

# Popups

Javier Fernández Rivera - [www.aurea.es](http://www.aurea.es)

Seccion de scripting: popups

Para acceder a los popups ir a, la barra de menu del mIRC, pulsar herramientas y luego popups o tambien puedes acceder con el icono:



En este manual intentare que el lector vaya aprendiendo paso a paso el funcionamiento de los popups, pero para ello es imprescindible poder observar el resultado de los ejemplos mencionados a continuación, es necesario pues que solo aparezca tales ejemplos en el editor del mIRC, borrando así los anteriores, (una vez se vayan comprendiendo) y copiando los nuevos ejemplos. Cuando el lector alcance un conocimiento mas preciso de esta herramienta de scripting, ia puede cambiar lo que desee. Pero mientras tanto utilizar los ejemplos del manual como base de aprendizaje.

## La definición

Popups: son la herramienta que posee el mIRC para la creación de menus despleables.

Podemos crear cinco menus diferentes, cada uno de estos se visualizara en su ventana correspondiente, con lo cual haremos un menu para la ventana de : **estado**, **canal**, **privado**, **lista de nick**, **menu**. Con la diferencia que en estos, dos últimos no se da el caso de que sean ventanas.

Los resultados de la programación de los popups se pueden ver en el texto **popup** del menu perteneciente a la pestaña de **popups**, del editor del mirc.

Atendiendo a las distintas secciones del menu **ver** de la pestaña **popups** del **editor**, destacamos los siguientes menus programables del mIRC.

**Los de estado:** se programan en `herramientas\popups\ver\estado\` y se despliega al hacer click en el boton derecho sobre la ventana de estado.

**Los del canal:** se programan en `herramientas\popups\ver\canal\` y se despliega al hacer click en el boton derecho de la ventana del canal actual.

**Los del privado:** se programan en `herramientas\popups\ver\privado chat\` y se despliega al hacer click en el boton derecho de la ventana de un privado o chat actual.

**Los de la lista de nicks:** se programan en `herramientas\popups\ver\lista de nicks\` y se despliega al hacer click en el boton derecho, sobre la lista de nicks que tiene un canal en el lado derecho de la ventana.

**Los de la barra de menu:** se programan en `herramientas\popups\ver\menu\` y se despliega al hacer click en el boton izquierdo sobre el **texto-titulo** de la barra de menu situada en la parte superior del mIRC. Un ejemplo de este caso es:

**Mi script**

**Información**

**.Hora: echo -s son las \$time**

**.Dia: echo -s Hoy es \$day**

-

**Ayuda**

**.Del mirc: help**

En este ejemplo no voy a entrar a explicarlo puesto que se hará de forma mas detenida posteriormente, decir solamente que la primera linea **Mi script** seria el llamando anteriormente **texto-titulo** es lo que servirá de titulo, a la barra de menu situada en la parte superior del mIRC.

Aunque no entiendas el ejemplo, pasalo por alto y abordalo cuando ya hayas comprendido el resto del documento.

Dentro del editor del mIRC y en la pestaña **popups**, vemos como al marcar sobre el menu **ver**, se despliegan los ya dichos cinco menus programables, de los que dispone el mIRC para facilitar la tarea de acceso a los comandos. Al hacer click en uno de ellos, el mIRC nos abrirá un nuevo campo de edición sobre el cual podemos programar dicho menu marcado con anterioridad.

## La sintaxis de un popup

**Nombre del menu: comados**

Donde en **Nombre del menu** podremos el nombre que queramos ver desplegado al apretar el boton derecho sobre uno de los cinco campos programables de los popups, a continuación pondremos : que indicaran que a partir de hay empiezan los comandos que se ejecutaran al hacer click sobre el **nombre del menu** y por ultimo, despues de los dos puntos pondremos los **comandos** el primero, antepuesto con el signo / o con un espacio como esta indicado en la sintaxis.

## Practica

Vamos a empezar a crear un primer menu, para ellos vallamos a el **editor** del mIRC, a la pestaña **popups** y al campo de edición correspondiente a la ventana de **estado**. Ahora pondremos:

**Información: echo –s Son las \$time y hoy es \$day**

Veremos como al hacer click sobre la ventana de estado aparece un menu con el texto **Información** este texto es el perteneciente a lo escrito antes de los dos puntos de la sintaxis del popup del ejemplo anterior, como ya dije el texto **información** es un ejemplo, lo que quiere decir que se puede poner cualquier otro texto, lógicamente ese texto ha de ir relacionado con los comandos que se ejecutarán, Ahora bien, si apretamos sobre ese menu llamado **información** veremos como aparece el texto que es impreso por acción del comando citado a continuacion de los 2 puntos.

Ahora pasaremos a crear un segundo menu, en el mismo lugar que el ya creado:

**Información: echo –s Son las \$time y hoy es \$day**

**Ayuda: help**

Ahora si apretamos el botón derecho sobre el status nos aparecera un menú con dos líneas de texto y al hacer click sobre el texto **Ayuda** el mIRC nos abrirá el archivo de ayuda.

A continuación vamos a insertar una línea divisoría, que separará en dos campos el texto **Información y Ayuda** esto lo haremos de la siguiente forma:

**Información: echo –s Son las \$time y hoy es \$day**

-

**Ayuda: help**

Hasta ahora hemos creado menus, pero los popups también permiten la creación de submenus. La utilización de la línea divisoría se hará de igual forma en los submenus. Para hacer estos submenus haremos lo siguiente:

**Información**

**.Hora: echo –s son las \$time**

**.Dia: echo –s Hoy es \$day**

-

**Ayuda**

**.Del mirc: help**

En este ejemplo, se dan los submenus **hora** y **día** que se encuentran dentro del menu **Informacion** y por otro lado el submenu **Del mirc** que se encuentra dentro del menu **Ayuda**.

Ahora bien, para especificar la situacion de un submenu dentro de un menu

lo haremos primero poniendo el nombre del menu, a continuacion en una linea posterior, escribimos un punto **.** seguido del nombre del submenu, y así tantas veces se precise, a continuación del nombre del submenu se ponen los dos puntos, posteriormente un espacio y el comando.

Si queremos especificar dos submenus, lo haremos poniendo dos puntos pero de forma seguida **..** para tres submenus lo hacemos con **...** y de la misma forma para cuatro cinco o los que sean, iremos aumentando el numero de puntos en relación a el numero de submenus.

En bastante dificil la explicación de su estructura pero es mas facil, encontrarle la logica en un ejemplo:

**Información**

**.estado**

**..Hora: echo –s son las \$time**

**.mirc**

**..version: echo –s Tu mirc es la version \$version**

**..bits: echo –s Tu mirc tiene \$bits bits**

-

**Ayuda**

**.Del mirc: help**

En este ejemplo hay: dos menus (**información** y **ayuda**) dentro del menú información se dan dos submenus (**estado** y **mirc**) y a su vez dentro de estos otros tres submenus (**hora**, **versión** y **bits**), dentro del menu (**ayuda**) se da el submenu (**Del mirc**)

Observemos que los dos puntos **:** se escriben únicamente cuando ya se dejan de poner mas submenus, osea estos dos puntos se citan en el último submenu que es el que lleva el comando, todos los anteriores menus o submenus no portan ningún comando por lo tanto no se les ponen los **:**.

En los popups los comandos tambien pueden ser citados en forma de columna al igual que en los alias, con la utilización de las llaves.

**Información**

**.mirc: {**

**echo –s tu mirc es la versión \$version**

**echo –s este mirc tiene \$bits bits**

**}**

Esto aunque funcionaria perfectamente puesto que las llaves hacen que el mIRC lea de una forma mas ordenada y rapida los codigos, sería por un lado poco práctico y por otro lado, el principal inconveniente que tiene esto, es que en los pupups el mIRC te permite solo 30 kbs de capacidad, con los cual al poner los comandos de la forma anterior, posiblemente llegaremos a acercarnos a esa capacidad y por consiguiente caer en ese error, para evitarlo se utilizan los alias, estos portaran los comandos y luego el alias se pondra en el lugar del comando del popup, tal que asi:

Poner en el editor correspondiente a los alias

```
infomirc {  
echo -s tu mirc es la versión $version  
echo -s este mirc tiene $bits bits  
}
```

Poner en el editor correspondiente a los popups referentes a el status

**Información**  
.mirc: infomirc

Aquí, lo que hacemos es, primero creamos un alias con el nombre de **infomirc** que ejecutará los dos comandos citados a continuación de la 1ª llave del alias, En los popups escribimos en lugar de los dos comandos, el alias que porta los mismos dos comandos, osea **infomirc** . Esto hará que cuando nosotros hagamos un click en el boton derecho sobre la ventana de estado nos salga un menú que trae de texto información vamos a el y al poner el raton sobre el, nos sale un submenu que trae mirc, al apretar en el, se ejecuta el comando infomirc que en este caso es un alias creado con anterioridad . De esta forma podemos hacer unos popups mas ordenados y complejos.

Bueno con esto doy por finalizado el manual sobre popups, la mejor forma de aprender a manejarlos bien es con la práctica, osea que a por ellos.