Desarrollo de Aplicaciones Web - FIUBA. Ejercicios Mongo DB

Ejercicios

1) Crear un archivo llamado import.json y pegarle la información que contiene este link: https://github.com/brianducca/dockerDAII/blob/master/mongodb/import.json

2) Dentro de nuestro proyecto, crear una carpeta de nombre mongo-seed y meter el archivo import.json dentro. Luego crear un archivo import.sh y poner lo siguiente:

```
#! /bin/bash

mongoimport --host mongo --port 27017 --authenticationDatabase
admin --username root --password rootPass --db DAM --collection
Usuarios --type json --file /mongo-seed/import.json --jsonArray
```

3) Crear una carpeta de nombre “src” donde irá nuestra API de NodeJs para consumir la data de MongoDB

4) En la raiz del proyecto, crear un archivo de nombre “docker-compose.yml” y pegar lo siguiente:

```
version: '3'
services:
 mongo:
   image:                              mongo
   hostname:                           mongo
   container_name:                     mongo
   restart:                            always
   environment:
     MONGO_INITDB_ROOT_USERNAME:       root
     MONGO_INITDB_ROOT_PASSWORD:       rootPass
     MONGO_INITDB_DATABASE:            DAM
   volumes:
   -                                   mongodbData:/data/db
   networks:
   -                                   mongo-net
   ports:
   -                                   "27017:27017"
 nodeapp:
   image:
abassi/nodejs-server:10.0-dev
   hostname:                           nodeapp
```

```
  container_name:                       nodeapp
  restart:                              always
  environment:
    MONGO_HOSTNAME:                     mongo
    MONGO_PORT:                         27017
    MONGO_USERNAME:                     root
    MONGO_PASSWORD:                     rootPass
  volumes:
    -                                   ./src:/home/node/app/src
  networks:
    -                                   mongo-net
  depends_on:
    -                                   mongo
  ports:
    -                                   "8080:8000"
  command:                              nodemon src/index.js

mongoexpress-admin:
  image:                                mongo-express
  hostname:                             mongoexpress-admin
  container_name:                       mongoexpress-admin
  restart:                              always
  environment:
    ME_CONFIG_MONGODB_ADMINUSERNAME:    root
    ME_CONFIG_MONGODB_ADMINPASSWORD:    rootPass
  networks:
    -                                   mongo-net
  depends_on:
    -                                   mongo
  ports:
    -                                   "8081:8081"

mongo_seed:
  container_name: mongo_import
  image: mongo
  links:
       - mongo
  volumes:
       - ./mongo-seed:/mongo-seed
  command:
```

```
        /mongo-seed/import.sh
    networks:
        -                                   mongo-net

networks:
 mongo-net:
    driver:                             bridge

volumes:
 mongodbData:
    external: false
```

5) Dentro de la carpeta “src”, crear una carpeta de nombre “db” y dentro de ella en el index.js establecer la conexión a la base de datos, luego exportarlo para utilizarlo como middleware, de la siguiente manera:

```
var MongoClient = require('mongodb').MongoClient;
var ObjectID = require('mongodb').ObjectID;
var nombreBD = "DAM";

// App settings from ENV VARS
const {
    MONGO_USERNAME,
    MONGO_PASSWORD,
    MONGO_HOSTNAME,
    MONGO_PORT
} = process.env;

// Create the URL for connect to DB
var URL_CONNECTION =
    "mongodb://" + MONGO_USERNAME + ":" + MONGO_PASSWORD +
    "@" +
    MONGO_HOSTNAME + ":" + MONGO_PORT;

// create DB settings string
var dbSettings =
    " - MONGO_HOSTNAME: " + MONGO_HOSTNAME + "\n" +
    " - MONGO_PORT: " + MONGO_PORT + "\n" +
    " - MONGO_USERNAME: " + MONGO_USERNAME + "\n" +
    " - MONGO_PASSWORD: " + MONGO_PASSWORD + "\n";
```

```
console.log(dbSettings);

MongoClient.connect(URL_CONNECTION, { useNewUrlParser: true,
useUnifiedTopology: true }, function(err, client) {
    if (err) {
        console.error(err);
        console.log("Error" + err)
        throw err;
    }
    console.log("Conectado a mongo");
    mongoConection.db = client.db(nombreBD);
    mongoConection.objectId = ObjectID;

});

module.exports = mongoConection;
```

6) Crear una carpeta de nombre “usuario” y dentro de ella crear un archivo index.js

7) Crear un método GET que retorne todos los documentos de la colección Usuarios

8) Crear un método GET que retorne la cantidad de documentos de la colección

9) Crear un método GET que retorne el email de todos los usuarios cuyo dni sea mayor a 30.000.000

10) Crear un método GET que retorne vuelva el usuario más joven (que tiene el dni mayor a todos)

11) Crear un método GET que retorne todos los usuarios cuyo apellido comienza con ‘B’