**UNIDAD 3: REDIS**

1. [UNIDAD 3: Introducción](https://learn.nextu.com/mod/page/view.php?id=10294&pid=P_WEB_DATABASE)

**REDIS**

Los manejadores orientados a clave/valor se definen como aquellos donde cada valor es accedido vía el uso de una clave, por ejemplo, el valor Luis se puede acceder usando la clave nombre-1. REDIS es uno de los manejadores más conocidos en este renglón. Él se caracteriza por mantener los datos en memoria, lo que se conoce como persistencia en memoria; esta cualidad de Redis hace que sea útil en sistemas donde hay que mantener datos en caché o en sistemas donde existen procesos y es necesario mantener el estatus en memoria. La forma de interactuar con REDIS es, principalmente, a través de los lenguajes de programación y su correspondiente conector para este manejador; sin embargo, en esta unidad interactuaremos con REDIS a través de una consola de comandos que se llama redis-cli. Aprenderemos así a definir elementos y consultarlos, usando el proceso de dada una clave obtenemos su valor. También estudiaremos los tipos de datos, desde los más básicos basados en strings hasta elementos más complejos como los tipos hash, conjuntos y listas. Similarmente a otros manejadores, existen formas de asegurar nuestros datos restringiendo el acceso a los mismos a través de la asignación de contraseñas, aunque en REDIS este proceso es un poco más simple de realizar, te irás familiarizando con dicho mecanismo. Finalmente, analizamos el proceso general de respaldo y recuperación de los datos, que explota la persistencia de estos en memoria.

**Objetivos de aprendizaje**

1. Crear bases de datos NoSQL orientadas a clave/valor usando REDIS, a través del uso de una línea de comandos.
2. Implementar consultas simples sobre los datos en REDIS basadas en el uso de claves y valores.
3. Comprender el proceso de respaldo y recuperación en REDIS usado para mantener copias de seguridad de nuestros datos.
4. Aplicar los diversos comandos de REDIS para definir diferentes tipos de datos que sean útiles en determinados contextos y situaciones.
5. Aplicar los comandos usados para asegurar los datos a través de la asignación de contraseñas que deban usarse en la aplicación de los comandos REDIS.
6. Lección 1: Introducción a REDIS
   1. [Introducción a REDIS](https://learn.nextu.com/mod/lesson/view.php?id=10295&pid=P_WEB_DATABASE)

Redis 🡪 Remote Dirtionary Server

Características:

* Open Source
* Está programado en ANSI C
* Se define como in-memory: esto significa que los datos están en memoria, esto permite que los datos se accedan muy rápidamente.
* Por lo que se utiliza cuando necesitamos obtener datos de forma rápida
* Es un manejador del tipo NoSQL orientado a clave-valor.
* Sigue el paradigma cliente-servidor
* Permite replicación, lo que significa que podemos varios nodos redis, donde los datos de uno se pueden replicar en otro nodo.

Paradigma clave-valor

Se pueden crear formatos, por ejemplo

“clave-1”: “1#maria#30”

Otro ejemplo:

“Xml-1”: “<..>…<..>” 🡪 String que contiene un formato específico.

“bitmap-1”: 0x124a23

Funcionalidades de Redis:

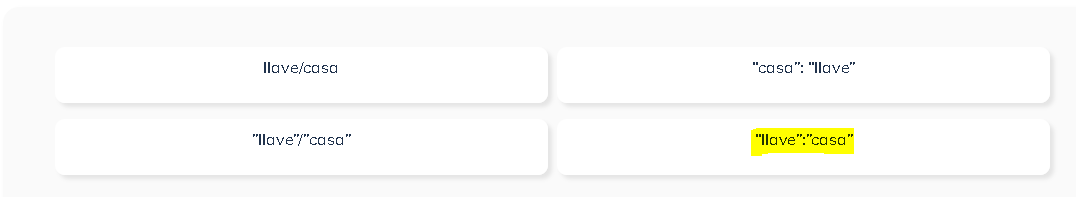
* Maneja la persistencia – en algunas versiones de Redis se pueden guardar los datos en memoria virtual, pero la idea de este manejador es que todos los datos estén en memoria RAM, permitiendo un rápido acceso a ellos.
* Redis tiene un buen rendimiento
* Soporta la replicación por cluster, pudiéndose formas un cluster de nodos redis que trabajen en conjunto.
* Maneja un protocolo de texto muy simple por lo que veremos que la conexión que se establece entre el cliente y el servidor pueden ser sencillas. Esto permite que instalar el conector de Redis de cualquier lenguaje no sea complicado. Lo que se debe hacer es conectarlo a un puerto específico y escribir un string seguido de los datos que se desean guardar. Esto hace que el driver este implementado en la mayoría de los lenguajes de programación.

Usos de Redis:

* Caché de valores que sean generados a partir de operaciones computacionales que consumen mucho tiempo. En este caso en lugar de guardar en disco las estructuras, datos o documentos generados, se guardan en el caché. Esto se puede ejecutar hasta que decidamos expirar el valor por vigencia. 🡪 este modelo se llama de carga floja, es para lo que se usa Redis el 90% de las veces
* En HTML para poder realizar carga de página más rápida.
* Procesos y estatus en lotes. Ejemplo: planillas 🡪 toma tiempo en procesarse 🡪 diferentes tipos de datos.
* De forma general, Redis se utiliza para almacenar datos que se van a almacenar datos en el momento que probablemente vayan a expirar y que luego no se van a necesitar.
  1. Actividad interactiva 1

**Prueba tus conocimientos**

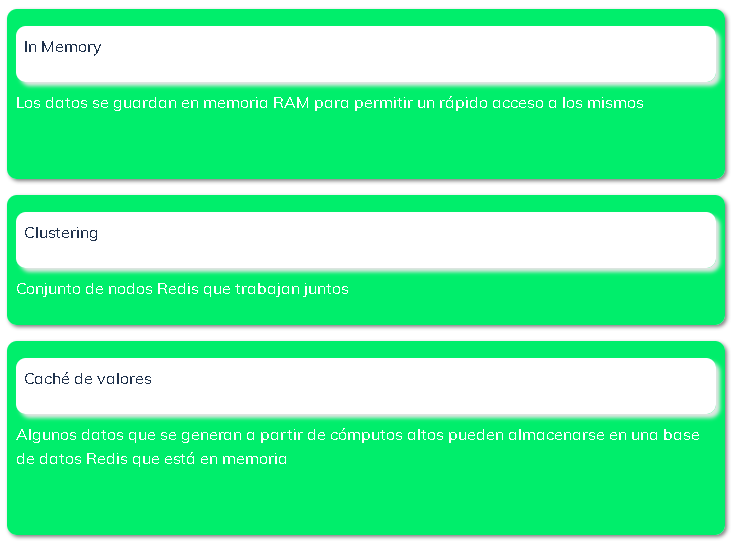
¿Cuál sería la forma de representar el objeto de tipo string llamado “casa” cuya clave es “llave” en el formato clave/valor?



* 1. Actividad interactiva 2

**Analiza y relaciona**

Realiza la correspondencia de los conceptos de la columna de la izquierda con las definiciones de la columna de la derecha.



* 1. Taller 1
  2. Comenzando con REDIS

Redis utiliza el puerto 6339 por defecto

Se usará un cliente predeterminado llamado redis-cli

El prompt redis-cli facilita el uso de los comandos indicando los argumentos permitidos.

Existen clientes gráficos pero no de manera oficial, existen clientes que funcionan para un sistema operativo pero no para otro o tienen algunas limitaciones.

Para ver si el servidor redis esta corriendo

127.0.0.1:6379> PING

PONG

127.0.0.1:6379> PING "HOLA"

"HOLA"

127.0.0.1:6379> PING "Hola Mundo!"

"Hola Mundo!"

Para guardar valores en la base de datos:

SET

Ejemplo:

127.0.0.1:6379> SET id-1 "Luis"

OK

Para obtener los valores almacenados:

127.0.0.1:6379> GET id-1

"Luis"

127.0.0.1:6379> GET id-2

(nil)

Para borrar todos los datos:

127.0.0.1:6379> FLUSHALL

OK

127.0.0.1:6379> GET id-1

(nil)

127.0.0.1:6379>

* 1. Taller 2
  2. ¿Sabías qué?

**Cápsula de conocimiento**

**Resumen de Comandos - Primeros Comandos**

La finalidad de este material es presentarte un resumen de los primeros comandos en REDIS.

* **Entrar a la línea de comando de REDIS:**

> redis-cli

* **Verificar el servidor con PING:**

redis-cli> PING <palabra-opcional>

* **Crear una entrada en la base de datos:**

redis-cli> SET <clave> <valor>

* **Obtener una entrada en la base de datos:**

redis-cli> GET <clave>

* **Eliminar todas las entradas:**

redis-cli> FLUSHALL

Esperamos que esta información sea de utilidad en la creación de tus bases de datos REDIS.

1. Lección 2: Persistencia, Strings y Estructuras Tipo Hash
   1. [Usando Comandos Básicos](https://learn.nextu.com/mod/lesson/view.php?id=10296&pid=P_WEB_DATABASE)

Se utiliza para juegos en línea

Ejemplo de juego con puntos y nombre de jugadores:

127.0.0.1:6379> SET jugador-1 "Luis;20"

OK

127.0.0.1:6379> SET jugador-2 "Ana;30"

OK

127.0.0.1:6379> GET jugador-1

"Luis;20"

127.0.0.1:6379> GET jugador-2

"Ana;30"

127.0.0.1:6379>

127.0.0.1:6379> SET jugador-1 "Luis;25"

OK

127.0.0.1:6379> GET jugador-1

"Luis;25"

127.0.0.1:6379> DEL jugador-2 🡪 Para borrar un clave en particular

(integer) 1

127.0.0.1:6379> GET jugador-2

(nil)

127.0.0.1:6379> GET jugador-1

"Luis;25"

127.0.0.1:6379>

127.0.0.1:6379> RENAME jugador-1 jugador-001 🡪 Para renombrar una clave

OK

127.0.0.1:6379> GET jugador-1

(nil)

127.0.0.1:6379> GET jugador-001

"Luis;25"

127.0.0.1:6379> SET tormenta-activa "true" EX 10 🡪 Para establecer Tiempo de expiración en segundos, en este caso 10 segundos

OK

127.0.0.1:6379> GET tormenta-activa

(nil)

127.0.0.1:6379> SET tormenta-activa "true" EX 10

OK

127.0.0.1:6379> GET tormenta-activa

"true"

127.0.0.1:6379> SET tormenta-activa "true" EX 100

OK

127.0.0.1:6379> TTL tormenta-activa 🡪 Para ver el tiempo que falta en segundos de una clave con tiempo de expiración

(integer) 95

127.0.0.1:6379> TTL tormenta-activa

(integer) 88

127.0.0.1:6379>

* 1. Actividad interactiva 1

**Prueba tus conocimientos**

Unos profesores de la Universidad del Futuro están creando una aplicación para monitorear el ritmo cardíaco de las personas a medida que realizan actividad física. Estos valores se van actualizando cada cierto tiempo. Existe una variable, “cambio-intensidad”, que indica cuándo las personas deben cambiar la intensidad del ejercicio.

**¿Cuál es la secuencia correcta de instrucciones Redis para realizar las siguientes tareas?**

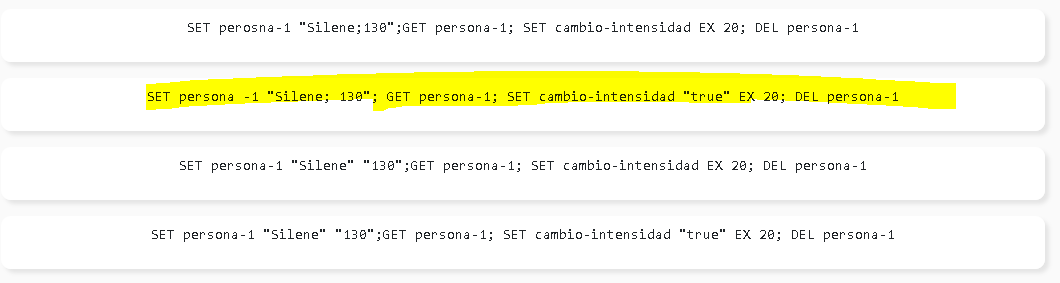
1. Crear un objeto con valor “Silene; 130” cuya clave es persona-1
2. Verificar el valor de persona-1
3. Crear la variable cambio-intensidad con un tiempo de vida de 20 segundos
4. Borrar a la persona-1 porque no continuó en la actividad física ya que se sintió mal.

SET persona-1 “Silene; 130”

GET persona-1

SET cambio-intensidad “true” EX 20

DEL persona-1



* 1. Explorando Tipos de Datos

Strings, Listas, Hash, Conjuntos.

MSET 🡪 Para asignar múltiples valores

Para redis todos los valores son de tipo string sin embargo los valores numéricos si se pueden incrementar.

Comandos de Strings:

127.0.0.1:6379> MSET contador 1 saludo Hola

OK

127.0.0.1:6379> GET contador

"1"

127.0.0.1:6379> INCR contador 🡪 Para incrementar un valor

(integer) 2

127.0.0.1:6379> GET contador

"2"

127.0.0.1:6379> APPEND saludo " Mundo" 🡪 Agregar string a un string

(integer) 10

127.0.0.1:6379> GET saludo

"Hola Mundo"

127.0.0.1:6379> TYPE saludo 🡪 Para ver tipo de variable

string

127.0.0.1:6379> TYPE contador

string

Comandos de Listas:

127.0.0.1:6379> RPUSH lista-1 "a" "b"

(integer) 2

127.0.0.1:6379> LRANGE lista-1 0 10

1) "a"

2) "b"

127.0.0.1:6379> RPUSH lista-1 "c"

(integer) 3

127.0.0.1:6379> LRANGE lista 0 10

(empty list or set)

127.0.0.1:6379> LRANGE lista-1 0 10

1) "a"

2) "b"

3) "c"

127.0.0.1:6379> LPUSH lista-1 "d"

(integer) 4

127.0.0.1:6379> LRANGE lista-1 0 10

1) "d"

2) "a"

3) "b"

4) "c"

127.0.0.1:6379> LPOP lista-1

"d"

127.0.0.1:6379> LRANGE lista-1 0 10

1) "a"

2) "b"

3) "c"

Comandos de HASH:

127.0.0.1:6379> HMSET usuario-1 nombre luis edad 20

OK

127.0.0.1:6379> HGET usuario-1 nombre

"luis"

127.0.0.1:6379> HINCRBY usuario-1 edad 2

(integer) 22

127.0.0.1:6379> HGET usuario-1 edad

"22"

127.0.0.1:6379>

Comandos de Conjuntos: la diferencia con los conjuntos es que si un nuevo valor es añadido, y es igual a uno de los valores dentro del conjunto, no se sumará ya que ya hay un valor.

127.0.0.1:6379> SADD colores "azul" "verde" "rojo"

(integer) 3

127.0.0.1:6379> SMEMBERS colores

1) "verde"

2) "azul"

3) "rojo"

127.0.0.1:6379> SISMEMBER colores "azul"

(integer) 1 🡪 Indica que es miembro del conjunto

127.0.0.1:6379> SISMEMBER colores "blanco"

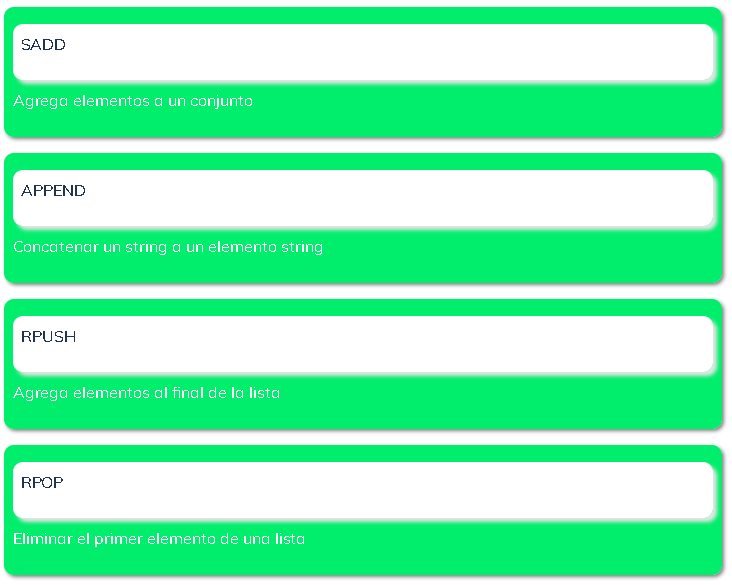
(integer) 0 🡪 Indica que no es miembro del conjunto

127.0.0.1:6379>

* 1. Taller 1
  2. Actividad interactiva 2

**Analiza y relaciona**

Haga la correspondencia de los comandos de la izquierda con su definición a la derecha.



* 1. ¿Sabías qué?

**Cápsula de conocimiento**

**Resumen de Comandos - Tipos de Datos**

El propósito de este recurso es que tengas una guía rápida de los comandos de Redis y su sintaxis.

* **Crear una entrada en la base de datos con tiempo de expiración:**

redis-cli> SET <clave> <valor> EX <segundos>

* **Eliminar una entrada en la base de datos:**

redis-cli>DEL <clave>

* **Verificar tiempo de expiración:**

redis-cli> TTL <clave>

* **Asignar múltiples entradas:**

redis-cli> MSET <clave-1> <valor-1> <clave-2> <valor-2> ...

* **Verificar el tipo:**

redis-cli> TYPE <clave>

* **Incrementar:**

redis-cli> INC <clave>

* **Asignar una lista por el inicio o izquierda:**

redis-cli> LPUSH <clave> <valor-1> <valor-2>

* **Asignar una lista por el final o derecha:**

redis-cli> RPUSH <clave> <valor-1> <valor-2>

* **Ver valores de la lista:**

redis-cli> LRANGE <clave> <indice-inferior> <indice-superior>

* **Asignar un hash:**

redis-cli> HMSET <clave> <nombre-campo1> <valor-campo1> <nombre-campo2> <valor-campo2> ...

* **Ver valores de un hash:**

redis-cli> HGET <clave> <nombre-campo>

* **Asignar un Conjunto (SET):**

redis-cli> SADD <clave> <valor-1> <valor-2> ...

* **Ver si pertenece al conjunto:**

redis-cli> SISMEMBER <clave> <valor>

* **Ver elementos del conjunto:**

redis-cli> SMEMBERS <clave>

Esperamos que esta lista de comandos te resulte útil al interactuar con el manejador Redis.

* 1. Ejercicio práctico

1. Lección 3: Aplicando Seguridad, y Publicando y Suscribiendo con Redis
   1. [Autenticando en REDIS](https://learn.nextu.com/mod/lesson/view.php?id=10297&pid=P_WEB_DATABASE)

El objetivo de Redis es almacenar variables en memoria para mejorar el rendimiento.

127.0.0.1:6379> SET A 123

OK

127.0.0.1:6379> SET B 456

OK

127.0.0.1:6379> GET A

"123"

127.0.0.1:6379> GET B

"456"

127.0.0.1:6379> INFO 🡪 Comando para obtener información de Redis

# Server

redis\_version:3.2.100

redis\_git\_sha1:00000000

redis\_git\_dirty:0

redis\_build\_id:dd26f1f93c5130ee

redis\_mode:standalone

os:Windows

arch\_bits:64

multiplexing\_api:WinSock\_IOCP

process\_id:10864

run\_id:02426307ef3b4cc550b7025e55672442f1872d0a

tcp\_port:6379

uptime\_in\_seconds:18474

uptime\_in\_days:0

hz:10

lru\_clock:11974100

executable:C:\Users\jrodrigue253\Desktop\Redis\redis-server.exe

config\_file:

# Clients

connected\_clients:1

client\_longest\_output\_list:0

client\_biggest\_input\_buf:0

blocked\_clients:0

# Memory

used\_memory:689264

used\_memory\_human:673.11K

used\_memory\_rss:651488

used\_memory\_rss\_human:636.22K

used\_memory\_peak:690056

used\_memory\_peak\_human:673.88K

total\_system\_memory:0

total\_system\_memory\_human:0B

used\_memory\_lua:37888

used\_memory\_lua\_human:37.00K

maxmemory:0

maxmemory\_human:0B

maxmemory\_policy:noeviction

mem\_fragmentation\_ratio:0.95

mem\_allocator:jemalloc-3.6.0

# Persistence

loading:0

rdb\_changes\_since\_last\_save:7

rdb\_bgsave\_in\_progress:0

rdb\_last\_save\_time:1622586764

rdb\_last\_bgsave\_status:ok

rdb\_last\_bgsave\_time\_sec:0

rdb\_current\_bgsave\_time\_sec:-1

aof\_enabled:0

aof\_rewrite\_in\_progress:0

aof\_rewrite\_scheduled:0

aof\_last\_rewrite\_time\_sec:-1

aof\_current\_rewrite\_time\_sec:-1

aof\_last\_bgrewrite\_status:ok

aof\_last\_write\_status:ok

# Stats

total\_connections\_received:1

total\_commands\_processed:52

instantaneous\_ops\_per\_sec:0

total\_net\_input\_bytes:2275

total\_net\_output\_bytes:5892638

instantaneous\_input\_kbps:0.00

instantaneous\_output\_kbps:0.00

rejected\_connections:0

sync\_full:0

sync\_partial\_ok:0

sync\_partial\_err:0

expired\_keys:0

evicted\_keys:0

keyspace\_hits:26

keyspace\_misses:2

pubsub\_channels:0

pubsub\_patterns:0

latest\_fork\_usec:316863

migrate\_cached\_sockets:0

# Replication

role:master

connected\_slaves:0

master\_repl\_offset:0

repl\_backlog\_active:0

repl\_backlog\_size:1048576

repl\_backlog\_first\_byte\_offset:0

repl\_backlog\_histlen:0

# CPU

used\_cpu\_sys:0.22

used\_cpu\_user:0.06

used\_cpu\_sys\_children:0.00

used\_cpu\_user\_children:0.00

# Cluster

cluster\_enabled:0

# Keyspace

db0:keys=2,expires=0,avg\_ttl=0

127.0.0.1:6379>

127.0.0.1:6379> CONFIG GET requirepass 🡪 Comando para ver si ha auth

1) "requirepass"

2) ""

127.0.0.1:6379>

127.0.0.1:6379>

127.0.0.1:6379> CONFIG SET requirepass abcdef 🡪 Set auth

OK

127.0.0.1:6379> GET A

(error) NOAUTH Authentication required.

127.0.0.1:6379> SET C 123456

(error) NOAUTH Authentication required.

127.0.0.1:6379>

127.0.0.1:6379>

127.0.0.1:6379> AUTH abcdefg 🡪 Command to authenticate

(error) ERR invalid password

127.0.0.1:6379> AUTH abcdef

OK

127.0.0.1:6379> GET A

"123"

127.0.0.1:6379> SET C 123456

OK

127.0.0.1:6379>

127.0.0.1:6379> GET A

"123"

127.0.0.1:6379> GET B

"456"

127.0.0.1:6379> GET C

"123456"

127.0.0.1:6379> CONFIG SET requirepass "" 🡪 Delete Auth

OK

127.0.0.1:6379> GET A

"123"

127.0.0.1:6379>

* 1. Actividad interactiva 1

**Prueba tus conocimientos**

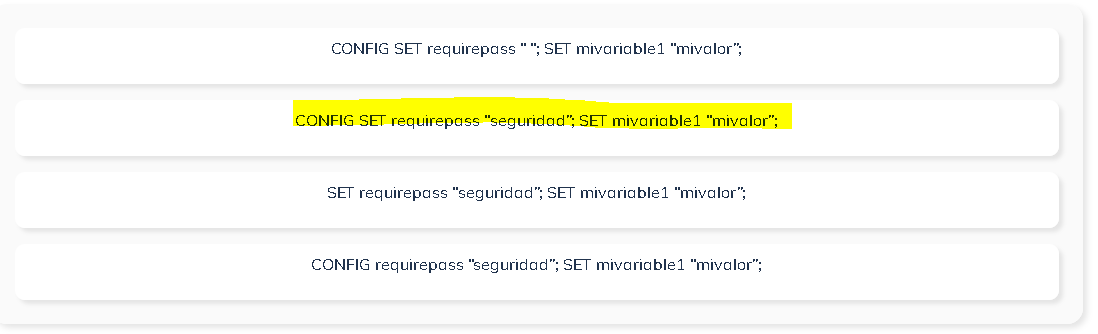
Supongamos que el usuario Luis se ha conectado a la consola redis-cli sin usar autenticación. ¿Cuáles son los comandos que debe ejecutar para realizar las siguientes tareas?

* Activa la autenticación usando “requirepass” y la contraseña “seguridad”.
* Asigna un valor a “mivariable1” de “mivalor”

CONFIG SET requirepass seguridad

AUTH seguridad

SET mivariable1 “mivalor”



* 1. Taller
  2. Respaldando y Recuperando Datos en REDIS

Persistencia en Redis es como se guardan los datos

Si se detiene el Servidor Redis este guardará todos los datos de memoria dentro de un dump, y cuando se inicie el servidor Redis va a utilizar este archivo para recuperar todas las variables.

Para saber dónde se almacenarán estos:

127.0.0.1:6379> CONFIG GET dir 🡪 Para saber el directorio

1) "dir"

2) "C:\\Users\\jrodrigue253\\Desktop\\Redis"

127.0.0.1:6379>

127.0.0.1:6379> CONFIG GET dbfilename 🡪 Para saber el nombre del dump

1) "dbfilename"

2) "dump.rdb"

127.0.0.1:6379>

Para guardar los datos se utiliza el comando:

127.0.0.1:6379> SAVE 🡪 Comando para guadar los datos en el archivo dump

OK

127.0.0.1:6379>

Para revisar si el dump existe:

C:\Users\jrodrigue253\Desktop\Redis>dir dump.rdb

Volume in drive C is SOE

Volume Serial Number is BA32-D3D1

Directory of C:\Users\jrodrigue253\Desktop\Redis

06/01/2021 04:57 PM 117 dump.rdb

1 File(s) 117 bytes

0 Dir(s) 54,835,269,632 bytes free

C:\Users\jrodrigue253\Desktop\Redis>

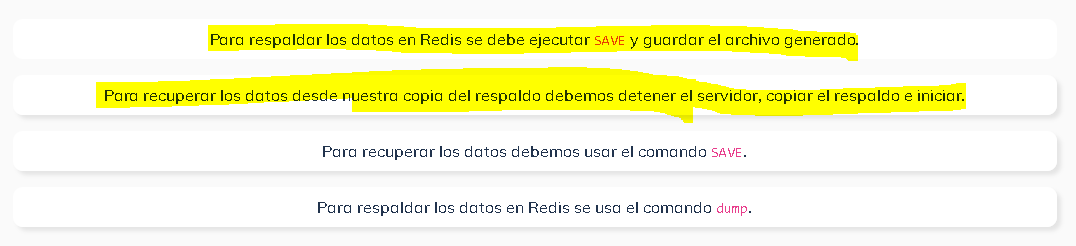
Para respaldar guardamos el archivo dump con una marca, por ejemplo nombre-fecha-hora

Antes de recuperar los datos se debe detener el servidor de Redis.

* 1. Actividad interactiva 2

**Prueba tus conocimientos**

¿Cuál de las siguientes opciones aplica en el caso de las actividades de respaldo y recuperación en Redis? Señala todas las que apliquen.



* 1. ¿Sabías qué?

**Cápsula de conocimiento**

**Resumen de Comandos - Autenticación y Respaldo**

El objetivo de este recurso es proporcionarte una referencia rápida a los comandos de Redis usados para la autenticación y el respaldo.

* **Obtener la contraseña de autenticación:**

redis-cli> CONFIG GET requirepass

* **Definir la contraseña de autenticación:**

redis-cli> CONFIG SET requirepass <contraseña>

* **Autenticar:**

redis-cli> AUTH <contraseña>

* **Obtener el directorio de respaldo:**

redis-cli> CONFIG GET dir

* **Definir el directorio de respaldo:**

redis-cli> CONFIG SET dir <ruta>

* **Obtener el nombre del archivo para el respaldo:**

redis-cli> CONFIG GET dbfilename

* **Definir el nombre del archivo para el respaldo:**

redis-cli> CONFIG SET dbfilename <nombre>

* **Guardar los datos:** El generará un archivo según la configuración y los datos serán recuperados desde el mismo sitio.

redis-cli> SAVE

Esperamos que este resumen te sirva como una referencia rápida cuando trabajes con la autenticación y respaldo en Redis.

5. UNIDAD 3: Prueba

# Bases de Datos No Relacionales > UNIDAD 3: Prueba

|  |  |
| --- | --- |
| **Comenzado en** | Tuesday, 1 de June de 2021, 17:07 |
| **Estado** | Terminados |
| **Finalizado en** | Tuesday, 1 de June de 2021, 17:16 |
| **Tiempo empleado** | 8 mins 15 segundos |
| **Puntos** | 10/10 |
| **Calificación** | **100** de un total de 100 |

Top of Form

### Pregunta 1

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Haga la correspondencia de los comandos de la izquierda con su definición a la derecha.

|  |  |
| --- | --- |
| MSET | Respuesta 1 |

Bottom of Form

|  |
| --- |
|  |
| INCR | Respuesta 2 |

|  |
| --- |
|  |
| LRANGE | Respuesta 3 |

|  |
| --- |
|  |
| HINCRBY | Respuesta 4 |

|  |
| --- |
|  |

#### Retroalimentación

La respuesta correcta es: MSET → asigna múltiples valores al mismo tiempo, INCR → incrementa un valor, LRANGE → lista los elementos de una lista, HINCRBY → incrementa el valor un campo dado de acuerdo al incremento

### Pregunta 2

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuál es la diferencia entre el tipo de datos hash y el tipo de datos conjunto? Señala todas las que apliquen.

Seleccione una o más de una:

a. Un valor del tipo hash se define usando el comando HMSET y el tipo conjunto con el SADD.

b. El tipo de datos hash no permite que dos valores este repetidos y el tipo de datos conjunto si lo permite.

c. El tipo de datos hash permite valores repetidos y el tipo de datos conjunto no lo permite.

d. El valor de un tipo hash se consulta con GET y el tipo de datos conjunto con SGET.

#### Retroalimentación

El tipo de datos hash permite la creación de valores complejos en Redis y se define usando el comando HMSET , los valores pueden ser repetidos. Por el contrario, un valor del tipo conjunto se crea con el comando SADD y no permite valores repetidos.

Las respuestas correctas son: Un valor del tipo hash se define usando el comando HMSET y el tipo conjunto con el SADD., El tipo de datos hash permite valores repetidos y el tipo de datos conjunto no lo permite.

### Pregunta 3

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuál (es) de la (s) siguientes NO es una característica del paradigma clave valor?

Seleccione una:

a. Cada objeto, estructura o documento es representado con una clave y un valor

b. La clave se utiliza para acceder al objeto, estructura o documento

c. Un objeto, estructura o documento puede existir sin una clave

d. Cada objeto, estructura o documento se representa con la siguiente sintaxis clave : valor

#### Retroalimentación

Un objeto, estructura o documento necesita un clave que permita accederlo.

La respuesta correcta es: Un objeto, estructura o documento puede existir sin una clave

### Pregunta 4

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuál (es) de la (s) siguientes NO es una característica de REDIS?

Seleccione una o más de una:

a. Es open source

b. No permite replicación de los datos

c. Es un manejador de base de datos NoSQL orientado a documentos JSON

d. Está escrito en el lenguaje de programación  Erlang

#### Retroalimentación

REDIS es un manejador de base de datos NoSQL orientados a clave/valor que es open source y está escrito en el lenguaje de programación ANSI C, el cual permite la replicación de los datos en otros nodos dentro de un cluster.

Las respuestas correctas son: No permite replicación de los datos, Es un manejador de base de datos NoSQL orientado a documentos JSON, Está escrito en el lenguaje de programación  Erlang

### Pregunta 5

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Siguiendo el principio de “In Memory” de REDIS, supongamos que cargamos un valor en la base de datos SET curso-1 “Curso de Base de Datos”, luego de realizar esto, nos salimos de la consola REDIS. A la media hora volvemos a entrar. ¿ Cuál será la salida de ejecutar GET Curso-1?

Seleccione una:

a. >PONG

b. >OK

c. >“Curso de Base de Datos”

d. >nil

#### Retroalimentación

El principio de “In Memory” de REDIS sugiere que los datos están en memoria, por  lo tanto luego de salir y volver entrar se podrá todavía recuperar el valor asociado a la clave curso-1.

La respuesta correcta es: >“Curso de Base de Datos”

### Pregunta 6

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuál es la sintaxis del comando para activar la autenticacion en REDIS usando la clave “seguridad”?

Seleccione una:

a. CONFIG SET  requirepass “seguridad”

b. CONFIG GET requirepass “seguridad”

c. CONFIG PING requirepass “seguridad”

d. CONFIG HMSET requirepass “seguridad”

#### Retroalimentación

El comando correcto para activar la seguridad con una clave “seguridad” es CONFIG SET requirepass “seguridad”

La respuesta correcta es: CONFIG SET  requirepass “seguridad”

### Pregunta 7

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

La persistencia en Redis significa que \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Seleccione una:

a. Los datos permanecerán en memoria hasta que expire su tiempo de vida o se eliminen con una instrucción.

b. Los datos permanecerán en memoria hasta que se ejecute un comando de SAVE.

c. Los datos permanecerán en memoria hasta que nos salgamos del sistema usando el comando EXIT.

d. Los datos se guardan en un archivo cada cierto tiempo y para obtenerlos debemos usar el comando GET.

#### Retroalimentación

En REDIS los datos permanecen en memoria hasta que finalice su tiempo de vida (TTL) o se borran intencionalmente.

La respuesta correcta es: Los datos permanecerán en memoria hasta que expire su tiempo de vida o se eliminen con una instrucción.

### Pregunta 8

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Haga la correspondencia de los comandos en la izquierda con la salida que se despliega en la consola en la derecha.

|  |  |  |
| --- | --- | --- |
| “1” | Respuesta 1 | |
|  | |
| “19” | | Respuesta 2 | |

|  |
| --- |
|  |
| [“1”, “2”] | Respuesta 3 |

|  |
| --- |
|  |
| “casa” “edificio” “town house” | Respuesta 4 |

|  |
| --- |
|  |

#### Retroalimentación

La respuesta correcta es: “1” → SET dia 1; GET dia, “19” → HMSET estudiante-1 nombre "Victoria" edad "19" ; HGET estudiante-1 edad, [“1”, “2”] → RPUSH numeros-1 "1" "2"; LRANGE numeros-1 0 10, “casa” “edificio” “town house” → SADD viviendas "casa" "edificio" "town house"; SISMEMBER viviendas

### Pregunta 9

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Para recuperar los datos en REDIS debemos realizar lo siguiente \_\_\_\_\_\_\_\_\_\_\_\_\_\_  (sin importar el orden de las actividades). Señala todas las que apliquen.

Seleccione una o más de una:

a. Copiar el archivo dump.dbp en otro directorio y renombrarlo.

b. Parar el servidor Redis.

c. Sobreescribir el archivo dump.dbp con nuestro archivo de respaldo

d. Iniciar el servidor Redis.

#### Retroalimentación

Para recuperar los datos lo que debemos copiar el archivo, donde respaldamos la información, en el directorio predestinado de REDIS donde se dejan los respaldos. De esta forma cuando se inicie el servidor, se colocarán los datos de dicho respaldo en memoria.

Las respuestas correctas son: Copiar el archivo dump.dbp en otro directorio y renombrarlo., Parar el servidor Redis., Sobreescribir el archivo dump.dbp con nuestro archivo de respaldo, Iniciar el servidor Redis.

### Pregunta 10

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuál es la manera más común de interactuar con REDIS?

Seleccione una:

a. Usando redis-cli

b. A través de un lenguaje de programación usando el conector correspondiente

c. A través de la interfaz gráfica oficial de Redis

d. A través de un PING

#### Retroalimentación

Realmente los clientes que se utilizan para interaccionar con REDIS son los lenguajes de programación, a través de los conectores.

La respuesta correcta es: A través de un lenguaje de programación usando el conector correspondiente