



Licenciatura en Sistemas de la Información

Cátedra: Bases de Datos NoSQL

Profesores: Walter Bel - Ernesto Ledesma

Alumno: Mateo Buris

TP 1 - Redis

2021

1- Compruebe que el servidor esté funcionando, utilizando el comando PING (el servidor devolverá PONG si está funcionando bien).

```
mateo@mateo: ~  
127.0.0.1:6379> ping  
PONG  
127.0.0.1:6379> █
```

2- Inserte 5 valores cualesquiera a la base de datos 0, utilizando el comando SET clave valor y compruebe que hayan quedado guardados usando el comando GET clave.

```
mateo@mateo: ~  
127.0.0.1:6379> set 1 A  
OK  
127.0.0.1:6379> set 2 B  
OK  
127.0.0.1:6379> set 3 C  
OK  
127.0.0.1:6379> set 4 D  
OK  
127.0.0.1:6379> set 5 E  
OK  
127.0.0.1:6379> get 1  
"A"  
127.0.0.1:6379> get 2  
"B"  
127.0.0.1:6379> get 3  
"C"  
127.0.0.1:6379> get 4  
"D"  
127.0.0.1:6379> get 5  
"E"  
127.0.0.1:6379> █
```

3- Cambie a la base de datos 1, utilizando el comando SELECT num_db, e ingrese nuevamente 5 valores.

```
mateo@mateo: ~  
127.0.0.1:6379> select 1  
OK  
127.0.0.1:6379[1]> set 1 F  
OK  
127.0.0.1:6379[1]> set 2 G  
OK  
127.0.0.1:6379[1]> set 3 H  
OK  
127.0.0.1:6379[1]> set 4 I  
OK  
127.0.0.1:6379[1]> set 5 J  
OK  
127.0.0.1:6379[1]> █
```

4- Determine la cantidad de elementos almacenados en la base de datos (1), utilizando el comando DBSIZE.

```
mateo@mateo: ~  
127.0.0.1:6379[1]> dbsize  
(integer) 5  
127.0.0.1:6379[1]> 
```

5- Elimine todos los valores de la base de datos (1), utilizando el comando FLUSHDB, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379[1]> flushdb  
OK  
127.0.0.1:6379[1]> dbsize  
(integer) 0  
127.0.0.1:6379[1]> 
```

6- Liste todas las claves de la base de datos (0), utilizando el comando KEYS *.

```
mateo@mateo: ~  
127.0.0.1:6379> keys *  
1) "1"  
2) "3"  
3) "4"  
4) "5"  
5) "2"  
127.0.0.1:6379> 
```

7- Determine si la clave “xyz” existe en la base de datos (0), utilizando el comando EXISTS clave.

```
mateo@mateo: ~  
127.0.0.1:6379> exists xyz  
(integer) 0  
127.0.0.1:6379> 
```

8- Cambiar el valor de una clave existente en la base de datos (0) y retornar el valor anterior, utilizando el comando GETSET clave nuevo_valor, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> getset 1 Z  
"A"  
127.0.0.1:6379> get 1  
"Z"  
127.0.0.1:6379> 
```

9- Eliminar una clave de la base de datos (0), utilizando el comando DEL clave, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> del 2  
(integer) 1  
127.0.0.1:6379> get 2  
(nil)  
127.0.0.1:6379> █
```

10- Mover una clave y valor de una base de datos a otra (de la 0 a la 1), utilizando el comando MOVE clave num_db, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> move 3 1  
(integer) 1  
127.0.0.1:6379> select 1  
OK  
127.0.0.1:6379[1]> get 3  
"C"  
127.0.0.1:6379[1]> █
```

11- Seleccione una clave de manera aleatoria, utilizando el comando RANDOMKEY, pruébelo más de una vez para ver el resultado.

```
mateo@mateo: ~  
127.0.0.1:6379> randomkey  
"1"  
127.0.0.1:6379> randomkey  
"4"  
127.0.0.1:6379> randomkey  
"5"  
127.0.0.1:6379> randomkey  
"5"  
127.0.0.1:6379> randomkey  
"4"  
127.0.0.1:6379> █
```

12- Determine el tipo de dato del valor asociada a una clave, utilizando el comando TYPE clave.

```
mateo@mateo: ~  
127.0.0.1:6379> type 1  
string  
127.0.0.1:6379> type 4  
string  
127.0.0.1:6379> █
```

13- Cree una lista de personajes de Star Wars que contenga al menos cinco nombres en la base de datos (0), utilizando el comando LPUSH clave valor1 valor2 valor3.

```
mateo@mateo: ~  
127.0.0.1:6379> lpush 2 Yoda Luke_Skywalker Leia_Organa Han_Solo Chewbacca  
(integer) 5  
127.0.0.1:6379> █
```

14- Listar el contenido completo de la lista, utilizando el comando LRANGE clave inicio fin (-1 representa el último elemento de la lista).

```
mateo@mateo: ~  
127.0.0.1:6379> lrange 2 0 -1  
1) "Chewbacca"  
2) "Han_Solo"  
3) "Leia_Organa"  
4) "Luke_Skywalker"  
5) "Yoda"  
127.0.0.1:6379> █
```

15- Obtenga el valor almacenado en el índice 0 de la lista creada en el punto anterior, utilizando el comando LINDEX clave 0. ¿Es el primero que se cargó? Muestra el último en cargar

```
mateo@mateo: ~  
127.0.0.1:6379> lindex 2 0  
"Chewbacca"  
127.0.0.1:6379> █
```

16- Agregue dos nombres más a la lista creada en el punto 13, utilizando LPUSH, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> lpush 2 Darth_Vader R2-D2  
(integer) 7  
127.0.0.1:6379> lrange 2 0 -1  
1) "R2-D2"  
2) "Darth_Vader"  
3) "Chewbacca"  
4) "Han_Solo"  
5) "Leia_Organa"  
6) "Luke_Skywalker"  
7) "Yoda"  
127.0.0.1:6379> █
```

17- Determine cuántos elementos tiene la lista, utilizando el comando LLEN clave.

```
mateo@mateo: ~  
127.0.0.1:6379> llen 2  
(integer) 7  
127.0.0.1:6379> █
```

18- Elimine el primer elemento de la lista, utilizando el comando LPOP, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> lpop 2  
"R2-D2"  
127.0.0.1:6379> lrange 2 0 -1  
1) "Darth_Vader"  
2) "Chewbacca"  
3) "Han_Solo"  
4) "Leia_Organa"  
5) "Luke_Skywalker"  
6) "Yoda"  
127.0.0.1:6379> █
```

19- Ingrese un valor a la lista antes y después de otro valor de la lista, utilizando el comando LINSERT clave BEFORE|AFTER valor nuevo_valor, verifique si las operaciones fueron exitosas.

```
mateo@mateo: ~  
127.0.0.1:6379> linsert 2 before Han_Solo R2-D2  
(integer) 7  
127.0.0.1:6379> lrange 2 0 -1  
1) "Darth_Vader"  
2) "Chewbacca"  
3) "R2-D2"  
4) "Han_Solo"  
5) "Leia_Organa"  
6) "Luke_Skywalker"  
7) "Yoda"  
127.0.0.1:6379> █
```

20- Ingrese en una lista un mismo valor en distintas posiciones, elimine la primera ocurrencia del mismo, utilizando el comando LREM clave cantidad valor, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> lrange 2 0 -1  
1) "Darth_Vader"  
2) "R2-D2"  
3) "Chewbacca"  
4) "R2-D2"  
5) "R2-D2"  
6) "Han_Solo"  
7) "Leia_Organa"  
8) "R2-D2"  
9) "R2-D2"  
10) "Luke_Skywalker"  
11) "R2-D2"  
12) "Yoda"  
127.0.0.1:6379> lrem 2 1 R2-D2  
(integer) 1  
127.0.0.1:6379> lrange 2 0 -1  
1) "Darth_Vader"  
2) "Chewbacca"  
3) "R2-D2"  
4) "R2-D2"  
5) "Han_Solo"  
6) "Leia_Organa"  
7) "R2-D2"  
8) "R2-D2"  
9) "Luke_Skywalker"  
10) "R2-D2"  
11) "Yoda"  
127.0.0.1:6379> █
```

21- Elimine todas las ocurrencias del valor repetido con una sola sentencia, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> lrem 2 5 R2-D2  
(integer) 5  
127.0.0.1:6379> lrange 2 0 -1  
1) "Darth_Vader"  
2) "Chewbacca"  
3) "Han_Solo"  
4) "Leia_Organa"  
5) "Luke_Skywalker"  
6) "Yoda"  
127.0.0.1:6379> █
```

22- Elimine el último valor de una lista, utilizando el comando RPOP clave, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> rpop 2  
"Yoda"  
127.0.0.1:6379> lrange 2 0 -1  
1) "Darth_Vader"  
2) "Chewbacca"  
3) "Han_Solo"  
4) "Leia_Organa"  
5) "Luke_Skywalker"  
127.0.0.1:6379> █
```

23- Cambiar el nombre de una clave, utilizando en comando RENAME clave nueva_clave, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> get 5  
"E"  
127.0.0.1:6379> rename 5 6  
OK  
127.0.0.1:6379> get 6  
"E"  
127.0.0.1:6379> █
```

24- Fuerce a Redis a pasar los datos de memoria a disco, utilizando el comando SAVE.

```
mateo@mateo: ~  
127.0.0.1:6379> save  
OK  
127.0.0.1:6379> █
```

25- Ordene la lista de elementos, utilizando el comando SORT clave (numérico) o SORT clave (alfanumérico).

```
mateo@mateo: ~  
127.0.0.1:6379> sort 2 alpha  
1) "Chewbacca"  
2) "Darth_Vader"  
3) "Han_Solo"  
4) "Leia_Organa"  
5) "Luke_Skywalker"  
127.0.0.1:6379> █
```


26- Ordene ahora la lista de manera descendente.

```
mateo@mateo: ~  
127.0.0.1:6379> sort 2 alpha desc  
1) "Luke_Skywalker"  
2) "Leia_Organa"  
3) "Han_Solo"  
4) "Darth_Vader"  
5) "Chewbacca"  
127.0.0.1:6379> █
```

27- Cambie el valor de uno de los elementos de la lista utilizando el comando LSET, verifique si la operación fue exitosa.

```
mateo@mateo: ~  
127.0.0.1:6379> lrange 2 0 -1  
1) "Darth_Vader"  
2) "Chewbacca"  
3) "Han_Solo"  
4) "Leia_Organa"  
5) "Luke_Skywalker"  
127.0.0.1:6379> lset 2 3 Yoda  
OK  
127.0.0.1:6379> lrange 2 0 -1  
1) "Darth_Vader"  
2) "Chewbacca"  
3) "Han_Solo"  
4) "Yoda"  
5) "Luke_Skywalker"  
127.0.0.1:6379> █
```

28- Ingrese una clave-valor solo si la misma no existe con un solo comando.

```
mateo@mateo: ~  
127.0.0.1:6379> setnx 3 Z  
(integer) 1  
127.0.0.1:6379> get 3  
"Z"  
127.0.0.1:6379> █
```

29- Ingrese una clave-valor de manera temporal con el comando SETEX, verifique si la operación fue exitosa, es decir si se ingresó y luego del tiempo expiró.

```
mateo@mateo: ~  
127.0.0.1:6379> setex 7 10 X  
OK  
127.0.0.1:6379> get 7  
"X"  
127.0.0.1:6379> get 7  
(nil)  
127.0.0.1:6379> █
```

30- Determine el tiempo restante para que expire una clave usando el comando PTTL.

```
mateo@mateo: ~  
127.0.0.1:6379> setex 7 30 X  
OK  
127.0.0.1:6379> pttl 7  
(integer) 22129  
127.0.0.1:6379> █
```

31- Cree un conjunto geoespacial cervceras con el comando GEOADD, de las cervcerías artesanales de la ciudad (Lagash, Drakkar, Biguá, Tractor, 7 Colinas, Ambar).

```
mateo@mateo: ~  
127.0.0.1:6379> geoadd cervcerias -32.4773091229112 -58.24828852686128 Lagash -32.48056662  
329105 -58.23392500394 Drakkar -32.48656672471113 -58.270978932476744 Bigua -32.48122361452  
758 -58.23797779598659 Tractor -32.47988810084559 -58.23535511198071 7_Colinas -32.48223439  
8152315 -58.23349841957728 Ambar  
(integer) 6  
127.0.0.1:6379> █
```

32- Consulte las coordenadas de Lagash con el comando GEOPOS.

```
mateo@mateo: ~  
127.0.0.1:6379> geopos cervcerias Lagash  
1) 1) "-32.47731059789657593"  
2) "-58.24828892547053272"  
127.0.0.1:6379> █
```

33- Agregue la FCyT al conjunto de cervceras

```
mateo@mateo: ~  
127.0.0.1:6379> geoadd cervcerias nx -32.47905177533222 -58.233335105381585 FCyT  
(integer) 1  
127.0.0.1:6379> █
```

34- Determine la distancia en metros y kilómetros desde la FCyT hasta Lagash y Drakkar, utilizando GEODIST.

```
mateo@mateo: ~  
127.0.0.1:6379> geodist cervcerias FCyT Lagash m  
"1666.5018"  
127.0.0.1:6379> geodist cervcerias FCyT Lagash km  
"1.6665"  
127.0.0.1:6379> geodist cervcerias FCyT Drakkar m  
"110.2801"  
127.0.0.1:6379> geodist cervcerias FCyT Drakkar km  
"0.1103"  
127.0.0.1:6379> █
```

35- Determine las cervecerías que se encuentran dentro del radio de 1km utilizando el comando GEORADIUS.

Posicionado en Plaza Ramirez

```
mateo@mateo: ~  
127.0.0.1:6379> georadius cervcerias -32.48455260005434 -58.23211617633781 1 km  
1) "Ambar"  
2) "FCyT"  
3) "Tractor"  
4) "7_Colinas"  
5) "Drakkar"  
127.0.0.1:6379> 
```

36- Suscribase a los siguientes canales: FCyT, IDTILAB, NSQL utilizando el SUBSCRIBE, para implementar el paradigma de mensajería Publish/Subscribe.

```
mateo@mateo: ~  
127.0.0.1:6379> subscribe FCyT IDTILAB NSQL  
Reading messages... (press Ctrl-C to quit)  
1) "subscribe"  
2) "FCyT"  
3) (integer) 1  
1) "subscribe"  
2) "IDTILAB"  
3) (integer) 2  
1) "subscribe"  
2) "NSQL"  
3) (integer) 3  
 
```

37- Ahora desde otro terminal de redis publique el mensaje “Hola a Mundo” en el canal NSQL con el comando PUBLISH.

```
mateo@m  mateo@mateo: ~  
127.0.0.1:6379> publish NSQL hola_mundo  
(integer) 1  
127.0.0.1:6379>   
127.0.0.1:6379> subscribe NSQL  
Reading messages... (press Ctrl-C to quit)  
1) "subscribe"  
2) "NSQL"  
3) (integer) 1  
1) "message"  
2) "NSQL"  
3) "hola_mundo"  
 
```

38- Desde otro terminal suscribase a los canales IDTILAB, “STAR WARS”

```
mateo@mateo: ~  
127.0.0.1:6379> subscribe IDTILAB "STAR WARS"  
Reading messages... (press Ctrl-C to quit)  
1) "subscribe"  
2) "IDTILAB"  
3) (integer) 1  
1) "subscribe"  
2) "STAR WARS"  
3) (integer) 2  
 
```

39- Ahora desde otro terminal de redis publique el mensaje “Hola a todos los miembros del laboratorio” en el canal NSQL con el comando PUBLISH.

```
mateo@mateo: ~  
127.0.0.1:6379> publish NSQL "Hola a todos los miembros del laboratorio"  
(integer) 1  
127.0.0.1:6379>   
127.0.0.1:6379> subscribe NSQL  
Reading messages... (press Ctrl-C to quit)  
1) "subscribe"  
2) "NSQL"  
3) (integer) 1  
1) "message"  
2) "NSQL"  
3) "Hola a todos los miembros del laboratorio"
```

40- Chequear los canales activos con el comando PUBSUB CHANNELS, abra otro terminal y suscribase al canal “STAR WARS” y verifique nuevamente los canales activos.

```
mateo@mateo: ~  
127.0.0.1:6379> pubsub channels  
1) "IDTILAB"  
2) "FCyT"  
3) "NSQL"  
127.0.0.1:6379> pubsub channels  
1) "IDTILAB"  
2) "FCyT"  
3) "STAR WARS"  
4) "NSQL"  
127.0.0.1:6379> 
```

41- Ahora cheque el número de suscriptores de cada canal activo con el comando PUBSUB NUMSUB.

```
mateo@mateo: ~  
127.0.0.1:6379> pubsub numsub FCyT IDTILAB NSQL "STAR WARS"  
1) "FCyT"  
2) (integer) 1  
3) "IDTILAB"  
4) (integer) 2  
5) "NSQL"  
6) (integer) 1  
7) "STAR WARS"  
8) (integer) 1  
127.0.0.1:6379> 
```

42- El comando UNSUBSCRIBE nos permite cancelar la subscripción a un canal o a todos, pero no se puede ejecutar desde el redis-cli, se utiliza cuando estamos trabajando en una aplicación.