



**universidad
de león**



Escuela de Ingenierías Industrial, Informática y Aeroespacial

GRADO EN INGENIERÍA INFORMÁTICA

Guía de instalación - Comparador Metal

Guía

Autor: Rubén Canillas Mañanes

Sistemas de Información de Gestión y BI

Tabla de contenido

Introducción	3
Primeros pasos	3
Preparación del sistema	5
Importar datos.....	5

Introducción

La práctica consiste en la creación de un sistema de recomendación, en este caso sobre discos de música metal, utilizando para ello un sistema de gestión de base de datos basado en grafos. El sistema será una aplicación web dinámica para la que se utilizarán las siguientes tecnologías:

SO Linux

Servidor Apache

PHP

Neo4j

HTML5

Primeros pasos

Para adquirir los conocimientos necesarios respecto a neo4j se han realizado los siguientes cursos de formación online:

Overview of Neo4j 4.x

Querying with Cypher in Neo4j 4.x

Creating Nodes and Relationships in Neo4j 4.x

Using Indexes and Query Best Practices in Neo4j 4.x

Importing Data with Neo4j 4.x

Como dataset inicial se usará el siguiente, al cual, se le realizará modificaciones para adaptarlo al diseño del programa.

<https://www.kaggle.com/kerneler/starter-metal-albums-artwork-eb523918-f/data>

El sistema operativo sobre el que se realizara el despliegue es xubuntu-20.04.3-desktop-amd64.iso.

Tras instalarlo se ejecutarán los siguientes comandos:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install apache2
```

```
sudo apt-get install php7.4-cli
```

```
sudo apt-get install php7.4-mbstring
```

```
sudo apt-get install php7.4-bcmath
```

```
sudo apt-get install php7.4
```

```
php -r "copy('https://getcomposer.org/installer','composer-setup.php');"
```

```
php -r "if (hash_file('sha384', 'composer-setup.php') ===  
'906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8cba5b96ac8b64115a  
c402c8cd292b8a03482574915d1a8') { echo 'Installer verified'; } else { echo  
'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
```

```
php composer-setup.php
```

```
php -r "unlink('composer-setup.php');"
```

```
sudo mv composer.phar /usr/local/bin/composer
```

Descargar neo4j de la siguiente url: <https://neo4j.com/business-subscription/?edition=enterprise&release=4.3.7&flavour=unix>

```
cd /home/<user>/Descargas
```

```
tar -xf <filename>
```

```
sudo add-apt-repository ppa:linuxuprising/java
```

```
sudo apt-get update
```

```
sudo apt-get install openjdk-11-jdk
```

```
cd /var/www
```

```
sudo nano composer.json // su contenido debe ser {}
```

```
sudo composer require stefanak-michal/bolt ^2.5.2
```

```
sudo composer require laudis/neo4j-php-client
```

Preparación del sistema

Tras instalar todos los componentes necesarios se debe colocar el contenido de la carpeta www en el directorio /var/www, y comprobar los permisos de ejecución.

Para iniciar el sistema es necesario seguir los siguientes comandos:

En el directorio base de neo4j, en la carpeta bin, ejecutar './neo4j start'

```
sudo service apache2 start
```

Tras seguir estos pasos la aplicación web será accesible vía ip desde el navegador.

Importar datos

Con los archivos .csv en la carpeta import del directorio de neo4j y desde el navegador en la dirección localhost:7474, se deben ejecutar las siguientes sentencias

```
LOAD CSV WITH HEADERS from 'file:///users.csv' as row FIELDTERMINATOR ';'
MERGE (:User {email: row.email, name: row.nombre, surnames:
row.apellidos, date: row.fecha, pass: row.pass, points: 0})
```

```
LOAD CSV WITH HEADERS from 'file:///paises.csv' as row FIELDTERMINATOR
';' MERGE (:Country {name: row.country})
```

```
LOAD CSV WITH HEADERS from 'file:///genres.csv' as row FIELDTERMINATOR
';' MERGE (:Genre {name: row.name})
```

LOAD CSV WITH HEADERS from 'file:///artists.csv' as row FIELDTERMINATOR ';' MERGE (:Band {name: row.name, status: row.status})

LOAD CSV WITH HEADERS from 'file:///albums.csv' as row FIELDTERMINATOR '|' MERGE (:Album {name: row.album, image: row.album_url, points: 0, number: 0})

LOAD CSV WITH HEADERS from 'file:///hoja.csv' as row FIELDTERMINATOR '|' MATCH (b:Band { name: row.name }), (a:Album { name: row.album }), (g:Genre { name: row.main_genre }) MERGE (b)-[:FROM_ARTIST]-(a)-[:GENRE_IS]->(g)

LOAD CSV WITH HEADERS from 'file:///from.csv' as row FIELDTERMINATOR '|' MATCH (c:Country { name: row.country }), (b:Band { name: row.name }) MERGE (c)-[:FROM_COUNTRY]-(b)

LOAD CSV WITH HEADERS from 'file:///alternative.csv' as row FIELDTERMINATOR '|' MATCH (a:Album { name: row.album }), (g:Genre { name: row.alt_genre }) MERGE (a)-[:SECOND_GENRE_IS]->(g)

LOAD CSV WITH HEADERS from 'file:///votes.csv' as row FIELDTERMINATOR ';' MATCH (a:Album), (u:User) WHERE a.name = row.album AND u.email = row.email SET u += {points: u.points+1 } SET a += {points: toInteger(row.points), number: a.number+1 } CREATE (a)-[:VOTED]-(u)