

Antes de continuar, vamos a presentar una característica disponible desde la versión 4.0, a través de la cual incorporar nuevos comandos a Redis: los módulos. Veremos cómo cargarlos y usarlos, no así cómo crearlos, pues quedan fuera del ámbito del curso.

La lección comienza presentando el concepto de módulo. A continuación, se presenta cómo cargarlos y listar los módulos cargados. Finalmente, se muestra, como ejemplo ilustrativo, cómo compilar el módulo RedisEx que incorpora los módulos `rxkeys`, `rxstrings`, `rxhashes`, `rxlists`, `rxsets`, `rxzsets` y `rxgeo`.

Al finalizar la lección, el estudiante sabrá:

- Qué es un módulo.
- Cómo cargar módulos.
- Cómo listar los módulos cargados.
- Cómo compilar el módulo RedisEx.

Introducción

Un **módulo** (*module*) es un componente que extiende la funcionalidad de Redis añadiendo nuevos comandos. Se encuentran implementados principalmente en C.

En redismodules.com, podemos encontrar módulos implementados por Redis Labs y otras organizaciones o particulares. Entre otros, encontramos:

- **Redisearch**. Proporciona soporte para búsquedas de texto.
- **ReJSON**. Proporciona soporte para trabajar con valores de tipo JSON.
- **Redis Graph**. Proporciona soporte para grafos.
- **Redis-ML**. Proporciona soporte para modelos de aprendizaje de máquina (*machine learning*).
- **Redis Password**. Permite almacenar contraseñas.
- **RedisEx**. Proporciona varios módulos que extienden la funcionalidad de los tipos de datos nativos: `rxkeys`, `rxstrings`, `rxhashes`, `rxlists`, `rxsets`, `rxzsets` y `rxgeo`.

Carga de módulos

Para poder usar un módulo, es necesario cargarlo. Esto podemos hacerlo mediante el parámetro de configuración `loadmodule`:

```
loadmodule /ruta/del/archivo/módulo.so
```

Así, por ejemplo, si tenemos el módulo `rxstrings.so` en el directorio `/opt/redis/modules`, la importación la podemos hacer como sigue:

```
loadmodule /opt/redis/modules/rxstrings.so
```

También es posible importar el módulo en la línea de comandos, durante el arranque de la instancia. Para este fin, se dispone de la opción `--loadmodule`:

```
--loadmodule /ruta/del/archivo/módulo.so
```

La tercera opción es mediante el comando **MODULE LOAD**:

```
MODULE LOAD /ruta/del/archivo/módulo.so
```

Cuando se carga un módulo, se puede observar un mensaje en el registro de eventos similar al siguiente:

```
4744:M 09 Apr 10:16:44.284 * Module 'rxzsets' loaded from /opt/redis/modules/rxzsets.so
```

Pero cuidado, si carga el módulo dinámicamente y añade algún par clave-valor de un tipo específico de ese módulo y no carga el módulo con el parámetro `loadmodule`, muy probablemente no podrá

reiniciar la instancia, recibiendo un mensaje similar al siguiente:

```
The RDB file contains module data I can't load: no matching module 'ReJSON-RL'
```

Listado de módulos cargados

Para conocer los módulos cargados, podemos utilizar el siguiente comando:

```
MODULE LIST
```

Compilación de módulos

Generalmente, la manera correcta de trabajar con módulos consiste en descargar su código fuente, compilarlo, copiar los módulos a la carpeta de módulos de la instancia y, finalmente, importarlos o cargarlos. A continuación, vamos a presentar cómo hacerlo con el módulo **RedisEx** que incorpora los módulos **rxkeys**, **rxstrings**, **rxhashes**, **rxlists**, **rxsets**, **rxzsets** y **rxgeo**, los cuales vamos a presentar uno a uno a lo largo del curso.

Descarga del código fuente

Para descargar su código fuente, podemos utilizar por ejemplo **git**:

```
$ git clone --depth 1 https://github.com/RedisLabsModules/redux.git
```

Compilación del módulo

A continuación, hay que entrar en la carpeta del proyecto y compilar con **make**:

```
$ cd redux
$ make
```

Si todo ha ido bien, encontraremos los archivos de los módulos, los de extensión **.so**, en la carpeta **src**:

```
$ ls -l src/*.so
src/rxgeo.so
src/rxhashes.so
src/rxkeys.so
src/rxlists.so
src/rxsets.so
src/rxstrings.so
src/rxzsets.so
$
```

Copia de módulos

Lo siguiente es copiar los archivos de los módulos en la carpeta de módulos de la instancia que, por convenio y buenas prácticas, se suele usar **/opt/redis/modules**:

```
$ cp src/*.so /opt/redis/modules
```

Importación de módulos

Finalmente, podemos importarlos, por ejemplo, mediante el comando **MODULE LOAD**.

A continuación, se muestra cómo cargar el módulo **rxstrings** presentado en la lección anterior:

```
127.0.0.1:6379> MODULE LOAD /opt/redis/modules/rxstrings.so
OK
127.0.0.1:6379> MODULE LIST
1) 1) "name"
   2) "rxstrings"
   3) "ver"
   4) (integer) 1
127.0.0.1:6379>
```

A partir de este momento, ya podemos utilizar los comandos **PREPEND** y **CHECKAND**.

Si deseamos que se carguen en todo arranque de la instancia, configurar un parámetro **loadmodule** para cada módulo.