



**Universidad Nacional Autónoma  
de México  
Facultad de Ingeniería**



**PROYECTO FINAL  
Laberinto**

**Profesor:**

Manuel Enrique Castañeda Castañeda

**Materia:**

Estructura de Datos y Algoritmos I

**Grupo:**

12

**Semestre:**

2121-2

**Integrantes:**

- Lemus Ambrosio Aline Andrea
- Sanchez Alvirde Andrés Iván
  - Rios Morales Manuel

## OBJETIVO

- Realizar un juego con diferentes niveles y premios usando los conocimientos adquiridos en el semestre.
- Hacer uso de una lista ligada circular para almacenar y mostrar los premios ganados por el jugador.

# INTRODUCCIÓN

Para la realización de este proyecto se debía de crear un juego usando los conocimientos adquiridos a lo largo del semestre, el juego escogido fue un laberinto en el cual podíamos incluir: niveles y premios, para ello se decidió realizar un laberinto el cual nos permitiría agregar diferentes niveles y durante el recorrido de cada nivel agregar un premio. Con este proyecto se usaron funciones, conceptos o métodos vistos en clase, los cuales permitieron tener un mejor control e idea de como realizar el laberinto.

## DESARROLLO

Para la realización de este laberinto fue necesario tener en cuenta cuantos niveles queríamos agregar, teniendo en cuenta esto sabríamos también cuantos premios poner. Para el juego fue necesario usar funciones para realizar cada parte del laberinto como los niveles, el texto, los premios, las instrucciones, etc., en algunos la forma de realización es parecida, más que nada en los textos.

Para realizar la portada, el menú, el texto de los niveles y de los premios, e incluso las imágenes de cada premio se hace uso más que nada de los printf, ya que este nos permite darle forma a un texto usando símbolos ( \_, /, @, |, ;, etc.) y varios renglones para formarlos, el título tratamos de hacer que las letras parecieran salir del fondo, para el resto de los textos solo hacemos las letras como si estuvieran infladas, son conocidas como letra en forma de globo, por último para hacer estos textos llamativos lo que hicimos fue agregarles color..

```

74 [
75 ] while(menu<4);
76 }
77
78 void menujuego(){
79     printf("\033[1;32m");
80     printf("\n");
81     printf("1. JUEGOS\n");
82     printf("2. CONTROLES\n");
83     printf("3. Opciones\n");
84     printf("4. Salir\n");
85     printf("\033[0;0m");
86     printf("\n");
87     printf("1. JUEGOS\n");
88     printf("2. CONTROLES\n");
89     printf("3. Opciones\n");
90     printf("4. Salir\n");
91     printf("\033[0;0m");
92     printf("\n");
93     printf("1. JUEGOS\n");
94     printf("2. CONTROLES\n");
95     printf("3. Opciones\n");
96     printf("4. Salir\n");
97     printf("\033[0;0m");
98     printf("\n");
99     printf("1. JUEGOS\n");
100    printf("2. CONTROLES\n");
101    printf("3. Opciones\n");
102    printf("4. Salir\n");
103    printf("\033[0;0m");
104    printf("\n");
105    printf("1. JUEGOS\n");
106    printf("2. CONTROLES\n");
107    printf("3. Opciones\n");
108    printf("4. Salir\n");
109    printf("\033[0;0m");
110    printf("\n");
111    printf("1. JUEGOS\n");
112    printf("2. CONTROLES\n");
113    printf("3. Opciones\n");
114    printf("4. Salir\n");
115    printf("\033[0;0m");
116    printf("\n");
117 }

```

En este mismo apartado se diseña la funcionalidad de los controles, es decir, con ayuda del if y else, se crea un espacio para que el juego detecte el movimiento del jugador ya que cada que se introduce una letra para moverse, en el código se le suma a la letra que se ingrese para así poder moverse en el juego y, en cuanto se detecta que el jugador llega a “chocar” con algún obstáculo se imprime que el camino está bloqueado, para que de este modo le permita volver a elegir una dirección, ya que en cada movimiento se detecta si el jugador pasa por un espacio en blanco o si llega a topar con los obstáculos del tablero.

En este mismo apartado se diseña la funcionalidad de los controles, es decir, con ayuda del if y else, se crea un espacio para que el juego detecte el movimiento del jugador ya que cada que se introduce una letra para moverse, en el código se le suma a la letra que se ingrese para así poder moverse en el juego y, en cuanto se detecta que el jugador llega a “chocar” con algún obstáculo se imprime que el camino está bloqueado, para que de este modo le permita volver a elegir una dirección, ya que en cada movimiento se detecta si el jugador pasa por un espacio en blanco o si llega a topar con los obstáculos del tablero.

[illegible]

En cuanto a los problemas presentados durante la creación del código, la mayor problemática que tuvimos fue el cómo implementar la lista ligada circular al juego, debido a que en un principio se nos ocurrió agregar los regalos directamente a la lista, es decir, guardar todo `artAscii` en la propia lista cosa que fue imposible debido a que daba muchos errores al compilar debido a que no se almacenaba o simplemente era demasiado grande como para guardarlo debido a que eran varias líneas de código.

Lo que hicimos para solucionarlo fue almacenar los premios por nombre. Lo que se hizo es que al momento de que el jugador toca el símbolo "\$" le aparece un pequeño menú abajo del tablero donde se le da al jugador la opción de escoger entre dos diferentes premios (los premios agregados en el juego son: un cactus, un globo, una figura de batman, unos lentes, un helado y una luna), para seleccionar tu premio solo escribes el nombre del que deseas, luego de esto se le informa al

jugador que su regalo se ha guardado de forma exitosa en el apartado “objetos”, para ello se usa la función insertarPremio, esta función una lista doble ligada circular, reserva la memoria dinámica, luego con el gets recibe el nombre que colocaste y ya de ahí se almacenará en la estructura premios en el dato que se llama premio[50] y así con cada premio correspondiente.

Otro problema que presentamos era a la hora de imprimirlos ya que no sabíamos como hacer el llamado a la lista para que imprimiera el artAscii, logramos imprimir los objetos que recolectó y que se almacenaron pero no el arte.

Al final lo solucionamos usando la libreria string.h haciendo uso del strcmpi para comparar el dato almacenado con el nombre del premio correspondiente que a la hora de verificarse mostraba el premio.

Un problema pequeño que nos dio a la hora de recolectar el premio, ya que pasarás las veces que pasaras, siempre ibas a ganar un premio, cosa que no tenía que ser así ya que solo era ganar un premio. Se soluciono fácil colocando un contador y ya no sucedió más.

## CONCLUSIONES

Finalmente este juego nos permitió poner en práctica lo aprendido durante clase, así mismo nos permitió crear un juego que podemos compartir con nuestros amigos o familiares para poder pasar un buen rato. Con este proyecto hemos podido reforzar nuestros conocimientos ya que pusimos en práctica lo aprendido en clase y en caso de habernos trabado teníamos la posibilidad de apoyarnos con los videos de clase o con videos externos, por lo que pudimos realizar nuestro juego con éxito.

Por último podemos demostrar que la programación orientada a objetos es una herramienta que además de permitirnos resolver problemas mediante algoritmos eficaces, nos permite también usar nuestra creatividad para realizar juegos, claro que podemos usar lo visto en esta materia para realizar algo más grande, pero para ello aún necesitamos aprender más.

Código
<pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; #include &lt;string.h&gt;  void nivel1(); void nivel2(); void nivel3(); void menujuego();</pre>

```

void menucontroles();
void textoNiveles();
void insertarPremio();
void mostrarPremio();
struct premios{
    char premio[50];
    premios *siguiente;
    premios *anterior;
};
premios *primero=NULL;
premios *ultimo=NULL;
int contreg1=0,contreg2=0,contreg3=0;
int main()
{
    int menu,menu2,meta=0;
    do{
        system("cls");
        menu=0;
        menujuego();
        scanf("%d",&menu);
        switch(menu){
            case 1:
                do{
                    system("cls");
                    textoNiveles();
                    scanf("%d",&menu2);
                    switch(menu2){
                        case 1:
                            nivel1();
                            break;
                        case 2:
                            nivel2();
                            break;
                        case 3:
                            nivel3();
                            break;
                    }
                }while(menu2<4);
                system("cls");
            break;
            case 2:
                system("cls");
                menucontroles();
                system("pause");
                system("cls");
            break;
            case 3:
                system("cls");
                if(contreg1==0 && contreg2==0 && contreg3==0){

```









```

        ultimo=q;
        primero->anterior=ultimo;
    }
    printf("Se ha agregado a tus objetos con exito!!\n");
}

void mostrarPremio(){
    printf("\033[0;33m");
    printf("\t\t\t\t\t _____ \n");
    printf("\t\t\t\t\t _ | _ | _ | | | _ |\n");
    printf("\t\t\t\t\t _| -| _|||- -| | _ |\n");
    printf("\t\t\t\t\t_| |_|_|_|_|_|_|_|_|_\n");
    printf("\033[0;0m");
    printf("\nEscribe el premio que deseas ver de los que tienes: \n");
    premios *indice=(premios*)malloc(sizeof(premios));
    indice=primero;
    if(indice!=NULL){
        do{
            printf("\n[%s]\t",indice->premio);
            indice=indice->siguiente;
        }while(indice!=primero);
        printf("\n");
    }
    else{
        printf("\t\t\t\t\t _____ \n");
        printf("\t\t\t\t\t || | | _ | | _ | | | _ | | | _ | |\n");
        printf("\t\t\t\t\t || | | | |- -| _||| _|| | | _ | -| _||| |- -| | _ |\n");
        printf("\t\t\t\t\t_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_\n");
    }
    printf("\n");
    premios *q=(premios*)malloc(sizeof(premios));
    fflush(stdin);
    gets(q->premio);
    if(strncmp(q->premio,"globo")==0||strcmp(q->premio,"Globo")==0){
        printf("\n");
        system("cls");
        printf("\033[0;31m");
        printf("\t      ,,,,,,,,,,\n");
        printf("\t      ..... \n");
        printf("\t      ..... \n");
        printf("\t      ..... \n");
        printf("\t..... \n");
        printf("\t;;;@:.....\n");
        printf("\t;;;@@:.....\n");
        printf("\t;;;@@:.....\n");
        printf("\t `;;@:.....@\n");
        printf("\t `;;;;;;;;@@"";\n");
        printf("\t `;;;;;;;;@@"";\n");
    }
}

```

```
printf("\t\t\t\t\t@;\n");  
printf("\t\t\t\t\t\n");  
printf("\t\t\t\t\t\n");  
printf("\t\t\t\t\t\n");  
printf("\033[0;0m");  
printf("\t\t\t\t\t\n");  
printf("\t\t\t\t\t\n");  
printf("\t\t\t\t\t\n");  
printf("\t\t\t\t\t\n");  
printf("\t\t\t\t\t\n");  
printf("\t\t\t\t\t\n");  
printf("\t\t\t\t\t\n");  
printf("\n");  
printf("[UN GLOBO MUY BONITO Y COLORIDO, NO LO PIERDAS!!]\n");  
printf("\n");  
  
if(strcmpi(q->premio,"cactus")==0||strcmpi(q->premio,"Cactus")==0){  
    printf("\n");  
    system("cls");  
    printf("\033[0;32m");  
    printf("\t\t\t\t\t,\n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\t\t\t\t\t;;\n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\t\t\t\t\t,, ;'\n");  
    printf("\t\t\t\t\t';' '\n");  
    printf("\t\t\t\t\t,' , , ,'\n");  
    printf("\t\t\t\t\t,' , , ,'\n");  
    printf("\t\t\t\t\t,' , , ,'\n");  
    printf("\t\t\t\t\t,\" \" \" \"'\n");  
    printf("\t\t\t\t\t\" \" \" \"'\n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\t\t\t\t\t;' \n");  
    printf("\033[0;0m");  
    printf("\n");  
    printf("[QUE BONITO CACTUS, PERO TEN CUIDADO CON SUS PINCHOS]\n");  
    printf("\n");  
  
}  
if(strcmpi(q->premio,"Lentes")==0||strcmpi(q->premio,"lentes")==0){
```

[illegible]

[illegible]

```

printf("\t          aaaaaaaaaa aaaaaaa          *\n");
printf("\t          aaaaaaa  aaaaaaaaaa\n");
printf("\t          aaaaaa a aaaaaa aaaaaa\n");
printf("\t          aaaaaaa  aaaaaaa\n");
printf("\t          aaaaaaaa          a\n");
printf("\t          aaaaaaaaaa          aa\n");
printf("\t          aaaaaaaaaaaaaaaaaa          aaaa\n");
printf("\t          aaaaaaaaaaaaaaaaaa          aaaaaa  *\n");
printf("\t      *          aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n");
printf("\t          aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\n");
printf("\t      *          aaaaaaaaaaaaaaaaaa\n");
printf("\033[0;0m");
printf("\n");
printf("\t[LA LUNA ES HERMOSA,NO ES ASI?\n");
printf("\n");
}
}

```

```

void nivel1(){
    system("cls");
    char mapa1[50][50]=
        {"\t=====",
        "\to      |$  |  |      |",
        "\t| ===  === |===  |||=== | |",
        "\t| | | | | | | | | |",
        "\t| | | | | | | | | |=",
        "\t|==| | | | | | | | |",
        "\t| | | | | | | |==| |",
        "\t| | | | | | | = | | | | |",
        "\t| | | | |==| | | | |",
        "\t| | | | | | | | | |=",
        "\t| | | | | | | | | |=",
        "\t| | | | | | | | | |=",
        "\t| | | | | | | | | |=",
        "\t| | | | | | | | | |=",
        "\t|==| | | | |==|==|==| |",
        "\t| | | | | | | | | |",
        "\t|==| | | | |==| | | |",
        "\t| | | | | | | | | |",
        "\t|== | | |== == |==| |",
        "\t| | | | | | | |",
        "\t|=====",
        };
    int x,y,ancho,altura;
    char movimiento;
    x=1;
    y=2;
    ancho=32;
    altura=12;
    int regalo1x=1,regalo1y=12;
    for(int i=0;i<=18;i++)

```

```

        puts(mapa1[i]);
        while(x != altura || y != ancho)
        {
            movimiento = getchar();
            if(movimiento == 's' || movimiento == 'S')
            {
                if(mapa1[x+1][y] != '|' && mapa1[x+1][y] != '|')
                {
                    mapa1[x][y] = ' ';
                    x++;
                    mapa1[x][y] = 'O';
                }
            }
            else
            {
                printf("\nCamino bloqueado!!\n");
                system("pause");
            }
        }
        if(movimiento == 'w' || movimiento == 'W')
        {
            if(mapa1[x-1][y] != '|' && mapa1[x-1][y] != '|')
            {
                mapa1[x][y] = ' ';
                x--;
                mapa1[x][y] = 'O';
            }
        }
        else
        {
            printf("\nCamino bloqueado!!\n");
            system("pause");
        }
    }
    if(movimiento == 'a' || movimiento == 'A')
    {
        if(mapa1[x][y-1] != '|' && mapa1[x][y-1] != '|')
        {
            mapa1[x][y] = ' ';
            y--;
            mapa1[x][y] = 'o';
        }
    }
    else
    {
        printf("\nCamino bloqueado!!\n");
        system("pause");
    }
}
if(movimiento == 'd' || movimiento == 'D')
{
    if(mapa1[x][y+1] != '|' && mapa1[x][y+1] != '|')

```



```

"\t| | | | | | | | =| =| |",
"\t|==== ===== | |=| |=|      |",
"\t| | | | | | | | |=| |",
"\t|= =| | | | | | |=| |=| | |",
"\t| | | | | | | |=| | |",
"\t| | ===== | |====== |",
"\t| | | | |",
"\t=====",
};

```

```

    int x,y,ancho,altura;
    char movimiento;
    x=1;
    y=2;
    ancho=32;
    altura=1;
    int regalo2x=5,regalo2y=30;
    for(int i=0;i<=18;i++)
        puts(mapa2[i]);
    while(x != altura || y != ancho)
    {
        movimiento = getchar();
        if(movimiento == 's' || movimiento == 'S')
        {
            if(mapa2[x+1][y] != '=' && mapa2[x+1][y] != '|')
            {
                mapa2[x][y] = ' ';
                x++;
                mapa2[x][y] = 'O';
            }
            else
            {
                {
                    printf("\nCamino bloqueado!!\n");
                    system("pause");
                }
            }
        }
        if(movimiento == 'w' || movimiento == 'W')
        {
            if(mapa2[x-1][y] != '=' && mapa2[x-1][y] != '|')
            {
                mapa2[x][y] = ' ';
                x--;
                mapa2[x][y] = 'O';
            }
            else
            {
                {
                    printf("\nCamino bloqueado!!\n");
                    system("pause");
                }
            }
        }
    }
}

```



```

if(movimiento == 'a' || movimiento=='A')
{
    if(mapa2[x][y-1] != '|' && mapa2[x][y-1] != '=')
    {
        mapa2[x][y] = ' ';
        y--;
        mapa2[x][y]='o';
    }
    else
        {
            printf("\nCamino bloqueado!!\n");
            system("pause");
        }
}
if(movimiento == 'd' || movimiento=='D')
{
    if(mapa2[x][y+1] != '|' && mapa2[x][y+1] != '=')
    {
        mapa2[x][y] = ' ';
        y++;
        mapa2[x][y]='o';
    }
    else
        {
            printf("\nCamino bloqueado!!\n");
            system("pause");
        }
}
if(y==regalo2y && x==regalo2x && contreg2==0){
    printf("\n¡a OBTUVISTE UN REGALO!\n");
    contreg2++;
    system("pause");
    printf("\nEscoge entre: Helado o Lentas\n");
    insertarPremio();
    system("pause");
}
system("cls");
for(int i=0;i<=18;i++){
    puts(mapa2[i]);
}
system("cls");
printf("\033[0;33m");
printf("\t\t\t\t\t _____ \n");
printf("\t\t\t\t\t _ _ | _ | | _ | _ _ | _ | _ |\n");
printf("\t\t\t\t\t | | _ | | | _ | | | | _ |\n");
printf("\t\t\t\t\t _____ | | | _____ |\n");
printf("\033[0;0m");
system("pause");

```

```

        system("cls");
    }

    void nivel3(){
        system("cls");
        char mapa3[50][50]=
            {"\t=====",
            "\t|o      |",
            "\t| ===== |==|=====| |",
            "\t|= || | | | | |",
            "\t| || | ===== |==| |",
            "\t| === |==| | | ==|= |",
            "\t| || | | ===== | |",
            "\t| | | | | || ===== |=",
            "\t| | | |==|=| || | |",
            "\t|===== | | | |==|=",
            "\t| | |==|=| |==| | |<+>",
            "\t| | | | ||$| ==|=| |",
            "\t| ===== |==| ||= | |",
            "\t| | | | | || |=| | |",
            "\t| | | | | | | | |=| |",
            "\t| | | | | | | |=| |",
            "\t| | ===== | | ===== |",
            "\t| | | |",
            "\t=====",
            };

        int x,y,ancho,altura;
        char movimiento;
        x=1;
        y=2;
        ancho=32;
        altura=10;
        int regalo3x=11,regalo3y=21;
        for(int i=0;i<=18;i++)
            puts(mapa3[i]);
        while(x != altura || y != ancho)
        {
            movimiento = getchar();
            if(movimiento == 's' || movimiento=='S')
            {
                if(mapa3[x+1][y] !=' '&&mapa3[x+1][y] !='|')
                {
                    mapa3[x][y] = ' ';
                    x++;
                    mapa3[x][y]='O';
                }
            }
            else
            {
                printf("\nCamino bloqueado!!\n");
            }
        }
    }

```

```

        system("pause");
    }
}
if(movimiento=='w' || movimiento=='W')
{
    if(mapa3[x-1][y] != '=' && mapa3[x-1][y] != '|')
    {
        mapa3[x][y] = ' ';
        x--;
        mapa3[x][y]='O';
    }
    else
    {
        printf("\nCamino bloqueado!!\n");
        system("pause");
    }
}
if(movimiento == 'a' || movimiento=='A')
{
    if(mapa3[x][y-1] != '|' && mapa3[x][y-1] != '=')
    {
        mapa3[x][y] = ' ';
        y--;
        mapa3[x][y]='o';
    }
    else
    {
        printf("\nCamino bloqueado!!\n");
        system("pause");
    }
}
if(movimiento == 'd' || movimiento=='D')
{
    if(mapa3[x][y+1] != '|' && mapa3[x][y+1] != '=')
    {
        mapa3[x][y] = ' ';
        y++;
        mapa3[x][y]='o';
    }
    else
    {
        printf("\nCamino bloqueado!!\n");
        system("pause");
    }
}
if(y==regalo3y && x==regalo3x && contreg3==0){
    printf("\n¡aOBTUVISTE UN REGALO\n");
    contreg3++;
    system("pause");
}

```

```

        printf("\nEscoge entre: Batman o Luna\n");
        insertarPremio();
        system("pause");
    }
    system("cls");
    for(int i=0;i<=18;i++){
        puts(mapa3[i]);
    }
}

system("cls");
printf("\033[0;33m");
printf("\t\t\t\t\t _____ \n");
printf("\t\t\t\t\t | _ | _ | | | _ | _ | _ | \n");
printf("\t\t\t\t\t | | | | | | | | | | \n");
printf("\t\t\t\t\t | | | | | | | | | | \n");
printf("\033[0;0m");
system("pause");
system("cls");
}

```