

# **PERIFÉRICOS Y DISPOSITIVOS DE INTERFAZ HUMANA**



## **Universidad de Granada**

### **Práctica 2**

### **Uso de bibliotecas de programación de interfaces de usuario en modo texto**

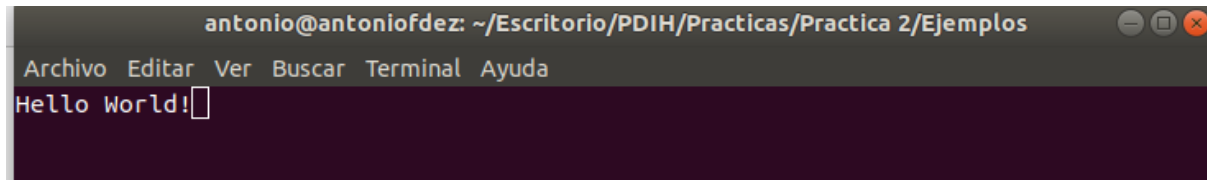
Antonio Fernández Ortega

## NCURSES Y EJEMPLOS

Primero de todo hemos instalado la librería ncurses.

```
antonio@antoniofdez:~/Escritorio/PDIH/Practicas/Practica 2$ sudo apt-get install libncurses5-dev libncursesw5-dev
```

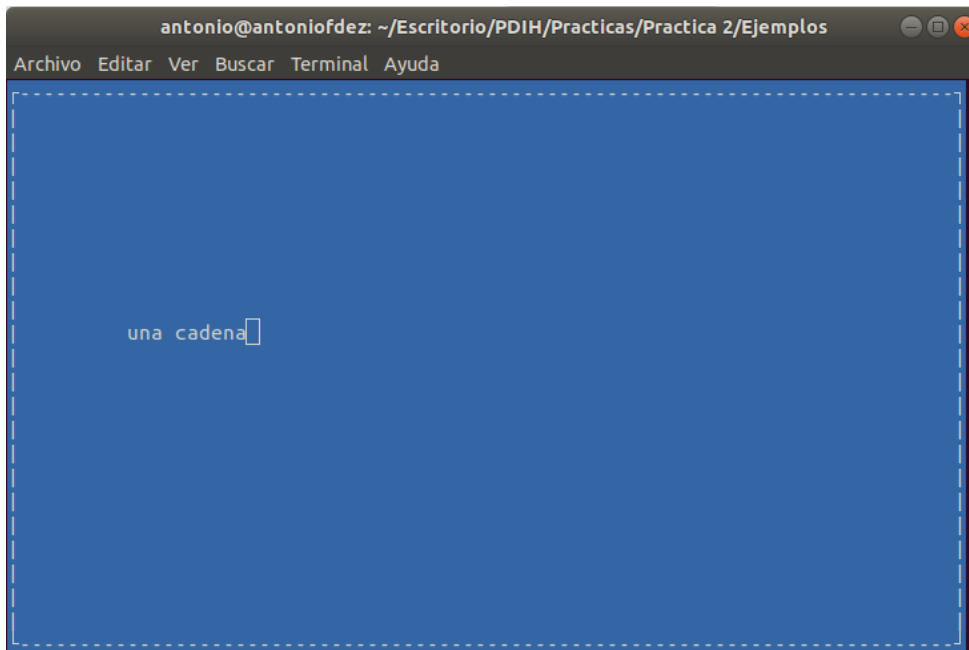
### hello.c



A terminal window titled "antonio@antoniofdez: ~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos" with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The terminal displays the text "Hello World!" followed by a cursor.

### ventana.c

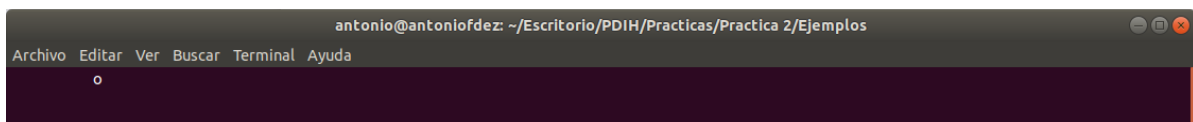
```
antonio@antoniofdez:~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos$ gcc ventana.c -lncurses -o ventana
antonio@antoniofdez:~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos$ ./ventana
antonio@antoniofdez:~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos$
```



A terminal window titled "antonio@antoniofdez: ~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos" with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). It contains a large ncurses window with a blue background and a dashed border. Inside the window, the text "una cadena" is displayed with a cursor.

### pelotita.c

```
antonio@antoniofdez:~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos$ gcc pelotita.c -lncurses -o pelotita
antonio@antoniofdez:~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos$ ./pelotita
antonio@antoniofdez:~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos$
```

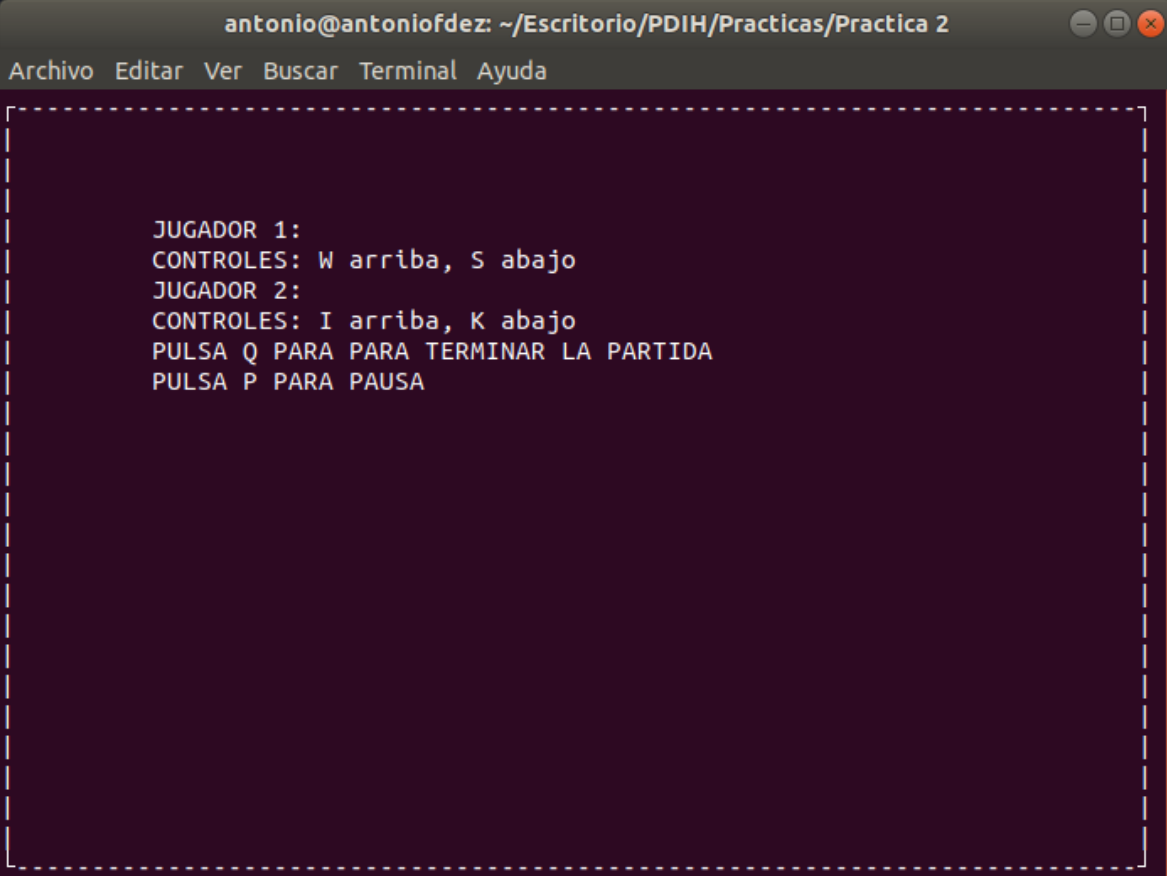


A terminal window titled "antonio@antoniofdez: ~/Escritorio/PDIH/Practicas/Practica 2/Ejemplos" with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The terminal displays a single character "o" followed by a cursor.

## pong.c

Función para mostrar la pantalla de inicio.

```
void inicio(){
    initscr();
    keypad(stdscr,true);
    noecho();
    curs_set(FALSE);
    mvprintw(4,10,"JUGADOR 1:");
    mvprintw(5,10,"CONTROLES: W arriba, S abajo");
    mvprintw(6,10,"JUGADOR 2:");
    mvprintw(7,10,"CONTROLES: I arriba, K abajo");
    mvprintw(8,10,"PULSA Q PARA PARA TERMINAR LA PARTIDA");
    mvprintw(9,10,"PULSA P PARA PAUSA");
    box(stdscr,'|','-' );
    getch();
    nodelay(stdscr,1);
    clear();
}
```

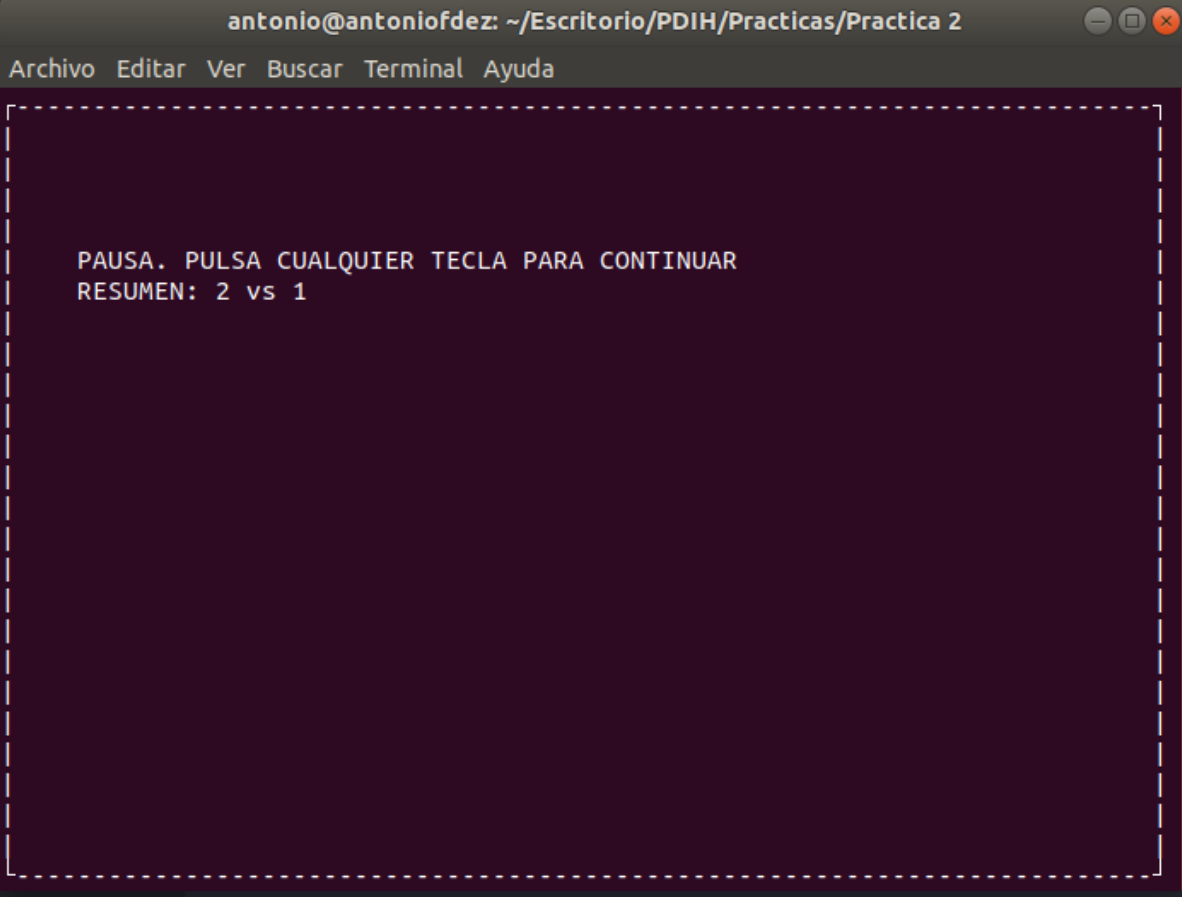


```
antonio@antoniofdez: ~/Escritorio/PDIH/Practicas/Practica 2
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

JUGADOR 1:
CONTROLES: W arriba, S abajo
JUGADOR 2:
CONTROLES: I arriba, K abajo
PULSA Q PARA PARA TERMINAR LA PARTIDA
PULSA P PARA PAUSA
```

Función para poner en pausa el juego.

```
void pausa(int puntos1, int puntos2){  
    clear();  
    mvprintw(5,5,"PAUSA. PULSA CUALQUIER TECLA PARA CONTINUAR");  
    mvprintw(6,5,"RESUMEN: %i vs %i",puntos1,puntos2);  
    box(stdscr,'|','-' );  
    nodelay(stdscr,0);  
    getch();  
}
```

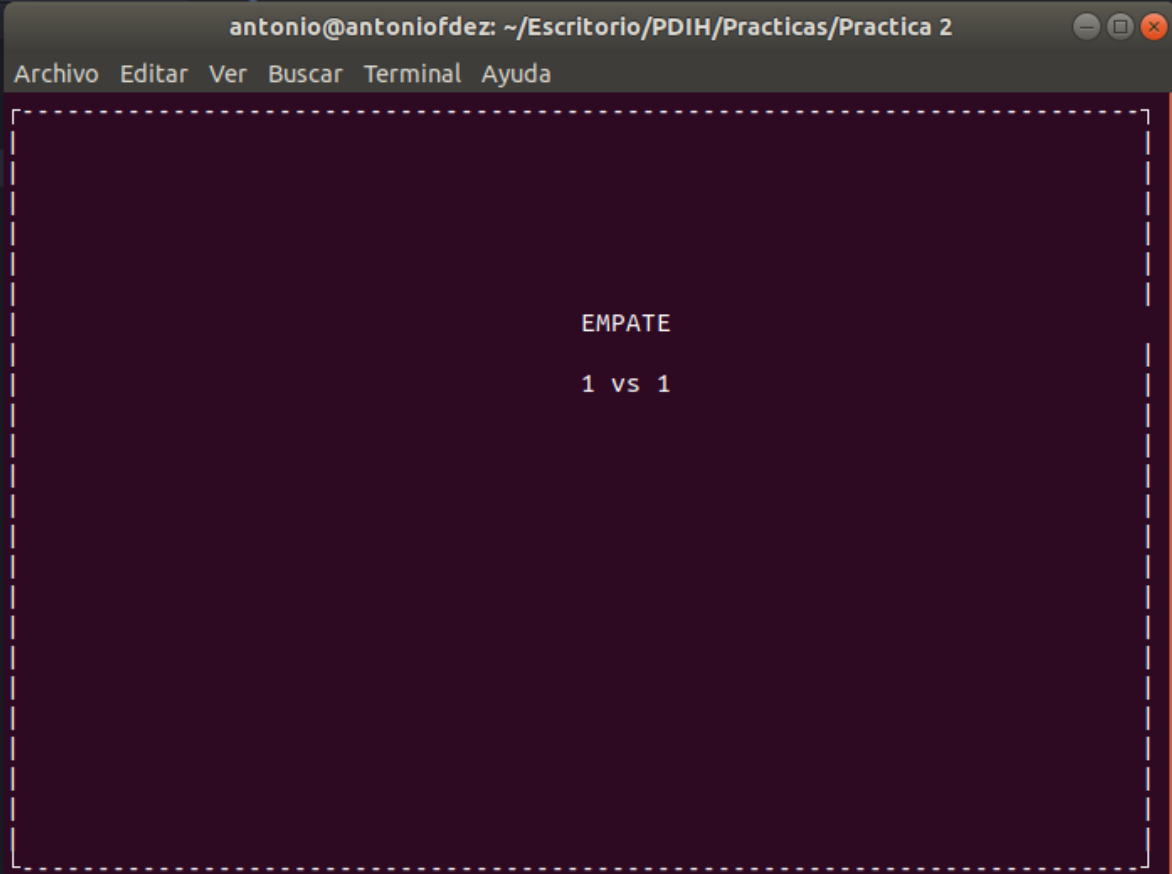


The screenshot shows a terminal window titled "antonio@antoniofdez: ~/Escritorio/PDIH/Practicas/Practica 2". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal content displays the pause screen with a dashed border. The text inside the border is:

```
PAUSA. PULSA CUALQUIER TECLA PARA CONTINUAR  
RESUMEN: 2 vs 1
```

Función para terminar la partida e indicar quién ha ganado.

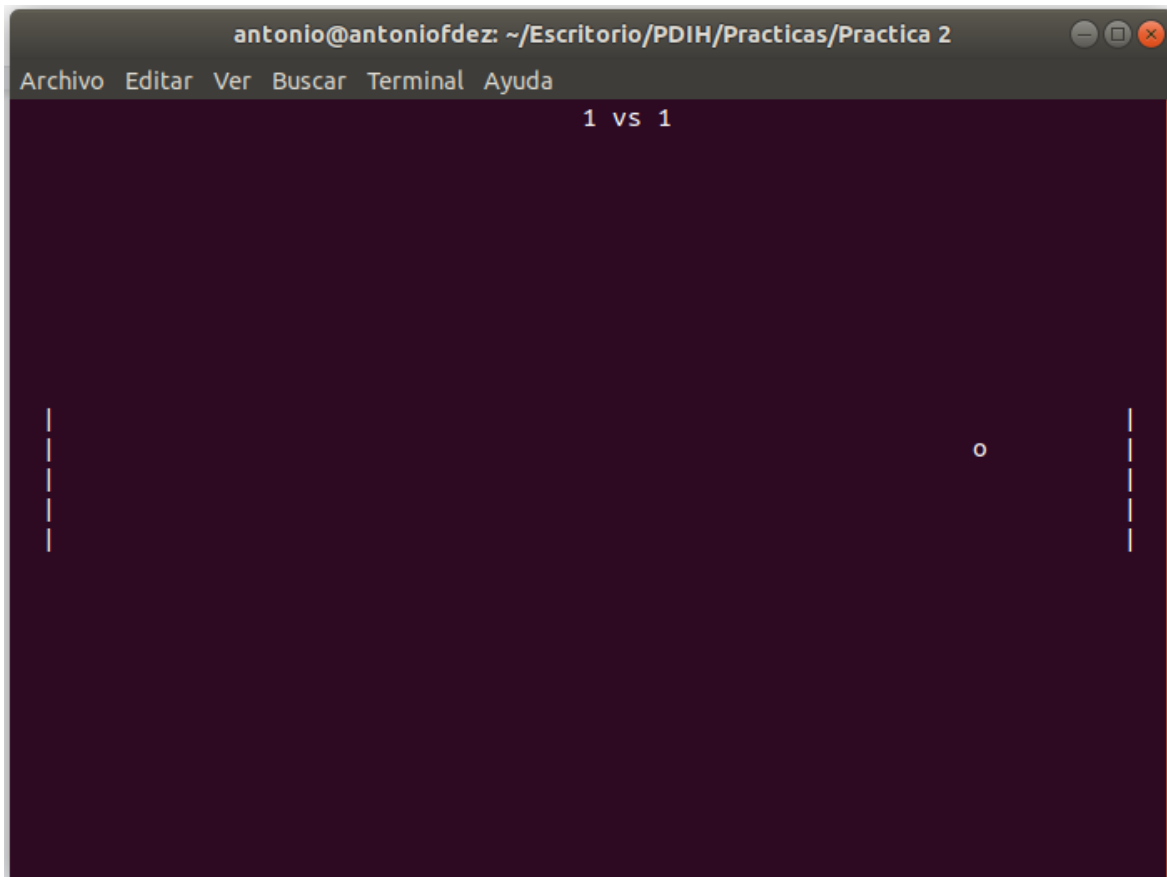
```
void final(int puntos1, int puntos2){
    nodelay(stdscr,0);
    clear();
    box(stdscr,'|','-' );
    if (puntos1 > puntos2){
        mvprintw(7,32,"GANADOR: JUGADOR 1");
    }
    else if(puntos1 < puntos2){
        mvprintw(7,32,"GANADOR: JUGADOR 2\n");
    }
    else{
        mvprintw(7,38,"EMPATE\n");
    }
    mvprintw(9,38, "%i vs %i",puntos1,puntos2);
    refresh();
    getch();
}
```



The screenshot shows a terminal window titled "antonio@antoniofdez: ~/Escritorio/PDIH/Practicas/Practica 2". The terminal has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The main area of the terminal is dark purple with a dashed white border. In the center, the text "EMPATE" is displayed on the first line, and "1 vs 1" is displayed on the second line.

Función para dibujar las barras de los jugadores.

```
void barras(int columnas, int pos1, int pos2){  
    int i;  
    for (i = 0; i < 5; i++){  
        mvprintw(pos1+i, 2, "|");  
        mvprintw(pos2+i, columnas-2, "|");  
    }  
}
```



## MAIN

comenzamos mostrando la pantalla de inicio mediante la función correspondiente y dimensionamos la pantalla.

Realizamos un bucle infinito que sólo termina en cuanto pulsemos la letra 'q' y se para en cuanto pulsamos la letra 'p'.

Colocamos el marcador y la bola y comienza a moverse la pelota conforme se ejecuta el bucle y según el delay que hayamos puesto.

Para comprobar cuando choca con las barras para que rebote y no haya recuento de puntos realizamos esta operación:

```
next_x = x + directionx;
next_y = y + directiony;

if(next_x == max_x-2 || next_x == 2){
    if(next_y == y1 || next_y == y1+1 || next_y == y1+2 || next_y == y1+3 || next_y == y1+4){
        directionx *= -1;
    }
    if(next_y == y2 || next_y == y2+1 || next_y == y2+2 || next_y == y2+3 || next_y == y2+4){
        directionx *= -1;
    }
}
```

Para comprobar cuando llega al borde y dar los puntos a los jugadores:

```
145
146     if(next_x >= max_x || next_x <= 0) {
147         directionx *= -1;
148
149         x = cols/2, y = fils/2;
150
151         if(next_x == max_x){
152             jugador1 += 1;
153         }else if(next_x == 0){
154             jugador2 += 1;
155         }
156     } else {
157         x += directionx;
158     }
159
160     if (next_y >= max_y || next_y < 0) {
161         directiony *= -1;
162     } else {
163         y += directiony;
164     }
165
```