**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
* Esta 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:

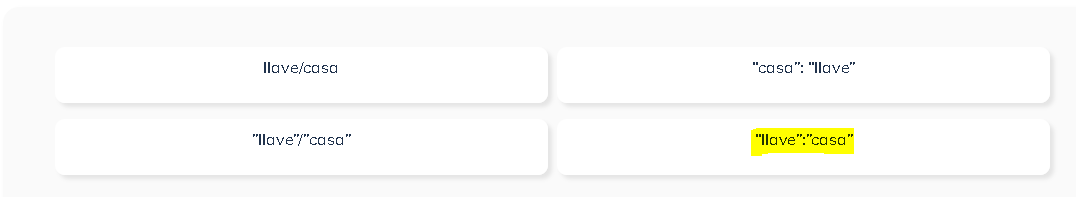
* Maneja la persistencia – en algunas versiones de Redis se pueden guardar los datos en memoria virtual, pero la idea de este manejador es que todos los datos estén en memoria RAM, permitiendo un rápido acceso a ellos.
* Redis tiene un buen rendimiento
* Soporta la replicación por cluster, pudiéndose 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**

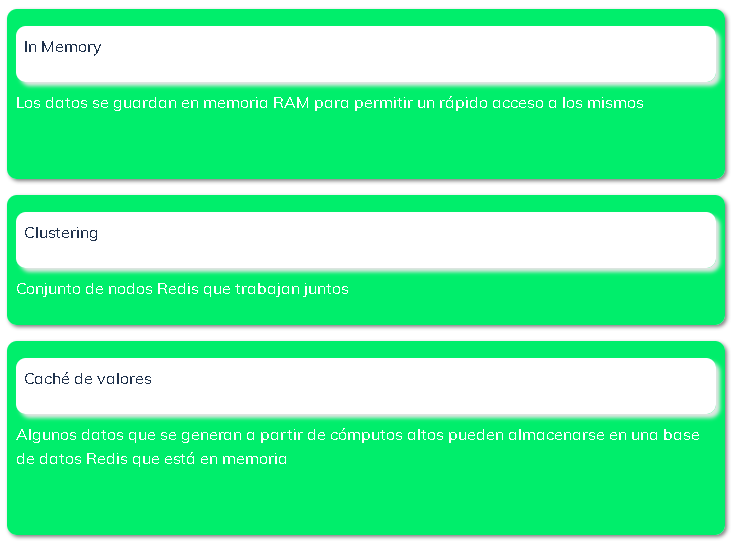
¿Cuál sería la forma de representar el objeto de tipo string llamado “casa” cuya clave es “llave” en el formato clave/valor?



* 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?**

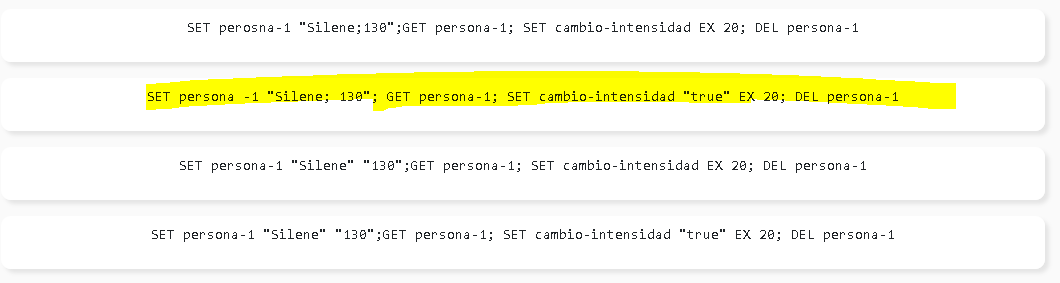
1. Crear un objeto con valor “Silene; 130” cuya clave es persona-1
2. Verificar el valor de persona-1
3. Crear la variable cambio-intensidad con un tiempo de vida de 20 segundos
4. Borrar a la persona-1 porque no continuó en la actividad física ya que se sintió mal.

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

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"

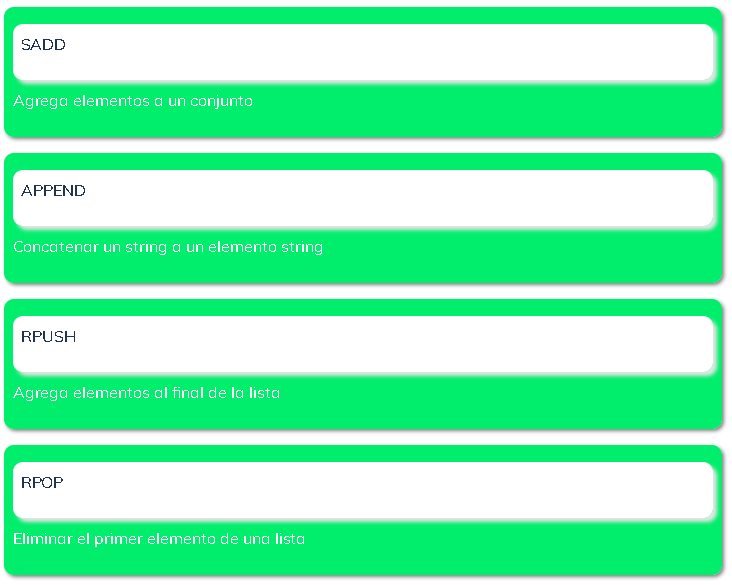
(integer) 0 🡪 Indica que no es miembro del conjunto

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)
   2. Actividad interactiva 1
   3. Taller
   4. Respaldando y Recuperando Datos en REDIS
   5. Actividad interactiva 2
   6. ¿Sabías qué?

5. UNIDAD 3: Prueba