

# REDES DE COMUNICACIONES II

## BOT 2 ESPIA

*Jorge Parrilla Llamas*  
([jorge.parrilla@estudiante.uam.es](mailto:jorge.parrilla@estudiante.uam.es))

## 1. INTRODUCCIÓN

El bot realizado consiste en la implementación de un pequeño espía (Espia) que realiza un seguimiento de ciertos parámetros de búsqueda en los canales y los devuelve a un usuario en concreto.

Dicho bot se conecta a todos los canales públicos disponibles en el servidor IRC indicado a la hora de realizar la ejecución y conexión al servidor. Para iniciar su funcionamiento, un usuario puede indicarle una cadena por privado mediante la cuál el bot activará una petición de rastreo de todos los canales en los que está conectado.

En caso de que el bot detecte la cadena buscada por el usuario, informará a dicho usuario por privado de que la cadena de texto que el usuario buscaba ha sido mencionada en X canal y por Y usuario.

## 2. FORMA DE EJECUCIÓN

El propio Makefile ejecuta el bot con los parámetros necesarios una vez se realiza un “make” sobre la raíz de la carpeta del BOT 2. En caso de necesidad, se puede ejecutar el bot con el siguiente comando:

```
./bot_2/bot_2 -a localhost:6667 -n EspiaBot
```

### - Banderas de ejecución

Como se puede ver en el comando de ejecución del bot, para una correcta ejecución se requieren tres parámetros:

- **a**: Dirección y puerto de conexión al servidor IRC separados mediante dos puntos.

Ejemplo: -a metis.ii.uam.es:6667

- **n**: Nick que se mostrará públicamente para el Bot en el canal.

Ejemplo: -n SiriBot

## 3. EJEMPLO DE FUNCIONAMIENTO

Como se mencionaba anteriormente, para iniciar su funcionamiento un usuario debe escribir un mensaje privado al Bot en el que le indique la cadena que quiere rastrear, por ejemplo, de la siguiente forma:

R: prueba

Una vez se envía dicho mensaje por privado al bot, el bot registra en la terminal que le ha llegado una nueva petición de rastreo:

Podemos observar que en la terminal se muestran los datos básicos de conexión, además de los canales a los que el bot se va a conectar.

En verde se pueden ver los mensajes o logs generados por el Bot y en naranja los mensajes que provienen del servidor al Bot.

```
./bot_2/bot_2 -a metis.ii.uam.es:6667 -n EspiaBot
#####
###      BOT ESPIA - JORGE PARRILLA      ###
#####
|| Direccion:  metis.ii.uam.es
|| Puerto:     6667
|| Nick:       EspiaBot
[B] Host obtenido OK
[B] Conexión con servidor OK
[B] Mensaje nick enviado OK
[B] Mensaje user enviado OK
[S] Conexión satisfactoria: Welcome to the Internet Relay Network EspiaBot!~bot
    espia@46.222.180.122
[B] Enviando comando LIST...
[B] Enviando JOIN a #Oops...
[B] Enviando JOIN a #delegados...
[B] Enviando JOIN a #IntArt...
[B] Enviando JOIN a #redes2pr...
[B] Enviando JOIN a #redes2...
[S] Ahora estás en el canal #IntArt 0
[S] Ahora estás en el canal #redes2pr 1
[S] Ahora estás en el canal #redes2 2
[S] Nueva petición de rastreo: x0soir3 | Término: R: prueba
```

Una vez el bot comienza a rastrear en este caso el ermino “prueba”, vemos que si por ejemplo en el canal público #redes2 mencionamos dicha palabra en una frase, el bot notifica al usuario que ha solicitado dicho rastreo que la palabra ha sido mencionada en ese chat:

[19:23:1x0soir3	R: prueba
[19:26:EspiaBot	Detectado término buscado ( prueba) - Canal: #redes2 - Usuario: x0soir3 - Mensaje: esto es una prueba de logeo de mensajes

El bot a su vez logea también en la terminal que se ha encontrado una coincidencia, en caso de que el usuario que la estaba rastreando ya no esté conectado se mostrará un error por la terminal.

```
[S] Ahora estás en el canal #redes2 2
[S] Nueva petición de rastreo: x0soir3 | Término: R: prueba
[B] Coincidencia encontrada: Usuario: x0soir3 Término: prueba
```

## 4. CONCLUSIONES

Ha sido una gran experiencia realizar tanto el Bot de Siri como el bot Espia, quizá el que más me ha gustado es el Espía ya que no era tan común como un simple Siri y además es algo que se me ha ocurrido a mi sobre la marcha.

## 5. LIBRERÍA EXTERNA UTILIZADA

Para el desarrollo de este bot he utilizado una librería externa de código totalmente libre bajo la licencia MIT para la gestión de arraylist bajo punteros, a pesar de que dicha librería estaba muy completa he tenido que modificarla un poco para adaptarla a mis requerimientos.

```
/**  
 * ArrayList implementation  
 * (c) 2011 @marekweb  
 *  
 * Uses dynamic extensible arrays.  
 */
```