

## Gestión de la Información en la Web MongoDB desde Python (consultas avanzadas)

**Fecha de entrega: martes 19 de enero de 2016, 13:55h**

### Entrega de la práctica

La entrega de la práctica se realizará a través del Campus Virtual de la asignatura mediante un fichero **grupoXX.zip** donde **XX** es el numero de grupo. Este fichero constará de un fichero **consultas\_mr\_agg.py** con el código del servidor web, cuyo esqueleto se puede descargar del Campus Virtual. Este fichero **consultas\_mr\_agg.py** comenzará con un comentario indicando los autores, el grupo y una **declaración de integridad** expresando que el código es fruto del trabajo de sus miembros. Además del servidor web, el fichero **ZIP** contendrá las plantillas/vistas necesarias para mostrar los datos adecuadamente (si las habéis utilizado).

### Objetivos mínimos

Ninguno. Esta práctica es opcional y únicamente afecta a la nota final. Los grupos que no entreguen esta práctica siguen teniendo la opción de aprobar en la convocatoria de febrero.

### Calificación

Cada ruta aporta 10/6 de la nota de la práctica, es decir, todas tienen el mismo peso.

### Otras consideraciones

El código entregado será resultado exclusivamente del trabajo de sus miembros. No se permite ningún tipo de colaboración entre grupos. Cualquier fragmento de código que aparezca repetido en distintas prácticas o copiado de alguna fuente externa implicará **automáticamente** una calificación de cero y se trasladará dicha situación a las autoridades académicas competentes.

---

En esta práctica consultaremos un servidor MongoDB usando Python mediante la librería *pymongo*. A diferencia de prácticas anteriores en esta ocasión utilizaremos consultas avanzadas para obtener valores agregados utilizando las técnicas que proporciona MongoDB: MapReduce y el *Aggregation Pipeline*.

El primer paso será descargar el fichero con la colección **users.json**. Debéis importar estos datos en la colección **users** dentro de la base de datos **giw**. Tendréis que inspeccionar dicha colección para conocer el esquema implícito de los documentos (todos siguen el mismo patrón). **Es muy importante respetar el nombre de la base de datos y de la colección.**

Para realizar la práctica debéis descargar el esqueleto básico del servidor web del Campus Virtual y usarlo como base. Este esqueleto incluye las 6 rutas en las que el servidor web debe responder a peticiones **GET** (no podéis cambiar las rutas, el método HTTP ni añadir nuevas rutas).

Antes de comenzar con la práctica es importante que consultéis la documentación de MongoDB y *pymongo* para entender cómo realizar este tipo de consultas:

- Documentación oficial de MapReduce en MongoDB:  
<https://docs.mongodb.org/manual/core/map-reduce/>
- MapReduce desde *pymongo*:  
<http://api.mongodb.org/python/current/examples/aggregation.html#map-reduce>

- Documentación oficial del *Aggregation Pipeline* en MongoDB:  
<https://docs.mongodb.org/v3.0/core/aggregation-pipeline/>
- *Aggregation Pipeline* desde *pymongo*:  
<http://api.mongodb.org/python/current/examples/aggregation.html#aggregation-framework>
- Capítulo 7 “Aggregation” del libro “MongoDB: the definitive guide” escrito por Kristina Chodorow, O'Reilly 2013.  
[http://cisne.sim.ucm.es/record=b3296286~S6\\*sp](http://cisne.sim.ucm.es/record=b3296286~S6*sp)

En esta práctica hay que implementar un servidor web que contesta en 6 rutas. Por simplicidad vamos a considerar consultas completamente “cableadas”, es decir, que no aceptan ningún parámetro del usuario. Recordad que todas las peticiones deben consultar la colección **users** de la base de datos **giw**.

#### 1.- /users\_by\_country\_mr

Esta petición calcula el número de usuarios que hay en cada país **utilizando una consulta MapReduce**. Devuelve una página web con una tabla de dos columnas: “País” y “Num. Usuarios”, donde cada fila es el resultado de un país. Las filas pueden aparecer en cualquier orden.

#### 2.- /users\_by\_country\_agg

Similar a la consulta anterior pero utilizando el *Aggregation Pipeline*. En este caso los resultados deben aparecer ordenados de manera **descendente por el número de usuarios**.

#### 3.- /spending\_by\_country\_mr

Calcula el gasto total de todos los pedidos realizados por los usuarios de los países, utilizando para ello una **consulta MapReduce**. Devuelve una página con una tabla de dos columnas: “País” y “Total gastos”, donde cada fila es el resultado de un país. Las filas pueden aparecer en cualquier orden.

#### 4.- /spending\_by\_country\_agg

Similar a la anterior consulta pero utilizando el *Aggregation Pipeline*. En este caso los resultados deben aparecer en **orden ascendente por el nombre del país**

#### 5.- /spending\_female\_3\_orders\_mr

Calcula el gasto total realizado por mujeres con **exactamente 3 pedidos**, (independientemente de qué país sean) utilizando una consulta **MapReduce**. Devuelve una página web con dicho gasto total.

#### 6.- /spending\_female\_3\_orders\_agg

Similar a la consulta 5, pero utilizando el *Aggregation Pipeline*.