
	<p>UNIVERSIDAD DE PALERMO Laboratorio II</p>	
---	--	---

## Trabajo Práctico NoSQL

### Informe sobre Bases de Datos Clave-Valor

#### Año / Cuatrimestre

2022 - 2C

#### Curso / Código / Modalidad

Laboratorio II – 020093 – **Presencial**

#### Docentes

Juan Lopreiato

#### Grupo N.º

2

#### Alumnos

Andrés Isaac Biso - Legajo 0125044

Matias Hernan Coria - Legajo 0127111

Fidel Honorato Schinelli - Legajo 0115710

Angel Matias Bouin - Legajo 103352

Felipe Mattiazzi

Fecha de Presentación: 07/11/2022

Calificación: \_\_\_\_\_

## Índice

Índice	2
<b>Consignas</b>	<b>3</b>
<b>Introducción</b>	<b>4</b>
¿Cómo funcionan las bases de datos de clave-valor?	5
¿Cuáles son las características de una base de datos clave-valor?	6
¿Cuándo usar una base de datos de clave-valor?	7
<b>Ventajas</b>	<b>7</b>
<b>Desventajas</b>	<b>8</b>
<b>Ejemplos de uso reales</b>	<b>8</b>
<b>Usos de este tipo de Base de Datos</b>	<b>10</b>
<b>Motores de ejemplo (open source y/o pagos)</b>	<b>10</b>
<b>Diferencias con otras DDBB</b>	<b>11</b>
<b>Escalabilidad</b>	<b>12</b>
<b>Links</b>	<b>13</b>

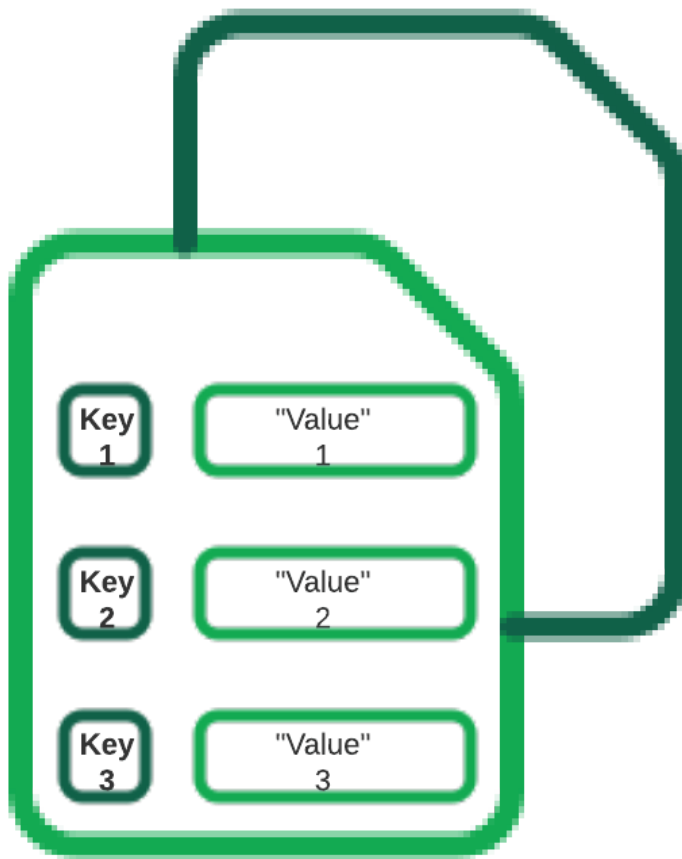
## Consignas

- Carátula
- Noción general del tema
- Ventajas
- Desventajas
- Para que se utiliza
- Ejemplos de uso reales
- Motores de ejemplo (open source y/o pagos)
- Diferencias con otras DDBB
- Escalabilidad
- Cualquier otro contenido con consideren apropiado

## Introducción

Las bases de datos de claves-valor, “key-value” - en inglés - o “store” clave-valor, son tipos de bases de datos en los que los datos se almacenan en un formato de “clave-valor” y se optimizan para leer y escribir esos datos. Los datos se obtienen mediante una o varias claves únicas para recuperar el o los valores asociados con cada clave; análogo a la estructura de datos conocida como diccionario o hash-table. Los valores almacenados pueden ser tipos de datos simples como cadenas y números u objetos complejos; y se leen y escriben en función de la clave para almacenar/recuperar su valor.

Las store de par clave-valor no son un concepto nuevo y han estado con nosotros por décadas. Uno de los stores conocidos es el antiguo Registro de Windows que permite que el sistema o las aplicaciones almacenen datos en una estructura de “clave-valor”, donde una clave se puede representar como un identificador único o una ruta única al valor.



### ¿Cómo funcionan las bases de datos de clave-valor?

Una base de datos de clave-valor asocia cada registro almacenado con una clave única que se utiliza para realizar un seguimiento del objeto. Las Bases de Datos Clave-Valor proporcionan a los desarrolladores una gran escalabilidad y flexibilidad para almacenar valores, con distintos tipos y tamaños.

En su forma más simple, un almacén de clave-valor es como un objeto de diccionario/array/map tal como existe en la mayoría de los paradigmas de programación, conformado de registros compuestos de un par clave-valor, la primera sirviendo como un identificador único para el registro, y la segunda siendo una

secuencia de bits continuos sin formato de longitud arbitraria. La diferencia siendo que las BBDD clave-valor se almacenan de forma persistente y son administradas por un Sistema de gestión de base de datos (DBMS).

Las bases de datos de clave-valor utilizan estructuras de índice compactas y eficientes para poder ubicar un valor de manera rápida y confiable por su clave, lo que las hace ideales para sistemas que necesitan poder encontrar y recuperar datos en tiempo constante. Redis, por ejemplo, es una base de datos clave-valor que está optimizada para rastrear estructuras de datos relativamente simples (tipos primitivos, listas, montones y mapas) en una base de datos persistente. Al admitir sólo una cantidad limitada de tipos de valores, Redis puede exponer una interfaz extremadamente simple para consultarlos y manipularlos, y cuando se configura de manera óptima, es capaz de un alto rendimiento.

### ¿Cuáles son las características de una base de datos clave-valor?

Una base de datos de clave-valor se define por el hecho de que permite que los programas o los usuarios de los programas recuperen datos de los registros mediante las claves que los identifican. Debido a que las bases de datos de valores clave tienen una definición muy general; no hay una lista global de características, pero hay algunas comunes:

- Recuperar un valor (si lo hay) almacenado y asociado con una clave dada
- Eliminar el valor (si lo hay) almacenado y asociado con una clave dada
- Establecer, actualizar y reemplazar el valor (si lo hay) asociado con una clave dada

Las aplicaciones modernas probablemente requerirán más que lo anterior, pero esto es lo mínimo para un almacén de clave-valor.

### ¿Cuándo usar una base de datos de clave-valor?

Hay varios casos en los que elegir un enfoque de almacenamiento de valor clave es una solución óptima:

- En caso de necesitar acceso aleatorio a datos en tiempo real, por ejemplo, atributos de sesión de usuario en una aplicación en línea, como juegos o finanzas.
- Mecanismo de almacenamiento en caché para datos de acceso frecuente o configuración basada en claves.
- Una aplicación basada en consultas simples basadas en claves.

### Ventajas

- **Esquemas flexibles:** Se pierde la estructura en tabla, pudiendo establecer diferentes niveles de jerarquía.
- **Sin esquemas predefinidos:** no es necesario predefinir el esquema de la base de datos, sino que se pueden añadir nuevos campos a posteriori.
- **Escalabilidad horizontal:** Permiten el funcionamiento en clusters.
- **Replicabilidad y alta disponibilidad:** Al funcionar en clusters es posible tener varias copias en un mismo documento en diferentes equipos de modo que si uno de los equipos falla automáticamente los otros toman el mando sin pérdida de datos ni rendimiento.
- **Particionado:** La posibilidad de particionar se basa en que un documento puede distribuirse en varios equipos, eliminando la necesidad de que un solo disco tenga la capacidad de guardar todo un conjunto de datos.

- **Velocidad de consultas:** Diferentes estructuras de datos y arquitecturas en clusters reducen los tiempos de consultas cuando la cantidad de datos que maneja es elevada.
- **Velocidad de procesamiento:** Cuando se quieren hacer operaciones sobre toda la base de datos las BD NoSQL llevan incorporados motores de procesamiento en paralelo que reducen los tiempos de lectura y escritura sobre grandes tablas.
- **Alto rendimiento y baja exigencia:** Se pueden desplegar y utilizar en equipos con recursos de HW modestos.

## Desventajas

- **Funciones de fiabilidad:** No existen las Claves Foráneas. No existe restricción referencial de que exista un registro en una tabla relacionada.  
Por esta razón, se debe implementar/solventar en el código del programa lo que agrega mayor complejidad a los sistemas y una debilidad en los esquemas de seguridad.
- **Aplicabilidad:** Relacionado con el punto anterior (funciones de fiabilidad). Limita la aplicabilidad a funciones delicadas como por ejemplo el sector bancario.
- **Incompatibilidad de consultas SQL:** En la mayoría de los casos las BBDD NoSQL son incompatibles a consultas SQL. Se debe incluir una consulta manual que puede ocasionar procesos lentos y complejos.

## Ejemplos de uso reales

- ❖ Listado de elementos más recientes.
- ❖ Sistemas de chat y mensajería.
- ❖ Cestas de la compra e-commerce.



- ❖ Almacenamiento de datos de sesión.
- ❖ Memoria cache de paginas web.
- ❖ Etcd en Kubernetes.

Para dar ejemplos concretos podemos mencionar los siguientes:

1. BioCatch es una empresa israelí de identidad digital que utiliza un seguimiento biométrico innovador para adelantarse a los estafadores en espacios de retail, gaming y finanzas digitales.

BioCatch utiliza módulos, características y varias estructuras de datos de Redis para crear una única base de datos de fuente confiable que brinda información de misión crítica en toda la organización. BioCatch captura datos de comportamiento, metadatos y API durante las sesiones de usuario activas. También crea subconjuntos de perfiles de comportamiento del usuario y perfiles predefinidos de comportamiento fraudulento.

2. El minorista de ropa Gap Inc. deseaba brindar a sus clientes de comercio electrónico información de envío en tiempo real para cada artículo que los compradores agregaron a sus carritos. La empresa enfrentó problemas con retrasos e información de inventario inexacta.

Este problema creó una mala experiencia del cliente que infló los costos y erosionó la lealtad a la marca.

Los desarrolladores de aplicaciones de Gap Inc. encontraron que la escalabilidad lineal y el rendimiento por debajo de los milisegundos de Redis Enterprise a una escala masiva fueron de gran ayuda, especialmente para los picos estacionales del Black Friday. En entornos de microservicios, los modelos de datos rápidos y flexibles protegen contra el aprovisionamiento excesivo de partes de la infraestructura que no se utilizan durante períodos más lentos.

## Usos de este tipo de Base de Datos

Se utilizan cuando se requiere una alta velocidad con un gran volumen de datos.

Son muy efectivas en la consulta y fáciles de escalar.

Estas bases de datos deben crear continuamente nuevas entradas y eliminar las antiguas.

En el ejemplo de uso real, etcd en Kubernetes, donde etcd es una base de datos de clave-valor distribuida desarrollada por el equipo de CoreOS, asegura que este replique la información a los nodos sucesores.

La coordinación del líder con los nodos del clúster a través de una base de datos etcd es especialmente conveniente en el caso de las aplicaciones distribuidas.

Kubernetes en sí es un sistema distribuido que se ejecuta en un clúster desde varios dispositivos. En consecuencia, se beneficia enormemente de una base de datos distribuida como etcd que almacena de forma segura los datos críticos

## Motores de ejemplo (open source y/o pagos)

**Amazon DynamoDB:** es de Amazon Web Services (AWS).

**Berkeley DB:** desarrollado por Oracle, ofrece interfaces para diferentes lenguajes de programación.

**Redis (Remote Dictionary Server):** Proyecto de código abierto. Constituye uno de los SGBD más utilizados y en una fase temprana fue empleado por Instagram y GitHub. Estructura master/slave.

Permite guardar información directamente en la memoria RAM. Esto acelera enormemente la respuesta por parte del servidor.

**Riak:** Riak existe en una variante de código libre, como solución empresarial y en forma de almacenamiento en la nube.

**Voldemort:** extendido SGBD, utilizado e impulsado por LinkedIn, entre otros.

### Diferencias con otras DDBB

#### Redis vs DDBB

Redis	RDBMS
Almacena los datos en la memoria primaria.	Almacena todo en la memoria secundaria.
Las operaciones de lectura y escritura son extremadamente rápidas debido al almacenamiento de datos en la memoria primaria.	Las operaciones de lectura y escritura son lentas debido al almacenamiento de datos en la memoria secundaria.
No puede almacenar archivos grandes o binarios, debido al menor tamaño de la memoria primaria aparte de ser muy costosa.	Puede almacenar archivos grandes, debido a que la memoria secundaria es de un tamaño abundante y mucho más barata por lo cual puede manejar este tipo de archivo fácilmente.
Se utiliza para almacenar una pequeña información textual a la que se debe acceder, modificar e insertar a una velocidad muy rápida, pero si intenta escribir datos masivos que superen la memoria primaria disponible comenzará a recibir errores.	Se utiliza para almacenar datos muchos más grandes que tienen un uso menos frecuente y no requieren ser rápidos.

## Escalabilidad

La escalabilidad indica la habilidad de un sistema para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida y estar preparado para hacerse más grande sin perder calidad y confiabilidad en los servicios ofrecidos.

Las bases de datos NoSQL están diseñadas para escalar usando clústeres de hardware en lugar de añadir costosos servidores y sólidos, lo cuál es muy conveniente para aplicaciones que tendrán un gran y quizás impredecible volumen de lectura y escritura en sus aplicaciones.

Tomemos por ejemplo el caso de AWS Dynamo DB, siguiendo la premisa de la escalabilidad en bases de datos NoSQL.

Dynamo fue pensado con la idea de tener una base de datos clave-valor altamente confiable y ultra-escalable.

Este es un servicio fully managed por AWS lo que significa que está construido para que el desarrollador no deba preocuparse por manejar la infraestructura de la base, y pueda concentrarse en el desarrollo de la aplicación. Esto también significa que el aprovisionamiento de los servidores, clusters y escalado será provisto de manera automática por AWS, librando de responsabilidades al desarrollador.

Este servicio está diseñado para poder escalar los recursos dedicados al mismo a cientos o incluso miles de servidores distribuidos en Availability Zones (distintas locaciones dentro de una misma región de de AWS) para satisfacer las necesidades de almacenamiento y rendimiento de la aplicación en cuestión. No hay límites predefinidos sobre la cantidad de data que cada tabla puede guardar. Los desarrolladores pueden guardar y recuperar cualquier cantidad de datos en DynamoDB y esta automáticamente se va a distribuir entre más servidores a medida que la data aumente en cantidad.

## Links

<https://www.mongodb.com/databases/key-value-database>

<https://redis.com/blog/5-industry-use-cases-for-redis-developers/>

<https://www.slideshare.net/arnabmitra/introduction-to-redis-71446732>

[http://www.thinkmind.org/download.php?articleid=immm\\_2012\\_4\\_10\\_20050](http://www.thinkmind.org/download.php?articleid=immm_2012_4_10_20050)

[https://faculty.washington.edu/wlloyd/courses/tcss562/papers/Spring2017/team7\\_NOSQL\\_DB/Survey%20on%20NoSQL%20Database.pdf](https://faculty.washington.edu/wlloyd/courses/tcss562/papers/Spring2017/team7_NOSQL_DB/Survey%20on%20NoSQL%20Database.pdf)