

# Redis

Ruben Gonzalez, *ITCR*, Rodrigo Acevedo, *ITCR*, Uriel Vargas, *ITCR*, and Edison Gonzalez, *ITCR*

**Abstract**—Redis que significa (Remote Dictionary Server) es un proyecto de estructura de datos en memoria de código abierto que implementa una base de datos distribuida de clave-valor en memoria con durabilidad opcional. Redis admite diferentes tipos de estructuras de datos abstractas, como cadenas, listas, mapas, conjuntos, conjuntos ordenados, hiperloglogs, mapas de bits, flujos e índices espaciales.

**Index Terms**—Redis, NoSQL

## I. ARQUITECTURA

**R**edis es una base de datos NoSQL (nuevo tipo de almacenes de datos no relacionales). Las bases de datos no relacionales no siguen esquemas y suelen utilizar un almacenamiento del tipo clave-valor y orientado a documentos. Estas bases de datos no garantizan completamente todas las características ACID (atomicidad, coherencia, aislamiento y durabilidad). En compensación, estos sistemas son muy escalables horizontalmente, manejan enormes cantidades de datos y no generan cuellos de botella en los sistemas.

Redis utiliza un almacenamiento en memoria. Esto permite que las operaciones de lectura y escritura se realicen de una forma asombrosamente rápida. Redis es capaz de soportar cientos de miles de transacciones por segundo, aportando un rendimiento altísimo. A pesar de ser una base de datos en memoria, Redis permite el almacenamiento persistente.

El sistema Redis cuenta con características importantes tales como:

### A. Publicación-Suscripción

Que implementan el paradigma de mensajería de publicación-suscripción donde los publicadores no están programados para enviar sus mensajes a suscriptores específicos. Más bien, los mensajes publicados se publican en canales, sin el conocimiento de cuantos suscriptores existen. Los suscriptores expresan interés en uno o más canales, y solo reciben mensajes que son de interés. Este desacoplamiento de editores y suscriptores puede permitir una mayor escalabilidad y una topología de red más dinámica.

### B. Replicación master/slave

Cuando las instancias maestra y esclava están bien conectadas, la maestra mantiene actualizado al esclavo enviando una secuencia de comandos al esclavo.

Cuando el enlace entre el maestro y el esclavo se rompe, por problemas de red o porque se detecta un tiempo de espera en el maestro o el esclavo, el esclavo se vuelve a conectar e intenta proceder con una resincronización parcial.

Cuando no se puede realizar una resincronización parcial, el esclavo solicitará una resincronización completa, en el que el maestro necesita crear una instantánea de todos sus datos, enviarlo al esclavo y luego continuar enviando la secuencia de comandos a medida que cambia el conjunto de datos.



Fig. 1. Logo de Redis

### C. Persistencia de disco

Por defecto, Redis guarda las instantáneas o “snapshots” del conjunto de datos del disco, en un archivo binario llamado `dump.rdb`. Se puede configurar Redis para que guarde el conjunto de datos cada N segundos si hay al menos M cambios en el conjunto de datos, o puede llamar manualmente los comandos `SAVE` o `BGSAVE`.

## II. HISTORIA

El proyecto desarrollado Redis (significa REMOTE DICTIONARY SERVER) se inició a principios de 2009, su principal desarrollador llamado Salvatore Sanfilippo de nacionalidad italiana. Redis inicialmente quería mejorar el rendimiento de LLOOGG, un producto de análisis web en tiempo real. [5]

En junio de 2009, Redis logra ser suficientemente estable y contaba con un conjunto de funciones básicas para proveer ese rendimiento (lo que necesitaba era un DB que manejara escrituras rápidas y una operación rápida de “obtener los últimos N elementos”) que buscaba para el servicio LLOOGG. El 19 de junio de 2009, se alcanzó un hito importante: Salvatore implementó Redis en el entorno de producción de LLOOGG y retiró la antigua instalación de MySQL.

Originalmente se inició Redis para escalar LLOOGG. Pero después de hacer funcionar el servidor básico, a Salvatore le gustó la idea de compartir el trabajo con otras personas, y Redis se convirtió en un proyecto de código abierto.

Durante los siguientes meses, Redis ganó popularidad rápidamente. Salvatore fomenta una gran comunidad, agregó funciones a un ritmo muy rápido y se ocupó de todos y cada uno de los informes de corrupción de bases de datos, inestabilidad, etc; con la mayor severidad.

En marzo de 2010, VMWare contrató a Salvatore para trabajar a tiempo completo en Redis. Poco después, VMWare contrató a Pieter Noordhuis, un colaborador clave de Redis, para darle un impulso adicional al proyecto. VMWare permitió que Redis siguiera siendo de código abierto. En 2013 y posteriormente en 2015, Salvatore Sanfilippo se unió a Pivotal Software y RedisLabs, respectivamente, y estas dos compañías se convirtieron en el patrocinio principal de Redis. [2]

### III. EMPRESAS QUE LO UTILIZAN

#### A. Twitter

Es una famosa red social creada originalmente en California. La red permite enviar mensajes de texto plano de corta longitud junto con otros archivos multimedia. El sistema de base de datos Redis es usado para almacenar multimedia e información de usuarios. 2



Fig. 2. Logo de Twitter

#### B. GitHub

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora. El software que opera GitHub fue escrito en Ruby on Rails. Actualmente GitHub es propiedad de Microsoft. <https://es.wikipedia.org/wiki/GitHub> El sistema de base de datos Redis es usado para almacenar repositorios y archivos de estos. 3



Fig. 3. Logo de GitHub

#### C. Pinterest

Es una plataforma o red social para compartir imágenes que permite a los usuarios crear y administrar, en tableros personales temáticos, colecciones de imágenes, como eventos, intereses, hobbies y mucho más. Permite a los usuarios, guardar y clasificar por categorías, imágenes en diferentes tableros. Pueden también seguir a otros usuarios con los mismos gustos e intereses.

El sistema de base de datos Redis es usado para el almacenamiento de los archivos multimedia de la aplicación además de información de usuarios y de fotos. 4



Fig. 4. Logo de Pinterest

#### D. Snapchat

Es una aplicación de mensajería para el teléfono inteligente con soporte multimedia de imagen, video y filtros de realidad aumentada. Su mayor seña de identidad es la mensajería efímera, donde las imágenes y mensajes pueden ser accesibles solo durante un tiempo determinado elegido por los usuarios. Fue creada por Evan Spiegel, Bobby Murphy y Reggie Brown.

Snapchat se encuentra entre las aplicaciones más utilizadas a nivel mundial junto a Facebook, WhatsApp, Twitter e Instagram. <https://es.wikipedia.org/wiki/Snapchat> El sistema de base de datos Redis es usado para el almacenamiento de los archivos multimedia de la aplicación tales como fotos y videos. 5



Fig. 5. Logo de Snapchat

#### E. Craigslist

Craigslist es un sitio web de anuncios clasificados con secciones dedicadas al empleo, vivienda, contactos personales, ventas, ítems, servicios, comunidad, conciertos, hojas de vida, y foros de discusión, entre otras. Fundada en 1995 por Craig Newmark, actualmente EBay es propiedad de Craigslist.

El sistema de base de datos Redis es usado para el almacenamiento de publicaciones y multimedia además de información de usuarios. 6



Fig. 6. Logo de Craigslist

#### F. Stack Overflow

Es un sitio web utilizado por una comunidad de desarrolladores informáticos, en el cual los usuarios hacen preguntas sobre distintos temas relacionados a la computación y otros usuarios conocedores del tema o profesionales responden sobre estas preguntas, creando una comunidad donde todos se ayudan. El sistema de base de datos Redis es usado para el almacenamiento de publicaciones de preguntas además de datos como códigos o información de usuario. 7



Fig. 7. Logo de Stack Overflow

### IV. VENTAJAS Y DESVENTAJAS

#### A. Ventajas

1) *Velocidad*: Guarda todos los datos en cache, por lo que cuando el microprocesador le pide un dato, lo busca en el

cache sin tocar el disco duro lo cual acelera el proceso de búsqueda. Así también en la inserción de muchos en el cache en tan poco tiempo.

2) *API de Redis*: Redis tiene API de cliente desarrolladas en todos los lenguajes populares como C, Ruby, Java, JavaScript y Python. También puede guardar los datos en formato JSON. [3]

3) *Representación compacta*: Puede guardar la información en una representación muy compacta de single-point-in-time en un archivo. Permite almacenar pares de clave y de valor tan grandes como de 512 MB: Puede tener enormes claves y valores de objetos tan grandes como 512 MB, lo que significa que Redis admitirá hasta 1 GB de datos para una sola entrada.

4) *Copias de seguridad*: Los archivos RDB (Redis DataBase file) se utilizan para generar copias de seguridad debido a su rapidez para recuperar los datos y en la creación de múltiples instancias para la múltiple restauración de versiones de la base de datos en caso de desastres. Gracias a esto puede con solo un archivo transferir toda la copia de una base de datos a lugares lejanos.

5) *Redis Hashing*: Redis usa su propio mecanismo de hashing llamado Redis Hashing. Redis almacena datos en forma de clave y mapa, es decir, campos de cadena y valores de cadena.

6) *Replicación de datos*: La replicación es el proceso de configurar nodos de caché maestro-esclavo. Los nodos esclavos siempre escuchan el nodo maestro, lo que significa que cuando se actualiza el nodo maestro, los esclavos también se actualizarán automáticamente. Redis también puede actualizar esclavos de forma asíncrona.

7) *Soporte de transacciones*: Redis admite transacciones, lo que significa que los comandos se pueden ejecutar como una cola en lugar de ejecutar uno a la vez. Normalmente, los comandos después de MULTI se agregaran a una cola y una vez que se ejecute EXEC, todos los comandos guardados en la cola se ejecutarán a la vez. [4]

## B. Desventajas

1) *No capaz de guardar mucha información*: No puede guardar tablas enteras ni muchos datos relacionados de una base de datos.

2) *Posibilidad de pérdida de datos*: Los archivos RDB no son buenos para minimizar la posibilidad de pérdida de datos en caso de que Redis deje funcionar por algún factor como corte de energía esto porque las instancias de RGB crear un snapshot cada 5 minutos por lo cual si ocurre algún problema se perderá los datos del tiempo de un snapshot hacia otro snapshot.

3) *Uso de forks*: El RGB necesita hacer forks para persistir en el disco utilizando un proceso secundario. El problema es que hacer forks puede tardar mucho tiempo si el conjunto de datos es muy grande si sucede este caso Redis dejará de funcionar a los clientes por milisegundos o segundos.

4) *Seguridad*: La seguridad de Redis no es muy buena ya que utiliza una seguridad muy básica hacia los niveles de instancias. Porque fue diseñada para ser utilizada en ambientes confiables con clientes confiables.

5) *Sin escalabilidad de instancias*: Las instancias únicas de Redis no son escalables esto porque solo corre en un solo núcleo del CPU en modo de un single-thread. Para obtener la escalabilidad muchas instancias de Redis deben ejecutarse, pero su distribución debe manejarla el desarrollador personalmente.

6) *Carece de rollback*: Cuando se esta haciendo una transacción Redis no permite hacer rollback de una transacción una vez esta se haya creado esto porque al hacer una transacción Redis garantiza que todos los comandos de la transacción no sean interrumpidos y que se ejecuten todos o ninguna. Por tal razón no permite rollback por lo tanto en caso de tener que hacer uno se debe hacer manual.

7) *Encriptación*: Redis no soporta encriptación de datos esto es una característica específica de los creadores porque no muchas personas necesitan soporte SSL y TLS (Transport Layer Security) así que recomiendan que se utilice solo en ambientes seguros y clientes confiables. Aun así ofrecen la utilidad de Spiped para crear encriptaciones simétricas entre sockets para una dirección a otra pero recomiendan utilizar otro servicio de encriptación.

## V. CASOS DONDE ES ÚTIL

- Para plataformas webs donde tendrán el acceso de miles de usuarios registrados por hora. Esto porque cada vez que un usuario visita la página se deben hacer consultas a la base de datos para conocer si el usuario con su cookie está registrado y autorizado para ver la página. Si se expande esto a miles de usuarios por hora y han estado dentro de la pagina varias veces se deberán hacer miles de consultas y procesar mucha información rápidamente. Redis puede realizar esta tarea gracias al uso del cache.
- Páginas web comerciales para el proceso rápido de listado de productos y actualización de este además de las miles de transacciones por minuto de usuarios en la compra y venta de productos esto porque en páginas comerciantes se necesita tiempos de respuesta rápida.
- Para uso de tablas de clasificación o de conteo ya que Redis trabaja muy bien con operaciones de incrementos y decrementos debido a la memoria cache.
- Para colas de mensajes ya que gracias al motor de almacenamiento de memoria de Redis en la creación de listas y establecimiento de operaciones para la creación y actualización de colas de mensajes.
- Para tiempo de series de datos porque utilizan una naturaleza secuencial donde se deben analizar datos, recolectarlos y ejecutar cálculos relacionados con estos sin sobrecargar la memoria y el tiempo de respuesta.
- Aplicaciones que utilizan información relacionada a localizaciones ya que se deben estar actualizando cada segundo ante miles de usuarios y por lo tal el tiempo de respuesta debe de ser de inmediato. [1]

## VI. ESTRUCTURA DE LOS DATOS

La diferencia con el sistema memcached (que también utiliza almacenamiento en memoria) es que Redis es un sistema de almacenamiento clave-valor. Esto quiere decir que los datos almacenados poseen estructura.[] Los diferentes tipos de estructuras de Redis son los siguientes:

### A. Cadenas de caracteres (Strings)

- Tipo de datos básico de las bases de datos clave-valor y el más simple de los soportados.
- Es una secuencia de bytes, lo que nos permite almacenar cualquier dato, como cadenas de texto, números, imágenes, vídeos, o un objeto serializado.



**Ruben Gonzalez** Nació en San José, Costa Rica en 1999. Actualmente, vive en Santo Domingo, Heredia. Se graduó en Kamuk School en 2016 y cursa por la mitad de la carrera de Ingeniería en Computación en el Tecnológico de Costa Rica desde el 2017.

### B. Conjuntos (Sets)

- Son colecciones de strings, sin ningún orden particular, en los que se garantiza que los elementos del mismo son únicos y no pueden estar duplicados. Redis garantiza que sólo existirá una vez, lo que nos permite poder añadir elementos sin tener que preocuparnos previamente de si ya existen o no.
- Las operaciones de añadir elementos al set, comprobar si un elemento existe, y eliminar elementos son muy rápidas.
- Redis permite además llevar a cabo operaciones de unión, intersección y diferencia entre los elementos de varios conjuntos.



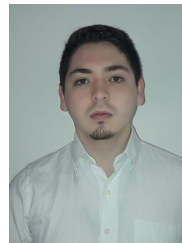
**Uriel Vargas** Nació en San José, Costa Rica en 1999. Actualmente, vive en Montes de Oca, San José. Se graduó en el Colegio Cedros en 2016 y cursa por la mitad de la carrera de Ingeniería en Computación en el Tecnológico de Costa Rica desde el 2017.

### C. Listas (Lists)

- Permiten almacenar secuencias de strings en el orden en el que fueron insertadas.
- En Redis las listas están optimizadas para qué añadir un elemento al inicio o al final de la lista tarde lo mismo independientemente de si la lista tiene 10 elementos, o 100 millones de elementos.

### D. Conjuntos ordenados (Sorted Sets)

- Son conjuntos agrupados con las propiedades de estos sets, que además permiten la ordenación de los elementos en base a un valor (score) numérico asociado a cada uno de ellos.
- La ordenación se lleva a cabo en el momento de añadir elementos al conjunto y no al solicitar los mismos, lo que hace que esta operación sea muy rápida.



**Edison Gonzalez** Nació en San José, Costa Rica en 1999. Actualmente, vive en Desamparados, San José. Se graduó en el Liceo de calle Fallas en 2016 y cursa por la mitad de la carrera de Ingeniería en Computación en el Tecnológico de Costa Rica desde el 2017.

### E. Tablas Hash (Hashes)

- Permite asociar a una clave una colección de pares clave-valor, habitualmente usado para representar objetos.
- Con un hash podemos almacenar cualquier entidad con sus atributos asociados (como una especie de tabla o diccionario), de forma que podamos recuperar únicamente el atributo de interés en cada momento.
- Redis almacena los hashes de forma muy eficiente, lo que optimiza el uso de memoria. [3]



**Rodrigo Acevedo** Rodrigo Acevedo Nació en San José, Costa Rica en 1995. Actualmente vive en el Carmen de Guadalupe, San José. Se graduó en el Centro Educativo Cristiano Reformado en 2012 y cursa los cursos finales de la carrera de Ingeniería en Computación en el Tecnológico de Costa Rica.

### REFERENCES

- [1] Top 5 redis use cases, Aug 2018.
- [2] Stefano Bernardi. Stefano bernardi, Aug 2017.
- [3] Carlos Galvan. Redis: Base de datos nosql. *Luis Peris*, Feb 2016.
- [4] Unni Mana. 10 advantages of redis - dzone database, Jan 2018.
- [5] Michael Russo. Redis, from the ground up.