Text

Description automatically generated with low confidence

Universidad Internacional de La Rioja

Facultad de Ingeniería y Tecnología

# Máster Universitario en Análisis y Visualización de Datos Masivos / Visual Analitics & Big Data

Uso Avanzado de BD NoSQL

|  |  |
| --- | --- |
| Actividad de estudio presentado por: | Juan David Escobar Escobar.  Andrés Felipe Leal Mora.  Juan Manuel Bautista Correa.  William Ramiro Ríos Henao. |
| Tipo de trabajo: | Actividad 3 |
| Modalidad: | Individual / Grupal |
| Profesor/a: | Marlon Cárdenas |
| Fecha: | Marzo 2022 |

Índice de contenidos

[Máster Universitario en Análisis y Visualización de Datos Masivos / Visual Analitics & Big Data 1](#_Toc96875841)

[1. Inicia la monitorización en MongoDB 3](#_Toc96875842)

[2. Inicia la monitorización en MongoDB 4](#_Toc96875843)

[3. Exportar la base de datos 5](#_Toc96875844)

[4. Restaurar una base de datos 5](#_Toc96875845)

[5. Caso de uso 6](#_Toc96875846)

[6. Conversion CSV a JSON 6](#_Toc96875847)

[7. Consultas y actualizaciones en MongoDB 11](#_Toc96875848)

[8. Neo4j 17](#_Toc96875849)

[9. Anexos 20](#_Toc96875850)

# Inicia la monitorización en MongoDB

1. **Inicia la monitorización en MongoDB**, ejecuta las siguientes instrucciones:
   1. En una consola diferente, visualiza las estadísticas con la instrucción
      * mongostat --discover

Text

Description automatically generated with medium confidence

* 1. 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.

Text, letter

Description automatically generated

* 1. **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 (*utiliza capturas para ilustrar tu respuesta*).

**Interpretación:** La operación mongostat –discover, reporta un log de estadísticas de todos los miembros replica set o sharded del cluster de MongoDB**.** La ventana de monitoreo registra en el log del total de registros insertados, seleccionados, actualizados o eliminados, también permite evaluar el uso de la cantidad de memoria utilizada en GB, el tamaño de entrada y salida de red utilizado en MB, y el porcentaje de algunos recursos de maquina utilizados.

# Inicia la monitorización en MongoDB

1. **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*.

Graphical user interface, text, application

Description automatically generated

* 1. 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*.

Text, table

Description automatically generated with medium confidence

# Exportar la base de datos

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

Table

Description automatically generated

# Restaurar una base de datos

1. **Restaurar una base de datos:**
   1. 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
  + - people
    - --people.bson
    - --people.metadata.json
  1. Descomprimid el fichero *act-grupal-people.zip* y ubicad los ficheros en la estructura de directorios indicada en el paso anterior.

Graphical user interface, text, application

Description automatically generated

# Caso de uso

1. **Caso de uso:** restricción deterrazas en Madrid por Covid-19.

Contratan a vuestro equipo para actualizar las restricciones de ciertos locales con 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 las siguientes URLs:

* [Open Data Censo de locales, sus actividades y terrazas de hostelería y restauración](https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=66665cde99be2410VgnVCM1000000b205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default) (Terrazas)
* <https://datos.madrid.es/FWProjects/egob/Catalogo/Economia/Ficheros/Estructura_DS_FicheroCLA_Terrazas.pdf>

Graphical user interface, text, application, email

Description automatically generated

# Conversion CSV a JSON

6.1. En el alcance del contrato se os pide:

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

Graphical user interface, text, application

Description automatically generated

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

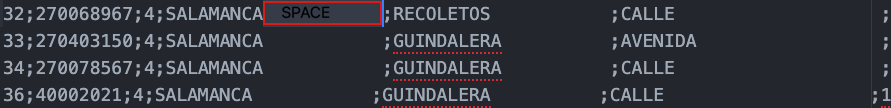


Se identifican valores asociados a columnas sin valor, que se ubican a la decrecha de la utila columna “SILLAS\_RA”, se proceden a eliminar dichas columnas.

Table

Description automatically generated

Se identifican espacios en blanco en columnas de tipo String, por lo cual se aplica la función rtrim() y ltrimp() a todas las columnas de tipo texto.



A picture containing text

Description automatically generated

Se identifican campos numéricos con valores Null y vacíos, los cuales son remplazados por cero y cateados a tipo int.

Text

Description automatically generated

Adicional se identificaron valores atípicos en la variable numérica “**sillas\_ra**”, por lo cual se remplazan los valores atípicos, nulos y vacíos por el valor 0.

Graphical user interface, text, application, email

Description automatically generated

Se detallan todas las operaciones de limpieza y análisis de datos adicionales en el siguiente enlace: <https://github.com/juadaves91/bd-bigdata-unir/blob/main/Activdad%203/Actividad3.ipynb>

Resultado CSV corregido

A screenshot of a computer

Description automatically generated with medium confidence

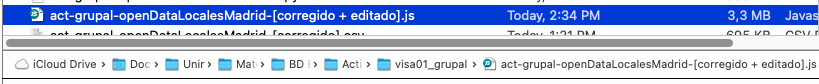
1. Editad el fichero generado y al principio del todo añadid la línea:

var datos\_insertar = […]

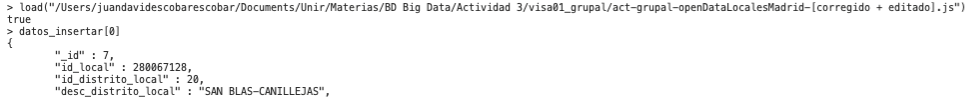
A screenshot of a computer

Description automatically generated with medium confidence

1. Guardad el fichero en una ruta conocida y cambiad su extensión a formato .js. ¿Sabéis qué tipo de fichero es un?j s? Java Script



1. Abrid el cliente *mongo* y ejecutad las siguientes instrucciones:
   1. load(“PATH\fichero.js”)
   2. datos\_insertar[0]



1. Describid brevemente qué hacen las instrucciones anteriores.

**Load()** = Carga y ejecuta un archivo de tipo JavaScript en el cliente de mongo.

**datos\_insertar[0]** = Imprime por consola el primer documento almacenado en el array o colección definido en el archivo JavaScript.

**Evidencia monitoreo:**

**A screenshot of a computer

Description automatically generated with low confidence**

**Interpretación:** La operación mongostat –discover, reporta un log de estadísticas de todos los miembros replica set o sharded del cluster de MongoDB**.** La ventana de monitoreo registra en el log un total de 2999 registros insertados, 0.1% de uso. 48.8KB de uso de salida red, 2.83MB de entrada de red, tamaño de la operación 5.36 GB y la fecha de la transacción.

1. Con lo que habéis hecho antes, ¿podríais realizar un *insert* masivo?

Las instrucciones anteriores permiten cargar un array o colección de objetos tipo json o documentos, los cuales podría enviar como parámetro a al método insertMany(**datos\_insertar**)

Text

Description automatically generated

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

mongoimport --verbose --db madrid --collection terrazas --file "/Users/juandavidescobarescobar/Documents/Unir/Materias/BD Big Data/Actividad 3/visa01\_grupal/act-grupal-openDataLocalesMadrid-[corregido].json"

# Consultas y actualizaciones en MongoDB

6.2. Las actualizaciones las llevaréis a cabo teniendo en cuenta bien la información del local en período estacional o la del resto del año. Las modificaciones son las siguientes:

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.

Table

Description automatically generated

1. 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.

Table

Description automatically generated

1. A las terrazas que se deban inspeccionar, asignad 2 mesas auxiliares y 8 sillas disponibles (*tanto en período estacional como el resto del año*).

Table

Description automatically generatedc

1. 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.

A picture containing table

Description automatically generated

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.

Text, table

Description automatically generated with medium confidence

1. De viernes a sábado, los locales que cierren a las 2:30:00, ahora tendrán que hacerlo a las 2:00:00.

Table

Description automatically generated

1. A todos los locales que estén sobre la calle Alcalá en Madrid se les debe inspeccionar.

Table

Description automatically generated

1. 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”}

Text, table

Description automatically generated

1. Cread una nueva colección llamada *zona1,* con todos los locales que pertenezcan al distrito de Villaverde.

Table

Description automatically generated

1. Cread una nueva colección llamada *zona2,* con todos los locales que pertenezcan al distrito de Salamanca y barrio Castellana.

*Las restricciones se aplican sin importar la estación del año.*

Table

Description automatically generated

# Neo4j

1. En Neo4J, proponed un modelo de datos donde sea posible visualizar los locales de cada barrio y la tipología de terrazas que tienen. Indicad los nodos y sus relaciones; una vez modelado, cread una única consulta que visualice el grafo total. 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**:
   1. Un grafo genérico que represente vuestro modelo (nodos, relaciones y atributos; se deben reflejar todos los campos utilizados).

Chart, bubble chart

Description automatically generated

* 1. Una breve descripción del modelo propuesto.

Modelo entidad relación generado a partir de entidades identificadas de la sabana de datos, act-grupal-openDataLocalesMadrid.xlsx. Las entidades que se generan son las siguientes:

* Terrazas
* Locales
* Periodo estacional
* Situación terrazas
* Municipios
* Barrios

Dichos nodos o entidades se asociaron mediante las siguientes relaciones:

* **Terraza – Local:** Contiene
* **Local – Municipio:** Pertenece
* **Municipio – Barrio:** EsContenedor
* **Terraza - Periodo\_Estacional:** Esta
* **Terraza - Situacion\_Terraza:** TieneSituacion
  1. La descripción del proceso utilizado para cargar los datos en Neo4J a partir de los datos en MongoDB.

**Paso 1.** Lo primero que se hizo fue un proceso de normalización de los datos, identificando primero las entidades que se encuentran en la sabana de datos act-grupal-openDataLocalesMadrid.xlsx, la siguiente tabla muestra la agrupación de cada uno de los 36 campos en las entidades o tablas identificadas:

Graphical user interface

Description automatically generated

Cada una de las tablas tiene un identificador excepto la tabla Barrio, la cual contiene únicamente el nombre o descripción del barrio, estandarizado previamente, mediante un proceso en Jupyter Nootbok.

**Paso 2.** El siguiente paso fue atreves de Python y la librería de Pandas generar, los conjuntos de datos de cada una de las entidades, y se exportan en archivos con formato csv, los cuales se listan a continuación, para ver mas detalle de la operación mencionada se puede referir al adjunto **Actividad 3. Ipny.**

**Table

Description automatically generated**

**Paso 3.** A continuación, se crean conjuntos de datos en Python los cuales soportan o almacenan los datos de las llaves que generan la relación entre 2 nodos, a continuación, se listan las tablas creadas.

Text

Description automatically generated

Los conjuntos de datos generados en el **Paso 2 y 3,** fueron exportados en formato CSV**,** en la ruta configurada por defecto para la BD de NEO4J creada para el ejercicio

Graphical user interface, text, application, email

Description automatically generated

**Paso 3.** En **Neo4j** se crea un nuevo proyecto **“Actividad 3”** y un nueva Base de datos llamada **“Censo\_M”**

Graphical user interface, application

Description automatically generated

**Paso 4.** En **Neo4j** Se generan las sentencias de creación de Nodos y relaciones, las cuales se encuentran descritas en el archivo anexo **ESCOBAR\_ESCOBAR\_JUAN\_DAVID\_actividad\_3\_(Grupal).txt**

Text

Description automatically generated

Text

Description automatically generated

**Paso 4.** Al momento de ejecutar las sentencias de creación de relaciones entre nodos, se puede generar un error de desbordamiento de memoria, por lo cual se opta por aumentar en tamaño de disponibilidad de memoria para Neo4j en el archivo de configuración, o trabajar con menos registros.

* 1. Una consulta CQL y capturas con la visualización del grafo generado con los datos.

Text

Description automatically generated

Chart, bubble chart

Description automatically generated

# Anexos

# **Script ETL**: <https://github.com/juadaves91/bd-bigdata-unir/blob/main/Activdad%203/Actividad%203.ipynb>

# **Script MongoDB**: <https://github.com/juadaves91/bd-bigdata-unir/blob/main/Activdad%203/ESCOBAR_ESCOBAR_JUAN_DAVID_Actividad_Grupal.txt>

# **Script Neo4j**: <https://github.com/juadaves91/bd-bigdata-unir/blob/main/Activdad%203/ESCOBAR_ESCOBAR_JUAN_DAVID_Actividad_Grupal.txt>

# **Modelo entidad relación**: <https://github.com/juadaves91/bd-bigdata-unir/blob/main/Activdad%203/Actividad%203.ipynb>