Persistencia de datos

Acceso a bases de datos

Índice

Fundamentos

- Habitualmente, la información persistente se encuentra en bases de datos, que pueden ser relacionales o NoSQL
- Python proporciona módulos para trabajar con ambos tipos de bases de datos.
- El módulo pymongo permite acceder a bases de datos MongoDB, una de las más utilizadas

Base de datos MongoDB

- Es una base de datos documental. Organizada en colecciones de documentos
- Se puede descargar desde:

  https://www.mongodb.com/download-center/community

- Existen herramientas gráficas para trabajar cómodamente con MongoDB, como compass:

  https://www.mongodb.com/download-center/compass

Instalación de pymongo

- Para instalar la librería, es necesario ejecutar la siguiente instrucción desde la línea de comandos:

```
> pip install pymongo
```

- Puede instalarse también desde el terminal de Spyder:

```
>!pip install pymongo
```

```
In [72]: !pip install pymongo
Collecting pymongo
  Downloading https://files.pythonhosted.org/packages/5b/df/d0f82279467c72dd0c8cd1908e04a7fb56145a5d222704722e2593af79f1/
pymongo-3.10.1-cp37-cp37m-win_amd64.whl (354kB)
Installing collected packages: pymongo
Successfully installed pymongo-3.10.1
```

Conexión con la base de datos

- Para operar con una base de datos de MongoDB utilizaremos el método MongoClient del objeto pymongo para conectar con el servidor, al que le pasaremos la cadena de conexión dicho servidor :

```
uri="mongodb://localhost:27017"
sr=pymongo.MongoClient(uri)
```

- Una vez establecida la conexión, puede obtenerse una referencia a una base de datos utilizando la expresión objeto_servidor.basedatos:

```
#referencia a base de datos "formacion"
db=sr.formacion
```

- Si la base de datos no existe, se creará en ese momento.

Colecciones

- Las bases de datos MongoDB se organizan en colecciones. Cada colección contiene un conjunto de objetos JSON.
- Para listar los nombres de todas las colecciones de la base de datos sería:

```
print(db.list_collection_names())
```

- Si queremos obtener una referencia a una colección concreta, por ejemplo "cursos", sería:

```
cursos=db.cursos
```

- Si la colección no existe se creará en ese momento

Inserción de documentos

- Para añadir documentos a una colección utilizaremos el método insert_one() de la colección, pasándole como parámetro el objeto JSON:

```
cursos.insert_one({"curso":"Java","duracion":120,"precio":350})
```

- Se puede añadir un conjunto de documentos, pasándole el array de objetos JSON al método insert_many():

```
cursos.insert_many([{"curso":"Java","duracion":120,"precio":350},
                    {"curso":"Python","duracion":80,"precio":250}])
```

- Cada documento que se añade a la colección incluye **una propiedad id** que es generada de forma automática y contiene un identificador único del documento

Búsqueda de documentos

- Un objeto colección dispone del método find() para recuperar los documentos que cumplen con el filtro pasado como parámetro. Si no se proporciona filtro, devuelve todos los documentos de la colección:

```
for c in cursos.find():
    print(c)
```

```
{'_id': ObjectId('5f1ef6785bb3c00d83b4fadc'), 'curso': 'Java', 'duracion': 120, 'precio': 350}
{'_id': ObjectId('5f1ef6785bb3c00d83b4fadd'), 'curso': 'Python', 'duracion': 80, 'precio': 250}
```

- En el caso de búsqueda por igualdad, el filtro tiene el siguiente formato:

```
{propiedad:valor}
```

- Para aplicar otro operador sería:

```
{propiedad:{operador:valor}}
```

```
#imprime los cursos con precio inferior a 300
for c in cursos.find({"precio":{"$lt":300}}):
    print(c)
```

- Los posibles operadores son: $lt, $lte, $gt, $gte, $ne y $regex

Eliminación y actualización de documentos

- Mediante los métodos de colección delete_one y delete_many, podemos borrar uno o varios documentos que cumplan la condición

```
>cursos.delete_many({"duracion":{"$lt":100}})
```

- Por su parte, la actualización la realizamos con update_one y update_many:

```
>cursos.update_one({"curso":"java"},{"$inc":{"precio":100}});

>cursos.update_many({"curso":{"$regex":"ava"}},{"$inc":{"precio":15}});
```

- Los operadores que se pueden utilizar con update para modificación de propiedades son: $inc, incrementa el valor en una cantidad, $mul, multiplica el valor por una cantidad.