



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
IIC2413 - BASES DE DATOS
PROFESOR: ADRIÁN SOTO
AYUDANTES: DANIELA CONCHA, TAMARA CUCUMIDES

Ayudantía I3 (Solución P4)

31 de mayo, 2019

Pregunta 4: MongoDB (I3 2017-2)

Piense en una base de datos en MongoDB con dos colecciones, una de usuarios de una red social y otra de estados publicados en ella:

```
// Usuarios
{
  "uid": 1,
  "name": "Marcelo Saldías",
  "age": 21,
  "description": {
    "estudia_en": "PUC",
    "Animes favoritos": ["Haikyuu", "Love Live"]
  }
}
// Estados
{
  "eid": 1,
  "uid": 1,
  "content": "Grande FUEL #BurnBlue"
  "likes": [1, 4, 7]
}
```

En que los **usuarios** tienen anidado un documento **description** que indica dónde estudian y sus **animes favoritos** (con un JSON Array que contiene los labels de los animes). Además cada **estado** emitido por el usuario posee un arreglo con los id de los usuarios que le han dado **like** a ese estado.

Se pide que entregue la siguiente consulta en MongoDB (sin usar un lenguaje de programación externo):

- a) Entregue el id y nombre de todos los usuarios que estudian en la "PUC" y tienen más de 20 años.

Ahora utilizando PyMongo se pide un procedimiento que entregue lo siguiente:

- b) Entregue cada a usuario junto al número total de likes que tiene.

Ahora se pide que entregue una secuencia de pasos para crear el índice correspondiente, junto al procedimiento para responder la siguiente consulta:

- c) Imprima el id de todos los estados que contienen el hashtag "#BurnBlue" pero no el hashtag "#ShockTheWorld", junto al nombre de todos los usuarios que le dieron like al estado.

Solución

Para todas las consultas tenemos:

```
from pymongo import MongoClient
```

```
client = MongoClient(MONGOSERVER, MONGOPORT)
db = client[MONGODBATABASE]
```

```
a) db.usuarios.find(
    {
        $and: [
            {description.estudia_en: "PUC"},
            {age: {$gt: 400}}
        ]
    }, {})
```

```
b) def likes_usuarios():
    # Recorremos todos los usuarios:
    for u in db.usuarios.find():
        # Filtramos los estados para solo recorrer
        # los que pertenecen a cada usuario:
        likes = 0
        for e in db.estados.find({"uid": u["uid"]}, {}):
            # Sumamos la cantida de likes asociados:
            likes += e["likes"].len()
        # Imprimimos
        print(f'El usuario ({u["uid"]}) {u["name"]} tiene {likes} likes.')
    # El usuario (1) Marcelo Saldías tiene 3 likes.
```

```

c) # Definimos el index
    estados = db.estados
    estados.drop_indexes()
    estados.create_index([("content", "text")])

def id_hashtag():
    # Filtramos los estados por su contenido:
    for e in estados.find("$text": {
        "$search": '"#BurnBlue" - "#ShockTheWorld"'}, {}):

        # Recorremos los usuarios que le dieron like al estado:
        usuarios_like = []
        for uid in e["likes"]:
            u = db.usuarios.find({"uid": uid}, {})
            # Guardamos sus nombres:
            usuarios_like.append(u["name"])

    print(f'Estado {e["eid"]} tiene likes de {usuarios_like}')
    # Estado 1 tiene likes de ["Marcelo Saldías", ...]

```