Datos del estudiante

**Nombre y apellidos**

**Fecha de entrega**

Uso avanzado de bases de datos NoSQL

Objetivos de la actividad

Esta actividad te va a permitir utilizar funcionalidades avanzadas de las bases de datos NoSQL, en concreto MongoDB y Neo4J. Realizarás una serie de ejercicios donde podrás monitorizar la actividad de la base de datos MongoDB y diseñarás un grafo que permita visualizar un conjunto de datos extraídos en Neo4J.

Logotipo

Descripción generada automáticamente

Figura 1. Fuente: Mongo DB.

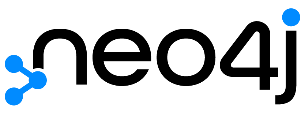


Figura 2. Fuente: Neo4j.

Pautas de elaboración

Para esta actividad, vas a procesar diferentes ficheros sobre MongoDB mientras monitorizas la actividad de la base de datos. Finalmente, con un conjunto de datos extraídos de MongoDB, tendrás que crear un grafo en Neo4J que muestre determinadas relaciones entre los datos. Hay que realizar la actividad desarrollando los apartados a continuación.

**Prepara el entorno de trabajo**

* **Opción 1**: descarga una imagen Docker de MongoDB versión 4.2 (o superior) y levanta un contenedor con dicha versión.
* **Opción 2**: descarga e instala MongoDB en su versión 4.2 (o superior) en tu ordenador local.
* **Instala las *Tools* de MongoDB:** en las siguientes URL podrás descargar las *tools* de MongoDB necesarias para este laboratorio:
* <https://docs.mongodb.com/database-tools/installation/installation/>
* <https://www.mongodb.com/try/download/database-tools>

Este paso seguro ya lo has hecho, elige en grupo la mejor opción para trabajar en este laboratorio. No es necesario documentar en el trabajo final.

* **Inicia la monitorización en MongoDB**, ejecuta las siguientes instrucciones:
  1. En una consola diferente, visualiza las estadísticas con la instrucción mongostat --discover
  2. En otra consola distinta, visualiza el *top* de las operaciones contra la base de datos, utiliza la instrucción mongotop 5 donde 5 es el tiempo en segundos que debe refrescar la visualización.
  3. **Tened estas dos consolas siempre activas y visibles mientras realizáis los siguientes puntos.** **Interpretad su salida** y describid **brevemente** qué muestran cuando se ejecutan dichas instrucciones (utilizad capturas para ilustrar vuestra respuesta).
* **Importar la base de datos:**
  1. Cread la base de datos *inspections* y dentro de ella la colección que corresponda a los datos del fichero *act-grupal-city\_inspections.json*.
  2. Cread la base de datos *countries* y dentro de ella dos colecciones que correspondan con los ficheros *act-grupal-countries-small.json* y act-g*rupal-countries-big.json*.

Reto: propón una única instrucción en el punto a y en el b. En el b amplía tu visión y saca provecho a las bondades de Mongoshell.

* **Exportar la base de datos:**
  1. Haced un *export* de la colección *inspections*. Llamad al fichero de salida *inspections.json*.

Nota: el ejecutable que permite hacer el *export* es *mongoexport*, la sintaxis es: mongoexport --collection=<name-collection> --db=<name-db> --out=<name-file>.json

* **Restaurar una base de datos:**

a. Para restaurar una base de datos, debéis indicar la ruta del directorio *dump* donde se almacenan los ficheros de la base de datos *(json y bson)*.

dump

* + - 1. people
      2. --people.bson
      3. --people.metadata.json

b. Descomprimid el fichero *act-grupal-people.zip* y ubicad los ficheros en la estructura de directorios indicada en el paso anterior.

Nota: recordad el uso de la instrucción: mongorestore --port xxxxx PATH\dump

* **Caso de uso**: restricción de terrazas en Madrid por covid-19. Contratan a vuestro equipo para actualizar las restricciones de ciertos locales y terrazas en Madrid por cuestiones del covid-19. Los datos para actualizar están en el fichero llamado *act-grupal-openDataLocalesMadrid.cvs* y se os pide que consolidéis dichos cambios en una base de datos MongoDB llamada *Madrid* con la colección *Terrazas.* La información del dataset a utilizar lo tenéis en la siguiente URL: Open Data Censo de locales, sus actividades y terrazas de hostelería y restauración (Terrazas)

<https://datos.madrid.es/FWProjects/egob/Catalogo/Economia/Ficheros/Estructura_DS_FicheroCLA_Terrazas.pdf>

* En el alcance del contrato se os pide:

a. Convertir el fichero CSV a formato JSON. Utilizad, por ejemplo, el servicio de <https://csvjson.com/csv2json>

b. Descargar el fichero generado en el paso anterior, en formato JSON indicando la opción Array (todos los documentos estarán dentro de un único *array*). En el fichero hay un problema de espacios en muchos campos String, intentad corregirlo en esta etapa.

c. Editar el fichero generado, y al principio del todo añadir la línea:

var datos\_insertar = […]

d. Guardar el fichero en una ruta conocida y cambiad su extensión a formato .js. ¿Sabéis qué tipo de fichero es un .js?

e. Abrir el cliente *mongo* y ejecutar las siguientes instrucciones:

load(“PATH\fichero.js”)

datos\_insertar[0]

f. Describir brevemente qué hacen las instrucciones anteriores.

g. Con el elemento datos\_insertar, ¿podríais realizar un *insert* masivo? ¿Cómo?

h. Desde la terminal, podríais proponer una alternativa que evite realizar los pasos anteriores (descripción de dos líneas máximo y no vale indicar el uso MongoCompass 😊).

* Realizad las siguientes actualizaciones sobre los datos (cuando los cambios afecten a campos \_es o \_ra, haced el *update* sobre los datos del local en período estacional):

1. Los locales del barrio Guindalera de Salamanca, por motivos de la desescalada, no podrán abrir y tendrán que permanecer cerrados. Cambiad la situación del local a *Cerrado* utilizando el *id* correspondiente. Recordad que debéis constatar los tipos de situaciones que existan y seguir con la misma codificación. A estos mismos locales, cambiad la situación de la terraza a *cerrada* siguiendo las mismas premisas anteriores.
2. A todas las terrazas que se ubiquen en la *acera,* añadid un campo llamado *inspeccionar* y estableced el siguiente valor:

Si dispone de más de 10 mesas, true.

Si dispone de menos de 10 mesas, false.

1. A las terrazas que se deban inspeccionar, añadid 2 mesas más auxiliares y 8 sillas más disponibles (tanto en período estacional como el resto del año).
2. A las terrazas que no deban ser inspeccionadas, añadid el campo *estado* con el valor:

1 si el número de sillas es menor que 10.

2 si el número de sillas está entre 10 y 20.

3 si cuenta con más de 20 sillas.

1. De lunes a jueves, ningún local podrá cerrar más allá de las 00:00:00, actualizad el horario de cierre a esta nueva hora límite.
2. De viernes a sábado, los locales que cierren a las 2:30:00 ahora tendrán que hacerlo a las 2:00:00.
3. A todos los locales que estén sobre la calle Alcalá en Madrid se les debe inspeccionar.
4. A todos los locales con terrazas abiertas, añadid un campo *revision* cuyo valor sea un documento con la siguiente información: {prox\_inspeccion: 10, puntuacion: 80, comentario: “separar las mesas”}
5. Cread una nueva colección llamada *Zona1,* con todos los locales que pertenezcan al distrito de Villaverde.
6. Cread una nueva colección llamada *Zona2,* con todos los locales que pertenezcan al distrito de Salamanca y barrio Castellana.

* En Neo4J, proponed un modelo de datos donde sea posible visualizar los locales de cada barrio y los tipos de terrazas que estos tienen. Indicad los nodos y sus relaciones, cread una única consulta que visualice el grafo total (incluidas las relaciones). Añadid tanto a nodos como a relaciones sus atributos, para ello elegid los campos que más convengan a vuestro modelo.

**De este punto se espera**:

* Un grafo genérico o diagrama que represente vuestro modelo y donde se reflejen todos los campos utilizados (nodos, relaciones y atributos).
* Una breve descripción del modelo propuesto.
* La descripción del proceso utilizado para cargar los datos en Neo4J a partir de los datos en MongoDB.
* Una consulta CQL y capturas con la visualización del grafo generado con los datos. Al menos debe haber una captura con un gran número de elementos del grafo.

Extensión y formato

Elaborad un informe con los elementos habituales (portada, índice, nombre, fecha, numeración de páginas, etc.), en el que deis respuesta a los apartados planteados anteriormente. Cada uno de los apartados se debe responder en una sección diferente. En cada apartado responded a los distintos literales que se proponen. **La respuesta debe incluir**: la respuesta misma al literal si la hay, la consulta realizada *(script)* y, cuando así se indique, una captura de pantalla con el resultado. **En el caso de que no respondáis a un literal concreto, ponedlo en la memoria y dejadlo en blanco. Todos los *scripts* de las consultas deben agruparse por sección en un único documento txt.**

Utilizad las páginas que necesitéis para redactar el informe. No se permiten anexos. Fuente utilizada Calibri 12 e interlineado 1,5. Al finalizar la actividad grupal, **todos los miembros del equipo harán la entrega del mismo informe final** a través del apartado «Envío de actividades» del aula virtual. El informe y *script* para entregar deben ir nombrados como se muestra a continuación:

* APELLIDO1\_APELLIDO2\_NOMBRE\_actividad\_grupal.pdf (en pdf)
* APELLIDO1\_APELLIDO2\_NOMBRE\_actividad\_grupal.txt

No uséis tildes, apóstrofes ni ningún otro carácter que pudiera resultar conflictivo.