

En este documento adjunto el código fuente de mi programa y muestro capturas de pantalla de su funcionamiento.

```
#include <stdlib.h>
#include <ncurses.h>
#define DELAY 50000

int main(void) {

// PREPARACION SOFTWARE

    int rows, cols;

    initscr();

    if (has_colors() == FALSE) {
        endwin();
        printf("Your terminal does not support color\n");
        exit(1);
    }

    start_color();
    init_pair(1,COLOR_WHITE,COLOR_BLUE);
    clear();

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

    WINDOW *window = newwin(rows,cols,0,0);
    wbkgd(window, COLOR_PAIR(1));
    box(window, '|', '-');

// ESTO ES EL MENU

    mvwprintw(window, rows/2, cols/2 - 10, "Bienvenido al juego Ping-Pong");
    mvwprintw(window, rows/2+3, cols/2 - 12, "Juego creado por Angel Lopez Martos");
    mvwprintw(window, rows/2+6, cols/2 - 20, "Los controles son ARRIBA (w), ABAJO (s),
SALIR (q)");
    mvwprintw(window, rows/2+9, cols/2 - 10, "Pulsa ENTER para comenzar");
    wrefresh(window);
    getch();

    int max_y = rows, max_x = cols; //Rango movimiento

    //Posicion inicial pelota
    int x = 0, y = 0;
    int next_x = 0;
```

```

int directionx = 1;
int next_y = 0;
int directiony = 1;

//Posicion J1

int j1_x = 0;
int j1_y = 0;

//Posicion J2

int j2_x = max_x - 1;
int j2_y = 1;

//Puntuaciones

int P1_goals = 0;
int P2_goals = 0;

int xc = 20, yc = 10;
int ch = 0;

initscr();
noecho();
cbreak();
curs_set(FALSE);
nodelay(stdscr, TRUE);

while(1) {

    clear();
    mvprintw(y, x, "o");

    int i = 0;

    //Pos ini jugadores

    for(i = 0; i < 5; i++){

        mvprintw(j1_y + i, j1_x, "|");

    }

    for(i = 0; i < 5; i++){

        mvprintw(j2_y + i, j2_x, "|");

    }

    //MOSTRAR CONTADOR

    mvprintw(0, max_y / 2 + 5, "*****");
    mvprintw(1, max_y / 2 + 5, "* Jugador 1: %d    Jugador 2: %d", P1_goals, P2_goals);
}

```

```

        mvprintw(2, max_y / 2 + 5, "*****");
        refresh();

ch = getch();
if (ch == 'o')
    xc -= 1;
else if (ch == 'p')
    xc += 1;

    usleep(DELAY);

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

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

    if (next_y >= max_y || next_y < 0) {
        directiony *= -1;
    } else {
        y += directiony;
    }

//CONTADOR

if (j1_x > next_x){

    if(next_y == j1_y || next_y == j1_y + 1 || next_y == j1_y + 2 || next_y == j1_y + 3 ||
next_y == j1_y + 4){

        //GOL DETENIDO

    }

    else {

        next_x = max_x/2;
        next_y = max_y/2;

        P2_goals++;

    }

}

if (j2_x < next_x){

    if(next_y == j2_y || next_y == j2_y + 1 || next_y == j2_y + 2 || next_y == j2_y
+ 3 || next_y == j2_y + 4){

        //GOL DETENIDO

```

```

    }

    else {

        next_x = max_x/2;
        next_y = max_y/2;

        P1_goals++;
    }
}

if(P1_goals == 1 || P2_goals == 1){

    if(P1_goals == 1){

        mvwprintw(window, rows/2, cols/2, "Jugador 1 gana, contador --->
Jugador 1: %d    Jugador 2: %d",P1_goals,P2_goals);
        //mvprintw(rows/2, cols/2 - 12, " Jugador 1: %d    Jugador 2:
%d",P1_goals,P2_goals);

        mvwprintw(window, rows/2 + 2, cols/2 - 5, "Pulse n para nueva
partida o q para salir");

    }

    else {

        mvwprintw(window, rows/2, cols/2, "Jugador 2 gana, contador --->
Jugador 1: %d    Jugador 2: %d",P1_goals,P2_goals);
        //mvprintw(1, max_y / 2 + 5, " Jugador 1: %d    Jugador 2:
%d",P1_goals,P2_goals);
        mvwprintw(window, rows/2 + 2, cols/2 - 5, "Pulse n para nueva
partida o q para salir");

    }

    wrefresh(window);
    //getchar();

    switch(getchar()){

        case 'q':

            endwin();

```

```

        exit(1);

    break;

    case 'n':

        //Posicion inicial pelota
        x = 0, y = 0;
        next_x = 0;
        directionx = 1;
        next_y = 0;
        directiony = 1;

        //Posicion J1
        j1_x = 0;
        j1_y = 0;

        //Posicion J2
        j2_x = max_x - 1;
        j2_y = 1;

        //Puntuaciones
        P1_goals = 0;
        P2_goals = 0;

        xc = 20, yc = 10;
        ch = 0;

    default:
        break;
}

}

switch (getch()){

    case 'w':
        if(j1_y >= 2){
            j1_y = j1_y - 1;
        }
        break;

    case 's':
        if(j1_y + 4 <= max_y - 4){

```

```
        j1_y = j1_y + 1;

    }
    break;

case 'o':
    if(j2_y >= 2){

        j2_y = j2_y - 1;

    }
    break;

case 'l':
    if(j2_y +4 <= max_y - 4){

        j2_y = j2_y + 1;

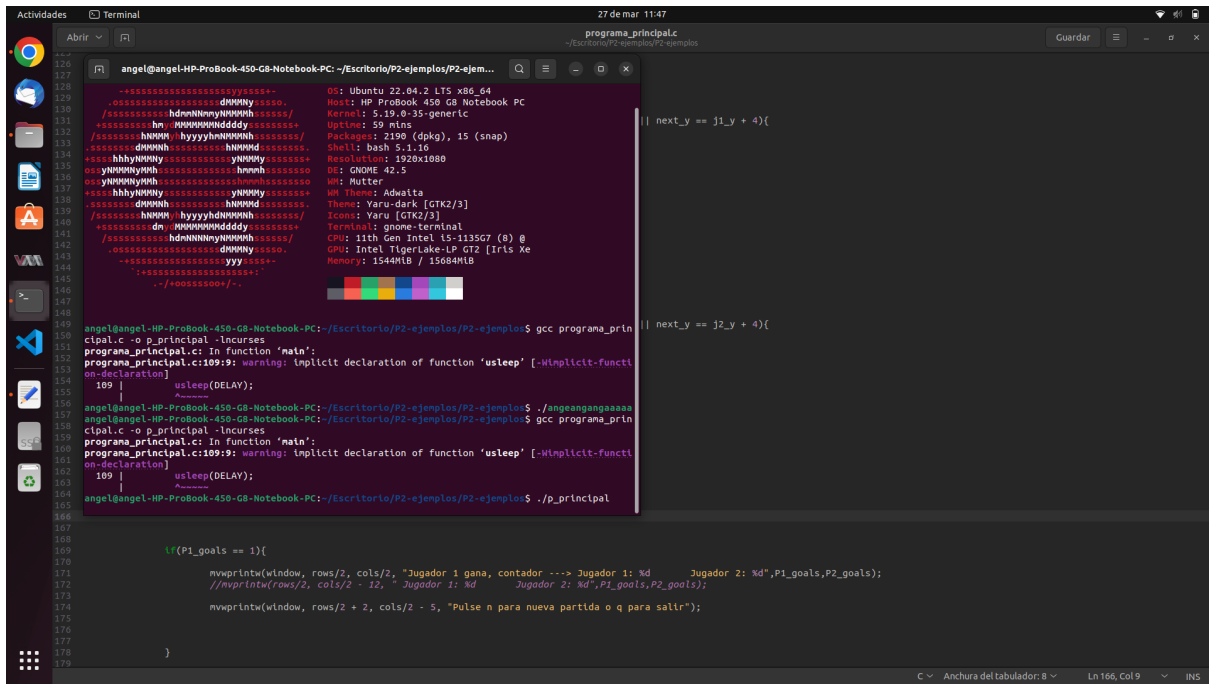
    }
    break;

default:
    break;

}

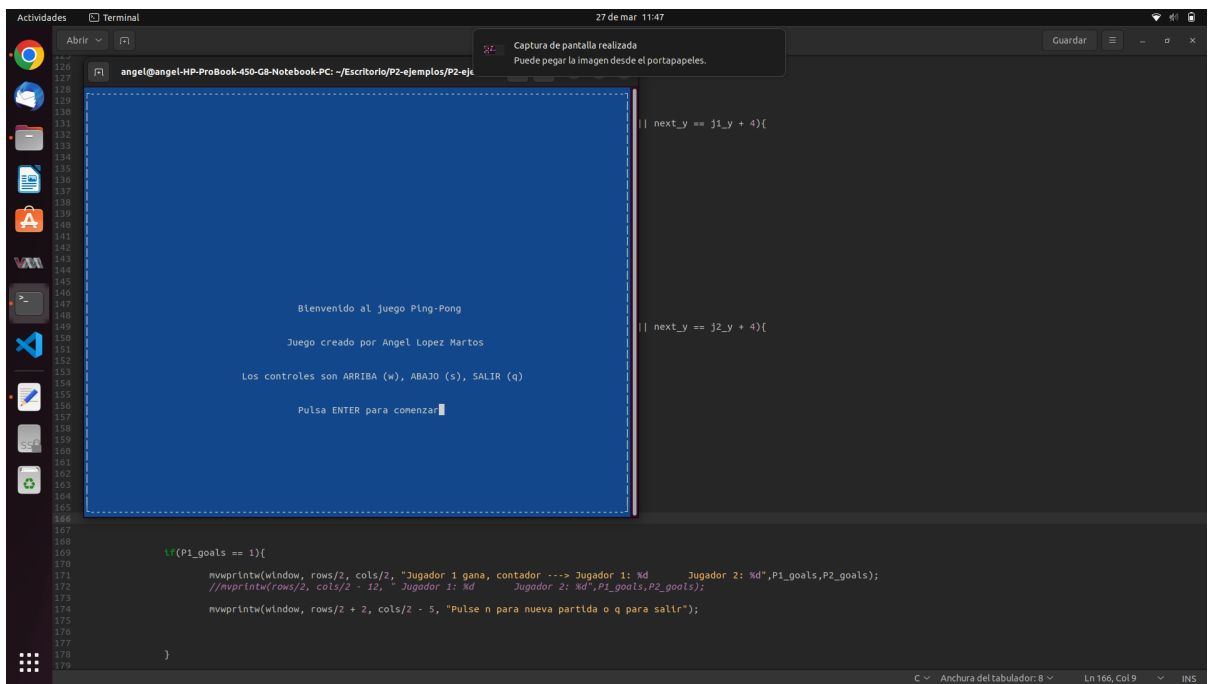
}

endwin();
return 0;
}
```



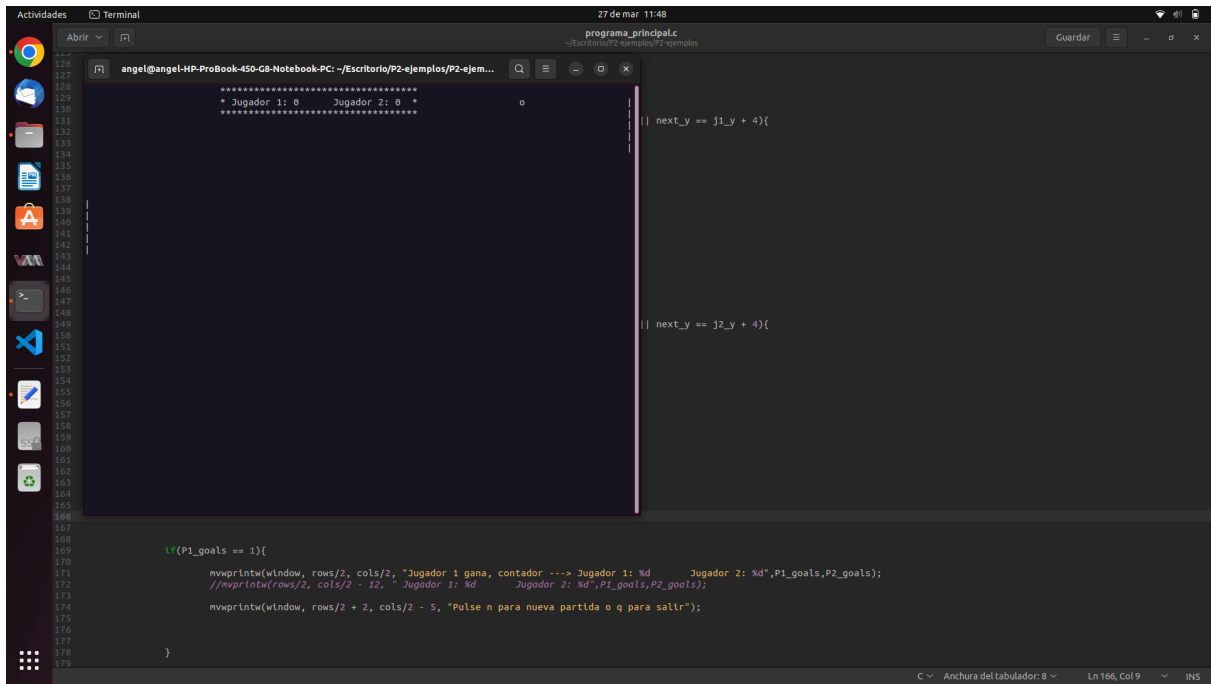
A terminal window titled 'programa\_principal.c' showing system information and C code compilation. The system information includes OS (Ubuntu 22.04.2 LTS), Host (HP ProBook 450 G8), Kernel (5.19.0-35-generic), Uptime (59 mins), Packages (2190), DE (GNOME 42.5), Shell (bash), WM (Mutter), Theme (Yaru-dark), Icons (Yaru), Terminal (gnome-terminal), CPU (11th Gen Intel i5-1135G7), GPU (Intel TigerLake-LP GT2), and Memory (1544MiB / 15684MiB). The C code is being compiled with 'gcc programa\_principal.c -o p\_principal -lncurses' and then executed with './p\_principal'. The code includes a loop for a game and a function 'usleep'.

(Correspondiente al momento de compilación y ejecución)



A terminal window showing the game menu. The menu text is: 'Bienvenido al juego Ping-Pong', 'Juego creado por Angel Lopez Martos', 'Los controles son ARRIBA (w), ABAJO (s), SALIR (q)', and 'Pulsa ENTER para comenzar'. The C code is visible in the background, showing the game loop and the 'usleep' function.

(Muestro menú con la explicación de los controles y opciones)



```
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179

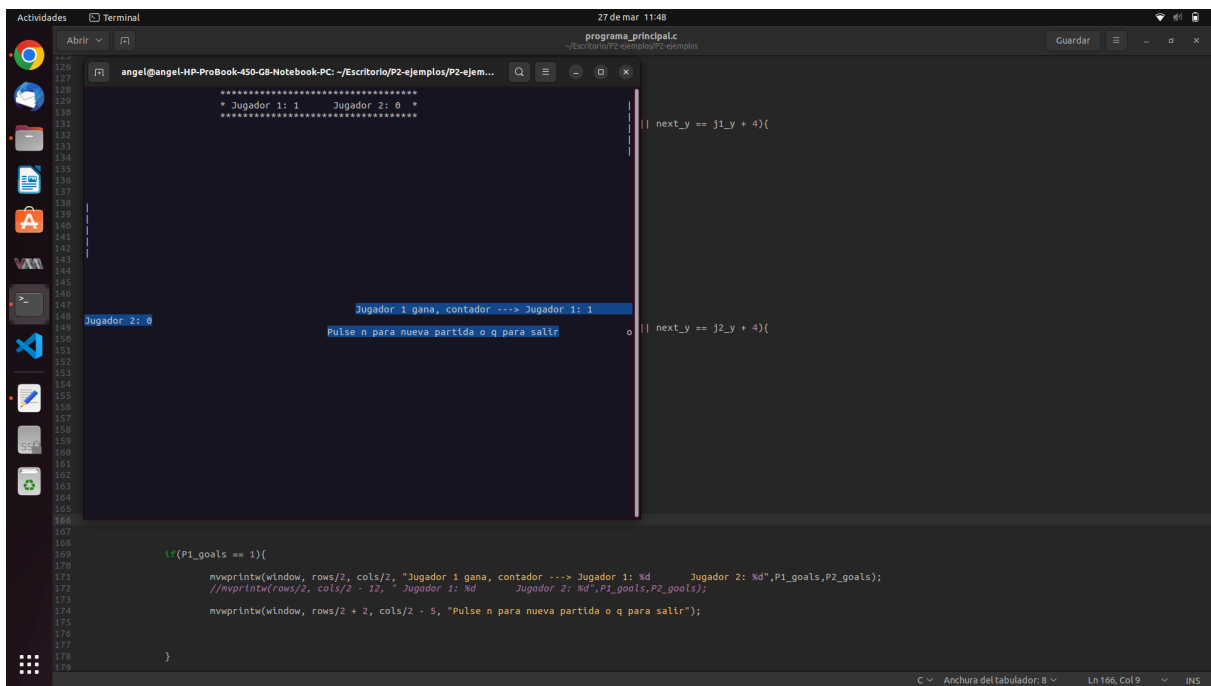
* Jugador 1: 0   Jugador 2: 0 *
*****

|| next_y == j1_y + 4){

|| next_y == j2_y + 4){

if(P1_goals == 1){
    mvwprintw(window, rows/2, cols/2, "Jugador 1 gana, contador --> Jugador 1: %d   Jugador 2: %d", P1_goals, P2_goals);
    //mvwprintw(rows/2, cols/2 - 12, " Jugador 1: %d   Jugador 2: %d", P1_goals, P2_goals);
    mvwprintw(window, rows/2 + 2, cols/2 - 5, "Pulse n para nueva partida o q para salir");
}
```

(Momento de interacción con el juego)



```
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179

* Jugador 1: 1   Jugador 2: 0 *
*****

|| next_y == j1_y + 4){

|| next_y == j2_y + 4){

Jugador 1 gana, contador --> Jugador 1: 1
Jugador 2: 0
Pulse n para nueva partida o q para salir

if(P1_goals == 1){
    mvwprintw(window, rows/2, cols/2, "Jugador 1 gana, contador --> Jugador 1: %d   Jugador 2: %d", P1_goals, P2_goals);
    //mvwprintw(rows/2, cols/2 - 12, " Jugador 1: %d   Jugador 2: %d", P1_goals, P2_goals);
    mvwprintw(window, rows/2 + 2, cols/2 - 5, "Pulse n para nueva partida o q para salir");
}
```

(Final de partida y menú con opciones)