

## Práctica 1 – Primer contacto con MongoDB

Nuestro primer objetivo es conseguir que MongoDB esté accesible y funcione.  
Comenzamos arrancando en **Linux**→**usuario local**

1. Nos movemos al área de trabajo 2 (ctrl+Alt+ → )
2. Abrir un terminal (arriba a la derecha; al lado del símbolo de Chrome; si ya hay un terminal y queremos otro, dar botón derecho sobre el terminal y elegir “Ventana Nueva”)
3. Crear un directorio donde se grabarán los datos: `rm -rf datos; mkdir datos`
4. Iniciamos el servidor tecleando `mongod --dbpath datos`
5. Volvemos al área de trabajo 1 ((ctrl+Alt+ ← )
6. Ahora vamos a importar los datos que usaremos en la práctica:
  1. Bajar del [campus](#) el fichero
  2. Abrimos un nuevo terminal y escribimos

```
mongoimport --db futbol --collection derbi --file /home/hlocal/Descargas/derbi.json
```

debe decirnos que ha importado 672 documentos

7. En el mismo terminal tecleamos `mongo futbol` sin más...ya estamos en el shell de mongo. Podemos teclear `show collections` y debemos ver que la colección derbi está. Y escribir `db.derbi.findOne()` para mirar un poco la estructura de la colección.

A partir de ahora, cuando lleguemos al laboratorio solo habrá que repetir estos pasos para empezar las prácticas.

1) Escribir en la solución (solucion.txt) la representación en JSON el siguiente documento XML

```
<persona>
  <nombre>Bertoldo</nombre>
  <edad>25</edad>
  <direccion>
    <ciudad>Santa Desidia</ciudad>
    <cp>10021</cp>
  </direccion>
  <telefonos>
    <telef type="fijo">666-666-666</telef>
    <telef type="movil">999-999-999</telef>
  </telefonos>
</persona>
```

2) Escribe una consulta que devuelva los tweets de la colección “derbi”, que son respuesta al usuario “Jezivi1” (la clave “in\_reply\_to\_screen\_name” indica a qué usuario es respuesta el tweet). (Ayuda: hay 2 tweets)

3) Consulta para contar el número de usuarios verificados (clave “verified” de “user” a valor true).  
Ayuda: para contar el número de elementos de una consulta añadir `.count()` al final  
Ayuda2: hay 33

4) Escribe una consulta que cuente el número de tweets que corresponden a usuarios con más de 800 000 seguidores (clave “followers\_count” de la clave “user”). Para cada uno de ellos mostrar solo el número de seguidores y el nombre de pantalla (screen\_name). Ayuda: solo hay un usuario con estas características

5) Listado de localidades y nombres de usuario (claves location y screen\_name, en user, respectivamente) ordenado por orden alfabético inverso según la localidad, y dentro de la misma localidad en orden alfabético normal por el nombre de usuario.

6) Consulta para devolver solo la clave user del usuario verificado con menos seguidores. (Ayuda: tiene 1132 seguidores)

7) Vamos a crear una colección con las siguientes instrucciones:

```
db.pru.drop()
db.pru.insert([ {a:1,b:1},{a:1,b:2},{a:2,b:1},{a:2,b:2},{a:3,b:1},{a:3,b:2} ])
```

Escribir una consulta que devuelva:

```
{ "a" : 3, "b" : 1 }
{ "a" : 2, "b" : 1 }
{ "a" : 1, "b" : 1 }
{ "a" : 3, "b" : 2 }
{ "a" : 2, "b" : 2 }
{ "a" : 1, "b" : 2 }
```

8) Afirmación: *Una consulta como `db.coleccion.find().limit(5).sort({nombre:1})` es muy eficiente, porque solo “hay que ordenar” un máximo de 5 documentos.*

¿Cierto o Falso? Justifica la respuesta con una frase.

9) Escribe la instrucción que falta en esta secuencia escrita en el terminal de Mongo:

```
x = { "a" : 1 };
y = "a";
.... // falta una instrucción aquí
print(x.a);
2 // mongo muestra el valor 2
```

Sabiendo que:

- La instrucción debe utilizar las variables x,y
- Se ha omitido la respuesta del shell en todas las instrucciones salvo la última (2)

10) Tenemos una colección con documentos que contienen un array a, como en el siguiente insert:

```
db.pru.insert({a:[10,20,30,40,50]})
```

queremos una consulta que devuelva los dos últimos valores del a para todos los documentos. En el caso anterior mostraría

```
{ "a" : [ 40, 50 ] }
```

Debe ser un find normal, no se permite usar bucles en javascript ni nada similar. Buscar en internet soluciones para mostrar parte de un array

**Entrega:** Subir al campus el fichero solución.txt, no olvidar el nombre de los dos componentes del grupo.