



# Programación de ventanas “curses” bajo Qt

Ignacio Alvarez García – Diciembre 2022

## INDICE

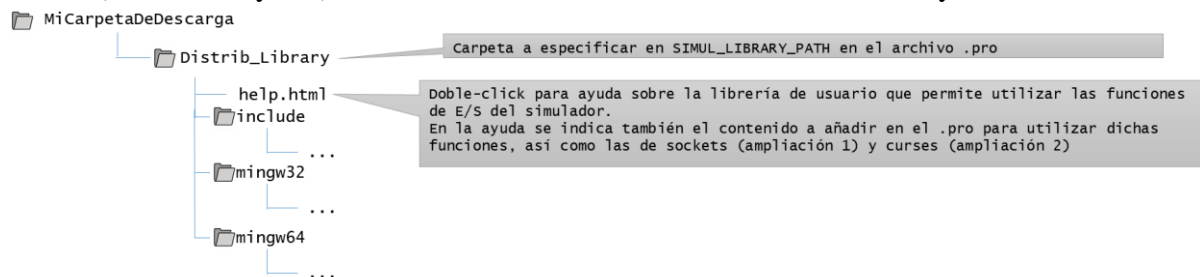
1.	Preparación del programa.....	1
2.	Inicialización de ventanas.....	1
3.	Creación de ventanas .....	1
3.1.	Creación de sub-ventanas .....	2
4.	Escritura en ventanas .....	2
5.	Entrada de teclado en ventanas.....	2
6.	Finalización .....	2
7.	Ejemplos y documentación.....	2



## 1. Preparación del programa

La librería ncurses se descarga junto al simulador del trabajo desde el enlace:  
[http://isa.uniovi.es/~ialvarez/Curso/descargas/Simulador\\_Trabajo\\_IlyC.zip](http://isa.uniovi.es/~ialvarez/Curso/descargas/Simulador_Trabajo_IlyC.zip)

Descomprimir en directorio deseado (que no incluya tildes ni ñ o similares). La librería curses (archivos .h y .dll) se encuentra en el subdirectorio Distrib\_Library



Añadir en archivo de proyecto .pro directorios de inclusión y enlazado (según el directorio de descarga elegido), y enlace a la librería pdcurses.dll, siguiendo las instrucciones de la ayuda disponible en Distrib\_Library/help.html

```
SIMUL_LIBRARY_PATH = CaminoCompletoAMiCarpetaDeDescarga/Distrib_Library
INCLUDEPATH += $$${SIMUL_LIBRARY_PATH}/include
contains(QT_ARCH, i386) {
    message("32-bit")
    LIBS += -L$$${SIMUL_LIBRARY_PATH}/mingw32 -lUserLibSimulator
} else {
    message("64-bit")
    LIBS += -L$$${SIMUL_LIBRARY_PATH}/mingw64 -lUserLibSimulator
}
# Si se desea utilizar curses:
LIBS += -lpdcurses
# Si se desea utilizar windows sockets, descomentar línea siguiente
# LIBS += -lws2_32
```

Se debe incluir “curses.h” en cada archivo .c que necesite utilizar las funciones de esta librería.

## 2. Inicialización de ventanas

```
initscr(); /* Inicializa */
cbreak(); /* Evita control-break (si se desea) */
noecho(); /* No hace eco automático de los caracteres
           pulsados, para que no aparezcan donde esté
           el cursor (si se desea) */

refresh(); /* Actualiza la pantalla */
start_color(); /* Permite colores */
init_pair(1, COLOR_YELLOW, COLOR_BLUE);
           /* Inicializa parejas de colores de carácter
           con fondo (hasta 16 parejas) */
init_pair(2, COLOR_RED, COLOR_BLACK);
...añadir más llamadas a init_pair() para todas las
combinaciones de color deseadas ...
```

## 3. Creación de ventanas

Crear una variable de tipo WINDOW\* por cada ventana a crear.  
Llamar a función newwin( ) para crear cada ventana, indicando su posición y tamaño  
Establecer atributos de la ventana con wattron( ) y wbkgd( )



Borrar contenido de la ventana con `wclear( )`

Establecer si se desea un borde para la ventana con `box( )`

```
WINDOW* wnd; /* Variable para manejar una ventana */

wnd=newwin(10,30,5,40); /* Crea nueva ventana en la fila 5,
                        columna 40, con tamaño 10 filas x 30 columnas */
wattron(wnd,COLOR_PAIR(1)); /* Asigna colores según el color establecido
                             para el índice 1 en init_pair()*/
wbkgd(wnd,COLOR_PAIR(1) | ' '); /* Llena la ventana con espacios del color
                                indicado */
wclear(wnd); /* Borra ventana */
box(wnd,ACS_VLINE,ACS_HLINE); /* Dibuja borde */
```

### 3.1. Creación de sub-ventanas

Crear una segunda variable de tipo `WINDOW*` por cada sub-ventana a crear, y llamar a la función `subwin( )` indicando la ventana “padre”. La posición y dimensiones son relativas a la ventana “padre”.

## 4. Escritura en ventanas

Utilizar la función `wprintw( )` para escribir en ventanas, con el mismo formato que `printf( )`. Previamente, es útil llamar a `wmove( )` para colocar el cursor en la posición deseada, y también se puede llamar a `wattron( )` para cambiar los colores utilizados.

Posteriormente, se debe llamar a `wrefresh( )` para que los cambios tengan efecto.

```
wmove(wnd,2,3); /* Colocar el cursor en fila 1, col 3*/
wprintw(wnd,"TEXTO y numero17=%d.",17); /* Escribir en la ventana en la
posición actual del cursor (formato como printf) */
wrefresh(wnd); /* La ventana no se dibuja de verdad hasta que se refresca */
```

## 5. Entrada de teclado en ventanas

Utilizar la función `wgetnstr( )` obtener entradas de texto en ventanas, con un funcionamiento similar a `gets( )`.

Previamente, es útil llamar a `wmove( )` para colocar el cursor en la posición deseada, y también se puede llamar a `wattron( )` para cambiar los colores utilizados.

```
char txt[20];

wmove(wnd,2,3); /* Colocar el cursor en fila 1, col 3*/
wattron(wnd,COLOR_PAIR(1));
wprintw(wnd,">> ");
wrefresh(wnd);
wattron(wnd,COLOR_PAIR(2));
wmove(wnd,2,6);
echo();
wgetnstr(wnd,txt,20); /* Espera la introducción de texto por teclado */
noecho();
```

## 6. Finalización

Terminar el sistema de ventanas llamando a la función `endwin( )`.

## 7. Ejemplos y documentación

Documentación completa en:

<https://tldp.org/LDP/lpg/node97.html>

<http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/pdf/NCURSES-Programming-HOWTO.pdf>