

# Data Import

El objetivo de esta práctica es trabajar con los 3 tipos de BBDD que hemos usado durante el curso: neo4j, mysql y mongodb. Los archivos están en mi repositorio de código, en el Tema 2, generate data.

**Requisitos previos:** Los estudiantes deben tener Docker instalado y las tres imágenes descargadas.

## Datos neo4j

Los datos de neo4j serán 3 archivos csv, a continuación damos una descripción de los mismos. Solo os hará falta 2 labels. El archivo **works\_at** es para definir relaciones.

### persons.csv

- **id:** Identificador único para cada persona. Tipo entero.
- **name:** Nombre de la persona. Tipo String.
- **age:** Edad de la persona. Tipo entero.

### empresas.csv:

- **id:** Identificador único para cada empresa. Tipo entero.
- **name:** Nombre de la empresa. Tipo String.
- **sector:** Sector en el que opera la empresa. Tipo cadena.

### works\_at.csv:

- **rol:** Rol o cargo de la persona en la empresa. Tipo String.
- **location\_id:** Clave foránea referida al identificador único de una localización asociada al proyecto. Tipo entero.

## Datos MySQL

Los datos de neo4j serán 4 archivos json, a continuación damos una descripción de los mismos.

### locations.json

- **id:** Identificador único para cada ubicación. Tipo entero.
- **name:** Nombre de la localización. Tipo cadena.
- **city:** Ciudad donde se encuentra la ubicación. Tipo string.

### skills.json:

- **id:** Identificador único para cada habilidad. Tipo entero.

- **name:** Nombre de la habilidad. Tipo string.

#### has\_skill.json

- **person\_id:**Clave foránea que hace referencia al identificador único de una persona.Tipo Integer.
- **skill\_id:** Clave foránea referida al identificador único de una habilidad.Tipo entero.
- **proficiency:** Nivel de competencia de la persona en la habilidad especificada. Tipo string.

#### pokemon.json

- **pokemon\_id:**identificador único de pokemon.Tipo Integer.
- **description:** descripción del pokemon. Tipo string.
- **pokeGame:** nombre del juego pokemon. . Tipo string.

## Datos de mongo

Los datos de mongodb serán 3 archivos csv y un json, a continuación damos una descripción de los mismos.

#### projects.csv

- **project\_id:** Identificador único para cada proyecto. Tipo entero.
- **name:** Nombre del proyecto. Tipo string.
- **description:** Descripción del proyecto. Tipo cadena.
- **location\_id:** Clave foránea referida al identificador único de una localización asociada al proyecto. Tipo entero.
- **company\_id:** Clave foránea referida al identificador único de una empresa asociada al proyecto.Tipo entero.

#### teams.csv

- **team\_id:**Identificador único para cada equipo. Tipo entero.
- **name:** Nombre del equipo.Tipo String.
- **description:** Descripción del equipo: Descripción del equipo. Tipo cadena.
- **project\_id:** Clave foránea que hace referencia al identificador único de un proyecto. Tipo entero.

#### works\_in\_team.csv

- **person\_id:** Clave foránea que hace referencia al identificador único de una persona. Tipo entero.
- **team\_id:** Clave foránea que hace referencia al identificador único de un equipo. Tipo entero.
- **rol:** Rol de la persona dentro del equipo. Tipo String.

### **favourite\_pokemon.json**

- **person\_id**: Clave foránea que hace referencia al identificador único de una persona. Tipo entero.
- **pokemon\_id**: Clave foránea que hace referencia al identificador único de un pokemon. Tipo entero.
- **dateCaptured**: fecha en la que el pokemon se capturó. Tipo String.

## Programa

Vuestro programa empezará cargando los datos a las diferentes BBDD a través de python, una vez terminado saldrá el siguiente menú.

---

Menú:

1. Personas y sus roles en una empresa concreta.
2. Personas con el mismo rol en diferentes empresas.
3. Empresas comunes entre dos personas.
4. Personas y sus funciones en un equipo específico.
5. Muestra todos los equipos con el número de personas que los componen.
6. Muestra los equipos con el número total de proyectos a los que están asociados.
7. Obtener todas las skills en las que una persona tiene al menos un nivel específico de proficiency
8. Encontrar todas las personas que tienen skill en al menos una skill en común con otra persona (es decir, encontrar personas con skills similares).
9. Encontrar el proyecto que tenga más personas en el equipo cuyos pokemons favoritos son de diferente tipo, mostrar todo los tipos distintos. Para ver si un pokemon es de un tipo o de otro tendrás que hacer una petición al API.
10. Dado una ubicación, obtén la lista de equipos que están ubicados allí junto con información de las personas que trabajan en ese equipo y los proyectos asociados.

### Ejemplo de resultado:

1. Equipo (nombre equipo) que trabaja en empresa (nombre empresa).  
Proyectos:
  - a. Proyecto [nombre proyecto] con skill [nombre skill]
  - b. ....Personas:
  - a. Persona [nombre persona] con rol [nombre rol]
  - b. ....
2. Equipo (nombre equipo) que trabaja en empresa (nombre empresa).  
Proyectos:
  - a. Proyecto [nombre proyecto] con skill [nombre skill] y [proficiency]
  - b. ....

Personas:

- c. Persona [nombre persona] con rol [nombre rol]
- d. ....

El usuario escribe por consola un número del 1 al 9 y se le mostrarán los datos que están en las BBDD, es posible que dependiendo del elemento del menú se le pida por consola la información necesaria: el nombre de la empresa, el proficiency, etc.

Cuando mostréis información no uséis únicamente el id, utilizad los nombres (name) de las entidades.

**Recomendación:** Hacer las queries primero directamente en la BBDD, luego hacéis el código.

## Entrega

La entrega será un enlace; un archivo txt con la URL de vuestro repositorio en GitHub.