

REDIS - ASPECTOS GENERALES

¿QUÉ VAMOS APRENDER?

1. Redis Arquitectura.
2. Redis - Instalación básica
3. Redis – Cluster
4. Redis – Tipo de datos y Como usarlos.
5. Redis – Cliente, como usar redis con el cliente Jedis.

¿QUÉ ES REDIS ?

Es un almacén de estructura de datos en memoria, usado como base de datos y cache. Soporta estructura de datos tales como: String, hashed, list, sets, sorted sets. Soporta operaciones atómicas sobre estas estructuras de datos tales como agregar data a String, incrementar un valor en un hash, agregar un elemento a una lista, y fijar operaciones como inserción, unión y diferencia.

Redis tiene un proceso de replicación incorporada, LUA Scripting y transacciones y niveles de persistencia en disco, y proporciona niveles de alta disponibilidad.

- Nos da acceso rápido a los datos NoSQL.
- Mejora los tiempos de respuesta de su aplicación con una infraestructura mínima. (interés de twitter)
- Proporciona Alta Disponibilidad, clustering , persistencia y opciones de seguridad.

Usado por compañías como: Twitter, Pinterest, StackOverflow , etc

BONDADES DE REDIS.

- Redis almacena un par de datos en memoria Clave – Valor.
- Clave – Valor, son almacenados en la memoria primaria (RAM).
- Las bases de datos tradicionales almacenan la data en memoria secundaria(disk).
- La memoria primara es mucho más rápido, pero más costosa.
- El inconveniente es que Redis no puede ser usado para almacenar archivos grandes. Pero en vez de ello se basa en el acceso rápido a los conjuntos de datos más pequeños.
- Altamente escalable.

REDIS - INSTALACIÓN

```
$ cd /opt/  
$ tar vxf redislabs-5.0.2-15-rhel7-x86_64.tar  
$ ./install.sh -s /var/run/redislabs
```

```
2018-04-16 09:01:48 [.] Checking root access  
2018-04-16 09:01:48 [!] Running as user root, sudo is not required.  
2018-04-16 09:01:49 [.] Appending to paths.sh  
2018-04-16 09:01:49 [.] Creating socket directory /var/run/redislabs  
2018-04-16 09:01:49 [.] Writing socket directory to /etc/opt/redislabs/config.js  
on  
2018-04-16 09:01:50 [?] Swap is enabled. Do you want to proceed? [Y/N]? y
```

Transaction Summary

=====

Install 2 Packages

Total size: 164 M

Installed size: 164 M

Downloading packages:

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Installing : redislabs-5.0.2-15.rhel7.x86_64 1/2

Firewalld is not running

Installing : redislabs-utils-5.0.2-15.rhel7.x86_64 2/2

Verifying : redislabs-5.0.2-15.rhel7.x86_64 1/2

Verifying : redislabs-utils-5.0.2-15.rhel7.x86_64 2/2

Installed:

redislabs.x86_64 0:5.0.2-15.rhel7 redislabs-utils.x86_64 0:5.0.2-15.rhel7

Complete!

2018-04-16 09:03:31 [?] Do you want to automatically tune the system for best performance [Y/N]? ☐

```
2018-04-16 09:06:45 [.] Removing systune from rc.local if necessary.
2018-04-16 09:06:45 [?] Cluster nodes must have their system time synchronized.
Do you want to set up NTP time synchronization now [Y/N]? y
2018-04-16 09:08:37 [.] Making sure NTP is installed and time is set.
2018-04-16 09:08:38 [!] Chrony service is already installed, skipping NTP installation.
2018-04-16 09:08:38 [$] executing: 'systemctl enable chronyd.service'
Created symlink from /etc/systemd/system/multi-user.target.wants/chronyd.service to /usr/lib/systemd/system/chronyd.service.
2018-04-16 09:08:38 [$] executing: 'systemctl start chronyd.service'
2018-04-16 09:08:38 [$] executing: 'chronyc -a makestep'
200 OK
2018-04-16 09:08:38 [?] This machine seems to have a firewall installed.
Would you like to open the ports utilized by Redis Enterprise on this machine's default firewall zone [Y/N]? y
2018-04-16 09:08:42 [$] executing: '/bin/firewall-cmd --add-service=redislabs'
Firewalld is not running
2018-04-16 09:08:42 [$] executing: '/bin/firewall-cmd --add-service=redislabs --permanent'
Firewalld is not running
2018-04-16 09:08:43 [!] Note: Log files will be stored on the root file system, in path /var/opt/redislabs/log

RedisLabs rest-api documentation has been deployed in /usr/share/doc/redislabs .

2018-04-16 09:08:44 [!] Installation is complete!
2018-04-16 09:08:44 [?] Would you like to run rlcheck to verify proper configuration? [Y/N]? y
2018-04-16 09:08:45 [$] executing: '/opt/redislabs/bin/rlcheck --suppress-tests=verify_bootstrap_status,verify_processes,verify_pidfiles'
Saving to file: /var/opt/redislabs/log/rlcheck.log
(Remark : Will stop on first failure)
##### Welcome to RedisLabs Enterprise Cluster settings verification utility #####
Skipping test: verify_bootstrap_status
Skipping test: verify_processes
Running test: verify_dmcproxy
PASS
Running test: verify_port_range
PASS
Skipping test: verify_pidfiles
Running test: verify_capabilities
PASS
Running test: verify_existing_sockets
PASS

Summary:
-----
ALL TESTS PASSED.

2018-04-16 09:08:46 [$] executing: 'chown redislabs:redislabs /var/opt/redislabs/log/rlcheck.log'
2018-04-16 09:08:46 [!] Please logout and login again to make sure all environment changes are applied
```


Summary:

ALL TESTS PASSED.

2018-04-16 09:08:46 [\$] executing: 'chown redislabs:redislabs /var/opt/redislabs/log/rlcheck.log'

2018-04-16 09:08:46 [!] Please logout and login again to make sure all environment changes are applied.

2018-04-16 09:08:46 [!] Point your browser at the following URL to continue:

2018-04-16 09:08:46 [!] <https://192.168.1.109:8443>

2018-04-16 09:08:46 [\$] executing: 'chmod 644 /tmp/install.log'

2018-04-16 09:08:46 [\$] executing: 'chown redislabs:redislabs /tmp/install.log'

2018-04-16 09:08:46 [!] Calling leash.py off

Cluster is not ready

[root@stationINS opt]#

redis enterprise

by redis labs

Version 5.0.2-15

Setup

© 2018 Redis Labs, Inc. All rights reserved.

node configuration

Persistent storage path	<input type="text" value="/var/opt/redislabs/persist"/>	Free = 23.32 GB																		
Ephemeral storage path ⓘ	<input type="text" value="/var/opt/redislabs/tmp"/>	Free = 23.32 GB																		
<input type="checkbox"/> Enable flash storage support ⓘ Read more																				
IP Addresses Usage ⓘ Read more	<table><thead><tr><th>IP Address</th><th>Interface</th><th>Internal traffic ⓘ</th><th>External traffic ⓘ</th></tr></thead><tbody><tr><td colspan="4">IPv4</td></tr><tr><td>192.168.1.109</td><td>ens33</td><td><input checked="" type="radio"/></td><td><input checked="" type="checkbox"/></td></tr><tr><td colspan="4"><div><div>+</div></div></td></tr></tbody></table>				IP Address	Interface	Internal traffic ⓘ	External traffic ⓘ	IPv4				192.168.1.109	ens33	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>	<div><div>+</div></div>			
IP Address	Interface	Internal traffic ⓘ	External traffic ⓘ																	
IPv4																				
192.168.1.109	ens33	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>																	
<div><div>+</div></div>																				
Cluster configuration	<div><input checked="" type="radio"/> Create new cluster <input type="radio"/> Join cluster</div> <div>Cluster name (FQDN) Read more <input type="text" value="mycluster"/></div> <div><input type="checkbox"/> Enable private & public endpoints support Read more <input type="checkbox"/> Enable rack-zone awareness Read more</div>																			

Next

cluster authentication

If you purchased a cluster key please insert it in the field below and click Next.
If you would like to use the free version just click Next.

Cluster key

By clicking "Next" you agree to the [terms and conditions](#) associated with this product.

Back Next

redis enterprise | by redis labs

set admin credentials

Email

walter.gonzales@edu.com.pe

Password

.....



Verify password

.....

Back

Next

○ please wait...

Creating cluster...

redis enterprise

cluster nodes da

gonzales@edu.com.pe



documentation Support Sign Out

create new database



redis database



memcached database

Runs on

RAM ▼

Deployment

Single Region ▼

Next

create database

Name	<input type="text" value="redis1db"/>
Protocol	Redis
Runs on	RAM
Memory limit (GB) Read more	<input type="text" value="0.1"/> GB 1.37 GB RAM unallocated
<input type="checkbox"/> Replication i	
<input type="checkbox"/> Redis Modules	None
Data persistence	<input type="text" value="None"/> ▼
Redis password	<input type="password"/> 🔑
Endpoint port number	<input type="text" value="14000"/>
<input type="checkbox"/> Database clustering Read more	
Data eviction policy i	<input type="text" value="volatile-lru"/> ▼
<input type="checkbox"/> Replica of i Read more	
<input type="checkbox"/> SSL Authentication	
<input type="checkbox"/> Periodic backup	

☐ Replica of ⓘ
Read more

☐ SSL Authentication

☐ Periodic backup

Alerts ⓘ

<input type="checkbox"/> Dataset size has reached	<input type="text"/>	% of the memory limit
<input type="checkbox"/> Throughput is higher than	<input type="text"/>	RPS (requests per second)
<input type="checkbox"/> Throughput is lower than	<input type="text"/>	RPS (requests per second)
<input type="checkbox"/> Latency is higher than	<input type="text"/>	msec
<input type="checkbox"/> Periodic backup has been delayed for longer than	<input type="text"/>	minutes

☐ Receive email alerts ⓘ

Cancel

Activate

[Hide advanced options](#)

database

redis1db ▼



metrics slowlog configuration

Activated on	4/15/2018 11:45:44 PM
Last changed	4/15/2018 11:45:44 PM
Protocol	Redis
Runs on	RAM
Endpoint ⓘ Get Replica of source URL	redis-14000.mycluster:14000 / 192.168.1.109:14000
Version	Redis version compliance 4.0.2
Memory	Memory limit 0.10 GB Used memory
Replication	Disabled
Redis Modules	None
Persistence	None
Redis password	
Database clustering	None

Google Traductor

Installing the Package | R

Enterprise-Class Redis for

←

→

↻

No es seguro | <https://192.168.1.109:8443/#/bdb/tabs/conf/1>

☆

⋮

Memory	Memory limit 0.10 GB
	Used memory 1.8 MB
Replication	Disabled
Redis Modules	None
Persistence	None
Redis password	
Database clustering	None
Eviction policy	volatile-lru
Replica of Read more	None
SSL Authentication	None
Periodic backup	None
Alerts	None

↶

Edit

Delete

Import

Export

Windows

Search

Task View

Shopping

Mail

File Explorer

Google Chrome

PowerPoint

Edge

Calendar

Photos

ESP

18:46

15/04/2018

4

#COMANDOS DE USO GENERAL

#para conectarse y ejecutar comandos use:

```
$redis-cli
```

#eliminando todo el contenido de la base de datos

```
>flushall
```

#Trabajando en un simple instancia

```
>set taco:1000 "Hola como estas"
```

```
>get taco:1000
```

```
>flushall
```

#para salir de redis use quit

```
>quit
```

#para limpiar console

```
>clear
```

#Para matar el proceso de server

```
$redis-cli -p 14000 shutdown
```

REDIS - ARQUITECTURA DE REDIS

ARQUITECTURA DE REDIS.

1. Vista general de la arquitectura.

2. Simple Instancia.

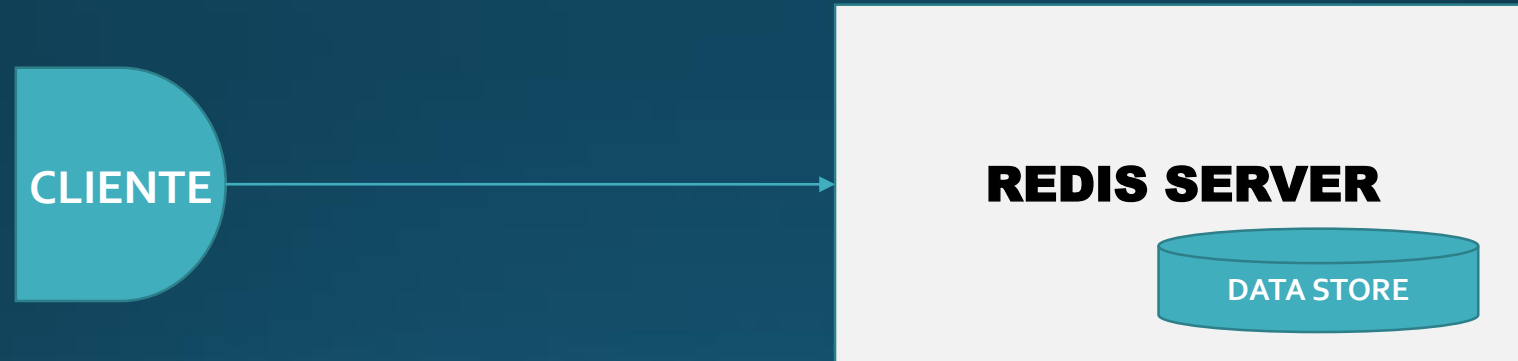
3. Múltiple instancia.

- Master.
- Esclavo.

- Redis almacena un par de datos en memoria Clave – Valor.
- Clave – Valor, son almacenados en la memoria primaria (RAM).
- Las bases de datos tradicionales almacenan la data en memoria secundaria(disk).
- La memoria primara es mucho más rápido, pero más costosa.
- El inconveniente es que Redis no puede ser usado para almacenar archivos grandes. Pero en vez de ello se basa en el acceso rápido a los conjuntos de datos más pequeños.
- Altamente escalable.

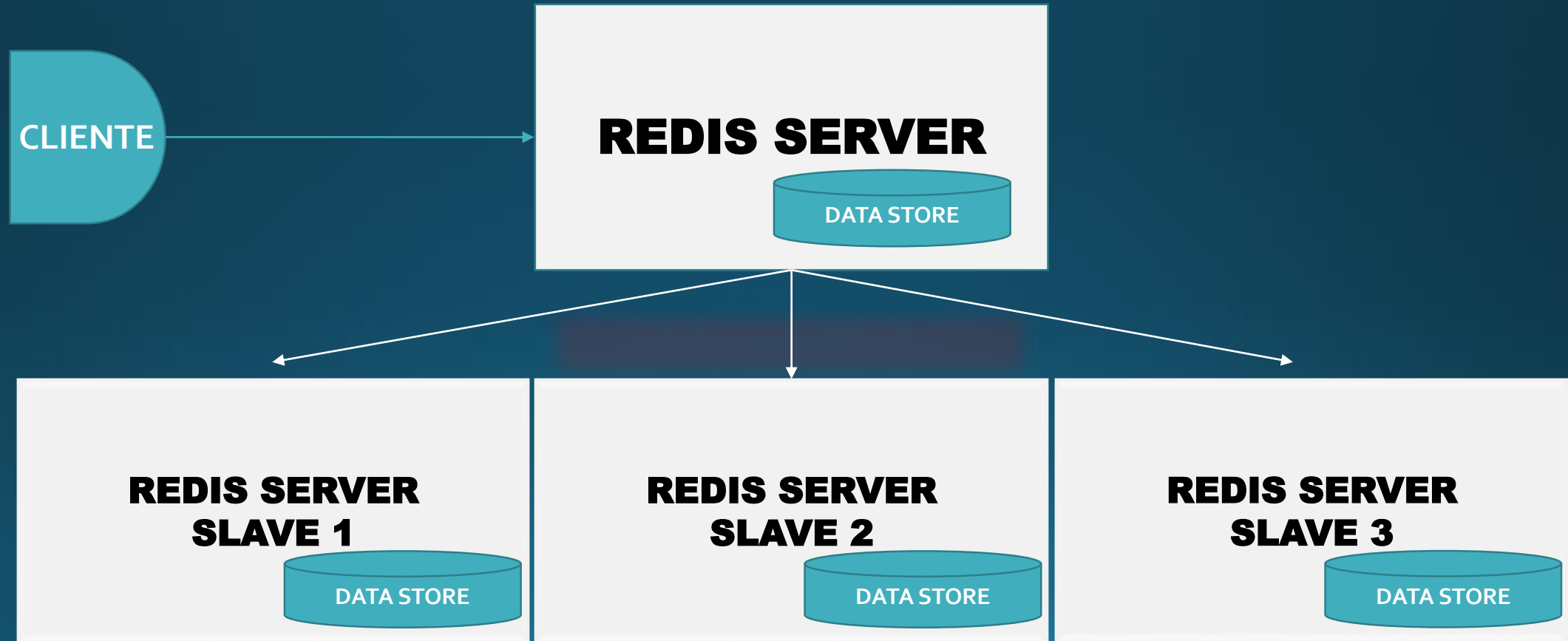
SIMPLE INSTANCIA

En la simple instancia existe un redis cliente que se comunica a in redis server que contiene almacén de data.



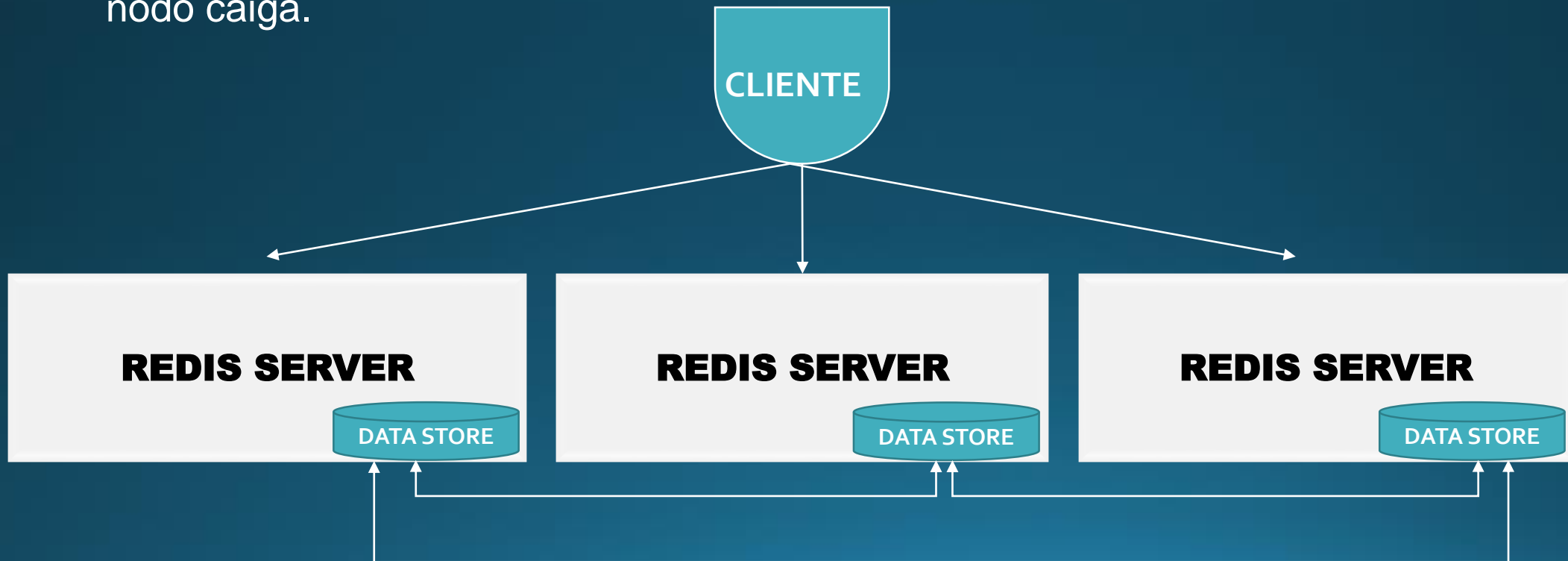
MULTIPLE INSTANCIA MASTER/SLAVE

1. Cliente se comunica con un master redis.
2. Redis master informa a los esclavos a través de la replicación(1:muchos)



MULTIPLE INSTANCIA CLUSTER

1. Cliente se comunica con 1:muchos - redis server
2. Los Redis servers se comunican entre sí compartiendo la data compartida a través de los servers
3. Cada nodo es master y slave a la vez de modo que permita la replicación de la data, garantizando así la disponibilidad de la data aun cuando algún nodo caiga.



BENEFICIOS - MÚLTIPLE INSTANCIA CLUSTER

1. La capacidad de dividir automáticamente el conjunto de datos entre varios nodos.
2. La capacidad de continuar las operaciones cuando un subconjunto de los nodos está teniendo errores o no se puede comunicar con el resto del clúster.
3. Mayor rendimiento: el rendimiento aumenta de manera lineal a medida que aumenta el número de particiones.
4. Mayor tamaño de memoria: aumenta de manera lineal a medida que aumenta el número de particiones.

REDIS - TIPOS DE DATOS

TIPOS DE DATOS PARA ALMACENAR INFORMACIÓN

Redis no es repositorio de clave valor, actualmente gestiona data estructurada, soportando diferentes tipos de valores. La siguientes lista los define los diferentes tipo que redis puede usar:

1. Strings.
2. Lists.
3. Sets
4. Hashes
5. Sorted Sets.
6. Bitmaps

STRING

1. String son los mas comunes y básico de tipos de datos en redis.
2. Redis puede almacenar tipo de dato que es serializable.
3. Operaciones comunes: GET, MGET, GETSET, SET MSET, INCR, DECR.
4. Casos de uso: el mas común tipo de dato usado para cualquier cosa donde crea conveniente.

SET

1. Redis Set es un colección desordenada de elementos.
2. No permite repeticiones de los miembros.
3. Set maneja operaciones como : unión , intersección diferencia.
4. Operaciones comunes : SADD, SCRAD, SISMEMBER, SDIFF, SINTER, SUNION, SPOP, SREM.
5. Uso : Rastrear IP que acceden al web site

LIST

1. Lista de string que son ordenados basado en el orden de ingreso.
2. La lista puede soportar sobre 4 billones de elementos por lista.
3. Operaciones comunes: BLPOP, BRPOP, LINDEX, LLEN, LPOP, RPOP, LPUSH, RPUSH, LREM, LRANGE.
4. Caso de Uso: Twitter la usa para los tweets posteados de los usuarios

HASHES

1. Son mapeos de campos y valores string
2. Ideal para representar Objetos.
3. Campos y valores pueden ser accedido simultáneamente o individualmente.
4. Operaciones comunes : HDEL,HEXIST, HGET, HGETALL,HLEN,HKEYS,HSET,HMSET,HVALS.
5. Uso : Para representación de objetos

SORTED SET

1. Redis sorted set son sets que tienen un score asociado a cada miembro.
2. Rápido acceso a la data almacenada en cualquier ubicación del sorted set.
3. El orden es calculado en la siguiente premisa.
 - **A > B** Si **A.score** es > **B.score**.
 - Si **A.score = B.score**, entonces **A > B** Si **A** alfanumérico es mas que **B**
4. Operaciones comunes: ZADD,ZCARD,ZCOUNT,ZINCRBY,ZRANGE,ZRANK,ZREM,ZSCORE.
5. Caso: Gestión de score para lideres o vendedores

<http://redis.io/topics/data-types-intro>

<http://redis.io/topics/data-types>^s

SORTED SET

1. Redis sorted set son sets que tienen un score asociado a cada miembro.
2. Rápido acceso a la data almacenada en cualquier ubicación del sorted set.
3. El orden es calculado en la siguiente premisa.
 - **A > B** Si **A.score** es > **B.score**.
 - Si **A.score = B.score**, entonces **A > B** Si **A** alfanumérico es mas que **B**
4. Operaciones comunes: ZADD,ZCARD,ZCOUNT,ZINCRBY,ZRANGE,ZRANK,ZREM,ZSCORE.
5. Caso: Gestión de score para lideres o vendedores

LINK DE REFERENCIAS

<http://redis.io/topics/data-types-intro>

<http://redis.io/topics/data-types>

REDIS - LÍNEA DE COMANDOS

#COMANDOS DE USO GENERAL

#para conectarse y ejecutar comandos use:

```
$redis-cli
```

#eliminando todo el contenido de la base de datos

```
>flushall
```

#Trabajando en un simple instancia

```
>set taco:1000 "Hola como estas"
```

```
>get taco:1000
```

```
>flushall
```

#para salir de redis use quit

```
>quit
```

#para limpiar console

```
>clear
```

#Para matar el proceso de server

```
$redis-cli -p 14000 shutdown
```

```
# -----TRABAJANDO CON STRING-----  
>get taco:1000  
>set taco:1000 "Hola como estas"  
>get taco:1000  
>mset taco:1000 "hola mundo, estan bien" taco:1001 "Estoy feliz"  
>mget taco:1000 taco:1001  
>setex taco:2001 20 "esto expirara en 60 segundos"  
>get taco:2001  
>ttl taco:2001  
>set numero "0"  
>INC numero  
>INC numero  
>INC numero  
>INC numero  
>INCRBY numero 50  
>get numero  
>DECR numero  
>DECR numero  
>DECRBY numero 50  
>get numero
```

```
# -----TRABAJANDO CON LIST-----  
#Agregando elementos a la lista  
>lpush departamentos "lima" "la libertad" "piura" "loreto"  
#Agregando un elemento duplicado.  
>lpush departamentos "loreto"  
#obteniendo el total de elementos de la lista.  
>lrange departamentos 0 -1  
#obteniendo los primeros elementos de la lista.  
>lrange departamentos 0 1  
#Agregando un elemento al inicio de la lista  
>lpush departamentos "Cerro de Pasco"  
#obteniendo los elementos de la lista  
>lrange departamentos 0 -1  
#Agregando un elemento al final de la lista  
>rpush departamentos "Madre de Dios"  
#eliminado elementos de la lista. lrem key count value.  
#count > 0: Remove elements equal to value moving from head to tail. Elimina la  
cantidad de elemtnos(count) que coincide con "value", pero de arriba hacia abajo.  
#count < 0: Remove elements equal to value moving from tail to head. Elimina la  
cantidad de elemtnos(count) que coincide con "value", pero de abajo hacia abajo.
```

```
# -----TRABAJANDO CON LIST-----  
#count = 0: Remove all elements equal to value.  
>lpush departamentos "Cusco" "Lima" "Cusco" "Arequipa" "Cusco"  
>lrem departamentos -2 "Cusco"  
#Obteniendo la cantidad de elementos  
>llen departamentos  
#Obteniendo el primer elemento de la lista.  
>lindex departamentos 0  
  
#Obteniendo el ultimo elemento de la lista  
>lindex departamentos -1  
  
#eliminando el primer elemento de la lista  
>lpop departamentos  
  
#eliminando el ultimo elemento de la lista  
>lpop departamentos  
#liberando memoria  
>flushall  
#Validando liberación  
>lrange departamentos 0 -1
```

-----TRABAJANDO CON SET-----

#Gestionando los conjuntos(SET) de datos, inserción aleatoria

>sadd departamentos "Piura" "Lima" "Pasco" "Madre de Dios"

#listando los miembros de conjunto de datos, se observa el ingreso aleatorio

>smembers departamentos

#Eliminando un registro el ordenamiento se reestructura

>srem departamentos "Loreto"

#Agregando un nuevo registro, no me permite el ingreso

>sadd departamentos "Loreto"

#Obteniendo la cantidad de elementos

>scard departamentos

#Preguntando si un elemento esta en el conjunto de datos 0=no 1=si

>sismember departamentos "Piura"

#Sacando al primer elemento de la lista

>spop departamentos

>smembers departamentos

#elimine todos los elementos del conjunto.

#agregar elementos a la lista

>sadd departamentos01 "Piura" "Lima" "Pasco" "Madre de Dios"

>sadd departamentos02 "Piura" "Lima" "Huanuco"

>sadd departamentos03 "Pasco" "Lima" "Huancayo"

#diferencia departamentos01,departamentos02

>sdiff departamentos01,departamentos02

#intersección de conjuntos set

>sinter departamentos01,departamentos02

#unión de conjuntos set

>sunion departamentos01,departamentos02

-----TRABAJANDO CON HASHSET-----

#Usando el HashSet para multiples hash <expresion> key [field(key') value [...]]

>hmset residencia Jose Lima Pedro Piura Miguel Huacho

#Insertando un elemento al hash multiple. hset key field value

hset residencia Wally barranca

#Inserten los mismo datos que pasa?. hset key field value

hset residencia Wally barranca

#Obteniendo el valor de hash <expresion> key field.

>hget residencia Jose

#Obteniendo el valor de hash,lugar de residencia, de todas las campos. hgetall key

>hgetall residencia

#Validando si existe un elemento. hexists key field

>hexists residencia Jose

#Obteniendo los key de todos los hash: hkeys key

>hkeys residencia

#Obteniendo los values de todos los hash: hvals key

>hvals residencia

#liberando memoria

>flushall

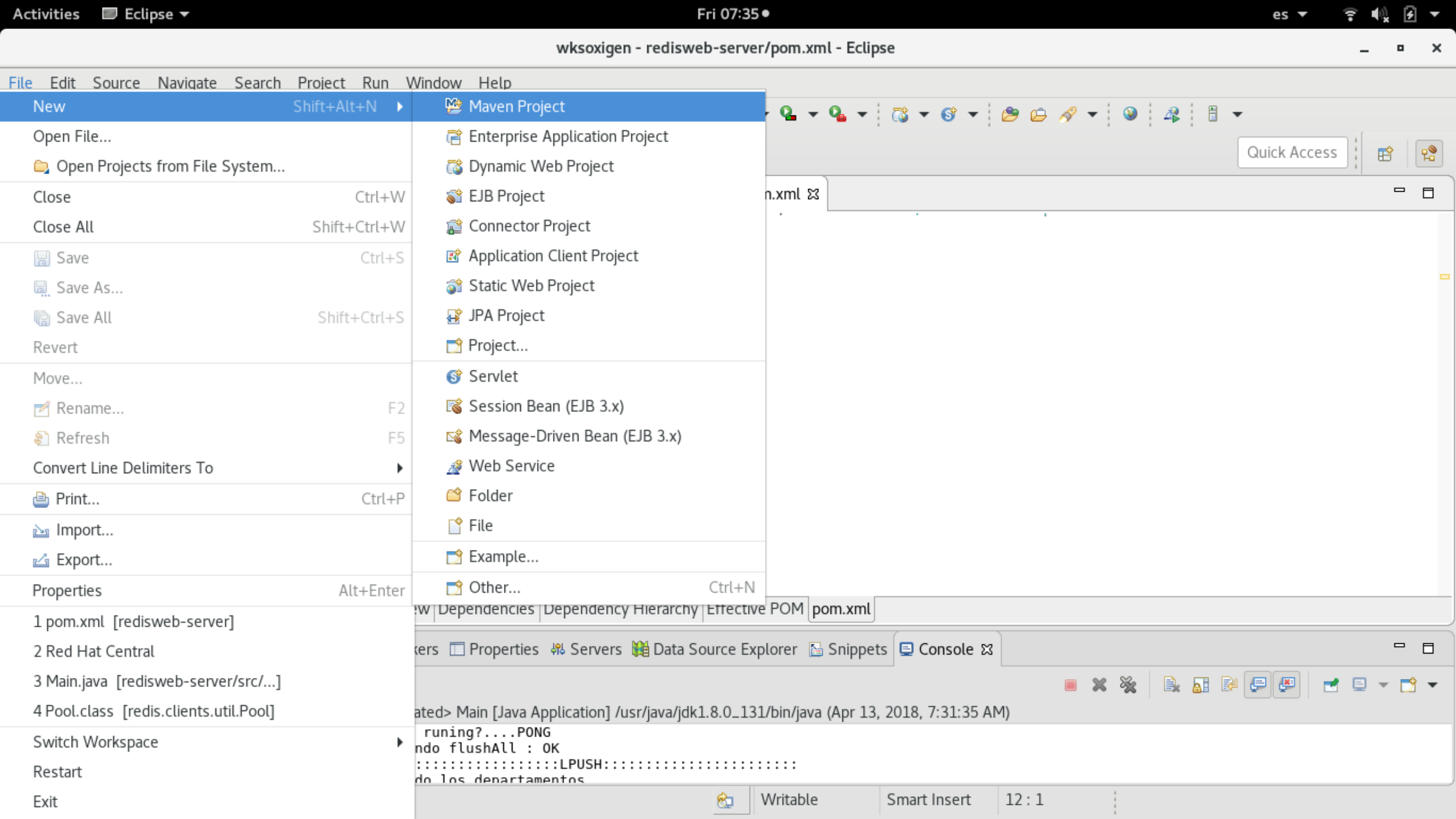
```
# -----TRABAJANDO CON SORTED SET-----  
#usando el zxx para gestionar elementos, cardinalidad y score de elementos en  
conjunto ordenado(sorted set).  
#zadd key [NX|XX] [CH] [INCR] score member [score member ...]  
#creando una hash con score asignados a los elementos. Si tienen el mismo score se  
usa el valor ordenando en base al alfabeto  
>zadd departamentos 1 "Lima" 1 "Piura" 2 "La Libertad" 3 "Lambayeque"  
#listando elementos del conjunto ordeando. zrange key start stop [WITHSCORES]  
>zrange departamentos 0 -1  
#Listando el primer elemento  
>zrange departamento 0 0  
#listando elementos con su score  
>zrange departamentos 0 -1 WITHSCORES  
#mostrando la cardinalidad, cantidad de elementos. zcard key  
>zcard departamentos  
#Mostrando la cantidad de elementos del conjunto ordenado (sorted set), definiendo  
su score. zcount key min max  
>zcount departamentos 1 3  
#Mostrando la cantidad de elementos con score 1 y 2  
>zcount departamentos 1 2
```

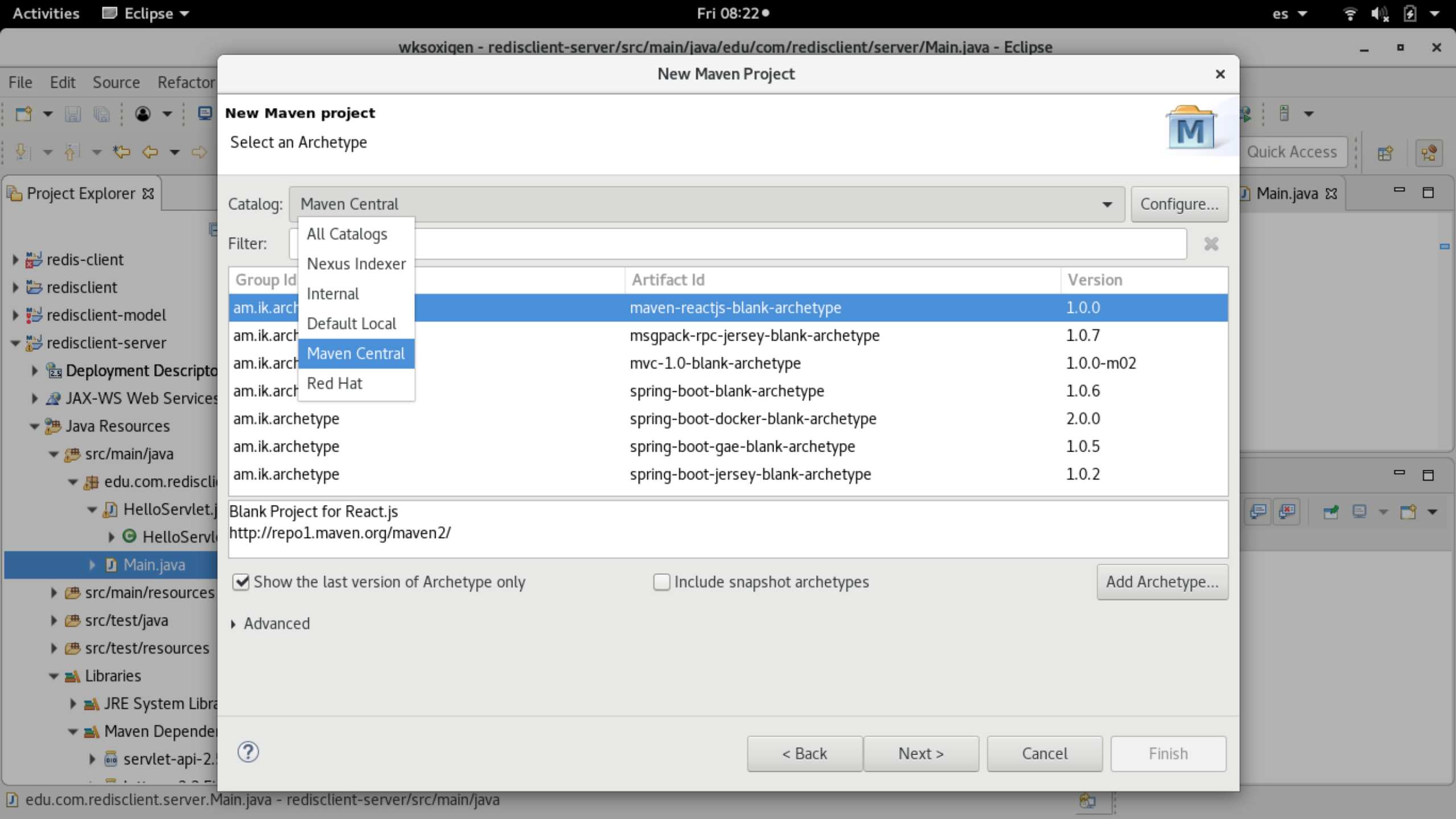


```
# -----TRABAJANDO CON SORTED SET-----  
#Mostrando la cantidad de elementos sin incluir los de score 1  
>zcount departamentos 2 3  
>zcount departamentos (1 3  
#mostrando el total de elementos del sorted set. zcount key min max  
>zcount departamentos -inf +inf  
#aumentar el score a un elemento del conjunto.  
#mostramos los elementos con su score  
>zrange departamentos 0 -1 WITHSCORES  
#Para aumentar el score de uno de los elementos o miembros. zincrby key increment  
member  
>zincrby departamentos 1 Lima  
#Vuelva a listar  
>zrange departamentos 0 -1 WITHSCORES  
#Buscar la posición actual de un elemento. zrank key member  
>zrank departamentos Lima  
#Mostrar el score de un elemento o miembro. zscore key member.  
>zscore departamentos Lima  
#Remover elemento o miembro. zrem key member [member ...]  
>zrem departamento Lima
```

REDIS - CLIENTE

1. Jedis es un cliente open source usado por java para realizar operaciones con redis.
2. Es uno de los clientes Java mas usados en la actualidad para interactuar con redis.
3. Soporte las todas las operaciones básicas de redis y también las avanzadas como clustering y temas de alta disponibilidad.
4. <https://redis.io/clients#java>





New Maven Project

New Maven project

Select an Archetype

Catalog: Maven Central

Filter:

- All Catalogs
- Nexus Indexer
- Internal
- Default Local
- Maven Central
- Red Hat

Group Id	Artifact Id	Version
am.ik.arch	maven-reactjs-blank-archetype	1.0.0
am.ik.arch	msgpack-rpc-jersey-blank-archetype	1.0.7
am.ik.arch	mvc-1.0-blank-archetype	1.0.0-m02
am.ik.arch	spring-boot-blank-archetype	1.0.6
am.ik.archetype	spring-boot-docker-blank-archetype	2.0.0
am.ik.archetype	spring-boot-gae-blank-archetype	1.0.5
am.ik.archetype	spring-boot-jersey-blank-archetype	1.0.2

Blank Project for React.js
<http://repo1.maven.org/maven2/>

☒ Show the last version of Archetype only

☐ Include snapshot archetypes

Add Archetype...

Advanced



< Back

Next >

Cancel

Finish



- redis-client
- redisclient
- redisclient-model
- redisclient-server
 - Deployment Descriptors
 - JAX-WS Web Services
 - Java Resources
 - src/main/java
 - edu.com.redisclient.server
 - HelloServlet.java
 - HelloServletTest.java
 - Main.java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - Libraries
 - JRE System Library
 - Maven Dependencies
 - javax.servlet-api-2.5.jar

New Maven Project

New Maven project

Select an Archetype



Catalog: Maven Central

Configure...

Filter: basic-w

Group Id	Artifact Id	Version
com.dyuproject.protostuff.archetype	basic-webapp	1.0.9
org.appfuse.archetypes	appfuse-basic-wicket-archetype	3.5.0

webapp archetype using protostuff, json and jquery
<http://repo1.maven.org/maven2/>

☒ Show the last version of Archetype only☐ Include snapshot archetypes

Add Archetype...

Advanced



< Back

Next >

Cancel

Finish

LINK DE REFERENCIAS

<https://redis.io/clients#java>

GRACIAS