

SGDI – Práctica 2 – Carga de datos

En esta práctica combinamos 4 tecnologías: 3 bases de datos NoSQL (orientada a documentos : MongoDB, clave-valor: Redis, Orientada a grafos: neo4j) y un analizados de sentimiento, IBM-Watson. Veamos la preparación para cada uno de ellos.

MongoDB

Preparación Mongo

- Descargar los ficheros minitweet.json y miniuser.json
- Crear una carpeta vacía para que se alojen los datos del servidor. Supongamos que esta carpeta se llama c:\hlocal\datos

Abrir 2 consolas (entorno de Mongo en el laboratorio), terminal si hablamos de Linux

- En la primera arrancamos el servidor con
mongod -dbpath c:\hlocal\datos
este paso no es necesario si decidimos utilizar Atlas. Si lo hacemos debemos dejar esta consola aparte (no cerrarla, destruiríamos nuestro servidor)
- En la segunda vamos a importar los datos. Para ellos utilizamos mongoimport (o lo hacemos desde MongoCompass). Por simplicidad suponemos que estamos en la carpeta que tiene los dos ficheros .json

```
mongoimport -d usa -c tweet --drop --file minitweet.json
mongoimport -d usa -c user --drop --file miniuser.json
```

El primer import crea la colección tweet dentro de la base de datos usa con 3820 tweets de la campaña electoral americana 2020. El segundo carga datos de los 3789 usuarios que han emitido estos tweets.

- En la segunda consola, en la que hemos importado los datos, podemos ahora arrancar la shell de mongo como cliente
mongo usa

```
> db.tweet.count()
3820
> db.user.count()
3789
```

datos,
con

ya dentro de la shell podemos comprobar que todo ha ido bien,

mirando el número de documentos de cada colección:

Examinando las colecciones

Antes de tratar los datos es conveniente “conocerlos” un poco.

Usuarios

Para los usuarios el esquema es muy sencillo, echar un vistazo y preguntar cualquier duda.

Tweets

En el caso de los tweets es más complejo porque además hay 2 tipos de documentos que se distinguen por el valor de la clave RT

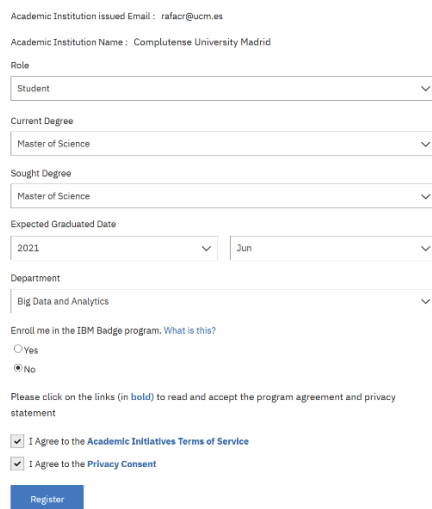
- RT:true. Es un retweet. En este caso es muy importante la clave RT_source que contiene el _id del tweet original
- RT:false. No es un retweet. En este caso contiene mucha información extra destacando el lenguaje del tweet (lang) el propio texto del tweet (texto), un array con los usuarios que se mencionan y mucha más información

IBM-WATSON

Acceso a la librería IBM-Watson para análisis de sentimiento

Para la práctica vamos a utilizar el análisis de sentimiento de la librería Watson. Para eso tenemos que seguir algunos pasos

1. Registrarse en <https://www.ibm.com/academic/home> . Debemos utilizar el Nick de la universidad. En la página final hacer click en las letras azules (links) para ver y aceptar las condiciones de servicio:



Academic Institution issued Email : rafacri@ucm.es

Academic Institution Name : Complutense University Madrid

Role

Student

Current Degree

Master of Science

Sought Degree

Master of Science

Expected Graduated Date

2021 Jun

Department

Big Data and Analytics

Enroll me in the IBM Badge program. [What is this?](#)

☐ Yes

☒ No

Please click on the links (in bold) to read and accept the program agreement and privacy statement

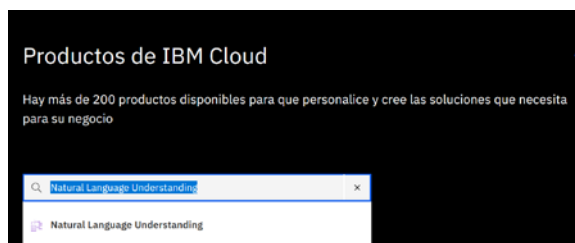
☒ I Agree to the **Academic Initiatives Terms of Service**

☒ I Agree to the **Privacy Consent**

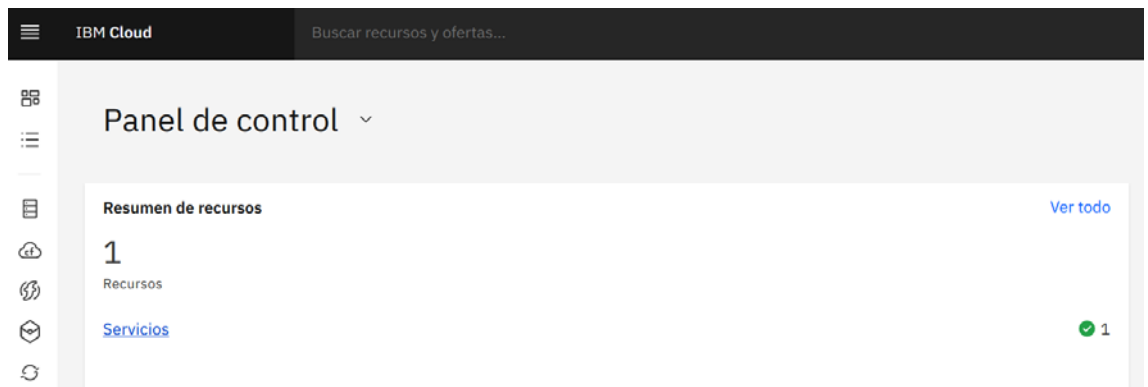
Register

2. Ahora nuestro correo (y el password que hemos creado) nos permiten acceder a los servicios de IBM gratuitos. En nuestro caso queremos entrar en <https://cloud.ibm.com/>

Dentro le daremos al botón “Crear Recurso” (arriba a la derecha). En la página que se abre, en el texto de búsqueda pondremos Natural Language Understanding

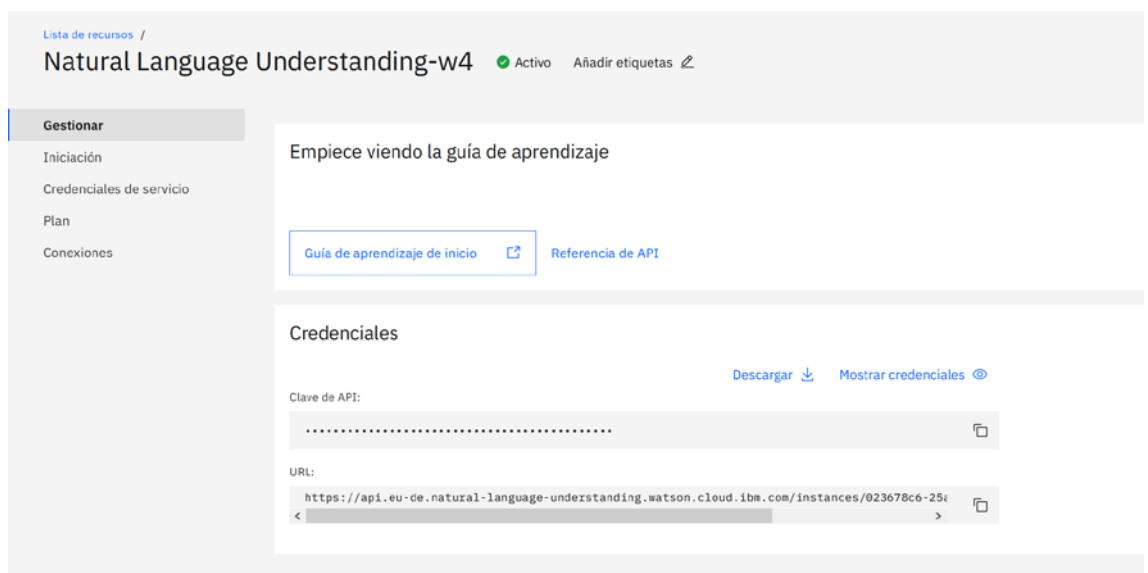


Allí elegiremos el “plan Lite” gratuito. De esta forma, volviendo a la página principal de cloud tendremos este servicio creado



Hacemos click en servicios, y en la lista que sale click en el que se llama “Natural language understanding...” es el único que está activo.

El objetivo final es acceder a una página de la forma:



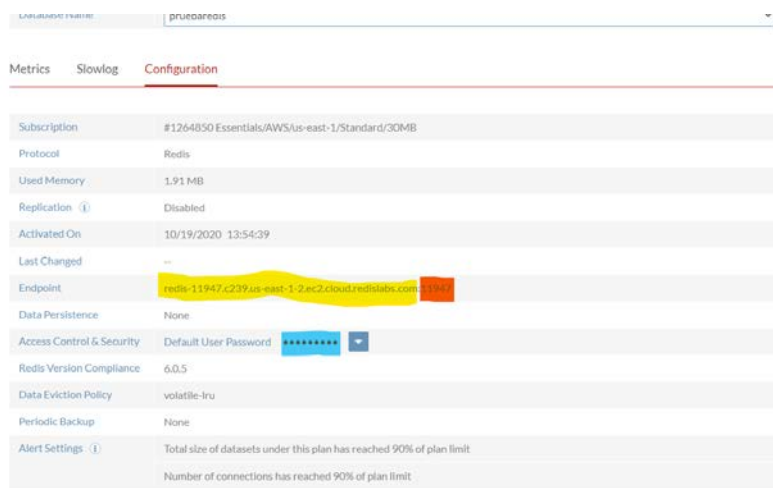
Aquí apuntaremos la URL y la Clave de API. Son los dos argumentos que usaremos en nuestro programa Python

REDIS

Para utilizar Redis necesitaremos disponer de una cuenta en www.redislabs.com. Una vez rellenos los datos y completado el proceso, debemos crear una base de datos. Debemos obtener algo así:



Al hacer click sobre el nombre de la base de datos vemos los detalles que queremos:



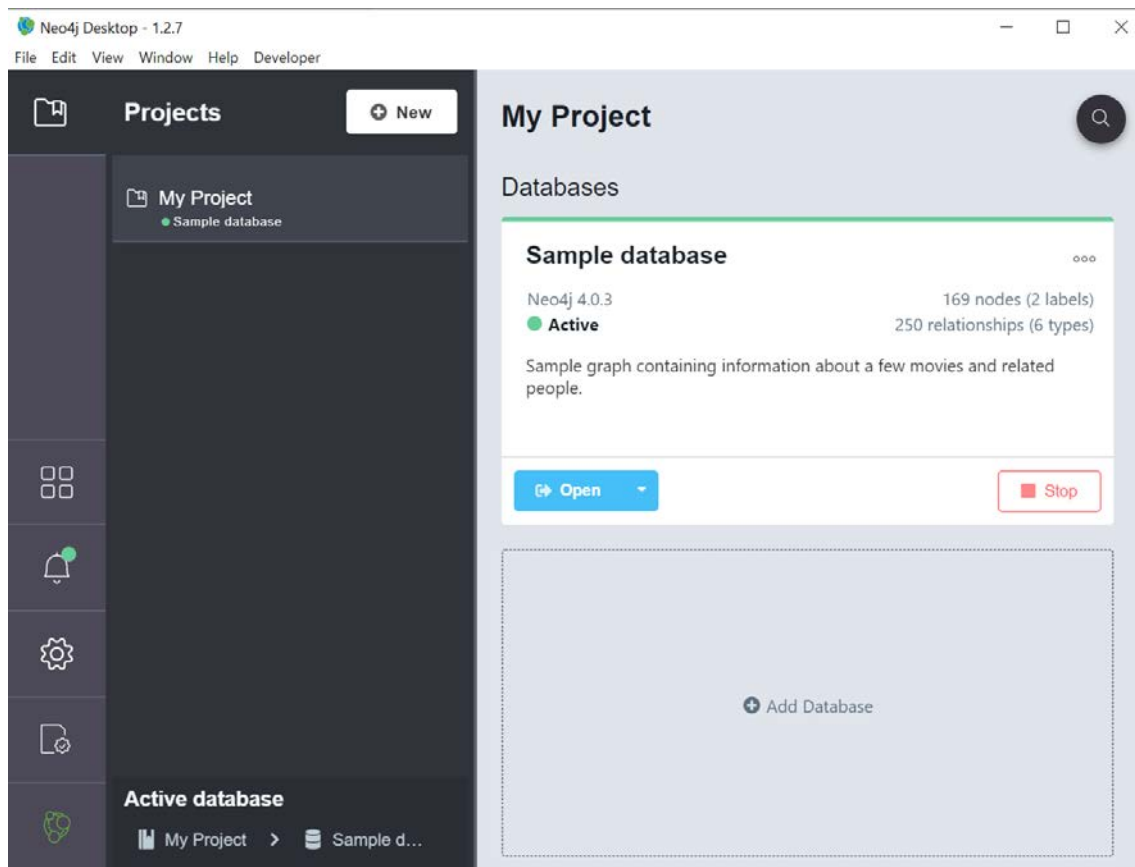
La primera parte del endpoint -en amarillo- tiene que ponerse en la variable “*redisconexion*” de la práctica (ver apartado *Cadena de conexión con Redis*), la segunda parte -en rojo- es el puerto y debe ponerse en la variable *redispuerto*. Finalmente, debemos copiar la password y ponerla en la variable *passwd*.

Neo4j

En el caso de Neo4j no hay servicio en la nube, debemos instalarnos el servidor. El cliente será nuestro propio navegador (recomiendo Chrome, para que es el que va mejor).

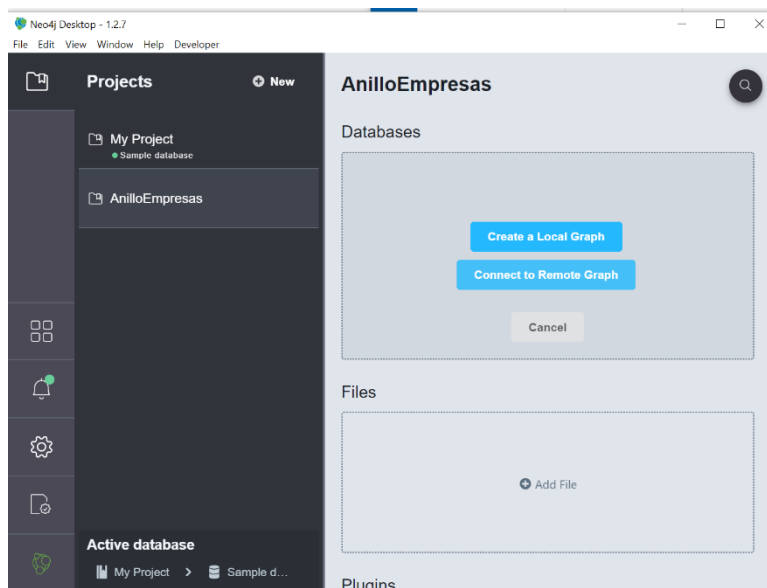
Instalación

La instalación es realmente sencilla. Basta con ir a <https://neo4j.com/> y buscar la sección "Downloads". Allí descargaremos el *Neo4j Desktop*. Tras instalarlo y abrirlo procedemos a crear un nuevo proyecto haciendo click en el botón *New* de la parte izquierda:



Por defecto el nombre del proyecto es *Project* pero si situamos el ratón sobre el nombre y pulsamos en el lápiz a su derecha podemos cambiarlo, por ejemplo a *Twitter*. Cada proyecto puede tener dentro varias bases de datos. En Neo4j el término "base de datos" se denomina, simplemente *grafo*. De toda formas, solo podemos tener un grafo activo cada vez.

Podemos crear un grafo nuevo dentro del proyecto *Twitter* simplemente pulsando en la parte derecha *Add a database*. Allí nos preguntará si queremos crear la base de datos en local o enlazar con una base de datos externa, por ejemplo con una alojada en el servicio cloud de Neo4j, [Aura](#):



Nosotros elegimos *Create a Local Graph*. Nos pedirá un nombre y un password, que debemos recordar. Llamemos a esta base de datos *retweets*. A continuación pulsamos *start* para iniciar el grafo. Ya tenemos nuestro primer grafo creado, de momento con cero vértices y cero aristas. Cuando terminemos debemos recordar que habrá que dar *stop* para parar la base de datos con seguridad.

En Neo4j, el Desktop en local hace el papel de servidor. Como ya lo hemos iniciado, lo siguiente sería abrir un cliente desde el que podamos hacer operaciones como insertar, hacer consultas, etc. Para ello pulsaremos en Open, que abrirá el cliente en el que haremos el resto del trabajo o (recomendado) vamos a un navegador y ponemos

<http://127.0.0.1:7474/browser/>

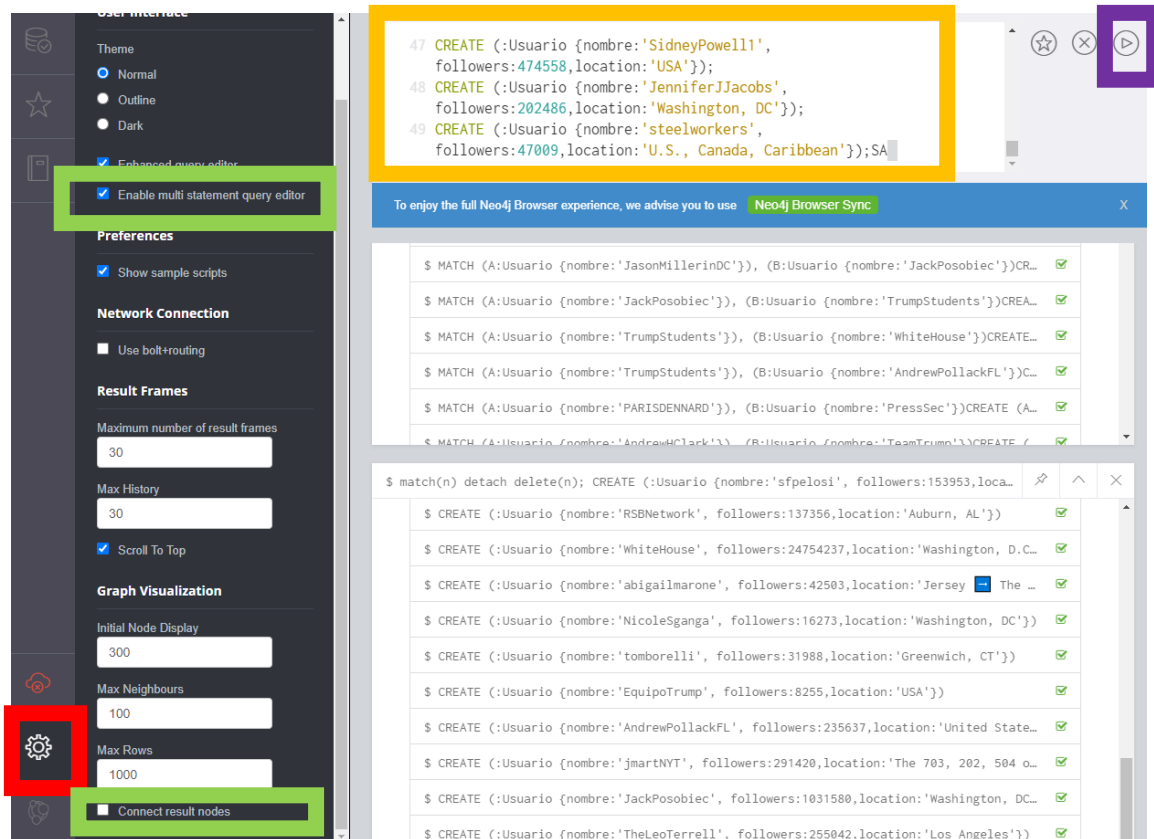
Y a partir de ahora, cuando queramos utilizar Neo4j el proceso será siempre el mismo:

1. Abrir el *Neo4j-Desktop*.
2. Seleccionar (o crear) el proyecto que deseemos, y dentro del proyecto iniciar (o crear e iniciar) el grafo que deseemos
3. Ir al navegador, <http://127.0.0.1:7474/browser/> , o pulsar Open para abrir el browser de columnas, o ir a Python o cualquier otro lenguaje para conectar con el servidor.

Carga de datos en Neo4j

En el navegador puede que nos pida login y passwd. Los valores por defecto son neo4j, neo4j. Luego nos pide una nueva password. En el laboratorio recomiendo poner “neo5js” sin comillas, para que si nos sentamos en otro lugar podamos acceder.

Nos interesan un par de opciones de configuración. Para ello, hacemos click en la rueda dentada de abajo a la derecha (en un cuadrado rojo en la imagen), y nos fijamos en las opciones marcadas en verde:



La primera opción nos permite introducir varias instrucciones simultáneamente, y es muy útil para copiar scripts. La marcamos. La segunda muestra las aristas entre nodos del resultado aunque no pertenezcan al resultado propiamente dicho. Esto a menudo es confuso, lo desmarcamos.

Ya estamos preparados para cargar los datos. En primer lugar copiamos todo el contenido del fichero `nodos_tweet.txt` a la caja naranja, y le damos a al botón de la caja morada para que se ejecute. Esto creará los nodos. Ojo, porque la primera instrucción borra lo que ya hubiera, y también porque puede tardar un poco. Después añadimos las aristas o relaciones copiando de nuevo el contenido del fichero `relaciones_tweet.txt`, y cuando acabe, ya tenemos nuestro grafo preparado. Las preguntas de la práctica se deben escribir en la caja naranja y darle al play (caja morada), y luego cuando veamos que están bien, trasladarlas al notebook.