

Periféricos y Dispositivos de Interfaz Humana Práctica 2



**UNIVERSIDAD
DE GRANADA**

Realizado por :
Sergio España Maldonado
Jose Luis Molina Aguilar

1. Instalación

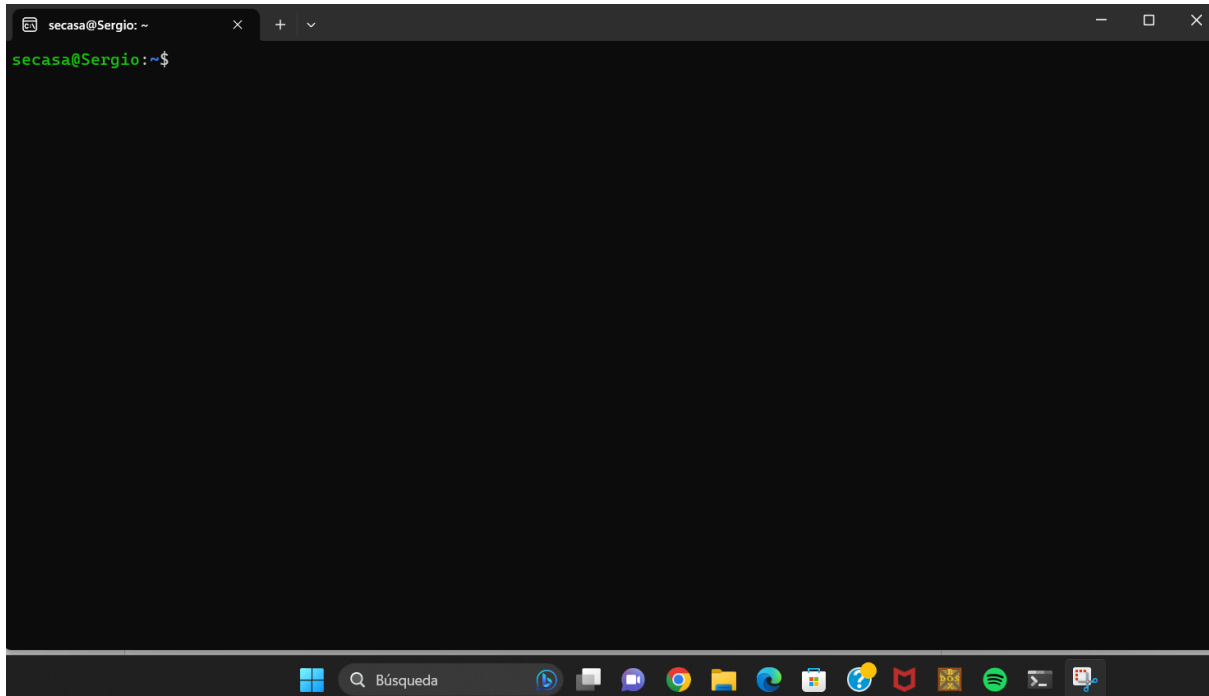
2. Ejemplos

- a. Hello
- b. Ventana
- c. Pelotita

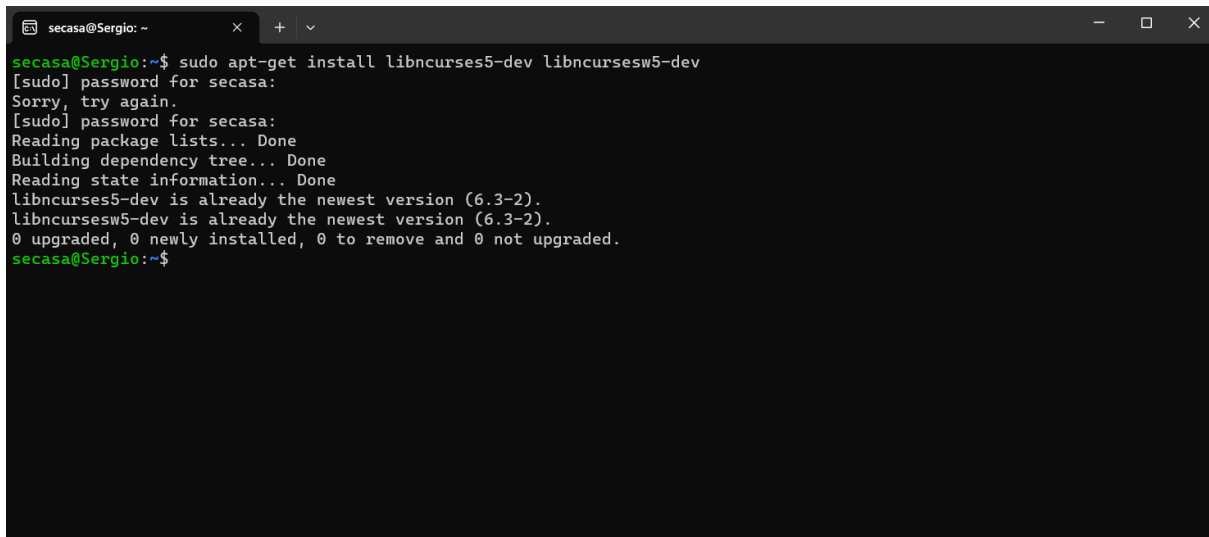
3. Pong

1.Instalación

Para la instalación utilizaremos la terminal de ubuntu en Windows.



A continuación, instalamos la librería de linux a través de el siguiente comando:



En nuestro caso, como ya lo tenemos instalado nos dice que ya tenemos la versión más reciente. Y ya con esto podríamos utilizar dicha librería en nuestro proyectos.

2. Ejemplos

En esta sección veremos el código y la ejecución de algunos códigos suministrados por el profesor y que se encuentran en el PDF de la práctica.

a. Hello.c

Código:

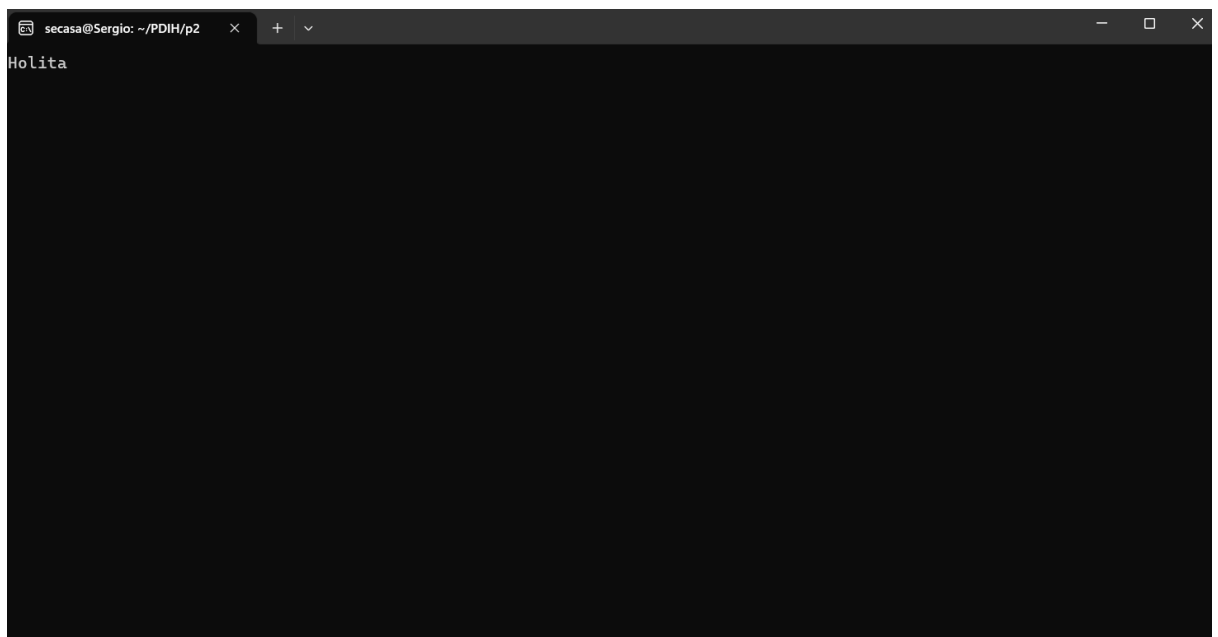
```
#include <ncurses.h>

int main(){

    initscr(); //inicializar modo curses
    printw("Holita"); //imprimir mensaje (aún no se verá)
    refresh(); //mostrarlo en pantalla
    getch(); //esperar la pulsación de una tecla
    endwin(); //terminar el modo curses
    return 0;
}
```

Ejecución:

```
secasa@Sergio:~/PDIH/p2$ gcc hello.c -o hello -lncurses && ./hello
```



b. Ventana.c

Código:

```
#include <stdlib.h>
#include <ncurses.h>

int main(void) {
    int rows, cols;

    initscr();
    if (has_colors() == FALSE) {
        endwin();
        printf("El terminal no tiene soporte de color \n");
        exit(1);
    }

    start_color();
    init_pair(1, COLOR_YELLOW, COLOR_GREEN);
    init_pair(2, COLOR_BLACK, COLOR_WHITE);
    init_pair(3, COLOR_WHITE, COLOR_BLUE);
    clear();

    refresh();
    getmaxyx(stdscr, rows, cols);

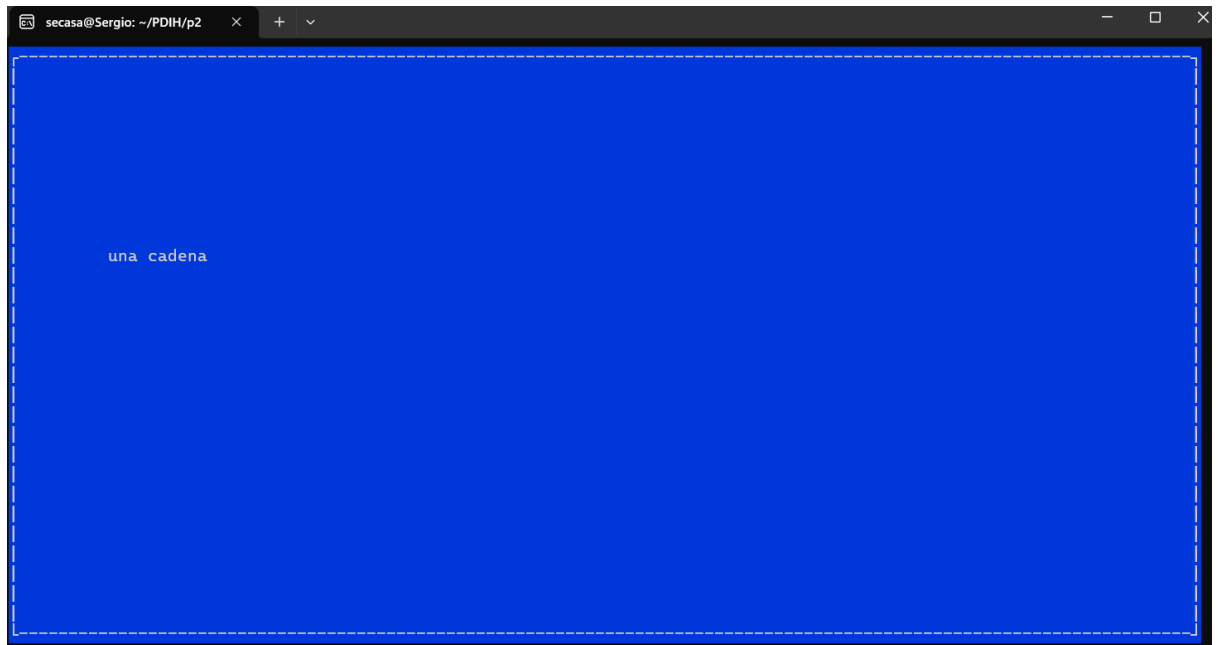
    WINDOW *window = newwin(rows,cols,0,0);
    wbkgd(window, COLOR_PAIR(3));
    box(window, '|', '-');

    mvwprintw(window, 10, 10, "una cadena");
    wrefresh(window);

    getch();
    endwin();
    return 0;
}
```

Ejecución:

```
secasa@Sergio:~/PDIH/p2$ gcc ventana.c -o ventana -lncurses &&  
./ventana
```



c. Pelotita.c

Código:

```
#include <ncurses.h>
#include <unistd.h>

#define DELAY 30000

int main(int argc, char *argv[]) {

    int x = 0, y = 0;
    int max_y = 50, max_x = 50;
    int next_x = 0;
    int direction = 1;

    initscr();
    noecho();
    curs_set(FALSE);

    while(1) {
        clear();
        mvprintw(y, x, "o");
        refresh();

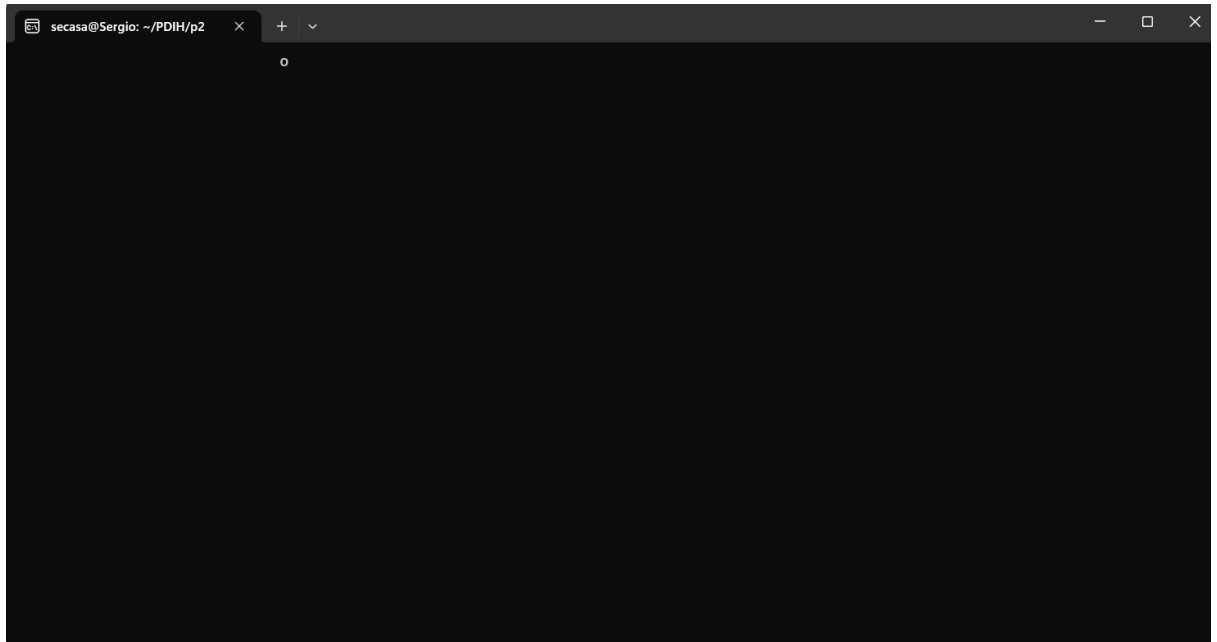
        usleep(DELAY);

        next_x = x + direction;

        if (next_x >= max_x || next_x < 0) {
            direction*= -1;
        } else {
            x+= direction;
        }
    }
    endwin();
}
```

Ejecución:

```
secasa@Sergio:~/PDIH/p2$ gcc pelotita.c -o pelotita -lncurses &&  
./pelotita
```



3. Pong

Código:

Función que resetea el juego:

```
void reset_game(Screen *screen, Ball *ball, Paddle *p1,Paddle *p2){  
    init_Ball(ball, screen->width / 2, screen->height / 2, 1, 1);  
    init_Paddle(p1, MARGIN, screen->height / 2, PAD_LEN, 0);  
    init_Paddle(p2, screen->width - MARGIN, screen->height / 2, PAD_LEN, 0);  
}
```


Función que tras pulsar “q”, “Q” o que alguno de los jugadores llegue a 5 de puntuación nos cambiará la ventana y nos mostrará el resultado final, la puntuación y las opciones para salir del juego o reiniciar la partida para jugar otra partida.

```
bool menuFinal(WINDOW *w,Screen *screen, Paddle *p1,Paddle *p2){
    char cadPuntos[50];
    box(w,'$','$');
    int shift = screen->width/2 - 10;
    sprintf(cadPuntos, "%d\t - %d", p1->score,p2->score);
    mvwprintw(w,screen->height/2-4, shift+10, cadPuntos);
    if(p1->score > p2->score)
        mvwprintw(w,screen->height/2-2,shift, " El ganador a sido el jugador1 ");
    else if(p1->score < p2->score)
        mvwprintw(w,screen->height/2-2,shift, " El ganador a sido el jugador2 ");
    else
        mvwprintw(w,screen->height/2-2,shift, " \t      Empate!");
        mvwprintw(w,screen->height/2,shift, " Pulsa ENTER para jugar de nuevo ");
        mvwprintw(w,screen->height/2+1,shift, " Pulsa Q para salir del juego ");

    switch (wgetch(w)){
        case KEY_ENTER:
            return true;
        case 'q':
        case 'Q':
            return false;
    }
}
```

A continuación veremos unas imágenes del código con inicialización y uso de algunas letras como la selección de las mismas, en la propia imagen se explica cada sección la función que tiene dentro del código :

```
//Colores para los diferentes objetos
start_color();
init_pair(1, COLOR_WHITE, COLOR_BLACK); // Color de la puntuación
init_pair(2, COLOR_CYAN, COLOR_BLACK); // Color barra central
init_pair(3, COLOR_YELLOW, COLOR_BLACK); // Color de la pelota
init_pair(4, COLOR_RED, COLOR_RED); // Color de las barras

//Dimensiones de la pantalla
getmaxyx(stdscr, screen.height, screen.width);

//Definicion de pelota
Ball ball;
init_Ball(&ball, screen.width / 2, screen.height / 2, 1, 1);

//Definicion de palas y jugadores
Paddle paddleP1, paddleP2;
init_Paddle(&paddleP1, MARGIN, screen.height / 2, PAD_LEN, 0);
init_Paddle(&paddleP2, screen.width - MARGIN, screen.height / 2, PAD_LEN, 0);

//pantalla de inicio de juego
WINDOW *init_screen = newwin(screen.height, screen.width, 0, 0);
box(init_screen, '|', '-');
int mid = screen.width/2 - 81/2;
mvprintw(2,mid, "-----");
mvprintw(3,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(4,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(5,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(6,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(7,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(8,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(9,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(10,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(11,mid, "| | | | | | | | | | | | | | | | | |");
mvprintw(12,mid, "-----");

//Controles de juego

mvprintw(15, mid , "-----CONTROLES-----");
mvprintw(16, 55, "Player 1");
mvprintw(17, 53.5, "UP -> W");
mvprintw(18, 53.5, "DOWN -> S");

mvprintw(16, 105, "Player 2");
mvprintw(17, 103.5, "UP -> ^ (flecha arriba)");
mvprintw(18, 103.5, "DOWN -> v (flecha abajo)");
```

```

while (!terminar){
    char cadPuntos[50];          // Cadena donde escribimos los puntos

    while(paddleP1.score < 5 && paddleP2.score < 5 && !terminar) {

        //Pintamos los bordes
        wattron(game_screen, COLOR_PAIR(2));
        box(game_screen, '|', '-');
        wattroff(game_screen, COLOR_PAIR(2));

        //Pintamos la pelota
        wattron(game_screen, COLOR_PAIR(3));
        mvwprintw(game_screen,ball.y, ball.x, "o");
        wattroff(game_screen, COLOR_PAIR(3));

        //Puntuacion
        sprintf(cadPuntos, "%d\t%d", paddleP1.score,paddleP2.score);
        mvwprintw(game_screen,1, screen.width / 2 - 3, cadPuntos);

        //Linea central divisoria
        mvwvline(game_screen, 1, screen.width / 2, ACS_VLINE, screen.height-2);

        //Linea horizontal
        //mvwhline(game_screen, screen.height / 2, 2, ACS_HLINE, screen.width-2);

        //Mover el conjunto de la pala
        wattron(game_screen, COLOR_PAIR(4));
        for (int i = 0; i < PAD_LEN ; i++){
            mvwprintw(game_screen, paddleP1.y + i, paddleP1.x, "|");
            mvwprintw(game_screen, paddleP2.y + i, paddleP2.x, "|");
        }
        wattroff(game_screen, COLOR_PAIR(4));
    }
}

```

```

//Recoger pulsacion de teclas
nodelay(game_screen, true);
switch (wgetch(game_screen)){
    case 'q':
    case 'Q':

        terminar = true;
        //werase(game_screen);

        break;

    case 'w':
    case 'W':
        if(paddleP1.y > 1){

            paddleP1.y--;
        }
        break;
    case 's':
    case 'S':
        if(paddleP1.y < screen.height - 2)
            paddleP1.y++;
        break;
    case KEY_UP:
        if(paddleP2.y > 1)
            paddleP2.y--;
        break;

    case KEY_DOWN:
        if(paddleP2.y < screen.height - 2)
            paddleP2.y++;
        break;

}

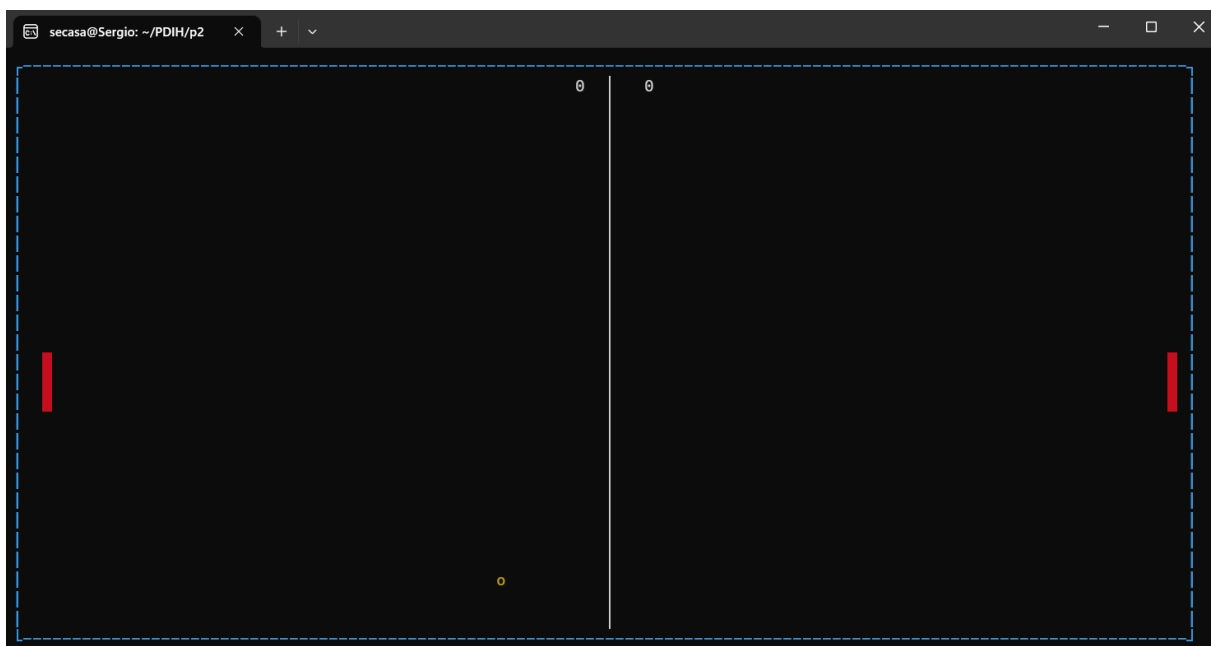
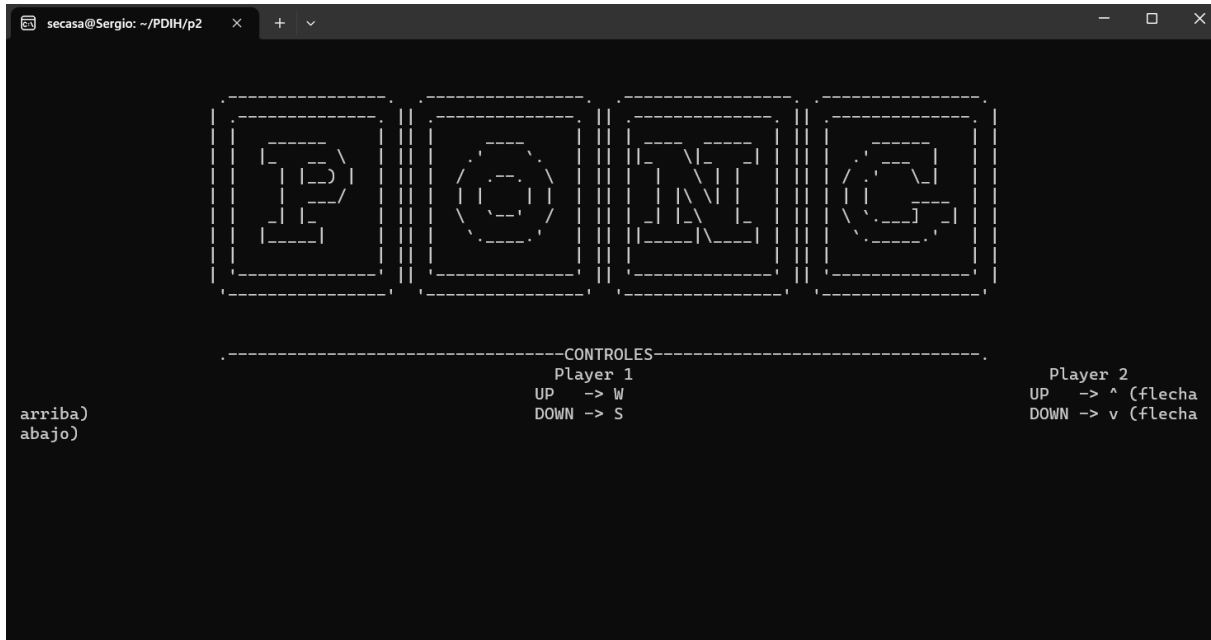
```

```

WINDOW *end_screen = newwin(screen.height, screen.width, 0, 0);
terminar = menuFinal(end_screen,&screen ,&paddleP1,&paddleP2);
if(terminar) reset_game(&screen, &ball, &paddleP1, &paddleP2);
else break;
wrefresh(end_screen);

```

Ejecución:



```
secasa@Sergio: ~/PDIH/p2
2 - 5
El ganador a sido el jugador2
Pulsa ENTER para jugar de nuevo
Pulsa Q para salir del juego
```