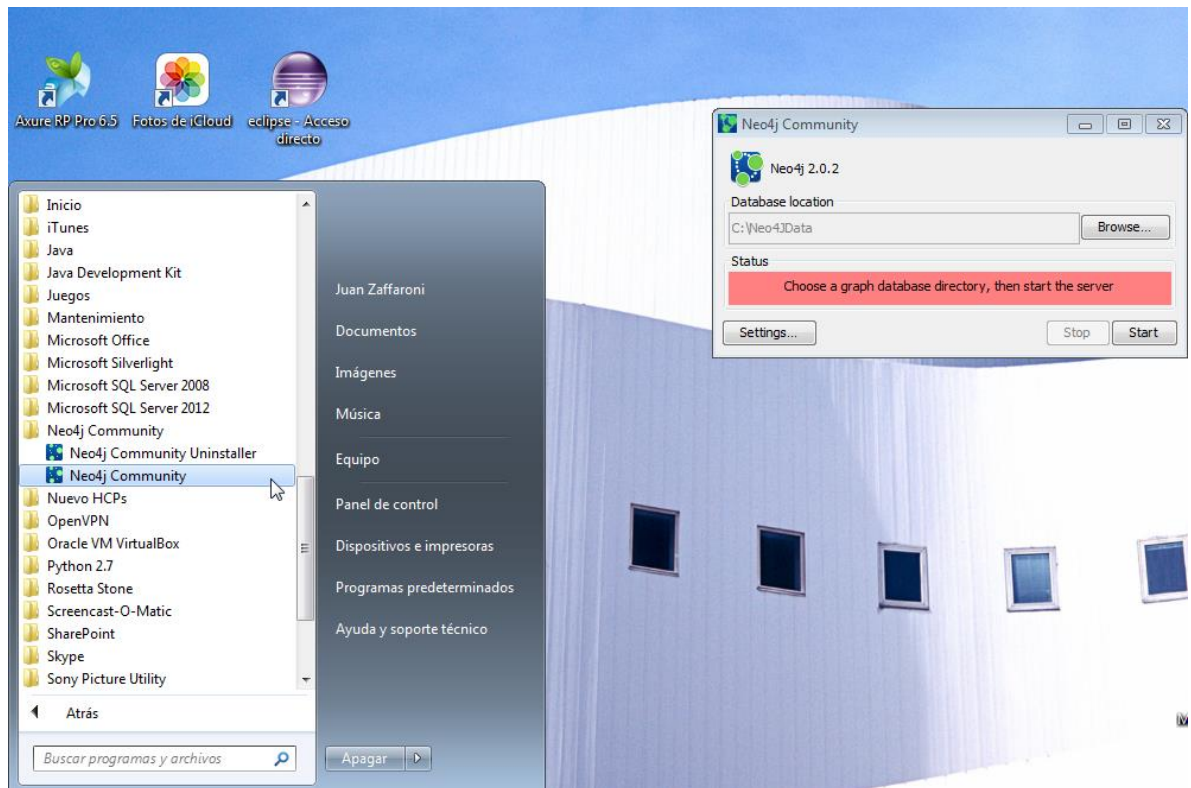
[www.neo4j.org/download](http://www.neo4j.org/download)



### 2. Instalar el Motor de BD

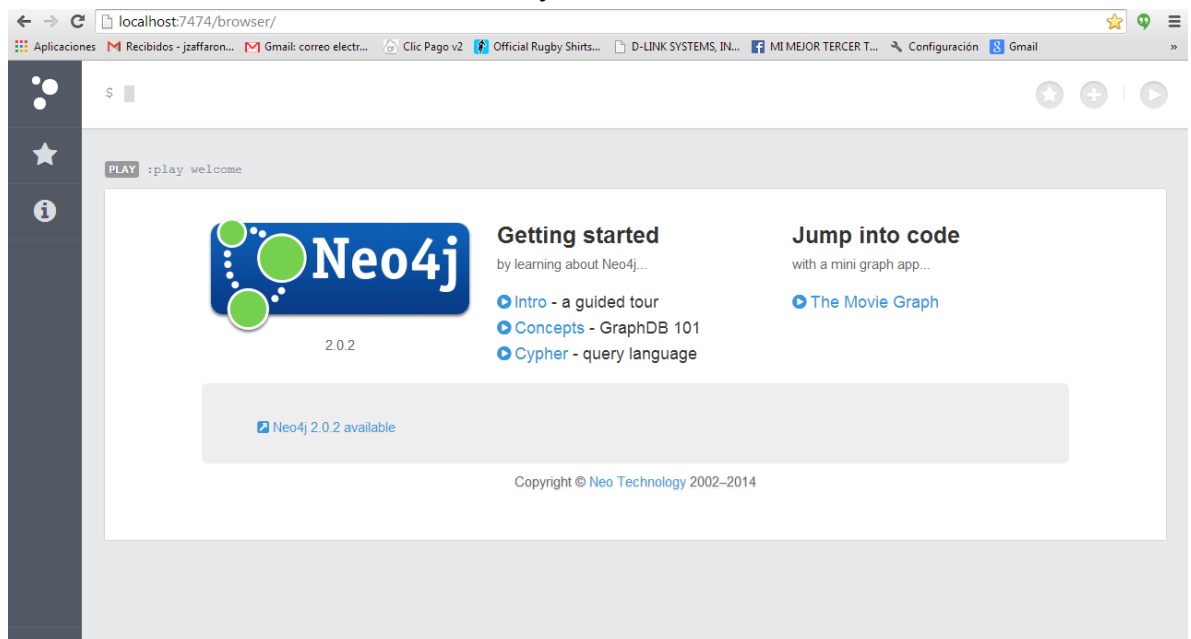


### 3. Iniciar Sesión del Motor de Neo4j

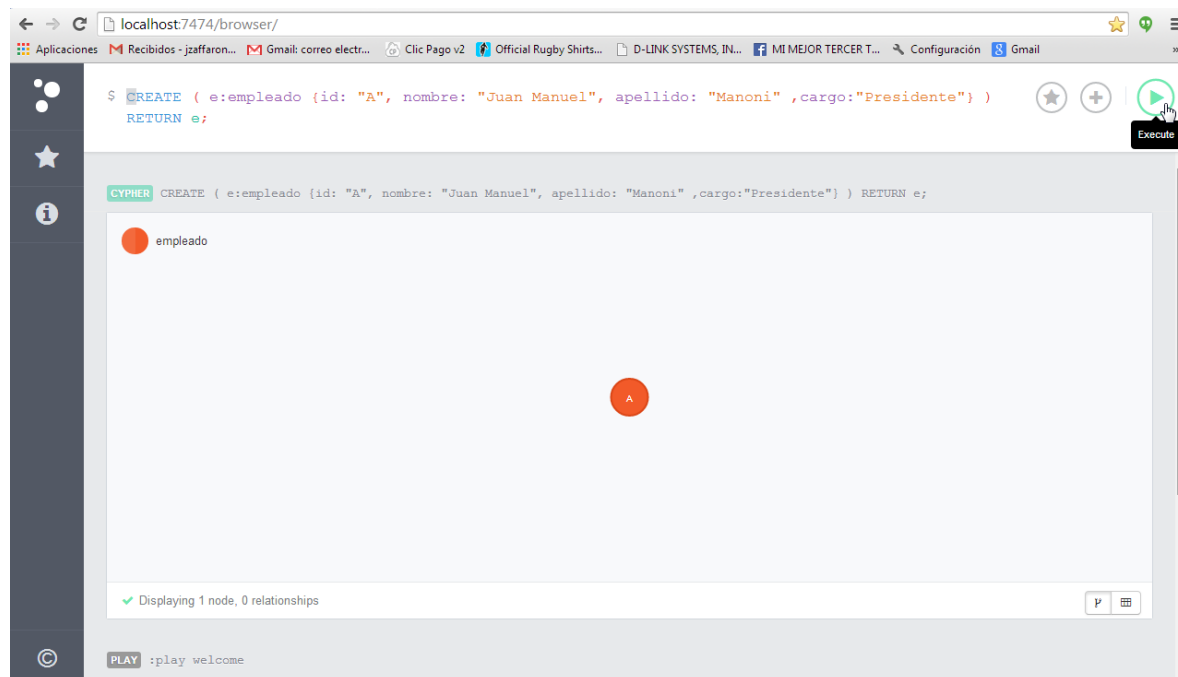


### 4. Definir locación para la BD en el ejemplo creé una carpeta C:\Neo4jdata antes de iniciar la Sesión del Motor y se la elegí como BD Location.

### 5. Acceder a la Consola del del Motor de Neo4j



## 6. Comenzar a crear la BD



## C. CREACIÓN DE LA BASE DE DATOS CON SUS NODOS Y RELACIONES

### 1. Creación de Nodos.

```
CREATE ( e:empleado {id: "A", nombre: "Juan Manuel", apellido: "Manoni",cargo:"Presidente"}) RETURN e;
```

```
CREATE ( e:empleado {id: "B", nombre: "Martín", apellido: "Zavasi",cargo:"Gerente Comercial" }) RETURN e;
```

```
CREATE ( e:empleado {id: "C", nombre: "Marina", apellido: "Basigalupi",cargo:"Gerente Financiero" }) RETURN e;
```

```
CREATE ( e:empleado {id: "D", nombre: "Soledad", apellido: "Zasiva",cargo:"Gerente de Operaciones" }) RETURN e;
```

```
CREATE ( e:empleado {id: "F", nombre: "Juan Cruz", apellido: "Biroccio",cargo:"Jefe de Planta" }) RETURN e;
```

```
CREATE ( e:empleado {id: "G", nombre: "Miguel", apellido: "Suparo",cargo:"Operador de Planta" }) RETURN e;
```

```
CREATE ( e:empleado {id: "E", nombre: "Mario", apellido: "Burgos",cargo:"Empleado Administrativo" }) RETURN e;
```

```
CREATE ( e:empleado {id: "H", nombre: "Miguel", apellido: "Prince",cargo:"Operador de Planta" }) RETURN e;
```

```
CREATE ( e:empleado {id: "I", nombre: "Gabriel", apellido: "García Marquez",cargo:"Responsable de Cuenta" }) RETURN e;
```

### 2. Creación de Relaciones.

```
MATCH (e: empleado), (e2:empleado) WHERE e.id = 'A' AND e2.id = 'B'  
CREATE (e) - [r: SupervisaA { fecha: "20100401" }] -> (e2)  
RETURN e, e2, r;
```

```
MATCH (e: empleado), (e2:empleado) WHERE e.id = 'A' AND e2.id = 'C'  
CREATE (e) - [r: SupervisaA { fecha: "20110610" }] -> (e2)  
RETURN e, e2, r
```

```
MATCH (e: empleado), (e2:empleado) WHERE e.id = 'A' AND e2.id = 'D'  
CREATE (e) - [r: SupervisaA { fecha: "20090315" }] -> (e2)  
RETURN e, e2, r
```

```
MATCH (e: empleado), (e2:empleado) WHERE e.id = 'D' AND e2.id = 'F'  
CREATE (e) - [r: SupervisaA { fecha: "20110109" } ] -> (e2)  
RETURN e, e2, r
```

```
MATCH (e: empleado), (e2:empleado) WHERE e.id = 'F' AND e2.id = 'G'  
CREATE (e) - [r: SupervisaA { fecha: "20120415" } ] -> (e2)  
RETURN e, e2, r
```

```
MATCH (e: empleado), (e2:empleado) WHERE e.id = 'C' AND e2.id = 'E'  
CREATE (e) - [r: SupervisaA { fecha: "20121010" } ] -> (e2)  
RETURN e, e2, r
```

```
MATCH (e: empleado), (e2:empleado) WHERE e.id = 'F' AND e2.id = 'H'  
CREATE (e) - [r: SupervisaA { fecha: "20121101" } ] -> (e2)  
RETURN e, e2, r
```

```
MATCH (e: empleado), (e2:empleado) WHERE e.id = 'D' AND e2.id = 'I'  
CREATE (e) - [r: SupervisaA { fecha: "20131001" } ] -> (e2)  
RETURN e, e2, r
```

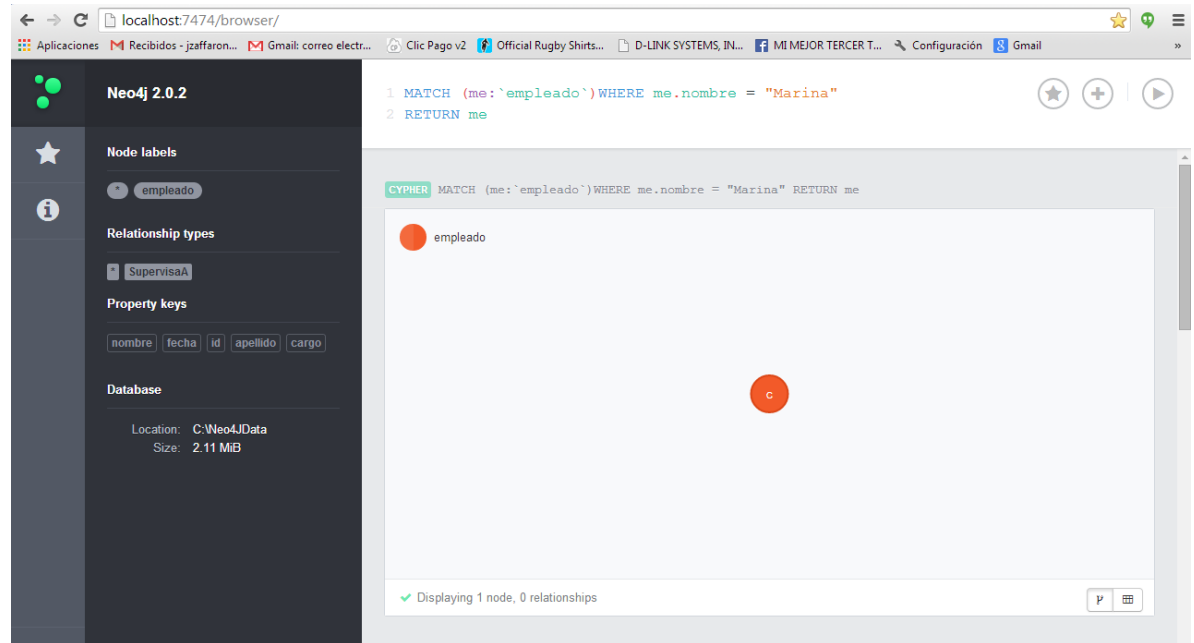
## D. CONSULTAS REALIZADAS SOBRE LA BASE DE DATOS

### 1. Consultas básicas de empleados

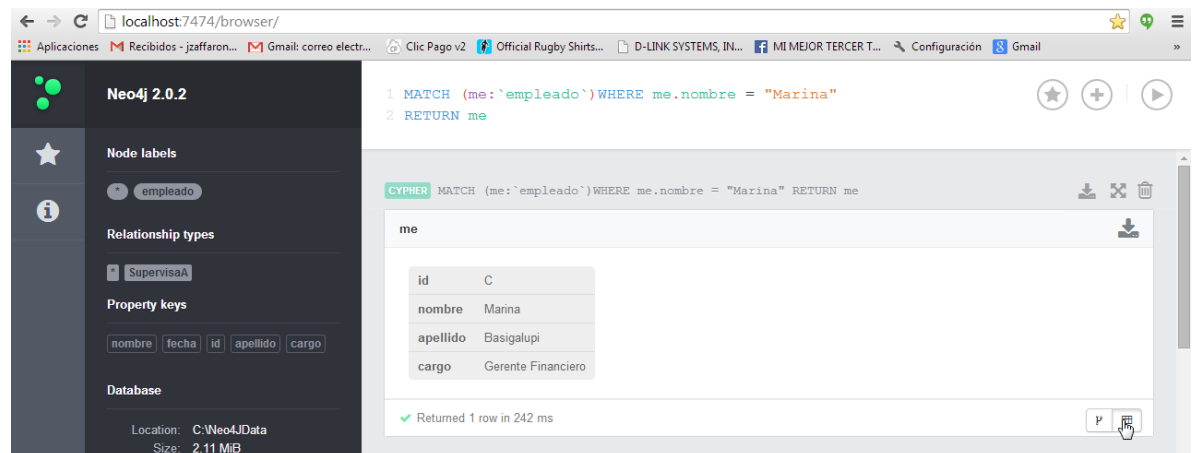
Consulta de Empleado cuyo nombre es “Marina”, retorna el nodo completo  
(en modo gráfico o ficha de valores)

**MATCH (me:`empleado`)  
WHERE me.nombre = "Marina"  
RETURN me**

Gráficamente obtendremos lo siguiente:



En modo Descripción de Datos obtendríamos lo siguiente



Consulta de Empleado cuyo cargo es “Presidente”, retorna los atributos del nodo cargo, apellido y nombre. *(en modo tabla de datos)*

**MATCH (me:`empleado`)  
WHERE me.cargo = "Presidente"  
RETURN me.cargo, me.apellido, me.nombre**

1 MATCH (me:`empleado`)  
2 WHERE me.cargo = "Presidente"  
3 RETURN me.cargo, me.apellido, me.nombre

CYPHER MATCH (me:`empleado`)  
WHERE me.cargo = "Presidente" RETURN me.cargo, me.apellido, me.nombre

me.cargo	me.apellido	me.nombre	
Presidente	Manoni	Juan Manuel	

Returned 1 row in 240 ms

Consulta de Empleados cuyo cargo es “Operador de Planta”, retorna los atributos “cargo, apellido, nombre y id” de todos los nodos. *(en modo tabla de datos)*

**MATCH (me:`empleado`)  
WHERE me.cargo = "Operador de Planta"  
RETURN me.cargo, me.apellido, me.nombre,me.id**

1 MATCH (me:`empleado`)  
2 WHERE me.cargo = "Operador de Planta"  
3 RETURN me.cargo, me.apellido, me.nombre,me.id

CYPHER MATCH (me:`empleado`)  
WHERE me.cargo = "Operador de Planta" RETURN me.cargo, me.apellido,

me.cargo	me.apellido	me.nombre	me.id	
Operador de Planta	Suparo	Miguel	G	
Operador de Planta	Prince	Miguel	H	

Returned 2 rows in 184 ms



## 2. Consultar los empleados y sus relaciones

Consulta todos los empleados definiendo cada jefe (empleado que supervisa a otro empleado) y su subordinados directos (empleados que son supervisados por otro empleado) ordenados por fecha, retorna apellido y nombre del jefe, apellido y nombre del empleado y la fecha de la relación.

**START jefe=node(\*)**

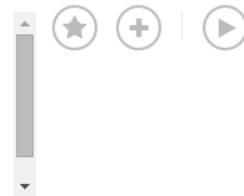
**MATCH (jefe)-[s:SupervisaA]->(empleado)**

**WHERE empleado.nombre IS NOT NULL**

**RETURN DISTINCT jefe.apellido, jefe.nombre, empleado.apellido, empleado.nombre, s.fecha**

**ORDER BY s.fecha**

```
1 START jefe=node(*)
2 MATCH (jefe)-[s:SupervisaA]->(empleado)
3 WHERE empleado.nombre IS NOT NULL
4 RETURN DISTINCT jefe.apellido, jefe.nombre, empleado.apellido,
empleado.nombre, s.fecha
5 ORDER BY s.fecha
```



**CYPHER** START jefe=node(\*) MATCH (jefe)-[s:SupervisaA]->(empleado) WHERE empleado.nombre IS NOT NULL

jefe.apellido	jefe.nombre	empleado.apellido	empleado.nombre	s.fecha	
Manoni	Juan Manuel	Zasiva	Soledad	20090315	
Manoni	Juan Manuel	Zavasi	Martín	20100401	
Zasiva	Soledad	Biroccio	Juan Cruz	20110109	
Manoni	Juan Manuel	Basigalupi	Marina	20110610	
Biroccio	Juan Cruz	Suparo	Miguel	20120415	
Basigalupi	Marina	Burgos	Mario	20121010	
Biroccio	Juan Cruz	Prince	Miguel	20121101	
Zasiva	Soledad	García Marquez	Gabriel	20131001	

✓ Returned 8 rows in 213 ms



### 3. Consultar el RIGHT de un EMPLEADO

Recordamos el RIGHT de un elemento es el conjunto de elementos que reciben un arco que parte del elemento indicado.

En nuestro ejemplo:  $\text{RIGHT}(A) = \{B, C, D\}$

A=Presidente

B=Gerente Comercial

C=Gerente Financiero

D=Gerente de Operaciones

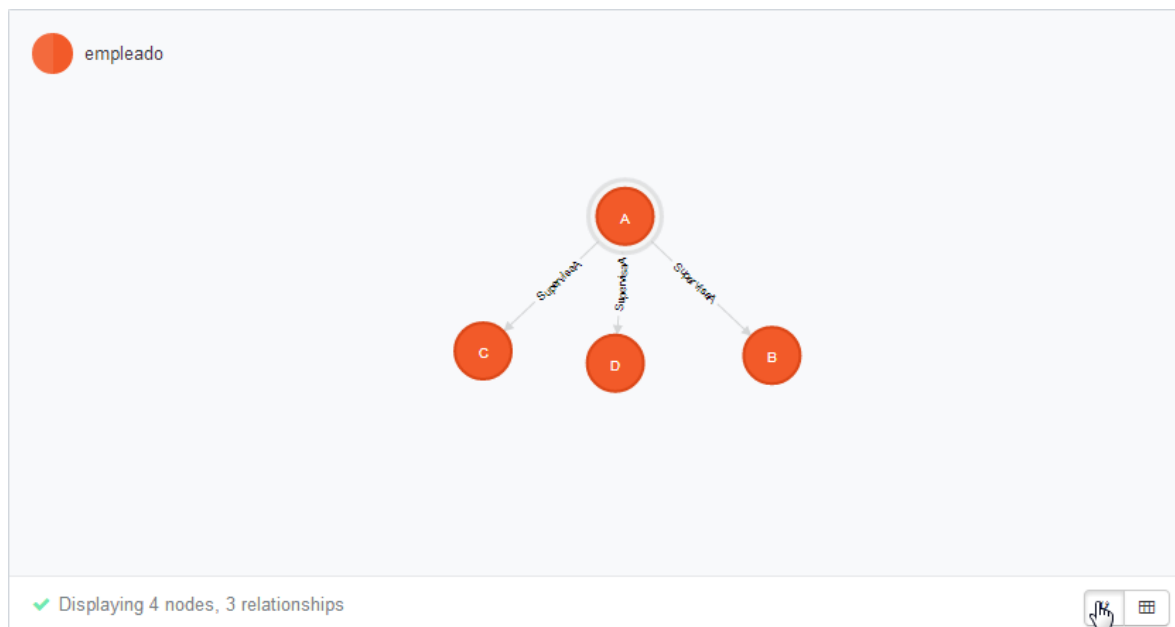
Por lo tanto, en nuestro caso el RIGHT de un empleado es el conjunto de empleados a los cuales SUPERVISA en forma directa.

**MATCH (jefe)-[s:SupervisaA]->(empleado)**

**WHERE jefe.id='A' AND empleado.nombre IS NOT NULL**

**RETURN DISTINCT jefe,empleado**

Gráficamente obtendremos lo siguiente:



En modo Descripción de Datos obtendríamos lo siguiente

```
1 MATCH (jefe)-[:SupervisaA]->(empleado)
2 WHERE jefe.id='A' AND empleado.nombre IS NOT NULL
3 RETURN DISTINCT jefe,empleado
4
```



CYPHER MATCH (jefe)-[:SupervisaA]->(empleado) WHERE jefe.id='A' AND empleado.nombre IS NOT NULL

id	nombre	apellido	cargo
A	Juan Manuel	Manoni	Presidente
B	Martin	Zavasi	Gerente Comercial
C	Marina	Basigalupi	Gerente Financiero

Returned 3 rows in 209 ms

#### 4. Consultar el IDEAL DERECHO de un empleado

Repasemos el concepto de IDEAL DERECHO de un elemento, es el conjunto de elementos con los cuales el elemento original tiene paso (relación directa o indirecta).

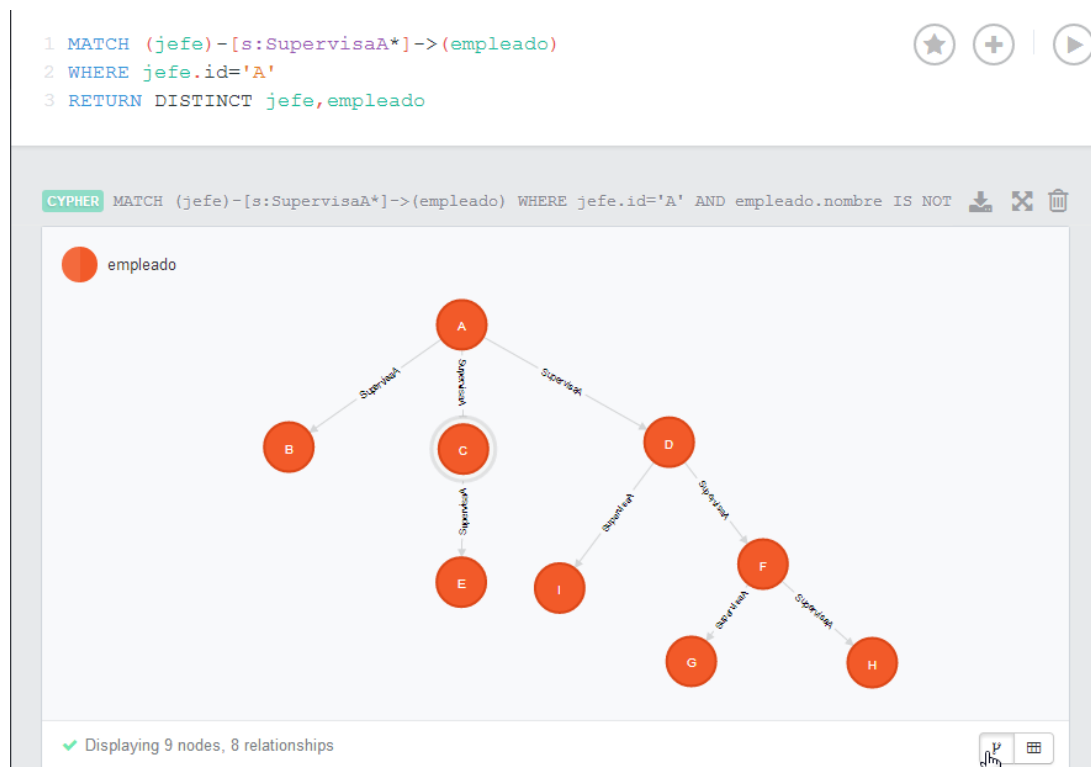
En nuestro ejemplo el IDEAL DERECHO de un empleado estará compuesto por todos los empleados que son supervisados por él de manera directa, o indirecta.

**MATCH (jefe)-[s:SupervisaA\*]->(empleado)**

**WHERE jefe.id='A'**

**RETURN DISTINCT jefe,empleado**

Gráficamente obtendremos lo siguiente:



En modo Descripción de Datos obtendríamos lo siguiente:

```
1 MATCH (jefe)-[s:SupervisaA*]->(empleado)
2 WHERE jefe.id='A'
3 RETURN DISTINCT jefe,empleado
```

CYPHER MATCH (jefe)-[s:SupervisaA\*]->(empleado) WHERE jefe.id='A' RETURN DISTINCT jefe,emple

id	A	id	B
nombre	Juan Manuel	nombre	Martin
apellido	Manoni	apellido	Zavasi
cargo	Presidente	cargo	Gerente Comercial

id	A	id	C
nombre	Juan Manuel	nombre	Marina
apellido	Manoni	apellido	Basigalupi
cargo	Presidente	cargo	Gerente Financiero

✓ Returned 8 rows in 248 ms

**5. Consultar el IDEAL DERECHO acotado a relaciones con prof. de paso de entre 1 y 2.**

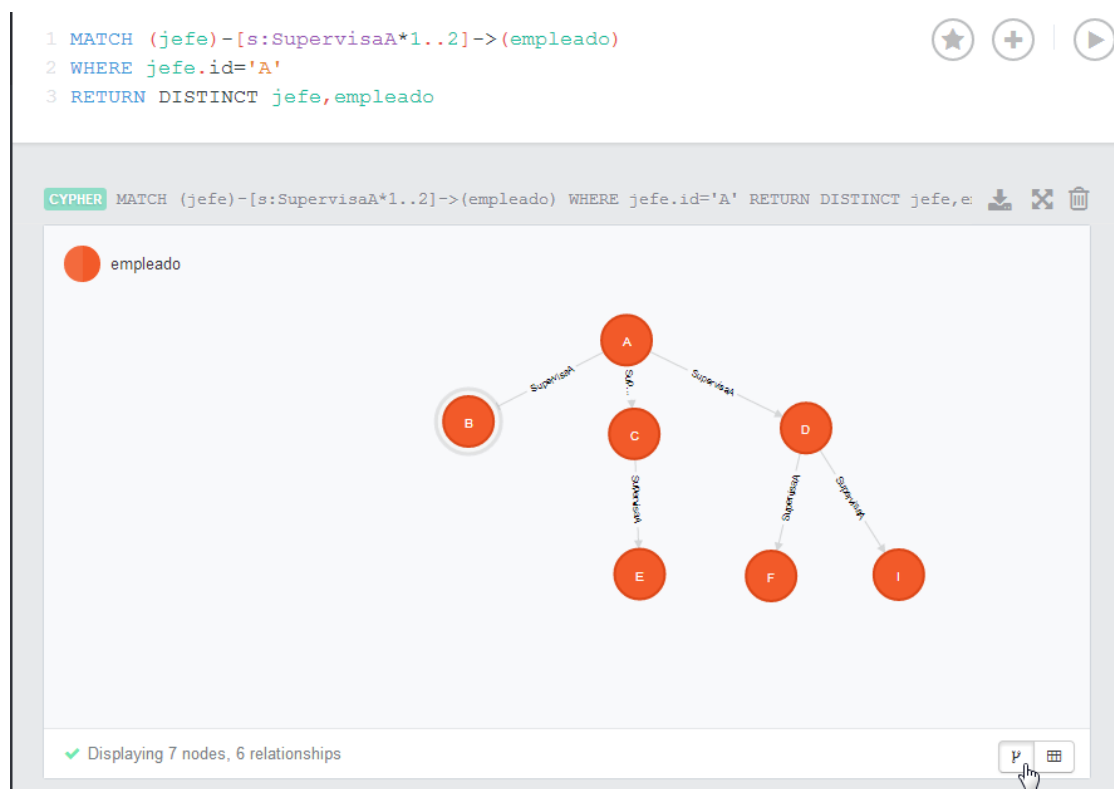
En este ejemplo obtenemos los empleados que son supervisados por el presidente directamente (longitud de paso 1) y los empleados supervisados por los gerentes (long. De paso 2).

**MATCH (jefe)-[s:SupervisaA\*1..2]->(empleado)**

**WHERE jefe.id='A'**

**RETURN DISTINCT jefe,empleado**

Gráficamente obtendremos lo siguiente:



En modo Descripción de Datos obtendríamos lo siguiente:

```

1 MATCH (jefe)-[s:SupervisaA*1..2]->(empleado)
2 WHERE jefe.id='A'
3 RETURN DISTINCT jefe,empleado

```

CYPHER

MATCH (jefe)-[s:SupervisaA\*1..2]->(empleado) WHERE jefe.id='A' RETURN DISTINCT jefe,e

<table> <tr><td>nombre</td><td>Juan Manuel</td></tr> <tr><td>apellido</td><td>Manoni</td></tr> <tr><td>cargo</td><td>Presidente</td></tr> </table>	nombre	Juan Manuel	apellido	Manoni	cargo	Presidente	<table> <tr><td>nombre</td><td>Martín</td></tr> <tr><td>apellido</td><td>Zavasi</td></tr> <tr><td>cargo</td><td>Gerente Comercial</td></tr> </table>	nombre	Martín	apellido	Zavasi	cargo	Gerente Comercial				
nombre	Juan Manuel																
apellido	Manoni																
cargo	Presidente																
nombre	Martín																
apellido	Zavasi																
cargo	Gerente Comercial																
<table> <tr><td>id</td><td>A</td></tr> <tr><td>nombre</td><td>Juan Manuel</td></tr> <tr><td>apellido</td><td>Manoni</td></tr> <tr><td>cargo</td><td>Presidente</td></tr> </table>	id	A	nombre	Juan Manuel	apellido	Manoni	cargo	Presidente	<table> <tr><td>id</td><td>C</td></tr> <tr><td>nombre</td><td>Marina</td></tr> <tr><td>apellido</td><td>Basigalupi</td></tr> <tr><td>cargo</td><td>Gerente Financiero</td></tr> </table>	id	C	nombre	Marina	apellido	Basigalupi	cargo	Gerente Financiero
id	A																
nombre	Juan Manuel																
apellido	Manoni																
cargo	Presidente																
id	C																
nombre	Marina																
apellido	Basigalupi																
cargo	Gerente Financiero																
<table> <tr><td>id</td><td>A</td></tr> <tr><td>nombre</td><td>Juan Manuel</td></tr> </table>	id	A	nombre	Juan Manuel	<table> <tr><td>id</td><td>E</td></tr> <tr><td>nombre</td><td>Mario</td></tr> </table>	id	E	nombre	Mario								
id	A																
nombre	Juan Manuel																
id	E																
nombre	Mario																

✓ Returned 6 rows in 295 ms

## E. ESTRUCTURA INTERNA DE LA BASE DE DATOS NEO4J

### Nodo

En uso
1ra. Relación
1ra. Propiedad

### Relación

En uso	
1er. Nodo	2do. Nodo
Tipo de relación	
1er. N. Rel previa	1er. N. Rel sigte
2do. N. Rel previa	2do. N. Rel sigte
1ra. Propiedad	

Cada nodo ocupa 9 bytes y la cada relación 33 bytes.

Las propiedades de los nodos y las relaciones están en una estructura de doble linkeo con un formato interno de KEY / VALUE para los pares tipopropiedad:valor.

Cada propiedad ocupa 41 bytes.

