Trabajo Tema 4 DDSI

22/01/2021

Víctor J. Rubia López

Pablo M. Moreno Mancebo

Antonio Galera Gazquez

Juan Andrés Peña Maldonado

Grupo 3

ÍNDICE

Descripción y descarga del SGBD 2

Descripción de DDL y DML usado 3

Sentencias que hemos usado 5

Mecanismo de conexión a la BD desde una aplicación 7

Crítica: ¿Sería adecuado usar esto en nuestro Sistema de Gestión de Torneos de Pádel? 9

Descripción y descarga del SGBD

En nuestro trabajo hemos decidido usar el Sistema Gestor Base de Datos MongoDB. A continuación, explicaremos qué hay que hacer para instalar MongoDB en nuestro sistema, que en este caso es Linux.

Usamos MongoDB ya que es una base de datos NoSQL escalable, tanto vertical como horizontalmente, mediante el uso de nodos. Además es bastante flexible, ya que no sigue ningún esquema, al contrario que las bases de datos relacionales. Este sistema, tiene una alta disponibilidad en cuanto a que posee un gran volumen de acceso, por ejemplo de usuarios o por distintas aplicaciones. La razón fundamental para haberlo elegido ha sido que es un software libre y que su documentación oficial es muy buena.

Abriremos una terminal, escribiremos y ejecutaremos los siguientes comandos:

```
$ sudo apt update
$ sudo apt install -y mongodb
```

Comprobaremos que efectivamente está instalado y activo con el siguiente comando:

```
$ sudo systemctl status mongodb
```

Nos aparecerá algo como lo siguiente. Efectivamente nos indica que está activo.

```
mongodb.service - An object/document-oriented database
   Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor
preset: enabled)
   Active: active (running) since Sat 2018-05-26 07:48:04 UTC; 2min 17s ago
     Docs: man:mongod(1)
 Main PID: 2312 (mongod)
    Tasks: 23 (limit: 1153)
   CGroup: /system.slice/mongodb.service
           └─2312 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config
/etc/mongodb.conf
```

Descripción de DDL y DML usado

- DDL

SENTENCIAS	EQUIVALENTE EN MONGODB	DESCRIPCIÓN

CREATE TABLE	`db.insertOne({})`
	
	`db.createCollection()`
	
	`db.insertMany({},...,{})`
		El primero de ellos crea implícitamente una entrada y además crea la tabla implícitamente, si esta no existe.
		
		El segundo crea una colección (asemejado a las tablas de SQL) con el nombre que se indique entre paréntesis.
		
		El tercero crea la tabla e inserta múltiples filas.

ALTER	`db.tabla.updateMany({})`	Podemos añadir campos usando $set y eliminarlos usando $unset
DROP	`db.tabla.drop()`	

- DML

SENTENCIAS	EQUIVALENTE EN MONGODB
SELECT	`db.tabla.find({},{})`
INSERT	`db.tabla.insertOne({})` `db.tabla.insertMany({},...,{})`
DELETE	`db.tabla.updateMany({},{})`
UPDATE	`db.tabla.deleteMany({})`

Sentencias que hemos usado

Seleccionamos la colección “jugador” y borramos la tabla

```
# colección
jugadores = db.jugadores

jugadores.delete_many({})
```

En vez de borrarlo, podríamos también mostrar sólamente el nombre de todos los jugadores que tengamos en la base de datos

```
for x in jugadores.find({}, {'_id': 0, 'nombre': 1}):
      print(x)
```

Podemos añadir múltiples jugadores a la base de datos

```
misJugadores= [{   "id_jugador": 1,
    "Nombre": "Pablo"
},
{   "id_jugador": 2,
    "Nombre": "Victor"
},
{   "id_jugador": 3,
    "Nombre": "Andres"
},
{   "id_jugador": 4,
    "Nombre": "Antonio"
}]

jugadores.insert_many(misJugadores)
```

Por último, imprimimos los resultados.

```
jugadores.insert_many(misJugadores)
for x in jugadores.find():
     print(x['nombre'])
```

Resultados:

```
pablo@pablomoreno:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("47b91b94-5797-4c49-a982-abab09d7380b") }
MongoDB server version: 3.6.8
Server has startup warnings:
2021-01-13T19:19:10.654+0100 I STORAGE  [initandlisten]
2021-01-13T19:19:10.654+0100 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2021-01-13T19:19:10.654+0100 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2021-01-13T19:19:11.204+0100 I CONTROL  [initandlisten]
2021-01-13T19:19:11.204+0100 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-01-13T19:19:11.204+0100 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2021-01-13T19:19:11.204+0100 I CONTROL  [initandlisten]
> use ddsi
switched to db ddsi
> db.jugador.insertMany(     [{   "id_jugador": 1,         "Nombre": "Pablo"     },     {   "id_jugador": 2,         "Nombre": "Victor"     },     {   "id_
id_jugador": 4,         "Nombre": "Antonio"     }] );
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fff32b73fac5dfe7b220915"),
                ObjectId("5fff32b73fac5dfe7b220916"),
                ObjectId("5fff32b73fac5dfe7b220917"),
                ObjectId("5fff32b73fac5dfe7b220918")
        ]
}
> db.jugador.find({})
{ "_id" : ObjectId("5fff32b73fac5dfe7b220915"), "id_jugador" : 1, "Nombre" : "Pablo" }
{ "_id" : ObjectId("5fff32b73fac5dfe7b220916"), "id_jugador" : 2, "Nombre" : "Victor" }
{ "_id" : ObjectId("5fff32b73fac5dfe7b220917"), "id_jugador" : 3, "Nombre" : "Andres" }
{ "_id" : ObjectId("5fff32b73fac5dfe7b220918"), "id_jugador" : 4, "Nombre" : "Antonio" }
> db.jugador.update( { "Nombre": "Pablo" }, { $set : { "Apellidos" : "Moreno Mancebo", "id_jugador" : 0 } } )
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.jugador.find({})
{ "_id" : ObjectId("5fff32b73fac5dfe7b220915"), "id_jugador" : 0, "Nombre" : "Pablo", "Apellidos" : "Moreno Mancebo" }
{ "_id" : ObjectId("5fff32b73fac5dfe7b220916"), "id_jugador" : 2, "Nombre" : "Victor" }
{ "_id" : ObjectId("5fff32b73fac5dfe7b220917"), "id_jugador" : 3, "Nombre" : "Andres" }
{ "_id" : ObjectId("5fff32b73fac5dfe7b220918"), "id_jugador" : 4, "Nombre" : "Antonio" }
> db.jugador.deleteMany({})
{ "acknowledged" : true, "deletedCount" : 4 }
> db.jugador.find({})
>
```

Mecanismo de conexión a la BD desde una aplicación

Para conectarnos a la base de datos MongoDB usaremos Python 3.8. Previamente, debemos instalar mediante pip su paquete para mongo.

```
pip install pymongo
```

A continuación, crearemos un archivo .py para hacer demostrar el mecanismo de conexión. Este archivo llevará en el encabezado la siguiente línea que nos permite importar los archivos de MongoDB necesarios para trabajar en Python.

```
from pymongo import MongoClient
```

En las siguientes líneas añadimos un main y establecemos conexión con la base de datos MongoDB que tenemos en nuestro sistema.

```
def main():
    mongoClient = MongoClient('localhost', 27017)
```

A continuación, seleccionamos que vamos a usar la base de datos llamada “ddsi”

```
    db = mongoClient.ddsi
```

Crítica: ¿Sería adecuado usar esto en nuestro Sistema de Gestión de Torneos de Pádel?

Como hemos estructurado el proyecto no seria conveniente usar mongo ya que no podriamos obtener una funcionalidad y el control con los conocimientos que tenemos de mongo de restricciones de integridad básicas y de redundancia de datos como los que nos ofrece sql para controlar la base de datos.