

# Base de Datos

## Clave-Valor

---

Sebastián Cazares Uribe

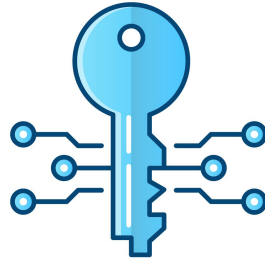
Carlos Manuel Olea Moreno

Carlos Francisco Porras Sainz

# ¿Qué es **Clave**?

La clave es el identificador único que le permite acceder al valor asociado con esa clave.

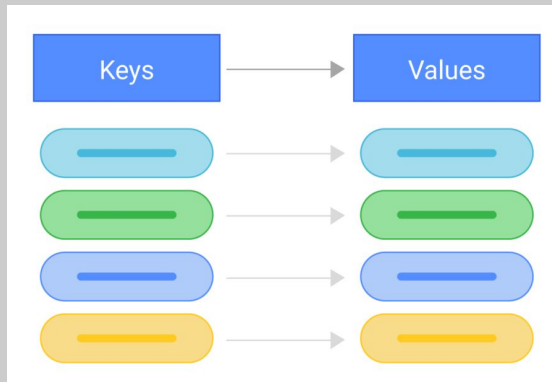
En teoría, la clave podría ser cualquier cosa. Pero esto puede depender del gestor de base de datos. Un gestor de base de datos puede imponer limitaciones mientras que otro puede imponer ninguna.



# ¿Qué es Valor?

El valor en un almacén de clave-valor puede ser cualquier cosa, como texto, un número, código de marcado como HTML, código de programación como PHP, una imagen, etc.

El valor también podría ser una lista, o incluso otro par clave-valor encapsulado en un objeto.



# Estructura Clave-Valor

Para base de datos en RIAK

Estructura de temporadas de los Simpsons

```
simpsons = {  
  'season 1': {  
    { 'episode 1': 'Simpsons Roasting on an Open Fire' },  
    { 'episode 2': 'Bart the Genius' },  
    # ...  
  },  
  'season 2': {  
    { 'episode 1': 'Bart Gets an "F"' },  
    # ...  
  },  
  # ...  
}
```

# Estructura **Clave-Valor**

## Para base de datos en Redis

Estructura de inserción en Redis

```
> set mykey somevalue  
OK  
> get mykey  
"somevalue"
```

# Bases de datos basadas en **Clave y Valor**

## 1. Clave-valor

- Riak KV
- Redis

### Posibles aplicaciones.

- Memorias caché de páginas web, donde la URL es la clave y el contenido el valor.
- Logs de operaciones, con los timestamp como clave y el contenido como valor.

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

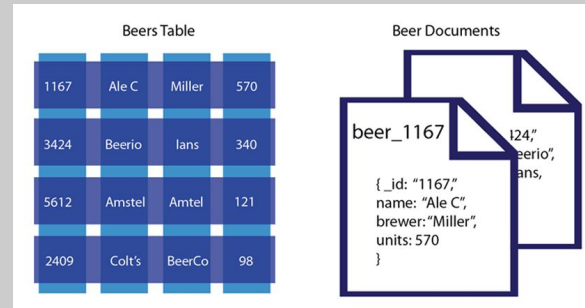
# Bases de datos basadas en **Clave y Valor**

## 2. Documentales

- MongoDB
- CouchDB
- CouchBase
- MarkLogic

### Posibles aplicaciones.

- Datos de sensores provenientes de diferentes fabricantes
- Fichas de clientes con diferentes características
- Catálogos de inventario de productos para una tienda o fábrica.



The diagram illustrates the difference between a relational table and a document-based representation. On the left, a table titled 'Beers Table' shows data in a grid. On the right, a document titled 'Beer Documents' shows a single document for beer\_1167 with its details in a JSON-like format.

1167	Ale C	Miller	570
3424	Beerio	Ians	340
5612	Amstel	Amtel	121
2409	Colt's	BeerCo	98

Beer Documents

```
beer_1167 {
  "_id": "1167",
  "name": "Ale C",
  "brewer": "Miller",
  "units": 570
}
```

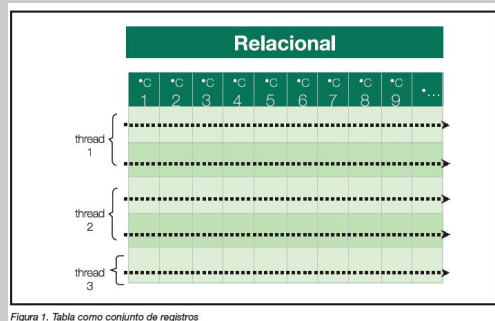
# Bases de datos basadas en **Clave y Valor**

## 3. Columnares

- Cassandra
- Hbase

### Posibles aplicaciones.

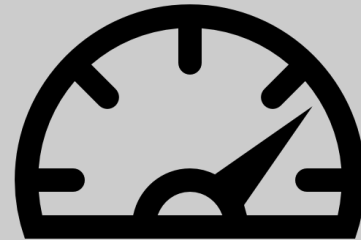
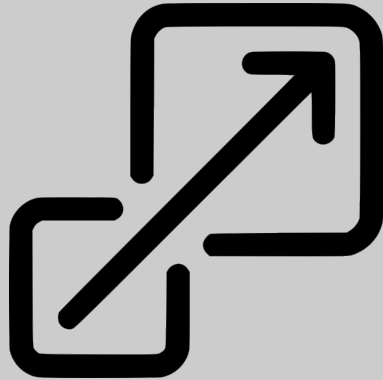
- Catálogos de productos con características predefinidas
- Datos de sensores homogéneos y con altas frecuencias de muestreo
- Aplicaciones de mensajería





# Ventajas **Clave-Valor**

Las bases de datos clave valor **almacenan** los datos en **diccionarios**. Esto nos garantiza tener la información, sin **importar** su **peso**, en una clasificación funcional que podemos recuperar y usar en cualquier momento. La **velocidad y escalabilidad** que **ofrecen** son de los puntos más **fuertes** de estas bases.



# Ventajas **Clave-Valor**

- Aumenta la eficacia y agilidad.
- Aumenta la seguridad en los datos.
- Maximiza tiempos.
- Mejoras en la productividad gracias a los factores anteriormente dictados.



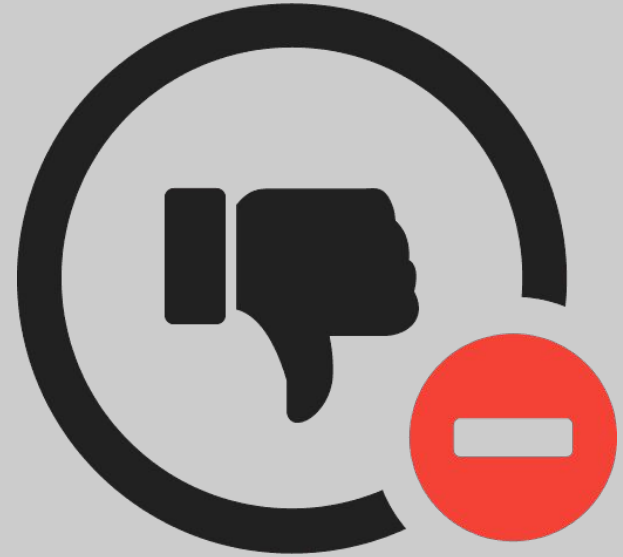
# Desventajas **Clave-Valor**

Comparte con el resto de las NoSQL la debilidad de **no** tener **estándares** en el **manejo de datos**. En ocasiones los **lenguajes de consultas** son **básicos**, impidiendo realizar trabajos de profundidad. Algunos **gestores carecen** de **herramientas** avanzadas para su uso y configuración personalizada. Adicionalmente las consultas y el **ordenamiento** de la **información** pueden ser realizados sólo partiendo de la **clave primaria**.



# Desventajas **Clave-Valor**

- No hay estándares en los datos lo que podría ocasionar incoherencias en los mismos
- Se limitan a aplicaciones específicas y con gran cantidad de datos.
- Incompatibilidad con consultas SQL Se necesita un lenguaje de consulta manual.



# Comparativas

Feature	NoSQL Databases	Relational Databases
Performance	High	Low
Reliability	Poor	Good
Availability	Good	Good
Consistency	Poor	Good
Data Storage	Optimized for huge data	Medium sized to large
Scalability	High	High (but more expensive)

# Cuando usar **SQL** o **NOSQL**

- Cuando los datos deben ser consistentes sin dar posibilidad al error utilizar una base de datos relacional: SQL.
- Cuando nuestro presupuesto no se puede permitir grandes máquinas y debe destinarse a máquinas de menor rendimiento: NoSQL.
- Cuando las estructuras de datos que manejamos son variables: NoSQL database.
- Análisis de grandes cantidades de datos en modo lectura: NoSQL.
- Captura y procesamiento de eventos: NoSQL.
- Tiendas online con motores de inteligencia complejos: NoSQL.



# CLAVE-VALOR

BD NoSQL