



UNIVERSIDAD
DE GRANADA



Cloud Computing: Servicios y Aplicaciones

Curso 2016 - 2017

Guión de prácticas 3

Bases de Datos NoSQL

El **objetivo** de esta práctica es familiarizarse con el uso de un sistema de gestión de bases de datos en entornos Big Data. Para ello, haremos uso de la aplicación más conocida como es MongoDB.

Para constatar el manejo de la herramienta anterior, el alumno deberá realizar las **tareas** que se describen a continuación y entregar documentación describiendo las tareas realizadas.

Tareas: Realizar los objetivos 1, 2 y 3 que aparecen a lo largo del documento de prácticas

Documentación: Es necesario entregar un informe con la siguiente estructura:

1. Portada con:
 - a. Nombre de la asignatura
 - b. Nombre de la práctica
 - c. Nombre del alumno
 - d. Dirección de correo electrónico
2. Salida de cada consulta realizada en las diferentes tareas.

El documento se entregará en formato PDF. El código fuente JavaScript asociado con las diferentes consultas se adjuntará también en una carpeta a tal efecto.

Tanto el código fuente como el documento, se empaquetará en un único fichero .zip con el nombre "practicaMongoDB" y se entregará a través de la plataforma docente de decsai.

Fecha límite para la entrega: 01 de Mayo de 2017 a las 23:59h.

Objetivo N° 1

Consulta de Documentos

Crear la colección `pedidos` en cada BD asociada a vuestro usuario, sobre la que se realizarán diversas operaciones CRUD. Para crear la colección abre y ejecuta el script `insertar_pedidos.js` (accesible en `/tmp/mongo`). Las tareas a realizar son las siguientes:

1. Visualiza la colección `pedidos` y familiarízate con ella. Observa los distintos tipos de datos y sus estructuras dispares.
2. Visualiza sólo el primer documento de la colección. Utiliza los métodos `.limit()` y `.findOne()`
3. Visualiza el cliente con `id_cliente = 2222`
4. Visualiza los clientes que hayan pedido algún producto de más de 94 euros
5. Visualiza los clientes de Jaén o Salamanca (excluye los datos de los pedidos). Utiliza los operador `$or` e `$in`
6. Visualiza los clientes no tienen campo `pedidos`
7. Visualiza los clientes que hayan nacido en 1963
8. Visualiza los clientes que hayan pedido algún producto fabricado por Canon y algún producto cuyo precio sea inferior a 15 euros
9. Datos personales (`id_cliente`, `Nombre`, `Direccion`, `Localidad` y `Fnacimiento`) de los clientes cuyo nombre empieza por la cadena "c" (No distinguir entre mayúsculas y minúsculas)
10. Visualiza los datos personales de los clientes (excluyendo `_id`). Limita los documentos a 4
11. Ídem anterior pero ordenando los documentos por `Localidad` (ascendente) e `id_cliente` (descendente)

Objetivo N° 2

Agregación

A partir de la colección `pedidos` utilizaremos consultas más complejas por medio de los operadores de agregación (pipeline). Por facilidad se indica la consulta en formato SQL estándar. Las tareas a realizar en este caso obtener:

1. N° total de clientes

```
SELECT COUNT(*) "NUMERO DE CLIENTES"
FROM pedidos;
```

2. N° total de clientes de Jaén

```
SELECT COUNT(*) "NUMERO DE CLIENTES"
FROM pedidos
WHERE Localidad = "Jaen";
```

3. Facturación total clientes por localidad

```
SELECT Localidad, SUM (Facturacion) "TOTAL"
FROM pedidos
GROUP BY Localidad;
```

4. Facturación media de clientes por localidad para las localidades distintas a "Jaen" con facturación media mayor de 5000. Ordenación por Localidad descendente. Eliminar el `_id` y poner el nombre en mayúsculas.

```
SELECT Localidad, AVG (Facturacion) "FACTURACION MEDIA"
FROM pedidos
WHERE Localidad <> "Jaen"
GROUP BY Localidad
HAVING AVG (Facturacion) > 5000
ORDER BY Localidad ASC;
```

5. Calcula la cantidad total facturada por cada cliente (uso de "unwind")

```
SELECT id_cliente "IDENTIFICADOR", nombre "NOMBRE COMPLETO",
       SUM (Precio_unidad * Pedidos) "TOTAL CLIENTE"
FROM pedidos
GROUP BY id_cliente, nombre
ORDER BY 2 DESC
```

Objetivo N° 3

MapReduce

Vamos a utilizar la base de datos libre GeoWorldMap de GeoBytes. Es una base de datos de países, con sus estados/regiones y ciudades importantes. Sobre esta Base de datos vamos a obtener el par de ciudades que se encuentran más cercanas en cada país, excluyendo a los EEUU.

Vamos a importar en nuestra BD de MongoDB un archivo con 37245 ciudades del mundo que está en formato csv (/tmp/mongo/Cities.csv)

```
mongoimport -u <user> -p <clave> --db <bd> --collection  
cities --type csv --headerline --file /var/tmp/Cities.csv
```

Las tareas a realizar en este caso son las siguientes:

1. Encontrar las ciudades más cercanas sobre la colección recién creada mediante un enfoque MapReduce conforme a los pasos que se ilustran en el tutorial práctico.
2. ¿Cómo podríamos obtener la ciudades más distantes en cada país?
3. ¿Qué ocurre si en un país hay dos parejas de ciudades que están a la misma distancia mínima? ¿Cómo harías para que aparecieran todas?
4. ¿Cómo podríamos obtener adicionalmente la cantidad de parejas de ciudades evaluadas para cada país consultado?.
5. ¿Cómo podríamos la distancia media entre las ciudades de cada país?.
6. ¿Mejoraría el rendimiento si creamos un índice?¿Sobre que campo? Comprobadlo.