

Mocoa, Fecha (16 – 10 – 2025)

Programa: Ingeniería de sistemas

Syllabus: Bases de datos no relacionales

Semestre: 7

Docente: Fredy Jacanamijoy

3204414573 – [fredy.jacanamijoy@itp.edu.co](mailto:fredy.jacanamijoy@itp.edu.co)

## Creación y Uso de Bases de Datos en Redis

Redis (REmote DIctionary Server) es una base de datos en memoria, open source, usada principalmente como caché, cola de mensajes o almacenamiento de datos de alta velocidad. A diferencia de los sistemas tradicionales como MySQL, Redis no crea bases de datos con nombre, sino que maneja espacios lógicos numerados.

### 1. Bases de datos lógicas en Redis

Redis no crea bases de datos con nombres personalizados. En su lugar, maneja bases numeradas por defecto del 0 al 15 (16 bases en total). Cada una actúa como un espacio independiente.

1. Ejemplo de selección de base de datos:

2. SELECT 1

3. SET usuario "Fredy"

4. GET usuario

Esto guarda la clave 'usuario' con valor 'Fredy' en la base lógica número 1. Si cambias a otra base, las claves no estarán disponibles, ya que cada base es independiente.

### 2. Base de datos por defecto

Si ejecutas un comando sin haber seleccionado una base con SELECT, Redis usa automáticamente la base número 0.

5. Ejemplo:

6. SET nombre carlos

El valor 'carlos' se almacena en la base 0. Si luego haces SELECT 1 y consultas GET nombre, no obtendrás resultados porque el dato pertenece a la base 0.

### 3. Cambiar el número de bases disponibles

Puedes modificar la cantidad de bases lógicas que Redis ofrece editando el archivo de configuración redis.conf y agregando o cambiando la siguiente línea:

7. databases 32

Esto aumentará la cantidad de bases lógicas a 32, numeradas del 0 al 31. Luego, debes reiniciar Redis para aplicar los cambios.

#### 4. Ejemplo en Python

Ejemplo de conexión a una base de datos específica usando Python:

```
import redis

# Conexión a la base de datos 2
r = redis.Redis(host='localhost', port=6379, db=2)

# Guardar y obtener un valor
r.set('proyecto', 'Emssanar')
print(r.get('proyecto')) # b'Emssanar'
```

#### 5. Comandos útiles en Redis CLI

8. Algunos comandos básicos para gestionar datos:

- 9. SET clave valor — Guarda un valor.
- 10. GET clave — Recupera un valor.
- 11. DEL clave — Elimina una clave.
- 12. EXPIRE clave segundos — Asigna un tiempo de vida a una clave.
- 13. DBSIZE — Devuelve el número de claves en la base actual.

#### 6. Recomendaciones finales

Redis no permite poner nombres personalizados a las bases de datos. Si necesitas separar datos por proyectos o módulos, puedes usar prefijos en las claves (por ejemplo, 'app1:usuario:123') o ejecutar instancias Redis separadas en diferentes puertos.