



**POLITÉCNICO
ESTELLA**

ASIGNATURA: BASES DE DATOS

**GRADO SUPERIOR EN DESARROLLO DE
APLICACIONES MULTIPLATAFORMA**

BDDA07 TAREA 1

Presentado por: Eugen Moga

Índice

Ejercicio 1 Preguntas teóricas	1
1.1 ¿Cuál es la diferencia entre un tipo de dato string y hash?	1
1.2 ¿Cuál es la diferencia entre un tipo de dato list y sorted set?	1
1.3 ¿A qué corresponde el parámetro TTL?	1
Ejercicio 2 Instalación	2
2.1 Servicio Redis Cloud	2
2.2 Redis Insight	3
Ejercicio 3 Datos de prueba	5
3.1 ¿Qué tipos de datos se cargaron en los registros de ejemplo?	6
3.2 Pruebas a cambiar el valor de TTL	7
3.3 Usando en el workbench	8
Ejercicio 4	8
4.1 Mediante un tipo de dato HASH	9
4.2 Queremos tener preparada las sentencias	9
4.3 Queremos realizar el ranking	10
4.4 Modifica el TTL 1 minutos	12
Ejercicio 5	13
5.1 Crea una cola llamada emails:pendientes	13
5.2 Procesa el primer mensaje (comando LPOP o RPOP)	13
5.3 Verifica cuántos mensajes quedan (comando LLEN)	14
5.4 Añade un mensaje prioritario al inicio de la cola	14
5.5 Procesa los mensajes restantes	15
5.6 Comprobar que quedan 0 mensajes	16

Ejercicio 1 Preguntas teóricas

1.1 ¿Cuál es la diferencia entre un tipo de dato string y hash?

String es una cadena de caracteres que almacena un único valor en forma de texto o binario mientras que Hash es como un diccionario que almacena multiples campos valor dentro de una misma clave

Ejemplo String: SET usuario:1 "Eugen Moga"

Ejemplo Hash: HSET usuario:1 nombre "Eugen" apellido "Moga" edad 30

1.2 ¿Cuál es la diferencia entre un tipo de dato list y sorted set?

List es una lista ordenada de elementos que puede tener duplicados y que al agregar elementos se pueden agregar al principio de la lista con LPUSH o al final de la lista con RPUSH mientras que Sorted Set es un conjunto ordenado de elementos únicos que no permiten duplicados y los elementos se ordenan automáticamente de menor a mayor.

1.3 ¿A qué corresponde el parámetro TTL?

El parámetro TTL hace referencia al tiempo de vida y es un valor que define cuanto tiempo permanece almacenado un dato antes de ser eliminado automáticamente. Esta funcionalidad se utiliza para gestionar la caducidad de datos sin intervención manual y es ideal para escenarios donde los datos tiene un ciclo de vida definido como las contraseñas temporales.

Ejercicio 2 Instalación

2.1 Servicio Redis Cloud

Accedo a [Redis](#) e inicio sesión con la cuenta de educación selecciono el servicio de Google Cloud y la ubicación en Europa y le doy a Get Stared

Una vez registrado voy a Data Access Control en la pestaña Roles y agrego un nuevo rol en el botón Add new role le pongo el nombre eugen, en ACL rule le doy acceso total seleccionando Full Access y selecciono la base de datos y le doy al botón Check Bold antes de guardar el rol.

The screenshot displays the Redis Cloud Data Access Control interface. On the left sidebar, the 'Data Access Control' option is highlighted. The main panel shows the 'Create New Role' form with the following fields and actions:

- Name:** A text input field containing 'eugen'.
- Databases:** A section with an 'Add' button and a note: 'Only databases with versions compatible with the ACL rule are shown.'
- ACL rule:** A dropdown menu set to 'Full-Access'.
- Database:** A dropdown menu showing 'Eugen-free-db: Eugen-free-db' with a checkmark button next to it.
- Associations:** A table with columns 'Database', 'Region', and 'ACL rule'.
- Buttons:** 'Cancel' and 'Save role' buttons at the bottom right.

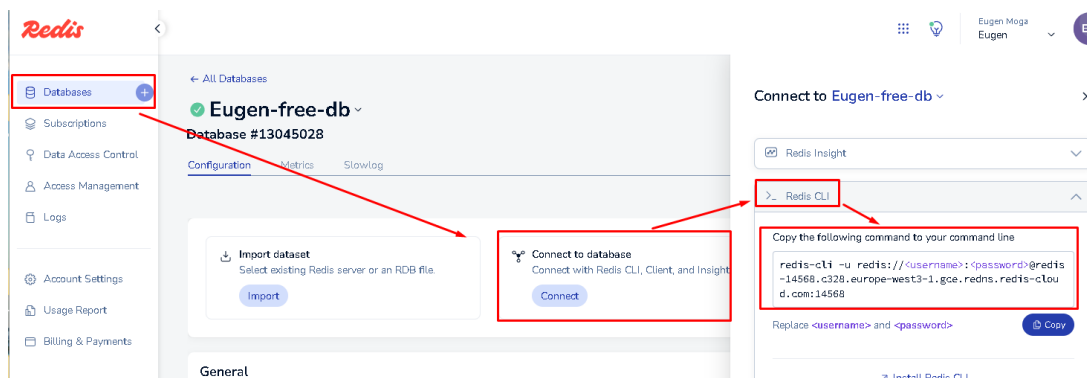
Red boxes and arrows indicate the sequence of actions: selecting 'Data Access Control' in the sidebar, navigating to the 'Roles' tab, entering 'eugen' as the role name, selecting 'Full-Access' for the ACL rule, choosing the database 'Eugen-free-db: Eugen-free-db', and clicking the checkmark button to confirm the selection.

2.2 Redis Insight

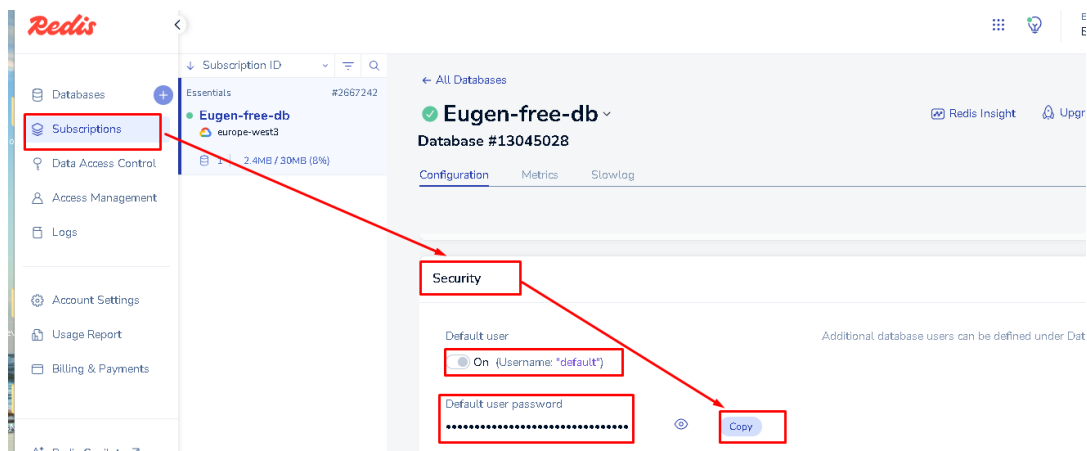
En la pestaña izquierda de Download Center selecciono Redis Insight for Windows y lo doy a Download y ejecuto el instalador



Para ver el comando de conexión voy a Database > Connect to databas y copio el comando que sale en Redis CLI



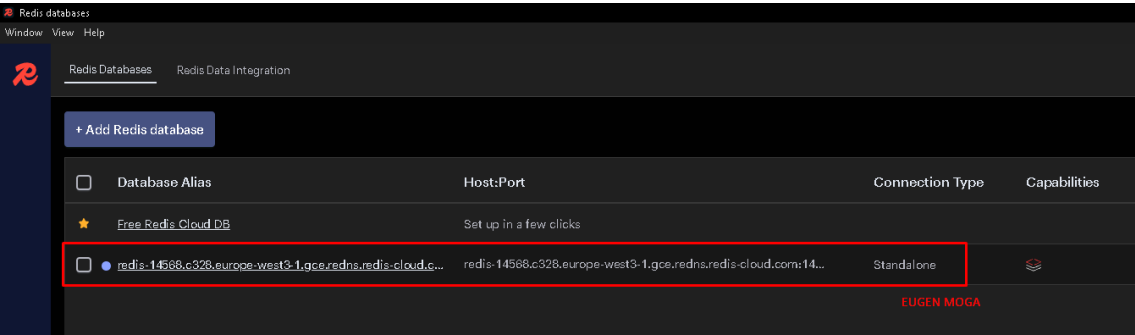
Para ver usuario y contraseña voy a Subscriptions selecciono la base de datos y en la parte de Security veo que el usuario es "default" y copio la contraseña



Ahora para establecer la conexión voy a la aplicación de escritorio Redis Insight y le doy a agregar Redis Database en Connection URL le pego el comando que he copiado de Redis CLI y le cambio el usuario y la contraseña y queda de la siguiente manera.

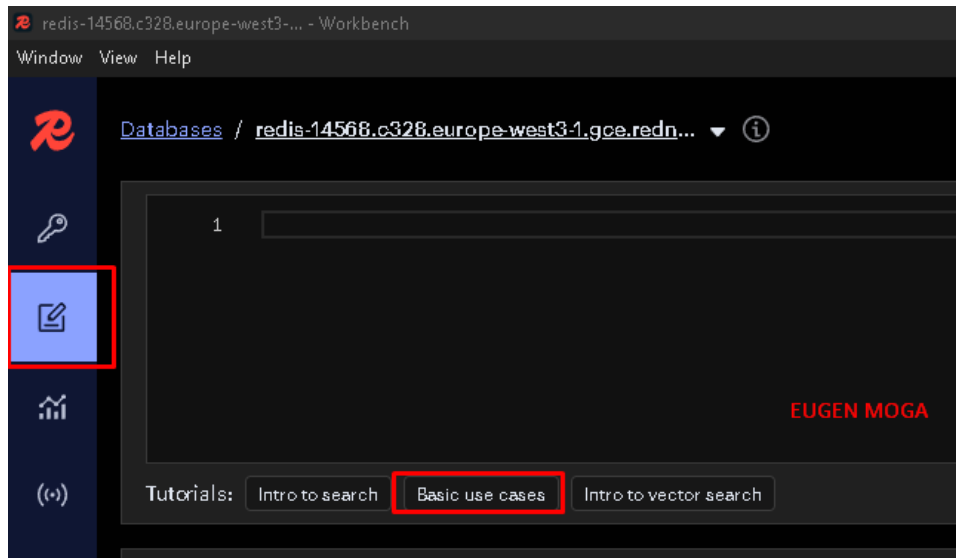


Y ya se puede ver la base de datos conectada desde la aplicación de escritorio

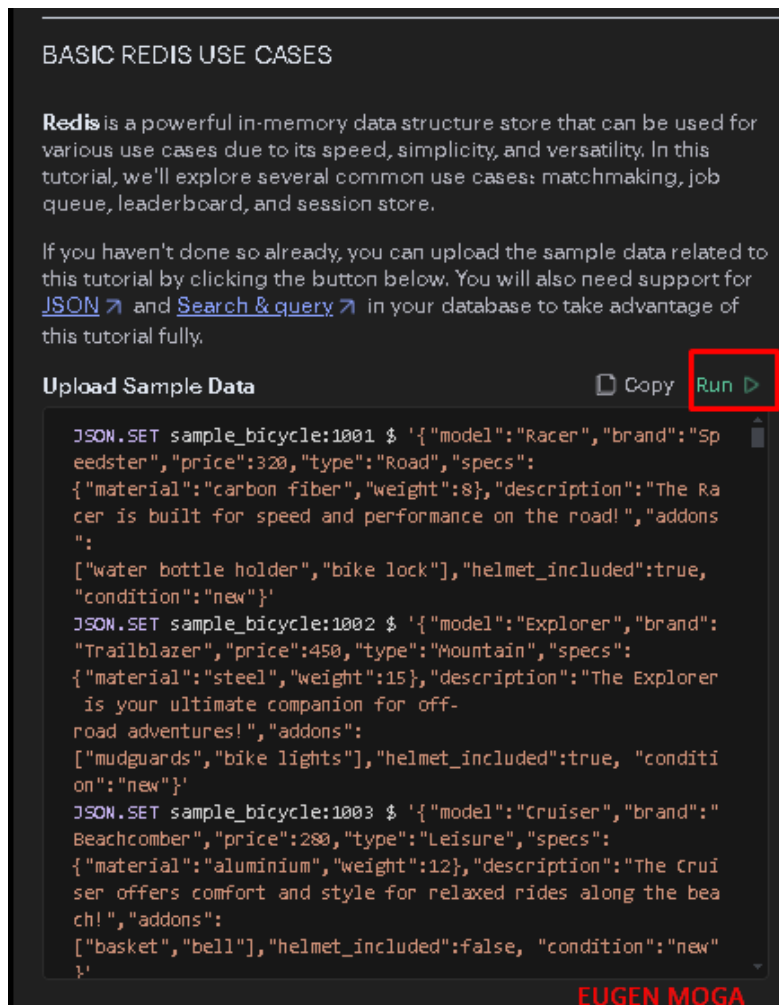


Ejercicio 3 Datos de prueba

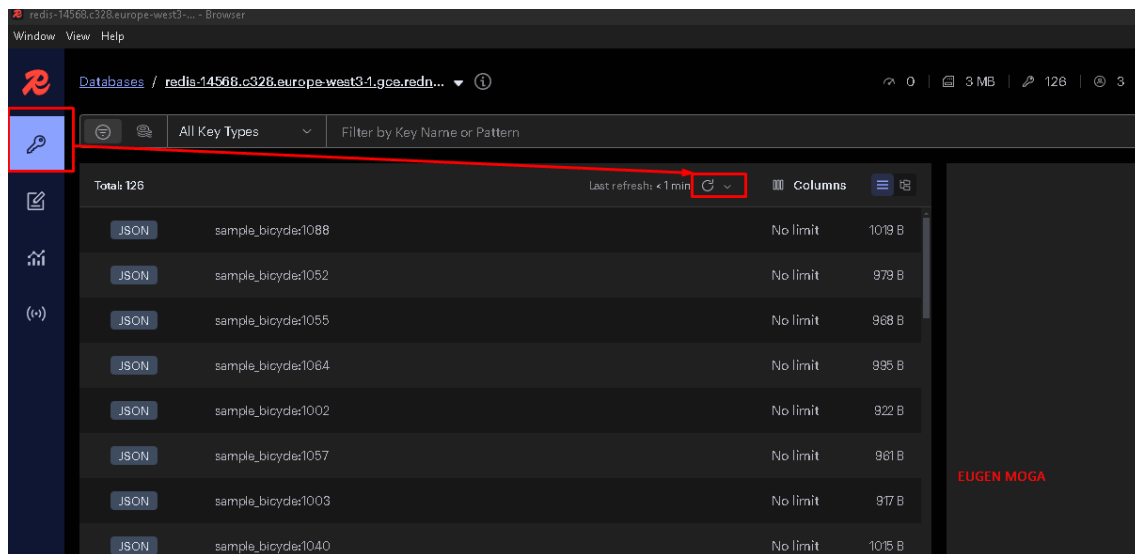
Una vez establecida la conexión hago clic sobre la base de datos se abre y le doy clic a Workbench > Basic use cases



En la parte derecha se abre el panel para insertar los datos de prueba y le doy al botón Run



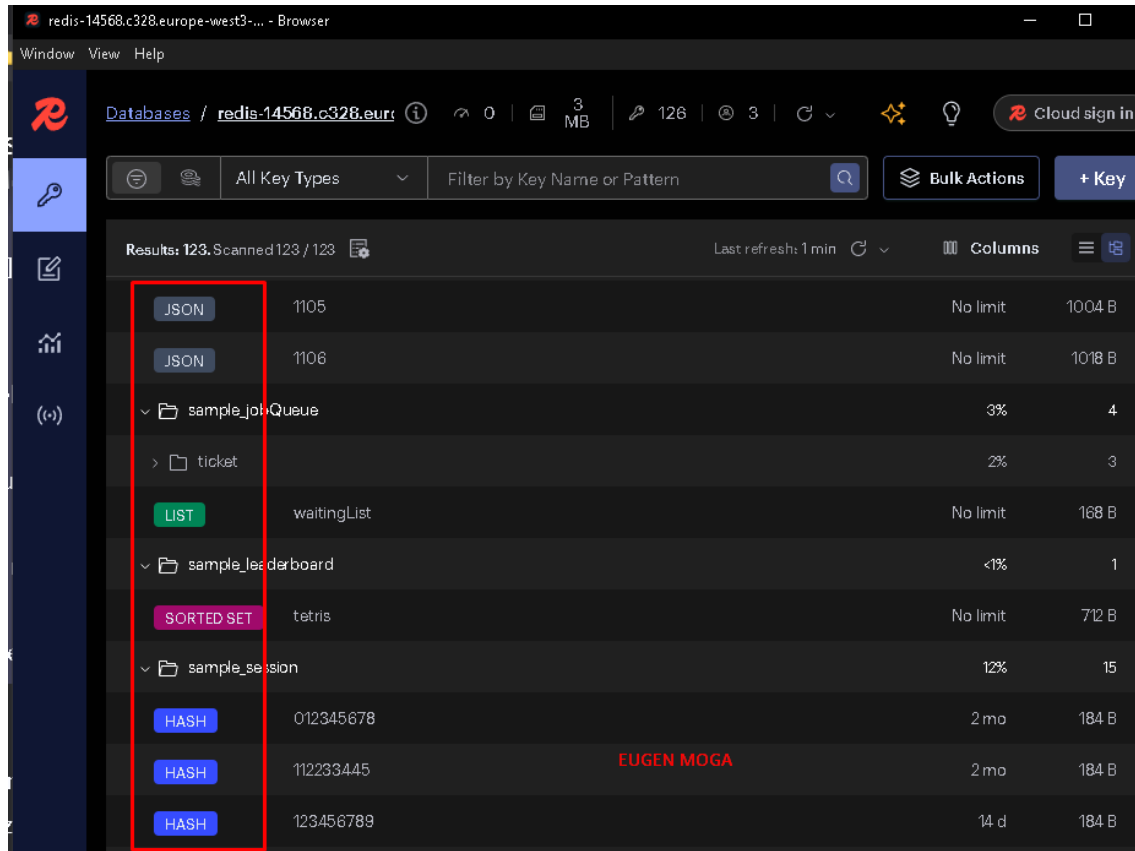
Para ver los datos voy a la pestaña Browser le doy click en el botón Refresh



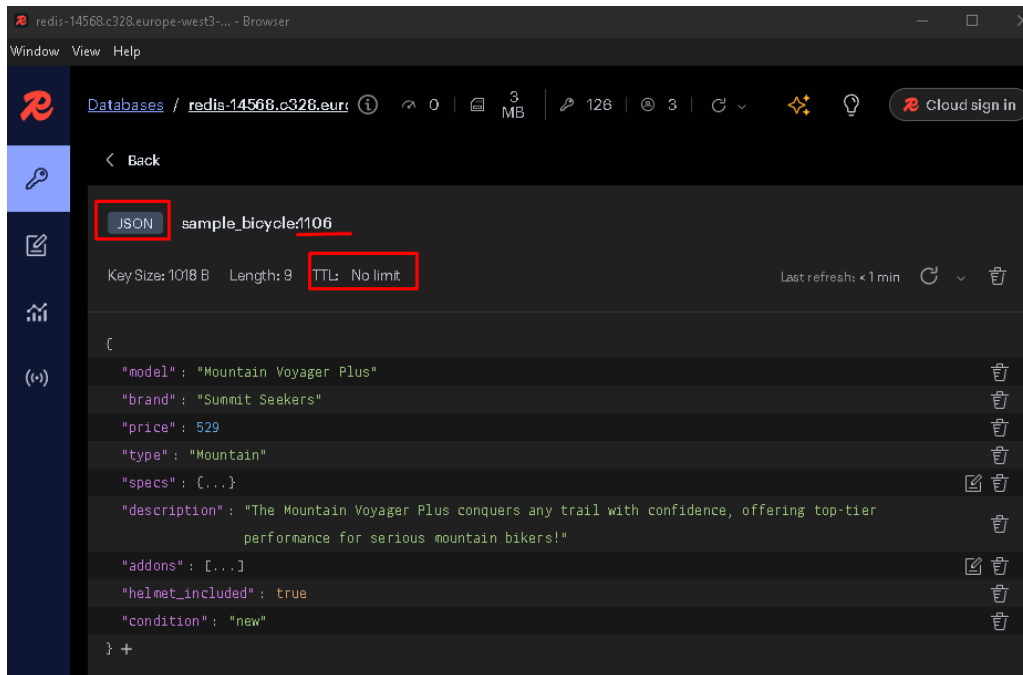
3.1 ¿Qué tipos de datos se cargaron en los registros de ejemplo?

Visualiza el detalle de algunos de ellos, en particular localiza la clave correspondiente y su TTL.

Se cargaron tipos de datos como JSON que son String, LIST, SORTED SET y HASH

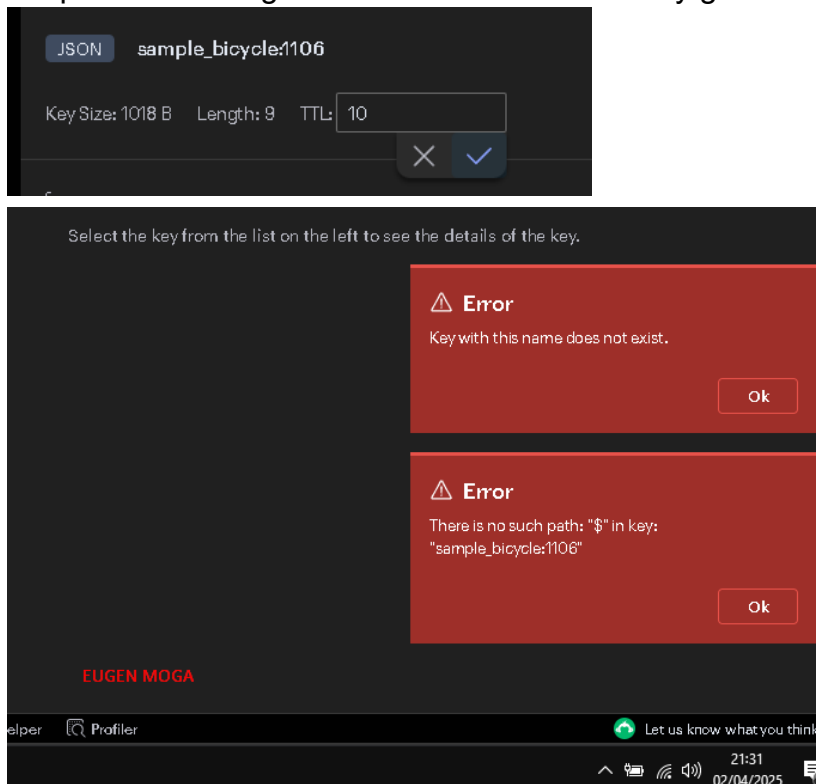


Para visualizar más en detalle algún dato hay que pulsar sobre el dato en concreto y se abre una vista detallada donde se pueden ver los datos que almacena donde la clave sería `sample_bicycle:1106` y el TTL en este caso no está definido.



3.2 Pruebas a cambiar el valor de TTL y verifica el efecto del cambio realizado. ¿En qué casos-ámbitos prácticos es interesante este parámetro?

En este caso establezco un TTL de 10 segundos y confirmo, esto hará que después de 10 segundos el elemento se borre y genere un error al refrescar



Establecer el parámetro tiempo de vida TTL es interesante usarlo en inicios de sesión o autenticación, promociones y ofertas, en bloqueos temporales para limitar intentos de inicio de sesión, contadores temporales para contar número de visitas a un página en un tiempo determinado.

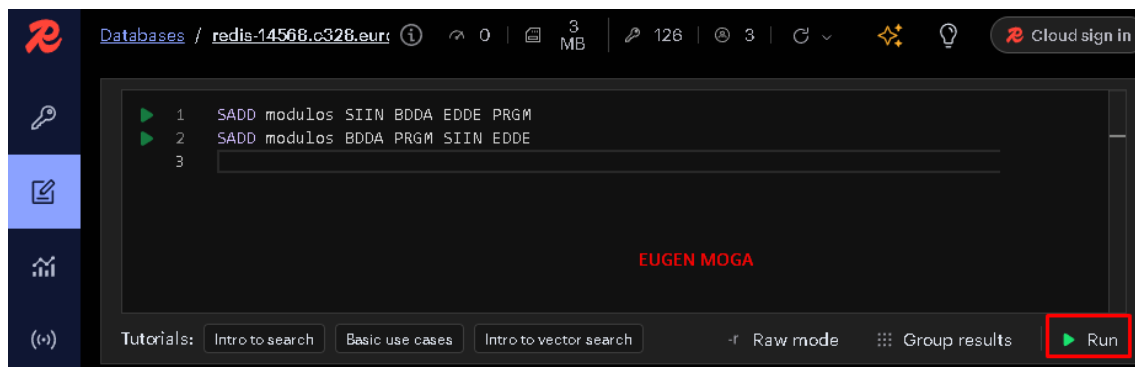
3.3 Usando en el workbench, ejecuta las siguientes sentencias:

SADD modulos SIIN BDDA EDDE PRGM

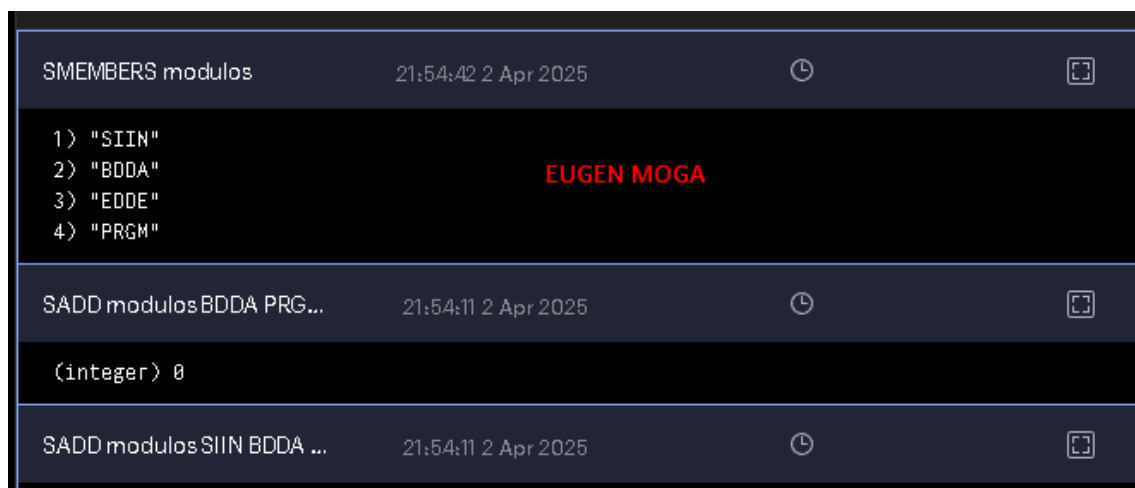
SADD modulos BDDA PRGM SIIN EDDE

¿Qué tipo de datos se han insertado? ¿Por qué la sentencia SMEMBRES modulos solo devuelve 4 registros?

Ejecuto los comandos y se insertan datos de tipo Set que son conjunto de datos no ordenados,



Al ejecutar la sentencia SMEMBRES modulos, solo muestra 4 registros porque los tipos de datos Set no permite duplicados



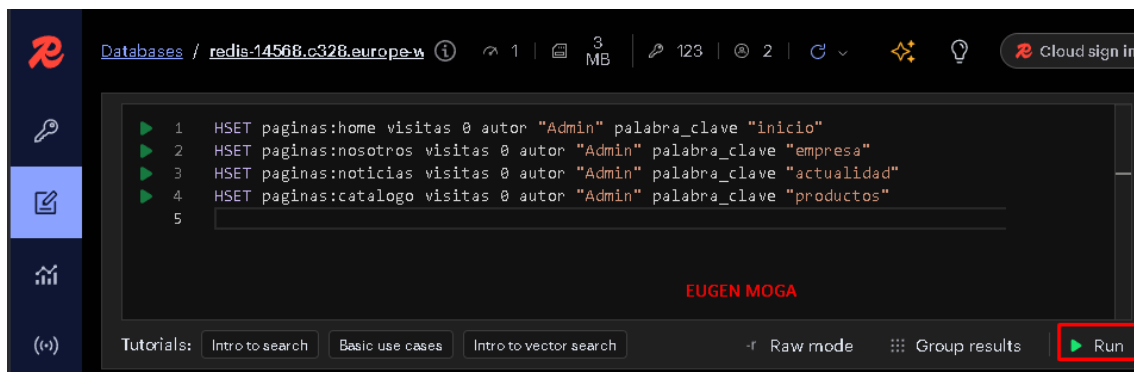
Ejercicio 4

Vamos a implementar un sistema de contador de visitas para una página web. Las páginas de nuestra web son “home”, “nosotros”, “noticias” y “catálogo”.

Queremos registrar para cada página el nº de visitas, el autor y una palabra clave sobre su contenido.

4.1 Mediante un tipo de dato HASH que llamaremos “paginas”, introduce para cada página un registro con los parámetros anteriormente mencionados.

Un tipo de dato HASH se inserta con HSET donde cada página web (home, nosotros, noticias y catalogo) es una clave dentro del Hash paginas y los atributos (visitas, autor y palabra clave) se almacenan como campos internos para cada clave



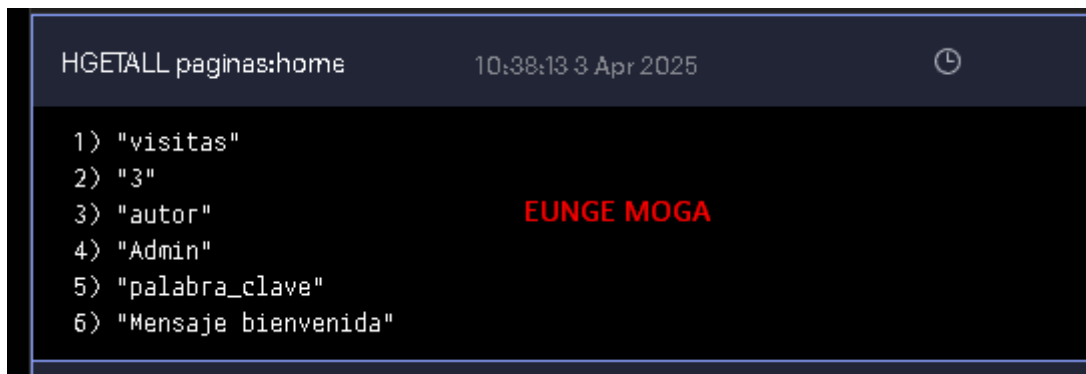
```
1 HSET paginas:home visitas 0 autor "Admin" palabra_clave "inicio"
2 HSET paginas:nosotros visitas 0 autor "Admin" palabra_clave "empresa"
3 HSET paginas:noticias visitas 0 autor "Admin" palabra_clave "actualidad"
4 HSET paginas:catalogo visitas 0 autor "Admin" palabra_clave "productos"
5
```

4.2 Queremos tener preparada las sentencias que permitan incrementar el nº de visitas de una página. Escribe para una de las páginas una sentencia que incrementa en 3 el nº de visitas (comando hincrby), y otra que actualiza la palabra clave asociada (comando hset). Muestra mediante sentencia los valores de la página modificada (comando hgetall).

Para incrementar el número de vistas de una página en tres, utilizó el comando HINCRBY (nombre clave en este caso: paginas): (seguido del nombre de la página que quiero incrementar: home) (y el nombre del atributo que voy a editar: visitas 3)

Para actualizar la palabra clave asociada utilizo el comando HSET (clave: paginas) (nombre página: home) (nombre atributo: palabra_clave “Mensaje bienvenida”)

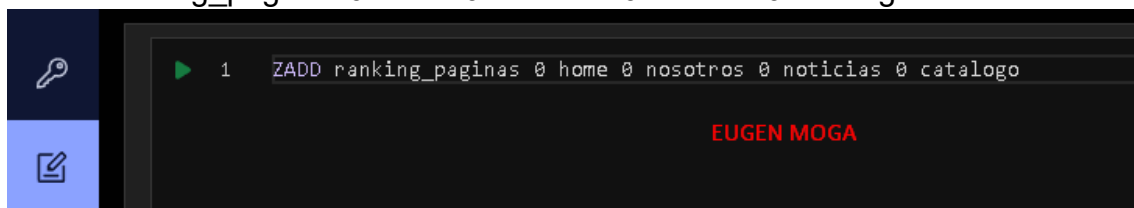
Para mostrar mediante sentencia los valores de la pagina modificada utilizó el comando HGETALL (clave: paginas) : (nombre página: home)



4.3 Queremos realizar el ranking de las visitas de las páginas, por ello usaremos un tipo de dato sorted set llamado `ranking_paginas`. Inicializa cada página con el contador a 0 (comando `ZADD`). Da la sentencia que aumentará en 1 el contador de una página dada (comando `ZINCRBY`). Tras hacer algún cambio manual del contador de cada página, sacar el ranking de las 3 primeras más visitadas junto al contador correspondiente (comando `ZRANGE`). Saca el rango actual de una de las páginas (comando `ZRANK`).

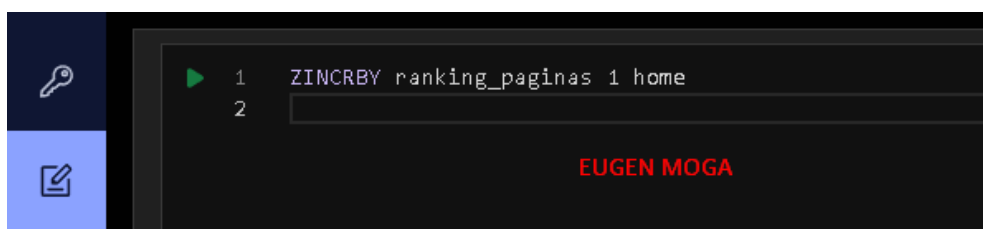
Inicializo el `ranking_paginas` con el contador a 0

`ZADD ranking_paginas 0 home 0 nosotros 0 noticias 0 catalogo`

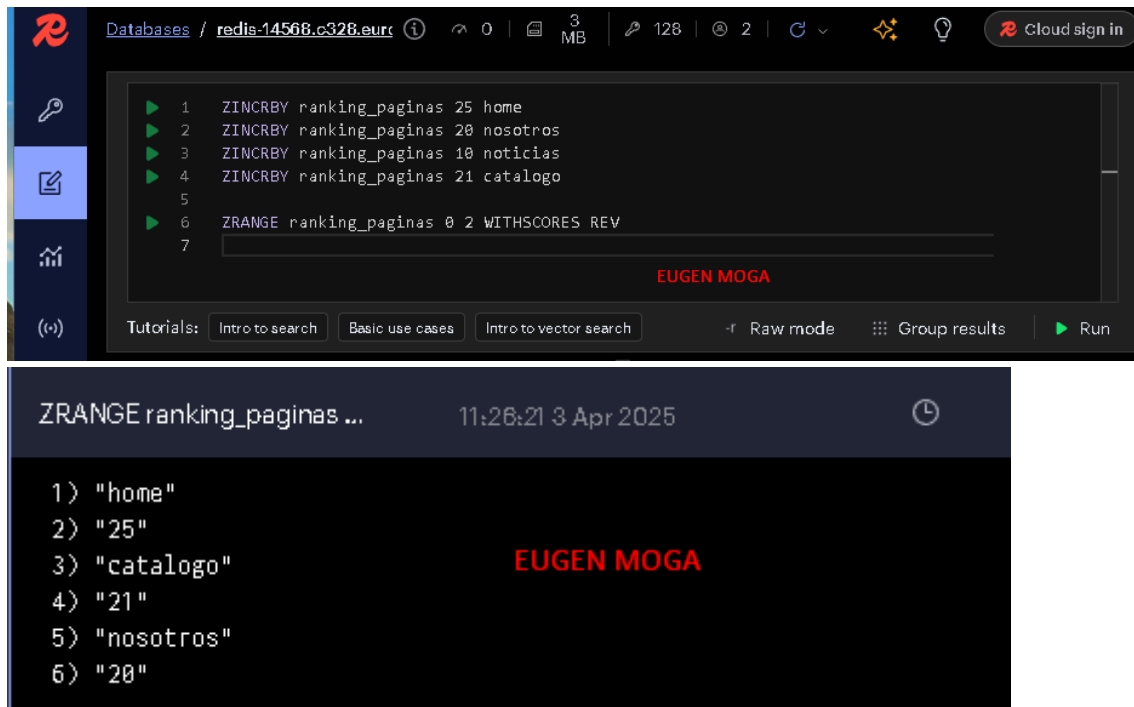


Incremento el contador de la página home

`ZINCRBY ranking_paginas 1 home`



Modifico manualmente el contador de las páginas. Y después saco el ranking de las 3 primeras páginas, con REV ordeno de mayor a menor
ZRANGE ranking_paginas 0 2 WITHSCORES REV



The screenshot shows a Redis CLI interface with a dark theme. The top bar displays the database name 'redis-14568.c328.eurc' and various status icons. The main area contains a list of commands: 1. ZINCRBY ranking_paginas 25 home, 2. ZINCRBY ranking_paginas 20 nosotros, 3. ZINCRBY ranking_paginas 10 noticias, 4. ZINCRBY ranking_paginas 21 catalogo, 5. (empty line), 6. ZRANGE ranking_paginas 0 2 WITHSCORES REV, 7. (empty line). Below the commands, the name 'EUGEN MOGA' is displayed in red. At the bottom, there are tabs for 'Intro to search', 'Basic use cases', and 'Intro to vector search', along with buttons for 'Raw mode', 'Group results', and 'Run'.

```
1 ZINCRBY ranking_paginas 25 home
2 ZINCRBY ranking_paginas 20 nosotros
3 ZINCRBY ranking_paginas 10 noticias
4 ZINCRBY ranking_paginas 21 catalogo
5
6 ZRANGE ranking_paginas 0 2 WITHSCORES REV
7
```

EUGEN MOGA

Tutorials: Intro to search Basic use cases Intro to vector search -r Raw mode Group results Run

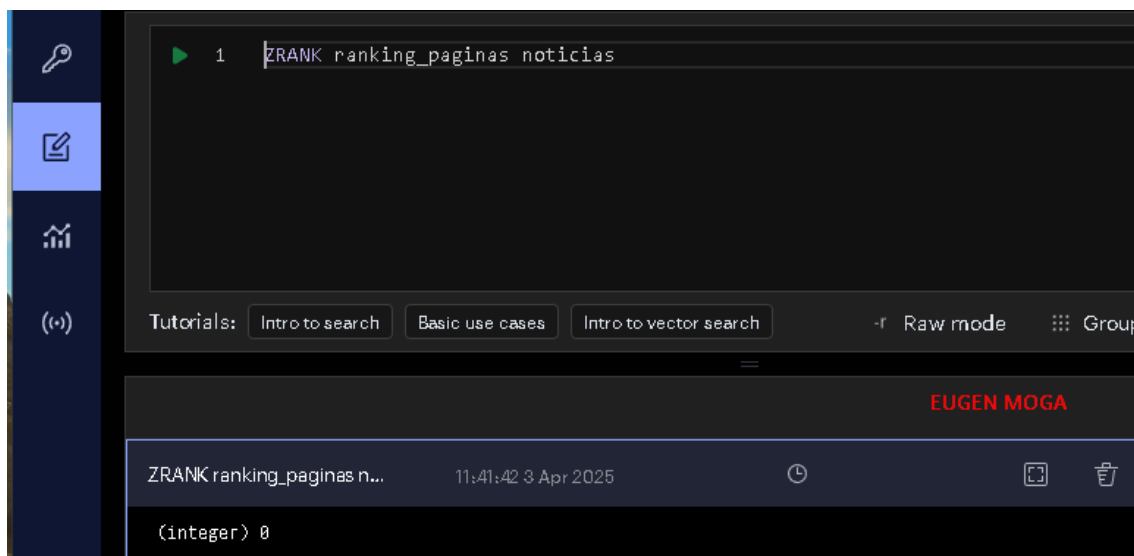
ZRANGE ranking_paginas ... 11:26:21 3 Apr 2025

```
1) "home"
2) "25"
3) "catalogo"
4) "21"
5) "nosotros"
6) "20"
```

EUGEN MOGA

Para sacar el rango actual de la pagina noticias utilizo el comando

ZRANK ranking_paginas noticias. Esto me devuelve la posición del elemento noticias en un sorted set ordenado de menor a mayor y el índice empieza en 0 para el elemento con menor score. Como el resultado es 0 quiere decir que está en última posición



The screenshot shows a Redis CLI interface with a dark theme. The top bar displays the database name 'redis-14568.c328.eurc' and various status icons. The main area contains a single command: 1. ZRANK ranking_paginas noticias. Below the command, the name 'EUGEN MOGA' is displayed in red. At the bottom, there are tabs for 'Intro to search', 'Basic use cases', and 'Intro to vector search', along with buttons for 'Raw mode', 'Group results', and 'Run'.

```
1 ZRANK ranking_paginas noticias
```

EUGEN MOGA

Tutorials: Intro to search Basic use cases Intro to vector search -r Raw mode Group results Run

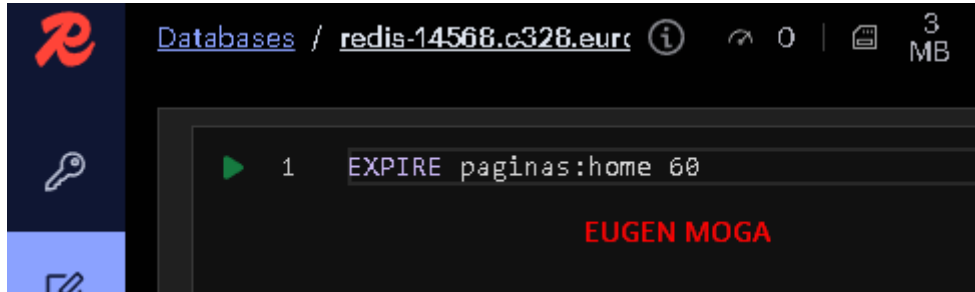
ZRANK ranking_paginas n... 11:41:42 3 Apr 2025

```
(integer) 0
```

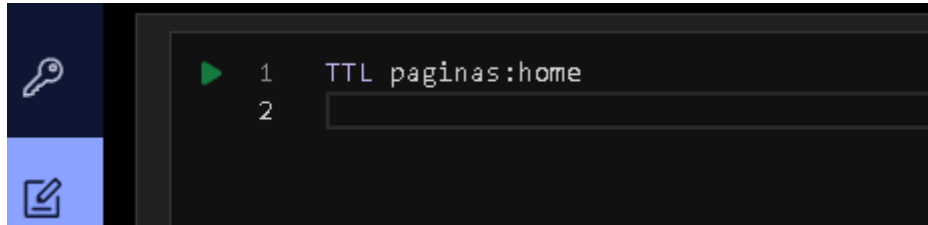
4.4 Modifica el TTL 1 minutos para los datos de la página home, y mira el efecto (comando EXPIRE).

Para modificar el TTL a 1 minuto para la pagina home utilizo el comando:

EXPIRE paginas:home 60



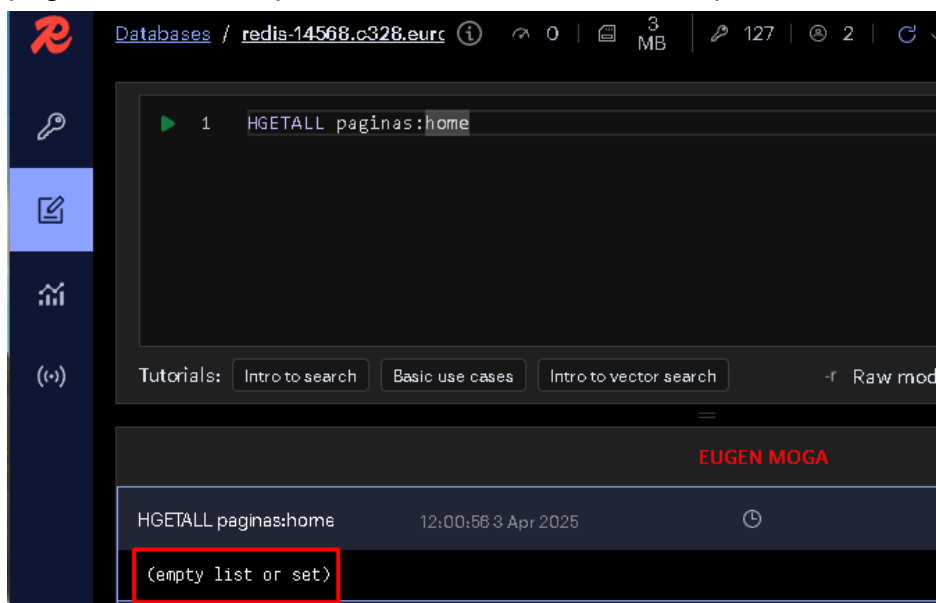
Para comprobar el TTL de un elemento utilizo TTL paginas:home



El resultado sale en segundos



Una vez finalizado el tiempo se elimina automáticamente la clave paginas:home. Se puede ver al hacer HGETALL que esta vacío

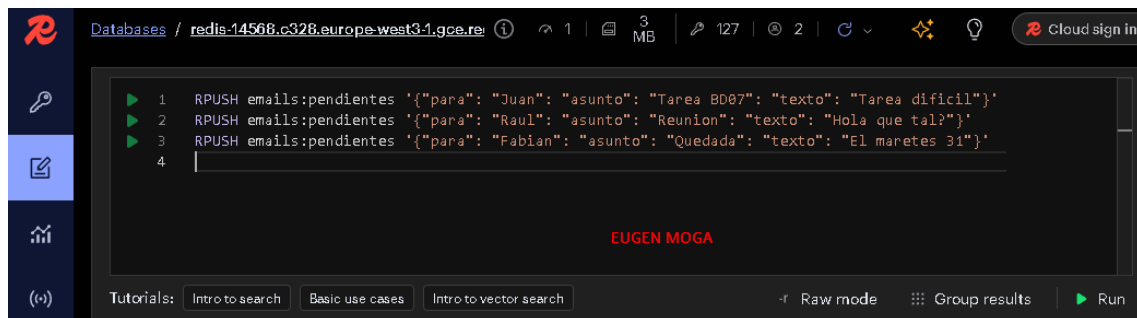


Ejercicio 5

Utilizando el tipo de datos listas de Redis, se pide simular una cola de correos.

5.1 Crea una cola llamada emails:pendientes y añade 3 mensajes de email al final de la cola (comando RPush o LPush). Para cada mensaje, insertamos una cadena JSON con los parámetros para, asunto y texto. Por ejemplo: `'{"para": "Profe", "asunto": "Muy importante", "texto": "hola"}'`

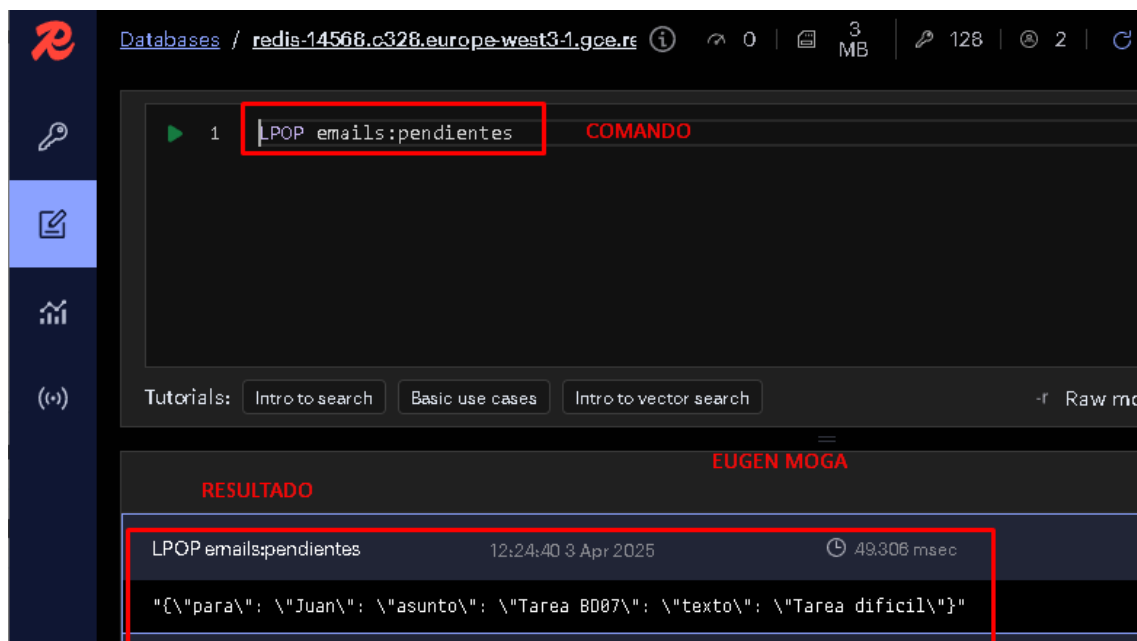
Para añadir al final de la cola utilizo el comando RPush emails:pendientes y agrego 3 mensajes modificando su contenido



```
Databases / redis-14568.c328.europe-west3-1.gcp.re
1 RPush emails:pendientes '{"para": "Juan": "asunto": "Tarea BD07": "texto": "Tarea difcil"}'
2 RPush emails:pendientes '{"para": "Raul": "asunto": "Reunion": "texto": "Hola que tal?"}'
3 RPush emails:pendientes '{"para": "Fabian": "asunto": "Quedada": "texto": "El maretes 31"}'
4
EUGEN MOGA
Tutorials: Intro to search Basic use cases Intro to vector search
-r Raw mode Group results Run
```

5.2 Procesa el primer mensaje (comando LPOP o RPOP).

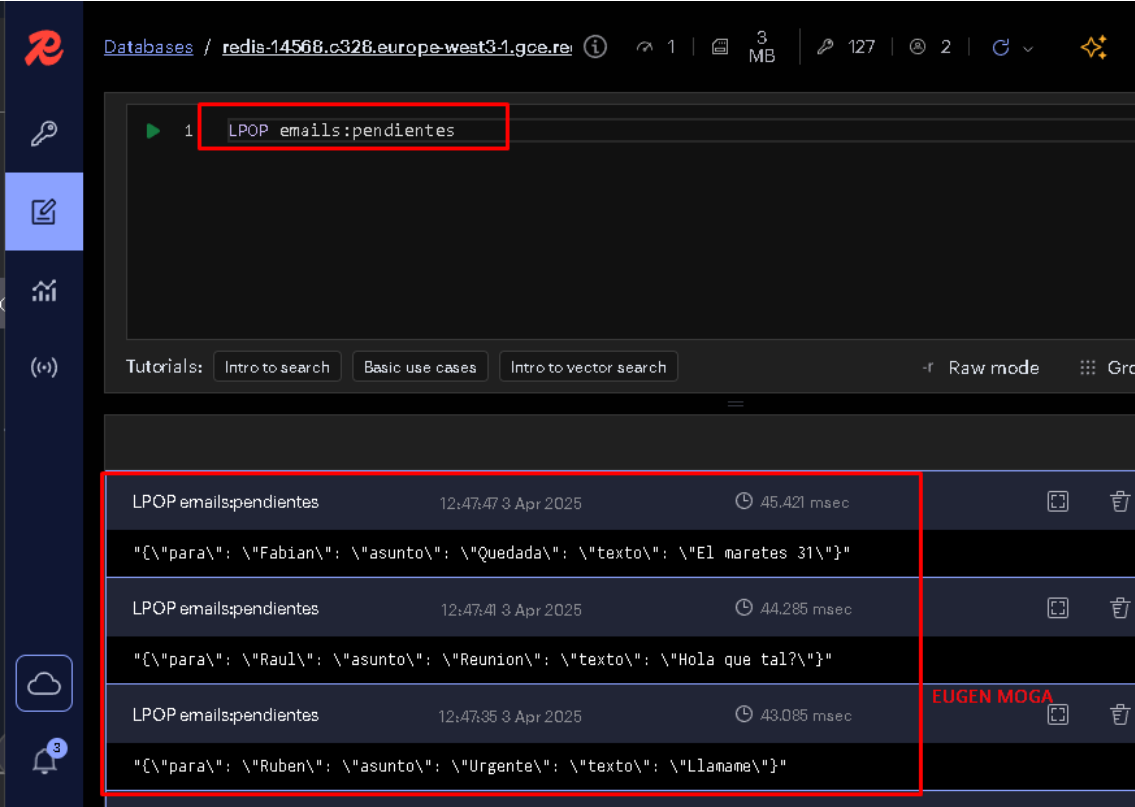
Para procesar el primer mensaje de la lista utilizo el comando LPOP emails:pendientes



```
Databases / redis-14568.c328.europe-west3-1.gcp.re
1 LPOP emails:pendientes
COMANDO
EUGEN MOGA
RESULTADO
LPOP emails:pendientes 12:24:40 3 Apr 2025 49.306 msec
{"para": "Juan", "asunto": "Tarea BD07", "texto": "Tarea difcil"}
```


5.5 Procesa los mensajes restantes.

Se pueden procesar uno a uno con el comando LPOP



Hasta que aparece el resultado (nil)

LPOP emails:pendientes	12:49:34 3 Apr 2025	44.909 msec	
(nil)			
LPOP emails:pendientes	12:47:47 3 Apr 2025	45.421 msec	
"{"para": "Fabian", "asunto": "Quedada", "texto": "El maretas 31"}			
LPOP emails:pendientes	12:47:41 3 Apr 2025	44.285 msec	EUGEN MOGA
"{"para": "Raul", "asunto": "Reunion", "texto": "Hola que tal?"}			
LPOP emails:pendientes	12:47:35 3 Apr 2025	43.085 msec	
"{"para": "Ruben", "asunto": "Urgente", "texto": "Llamame"}			

5.6 Comprobar que quedan 0 mensajes.

Para comprobar que no quedan mensajes utilizo el comando LLEN si el resultado que devuelve es 0 significa que no quedan mensajes

