



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2143 Bases De Datos (II/2017)

Guía Entrega 4

1. Objetivos

- Familiarizarse con la librería de *Python*: `flask` y `pymongo`.
- Aprender a crear una *API* con `flask`.

2. Requisitos

Para la ayudantía, **deben** tener instalado MongoDB¹, Python v3.x, las librerías `flask`² y `pymongo`³.

3. Conceptos Claves

- API
- *Framework*

4. Librería: flask

La librería `flask` es ...

- **Rutas:** En el curso de *Programación Avanzada*, viste una materia que se llamaba *decoradores* que servían para agregarle cosas a una función sin cambiar su estructura. En `flask`, las *rutas* corresponden a decoradores que tomarán una función para redirigirla a una *ruta*, valga la redundancia, que podrás ver en tu *url*. En general, las *rutas* se ven así:

```
"""
url -> localhost:puerto/index (puerto es un número)
"""
@app.route("index")
def home():
    return "Hola Mundo"

"""
url -> localhost:puerto/index/1 (ojo que es distinto que la ruta anterior)
"""
@app.route("index/1")
```

¹Documentación MongoDB

²pip3 install flask

³pip3 install pymongo

```
def show_data():
    return some_json

"""
url -> localhost:puerto/index/<id>/ (ojo que es distinto que la ruta anterior)
"""
@app.route("msj/<id>") #id es un parámetro
def show_data(id):
    find_msj(id); #busco el mensaje con ese id
    return some_json
```

- **Métodos GET y POST:** En esta entrega sólo deberán utilizar el método **GET** que les retornará un JSON.
- **Retorno:** En general, el retorno puede ser un *string*, un JSON o bien, un *template* HTML. En el caso de retornar un JSON, se utiliza `jsonify` y para el HTML, `render_template`.

5. Mongo

Para poder ejecutar la base de datos **MongoDB**, primero deben correr el servidor de Mongo⁴ y en otra pestaña de la terminal, cmd o lo que utilice su SO, pueden ejecutar Mongo escribiendo `mongo`.

- **Base de Datos y colecciones**
- **Importar un JSON en MongoDB:** Se usa el comando `mongoimport` que se verá en clases.
- **Consultas:** El formato de consultas es el siguiente:

```
db.find( { (1) }, { (2) } )
```

donde (1) representa el filtro que uno le quiere hacer a los datos y (2) los *campos* que se quieren mostrar.

- **Búsqueda en texto:** Para poder hacer una búsqueda en texto, deben primero crear un índice⁵ sobre el *campo* que querrán filtrar por palabras o frases. Para poder aplicar el filtro, la consulta se verá de la siguiente forma:

```
db.find( { "$text": { "$search": (3) } }, { } )
```

donde (3) puede filtrar por palabras o frases **exactas**, que contengan alguna de las palabras o bien, que **no** contengan esas palabras o frases⁶.

6. Librería: pymongo

Es la *API* de **MongoDB** para *Python* donde puedes acceder a tu base de datos y colecciones. Para probar que tu código esté funcionando bien, no te olvides de correr el servidor de **MongoDB**. Para empezar, el código parte más o menos así:

```
from pymongo import MongoClient

client = MongoClient('localhost', 27017) #viene por default
db = client.mi_base_de_datos
collection = db.collection
```

⁴sudo mongod (en el caso de macOS)

⁵Crear índice MongoDB

⁶Para más información, revisar acá.