Caché

Redis

Sprint 2 Lab 3



Felipe Izquierdo Romero

Índice

[1 Descripción del problema](#DescripciónDelProblema)

[2 Desarrollo](#Desarrollo)

[3 Opcional](#Opcional)

1. Descripción del problema

Crear un sistema de notificaciones donde un profesor publica mensajes u los alumnos los visualizan.

1. Desarrollo

Para completar el ejercicio desde la consola de Docker actuará como alumno y se subscribirá al canal “TablonAnuncios” y desde una consola de comandos seremos el profesor que recibirá los mensajes.

Primero se mostrará la parte de Docker

redis-cli

Para entrar ejecutar posteriormente los comandos de redis

SUBSCRIBE TablonAnuncios

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 1- Alumno subscrito

A continuación, en una consola de comandos

docker ps

Escogemos el identificador del contenedor de Docker de redis

docker exec -it <id\_contenedor> redis-cli

Ahora se ejecuta el comando PUBLISH para escribirl el mensaje como profesor

PUBLISH TablonAnuncios 'Hola alumnos'

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 2- Profesor escribiendo un mensaje

Después si volvemos a Docker veremos que recibimos un mensaje del profesor

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 3- Mensaje recibido

1. Opcional

Siguiendo con el código del ejemplo de las clases he modificado algunas líneas para mostrar este funcionamiento y escribir mensajes como si fuera el profesor y visualizarlos en la consola de Docker o en el propio navegador.

Aunque estas modificaciones no distinguen la diferencia de un usuario final entre los rolles de alumno o de profesor.

Primero he cambiado el canal donde se conecta el programa a recibir y/o publicar mensajes.

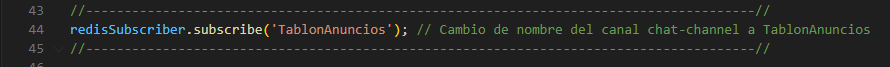


Ilustración 4- Cambio Subscripción a canal



Ilustración 5- Cambio Publicaciones al canal

Después para que se carguen los mensajes que se han escrito anteriormente en el tablón he pensado en guardar los mensajes en una lista y listar los mensajes cuando te conectas.



Ilustración 6- Guardar mensajes en una lista

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración 7- Listar mensajes

Para comprobar el funcionamiento del código ejecutaremos el servidor abriendo una consola de comandos en el directorio donde se encuentra y ejecuteremos

node server.js

Texto

Descripción generada automáticamente

Ilustración 8- Ejecutar servidor

Al entrar en <http://localhost:3000/> Veremos que no se ha cargado el mensaje publicado anteriormente ‘Hola alumnos’.

Esto es porque no hay ningún mensaje en la lista donde guardamos los mensajes al enviarlos a través de nuestro navegador.

Probaremos a poner el username de Profesor\_1 y escribir un mensaje y veremos cómo se cargan en el navegador al iniciar otra sesión nueva y en la consola de Docker que esta subscrita al canal.

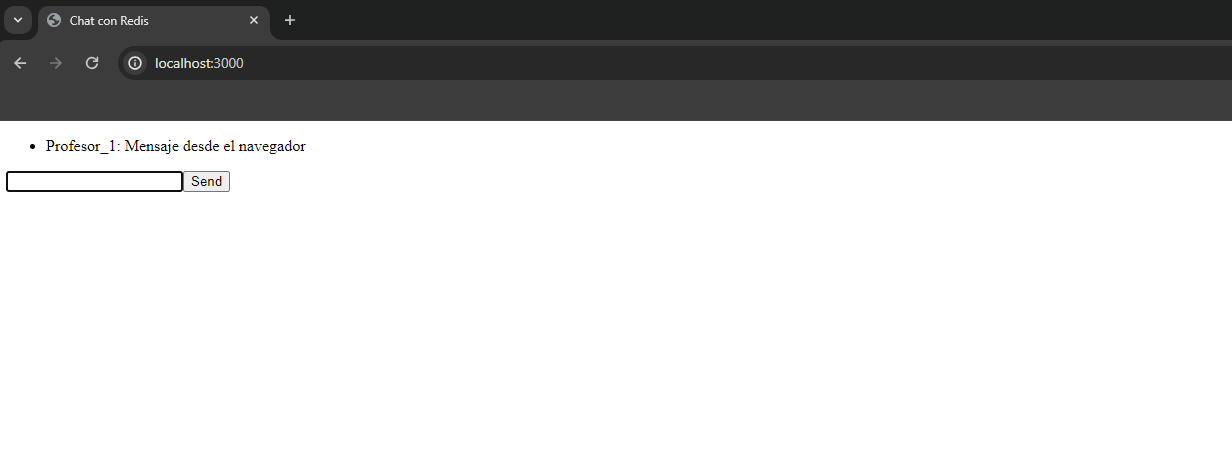


Ilustración 9- Mensaje desde el navegador

Texto

Descripción generada automáticamente

Ilustración 10- Recibido en Docker

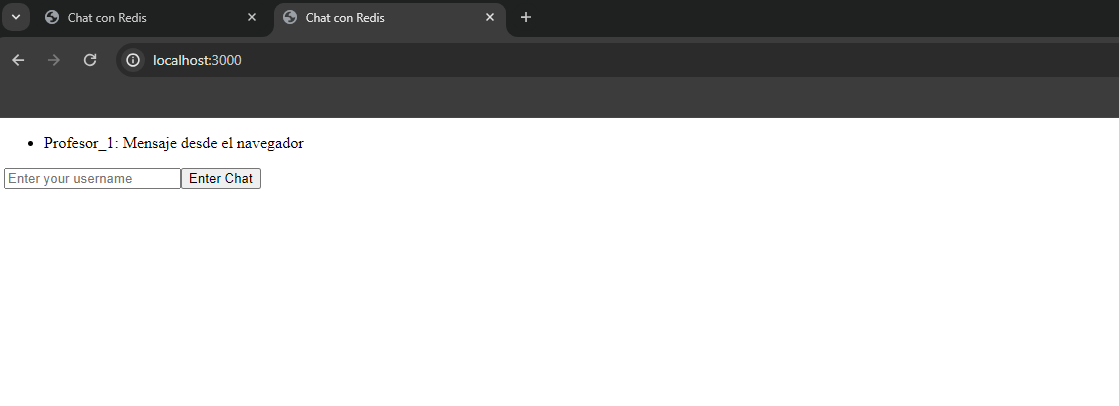


Ilustración 11- Carga de mensajes al iniciar nueva sesión