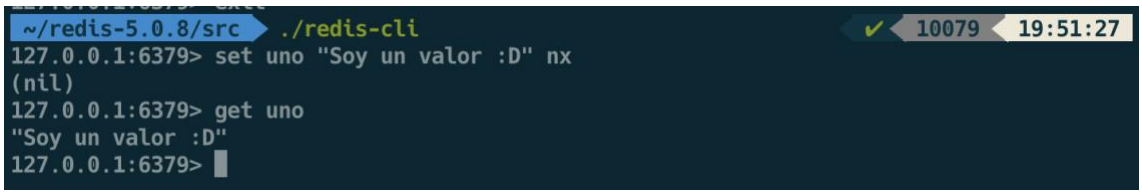
Bases de Datos NoSQL

Práctica 3: Redis

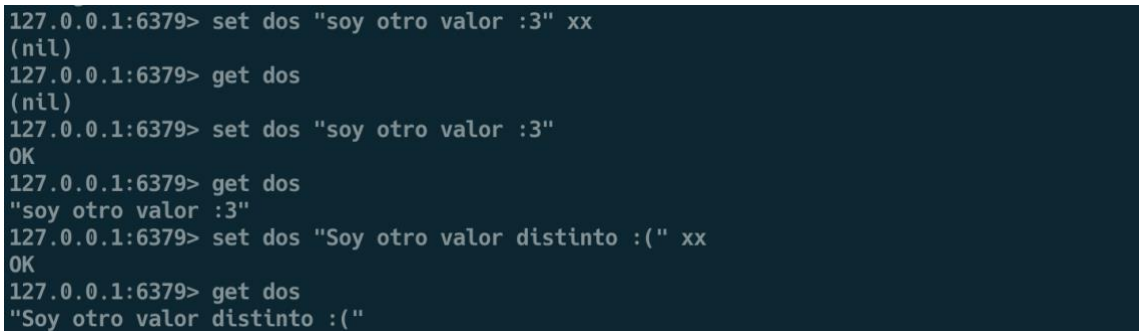
Javier Muñoz Haro

1. **Estructura de datos**

   1.1 Redis Strings



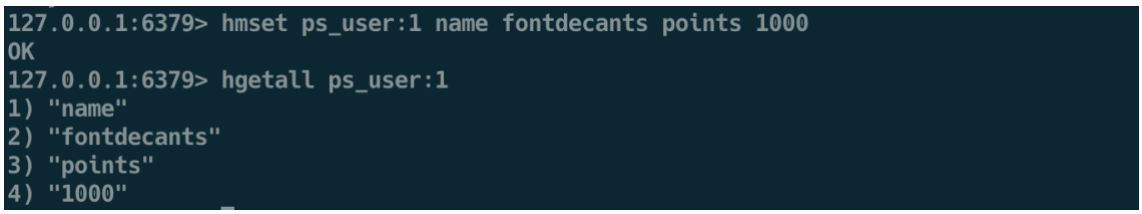
Este resultado es debido al parámetro que se pone al final “nx”. Este parámetro determina que, si la clave no existe, se cree y se le asigne el valor. Si consultamos el valor “uno” vemos que el valor ha sido guardado.



En este caso, el parámetro “xx” solo permite modificar el valor de la clave si esta ya existe. Al no existir de primeras la clave “dos”, el comando no almacena ningún valor. Posteriormente se hace un set sin argumento, por lo que se crea y se almacena un valor en la clave. Después una vez creada, se vuelve a usar el argumento sea “xx” y este funciona, dado que la clave ya está creada.

1.2 Redis Hashes

¿Qué comando tendríamos que utilizar si quisiésemos obtener todos los campos de una entrada?



1.3 Redis Sets

¿Qué comando tendríamos que utilizar si quisiésemos saber si existe un elemento en mi set?

```
127.0.0.1:6379> sadd colorset green blue yellow
(integer) 3
127.0.0.1:6379> sismember black
(error) ERR wrong number of arguments for 'sismember' command
127.0.0.1:6379> del colorset
(integer) 1
127.0.0.1:6379> sadd colorset green blue yellow
(integer) 3
127.0.0.1:6379> sismember colorset black
(integer) 0
127.0.0.1:6379> sismember colorset blue
(integer) 1
127.0.0.1:6379> spop colorset
"green"
```

Solo devuelve un valor del conjunto, el cual es obtenido de manera aleatoria.

El comando SPOP devuelve uno o varios elementos aleatorios de nuestro set. ¿Qué efectos tiene este comando sobre nuestro set? ¿Qué otro comando podríamos utilizar para evitar este efecto sobre nuestro set?

```
127.0.0.1:6379> smembers colorset
1) "yellow"
2) "blue"
```

Para evitar que un elemento del set se elimine de manera aleatoria, se deberán de usar los sorted sets.

1.4 Redis Sorted Sets

¿Cómo podemos saber los scores que tienen nuestros elementos asignados?

```
127.0.0.1:6379> zrange colorsortedset 0 -1
1) "green"
2) "blue"
3) "yellow"
127.0.0.1:6379> zrange colorsortedset 0 -1 withscores
1) "green"
2) "1"
3) "blue"
4) "2"
5) "yellow"
6) "3"
127.0.0.1:6379>
```

2. LUA

2.1 hello.lua

```
~/redis-5.0.8/src ./redis-cli                                    ✔ 10071 19:35:19
127.0.0.1:6379> set greetkey "Hello"
OK
127.0.0.1:6379> exit
```

```
~/redis-5.0.8/src ./redis-cli --eval ~/Universidad/Master/Segundo/BBDD\ NoSQL/P3/2.LU
A/hello.lua greetkey , "Carlos"
"Hello Carlos"
```

2.2 multiply.lua

```
~/redis-5.0.8/src ./redis-cli                                    ✔ 10076 19:40:08
127.0.0.1:6379> set multiply 3
OK
127.0.0.1:6379> exit
~/redis-5.0.8/src ./redis-cli --eval ~/Universidad/Master/Segundo/BBDD\ NoSQL/P3/2.LU
A/multiply.lua multiply , 5
(integer) 15
~/redis-5.0.8/src                                                ✔ 10078 19:40:47
```

3. Python

3.1 hello.py

```
~/redis-5.0.8/src python /Users/javiermunoz/Universidad/Master/Segundo/BBDD\ NoSQL/P3
/3.Python/hello.py "greetkey" "Carlos"
b'Hello' Carlos
```

3.2 helloLua.py

```
~/Universidad/Master/Segundo/BBDDNoSQL/P3/2.LUA python /Users/javiermunoz/Universidad
/Master/Segundo/BBDDNoSQL/P3/3.Python/helloLua.py "greetkey" "Carlos"
b'Hello Carlos'
~/Universidad/Master/Segundo/BBDDNoSQL/P3/2.LUA                  ✔ 10103 21:00:38
```

3.3 wordCount.py

```
Quijote 52

caballero 49

habia 48

Don 38

asi 29

armas 26

dijo 25

aquel 20

venta 19

nombre 17

~/Universidad/Master/Segundo/BBDDNoSQL/P3/3.Python               ✔ 10116 18:02:27
```