#Ejercicio 01 - MongoDB

Para poder hacer este ejercicio deberías haber leído y entendido antes los siguientes ejemplos:

- 01_mongodb (https://github.com/albertsgrc/curso-mean-jedi-2016Q2/blob/master/ejemplos/01_mongodb)

Introducción:

El propósito de este ejercicio es practicar algunos comandos de MongoDB y que os familiaricéis con su intérprete de comandos.

Primero, aseguraros de haber iniciado el servidor de mongodb con el comando:

```
mongod
```

Luego, situados en la misma carpeta en la que está este readme, ejecutaremos desde el terminal:

```
mongorestore ./base_datos
```

Esto cargará en el servidor mongodb una base de datos llamada `ejercicio_01` que hemos creado para el ejercicio.
Podéis acceder a la base de datos desde el intérprete de mongodb (ejecutando `mongo`) y ver sus colecciones.

Nota: Si no funciona correctamente, prueba la opción `-d ejercicio_01`

Ejercicios:

Sobre la base de datos cargada, escribe una operación de mongodb para cada uno de los siguientes puntos. Tened en cuenta que debéis programar las operaciones de forma independiente a la base de datos que os hemos dado, es decir, no podéis suponer nada respecto a los documentos y datos que habrá en la base de datos.

Podéis probar las operaciones que programéis una a una con el intérprete de mongodb, pero tened en cuenta que:

Deberéis subir en la entrega correspondiente del moodle (http://padawan.jediupc.com/moodle/mod/assign/view.php?id=159) todas las operaciones en un fichero con extensión `.js`, y en el mismo orden en el que aparecen sus respectivos enunciados aquí.

Si quieres resetear la base de datos y volverla a cargar puedes escribir el comando `db.dropDatabase()`
en el intérprete de mongo, habiendo antes ejecutado `use ejercicio_01` para borrarla, y repetir la carga
con `mongorestore ./base_datos`, tal y como se indica en el apartado de Introducción.

1. Inserción:

- **1.1**: Crea un nuevo documento sobre la colección `usuarios` con los siguientes atributos y valores:
    - `nombre: "Maria"` (String)
    - `edad: 22` (Number)
    - `estado_animo: "contenta"` (String)
    - `departamento: "Formación"` (String)
    - `habilidades: ["node", "c++", "c", "swift"]` (Array de Strings)
- **1.2**: Crea otro documento sobre la colección `usuarios`:
    - `nombre: "Sergio"` (String)
    - `edad: 35` (Number)
    - `esta_casado: true` (Bool)
    - `fecha_ultimo_casamiento: new Date()` (Date) (ver tipo Date (https://docs.mongodb.org/manual/reference/method/Date/))
    - `departamento: "RRHH"` (String)
- **1.3**: Crea un nuevo documento sobre la colección `tareas`:
    - `nombre: "Arreglar la bici"` (String)
    - `encargados: [ { nombre: "Alberto", obligado: false }, { nombre: "Fabian", obligado: true }, { nombre: "Peter", obligado: false } ]` (Array de objetos)
- **1.4**: Crea otro documento sobre la colección `tareas`:
    - `nombre: "Limpiar el suelo"`(String)
    - `encargados: []` (Array de objetos)
    - `cancelada: true` (Bool)

2. Consulta:

En todos los casos, si el usuario o tarea no tiene alguno de los atributos por los que se filtra no os preocupéis, asumid el comportamiento por defecto de mongodb en estos casos, que es el que se aceptará.

Todas las consultas **deben mostrarse en formato legible** (método `.pretty()` si es necesaria).

- **2.1**: Lista todos los usuarios
- **2.2**: Lista todos los usuarios con nombre 'Maria'
- **2.3**: Lista todos los usuarios con estado de ánimo 'contento' o 'contenta'
- **2.4**: Lista todos los usuarios con edad mayor a 25. Para cada usuario **sólo hay que mostrar el atributo `nombre`**.
- **2.5**: Lista todos los usuarios con edad entre 20 y 25 ambos incluidos. Para cada usuario **hay que mostrar todos los atributos excepto la `edad`**.
- **2.6**: Lista todos los usuarios ordenados decrecientemente por su edad
- **2.7**: Lista todos los usuarios que no son del departamento "Marketing" y que tienen edad mayor o igual a 20
- **2.8**: Lista todos los usuarios que tengan edad 30, 25 o 16 (ver operador $in (https://docs.mongodb.org/v3.0/reference/operator/query/in/))
- **2.9**: Lista todos los usuarios que tengan entre sus habilidades "mongodb" (ver link (http://stackoverflow.com/questions/5366687/how-to-check-if-an-array-field-contains-a-

unique-value-or-another-array-in-mongo))

- **2.10**: Obtiene la única tarea con nombre "Limpiar el suelo". No podéis usar el método `find()`. Pista: Hay otra que también empieza por find y que hace lo que se pide
- **2.11**: Obtiene el único usuario con _id `56a2a3ef43e14e025b742bbc`. Notad que el atributo `_id` **no tiene tipo `String`**.
- **2.12**: Lista todos los usuarios con fecha de último casamiento posterior o igual a la actual. Fecha se refiere al tipo `Date` de mongodb, no a fecha como día/mes/año
- **2.13**: Lista todos los usuarios con fecha de último casamiento anterior a la actual

3. Actualización:

- **3.1**: Cambia el nombre del usuario con nombre "Fulgencio" por "José"
- **3.2**: Haz que **todos** los usuarios pasen a tener estado de ánimo "radiante", tanto si tenian el atributo estado de ánimo como si no
- **3.3**: Añade un nuevo encargado `{ nombre: "Oscar", obligado: true }` a la tarea que tiene como segundo encargado uno con nombre `"Fabian"` (ver operador $push (https://docs.mongodb.org/manual/reference/operator/update/push/))
- **3.4**: Elimina el primer elemento del array de encargados de la tarea con nombre "Arreglar la bici" (ver operador $pop (https://docs.mongodb.org/manual/reference/operator/update/pop/))

4. Borrado:

- **4.1**: Elimina un único usuario con nombre "h4x0r" (el primero que mongodb encuentre)
- **4.2**: Elimina todos los usuarios con edad mayor a 29

En el fichero `salida_esperada.js` tenéis la salida que se espera para cada una de las operaciones de consulta, y dada una operación `.find()` en cada una de las colecciones una vez finalizadas todas las operaciones. Obviamente los valores de los atributos `_id` para documentos que hayáis insertado vosotros serán diferentes, ya que mongodb los genera en función de la fecha, la máquina, el id del proceso y un valor aleatorio. También será diferente la fecha de último casamiento, ya que depende de cuando hagáis la inserción.