Bases de datos 1

Dennis Patrick Juilland Prada - Ruth Lopez - Mauricio Henao - Johan Aparicio - Nicolas Trujillo

**Bases de datos no relacionales**

Diciembre 13 del 2022

línea horizontal

# Marcador de posición de imagen

# Introducción

Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto Escribe aquí tu texto.

## Historia

El desarrollo de Redis fue iniciado por Salvatore Sanfilippo a principios del 2009, para acelerar los tiempos de respuesta de un producto llamado LLOGG. Pronto fue ganando popularidad, hasta que en Marzo del 2012 la empresa VMWare contrató a Salvatore para trabajar a tiempo completo en Redis. Poco después VMWare contrató también a uno de los principales desarrolladores de Redis, Pieter Noordhuis. Desde entonces Redis ha evolucionado muy rápidamente, incluyendo funcionalidades que le hacen tan útil.

### ¿Qué es ?

La mejor definición que se me ocurre de Redis es que se trata de “un servidor de caché de memoria que conoce la estructura de los datos que alberga” o dicho de otra forma, “una caché de datos con estructura”.

La idea de servidores de caché de memoria en los que se alberga información para su rápido acceso desde distintos puntos es muy antigua, y existen productos muy consolidados en el mercado, como el famoso Memcached, utilizado por sitios con muchos millones de usuarios como Wikipedia, Twitter, Youtube, WordPress, etc.

Y si ya existen otros productos servidores de caché de memoria, ¿qué aporta Redis sobre ellos y por qué es tan revolucionario? El factor que diferencia a Redis respecto a una caché de memoria convencional es que conoce la estructura de los datos que alberga. Lo más fácil para entender el concepto es poner un ejemplo:

Imaginad que tenéis una aplicación web y queréis cachear en un servidor aparte (un servidor de memoria) la lista de los usuarios que están activos, para que pueda ser accedida desde diversas máquinas – todas las de un cluster:

Si queremos realizar esto con un producto que no sabe de estructuras de datos (como Memcached), cada vez que queremos modificar dicha lista de usuarios – por ejemplo para eliminar uno de ellos que ya no está activo - la tenemos que traer entera a uno de los servidores, modificarla y volverla a escribir entera. Esto es muy ineficiente, sobre todo cuando mayor es la lista.

Para realizar esto con Redis, tenemos funciones de la caché para albergar listas (entre otras estructuras de datos, sabe manejar listas), de forma que sólo tendríamos que indicar a Redis que elimine un determinado elemento de la lista. Más adelante indicaremos los tipos de estructuras de datos que maneja Redis.

### Cómo funciona

Con el auge de Internet, las aplicaciones modernas cada vez tienen que dar servicio a un mayor número de usuarios (imaginaros el número de peticiones por segundo que reciben los servidores de Facebook, Google o Twitter). Las arquitecturas tecnológicas que se utilizaban antiguamente para desarrollar aplicaciones no son suficientemente escalables como para soportar cientos de millones de usuarios, por lo que ha sido necesario re-inventar el concepto de "Arquitectura Tecnológica de Aplicaciones" para dar cabida a estas nuevas necesidades, apoyándose en productos innovadores que sustituyen a las antiguas arquitecturas Cliente-Servidor y las Bases de Datos Relacionales.

En febrero del 2011 estuve en el Spring I/O, y os conté como una de las charlas que me encantó se llamaba “Aplicaciones web ultra escalables con Redis”. Era uno de mis primeros contactos con la base de datos NoSQL (de la que también os hablé hace tiempo). Ha llovido mucho desde entonces, y de todos los nuevos productos con los que he trabajado en este tiempo hay uno que me parece especialmente revolucionario y visionario, REDIS.

### Características

Actualmente Redis soporta cinco tipos de estructuras de datos:

Cadenas de caracteres (Strings)

Listas (Lists)

Conjuntos (Sets)

Conjuntos Ordenados (Sorted Sets)

Tablas Hash (Hashes)

Aunque puede parecer una lista muy corta, sólo con estos cinco tipos de estructuras se pueden modelar infinitud de procesos y aplicaciones complejas, hasta el punto que muchos de los usuarios de Redis han terminado por sustituir completamente las bases de datos relacionales.

### Imagenes de código









