Proyecto final:

Rojito Adventure

Nombre de los alumnos /ID:

Fernando Toledo Pacciotta / 335097

Paris Ahmed Hernández Pérez / 333176

Nombre del centro:

Centro de Ciencias básicas

Nombre de la carrera

Ingeniería Robótica

Nombre del módulo:

Programación I

Nombre del Maestro:

David Alejandro Montoya Murillo

Grado y grupo:

3°A

Fecha de entrega: 9 de diciembre de 2022

índice:

Introducción:

La programación consiste en un proceso en el cual creamos conjuntos de instrucciones que le dicen a una computadora como realizar algún tipo de tarea. Pero esta noción no solo incluye escribir un código y que la computadora lo ejecute, también se debe tener en cuenta una serie de tareas necesarias para que dicho código funcione correctamente, no solo hablando de la sintaxis, si no también, que es lo que se necesita que realice la computadora.

El lenguaje entendido por una computadora se conoce como código máquina. Consiste en secuencias de instrucciones básicas que el procesador reconoce, codificadas como cadenas de números 1 y 0 (binario). En los primeros tiempos de la computación se programaba directamente en código máquina. Escribir programas así resultaba demasiado complicado, también era difícil entenderlos y mantenerlos una vez escritos. Con el tiempo, se fueron desarrollando herramientas para facilitar el trabajo.

Un aspecto importante de la programación son los denominados “paradigmas” que consisten en una clasificación de los lenguajes de programación según sus características, los diferentes idiomas de programación pueden pertenecer a múltiples paradigmas, tales como:

- Programación imperativa
    - Programación estructurada
    - Programación procedimental
    - Programación orientada a objetos
- Programación declarativa
    - Programación funcional
    - Programación lógica

C++ es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C y añadir mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido.

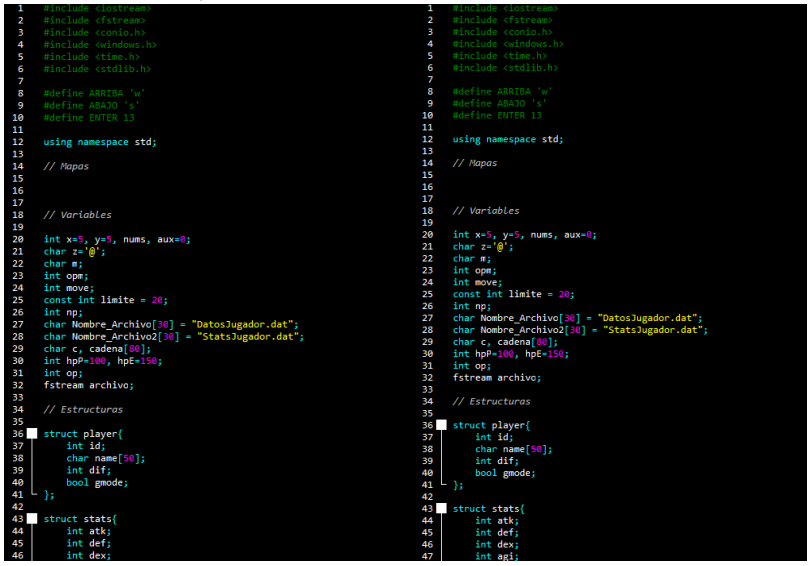
Nuestro proyecto fue enfocado en la realización de un juego en el lenguaje de c++, haciendo uso de lo visto a lo largo del curso e investigación por cuenta propia. Diseñamos el juego teniendo como inspiración los “roguelite” y tomando algunos aspectos de los rpgs como son los “stats” entre otros.

Desarrollo:

Requerimientos:

El programa fue realizado con el fin de ser un videojuego al estilo roguelite con ciertos aspectos de RPG, permitiendo a su vez, llevar un registro de los datos de cada jugador a la hora de crear una nueva partida, así como poder abrir una ya creada buscando a partir del nombre con el que el jugador se registró. también cuenta con algunos secretos esperando a ser encontrados, estos no afectan a la propia progresión del juego, pero permiten añadirle un cierto grado de aventura a un juego relativamente “simple”.

solución a los requerimientos:



```
#include <iostream>
#include <fstream>
#include <conio.h>
#include <windows.h>
#include <time.h>
#include <stdlib.h>

#define ARRIBA 'w'
#define ABAJO 's'
#define ENTER 13

using namespace std;

// Mapas



// Variables

int x=5, y=5, nums, aux=0;
char z='@';
char m;
int opm;
int move;
const int limite = 20;
int np;
char Nombre_Archivo[30] = "DatosJugador.dat";
char Nombre_Archivo2[30] = "StatsJugador.dat";
char c, cadena[80];
int hpP=100, hpE=150;
int op;
fstream archivo;

// Estructuras

struct player{
    int id;
    char name[50];
    int dif;
    bool gmode;
};

struct stats{
    int atk;
    int def;
    int dex;
```



```
#include <iostream>
#include <fstream>
#include <conio.h>
#include <windows.h>
#include <time.h>
#include <stdlib.h>

#define ARRIBA 'w'
#define ABAJO 's'
#define ENTER 13

using namespace std;

// Mapas



// Variables

int x=5, y=5, nums, aux=0;
char z='@';
char m;
int opm;
int move;
const int limite = 20;
int np;
char Nombre_Archivo[30] = "DatosJugador.dat";
char Nombre_Archivo2[30] = "StatsJugador.dat";
char c, cadena[80];
int hpP=100, hpE=150;
int op;
fstream archivo;

// Estructuras

struct player{
    int id;
    char name[50];
    int dif;
    bool gmode;
};

struct stats{
    int atk;
    int def;
    int dex;
    int agi;
```

```cpp
#include <iostream>
#include <fstream>
#include <conio.h>
#include <windows.h>
#include <time.h>
#include <stdlib.h>

#define ARRIBA 'w'
#define ABAJO 's'
#define ENTER 13

using namespace std;

// Mapas



// Variables

int x=5, y=5, nums, aux=0;
char z='@';
char m;
int opm;
int move;
const int limite = 20;
int np;
char Nombre_Archivo[30] = "DatosJugador.dat";
char Nombre_Archivo2[30] = "StatsJugador.dat";
char c, cadena[80];
int hpP=100, hpE=150;
int op;
fstream archivo;

// Estructuras

struct player{
    int id;
    char name[50];
    int dif;
    bool gmode;
};

struct stats{
    int atk;
    int def;
    int dex;
    int agi;
```


```cpp
#include <iostream>
#include <fstream>
#include <conio.h>
#include <windows.h>
#include <time.h>
#include <stdlib.h>

#define ARRIBA 'w'
#define ABAJO 's'
#define ENTER 13

using namespace std;

// Mapas



// Variables

int x=5, y=5, nums, aux=0;
char z='@';
char m;
int opm;
int move;
const int limite = 20;
int np;
char Nombre_Archivo[30] = "DatosJugador.dat";
char Nombre_Archivo2[30] = "StatsJugador.dat";
char c, cadena[80];
int hpP=100, hpE=150;
int op;
fstream archivo;

// Estructuras

struct player{
    int id;
    char name[50];
    int dif;
    bool gmode;
};

struct stats{
    int atk;
    int def;
    int dex;
    int agi;
```



```cpp
#include <iostream>
#include <fstream>
#include <conio.h>
#include <windows.h>
#include <time.h>
#include <stdlib.h>

#define ARRIBA 'w'
#define ABAJO 's'
#define ENTER 13

using namespace std;

// Mapas



// Variables

int x=5, y=5, nums, aux=0;
char z='@';
char m;
int opm;
int move;
const int limite = 20;
int np;
char Nombre_Archivo[30] = "DatosJugador.dat";
char Nombre_Archivo2[30] = "StatsJugador.dat";
char c, cadena[80];
int hpP=100, hpE=150;
int op;
fstream archivo;

// Estructuras

struct player{
    int id;
    char name[50];
    int dif;
    bool gmode;
};

struct stats{
    int atk;
    int def;
    int dex;
    int agi;
```



```cpp
#include <iostream>
#include <fstream>
#include <conio.h>
#include <windows.h>
#include <time.h>
#include <stdlib.h>

#define ARRIBA 'w'
#define ABAJO 's'
#define ENTER 13

using namespace std;

// Mapas



// Variables

int x=5, y=5, nums, aux=0;
char z='@';
char m;
int opm;
int move;
const int limite = 20;
int np;
char Nombre_Archivo[30] = "DatosJugador.dat";
char Nombre_Archivo2[30] = "StatsJugador.dat";
char c, cadena[80];
int hpP=100, hpE=150;
int op;
fstream archivo;

// Estructuras

struct player{
    int id;
    char name[50];
    int dif;
    bool gmode;
};

struct stats{
    int atk;
    int def;
    int dex;
    int agi;
```

```
294             cout << "   Defensa : "; cin >> y.def; lim=lim+y.def;
295             cout << "   Destreza : "; cin >> y.dex; lim=lim+y.dex;
296             cout << "   Agilidad     : "; cin >> y.agi; lim=lim+y.agi;
297             cout << "   Arcana : "; cin >> y.arc; lim=lim+y.arc;
298
299         fstream archivo;
300         archivo.open(Nombre_Archivo2, ios::binary|ios::in|ios::out);
301         archivo.seekp((numero) * sizeof(stats), ios::beg); // write escribe el registro una vez ya posicionado
302         archivo.write(reinterpret_cast<char*>(&y),sizeof(stats));
303         cout << "Datos grabados...\n ";
304         archivo.close();
305     }
306
307     void readO(){
308         fstream archivo;
309         char nombreB[50];
310         char nombreR[50];
311         bool encontrado = false;
312         cout << "Que jugador (nombre) buscas: "; cin >> nombreB;
313         for(int i=0; i<strlen(nombreB) ; i++){
314             nombreB[i]= toupper(nombreB[i]);
315         }
316         archivo.open(Nombre_Archivo, ios::binary|ios::in);
317         if (!archivo){
318             cerr << " No se pudo abrir el archivo " << endl;
319         }
320         else{
321             player x;
322             for(int j=0; j<limite; j++){
323                 archivo.read( reinterpret_cast<char *> (&x),sizeof(player));
324                 char nombreR[50];
325                 strcpy(nombreR,x.name);
326                 for(int i=0; i<strlen(nombreR) ; i++){
327                     nombreR[i]= toupper(nombreR[i]);
328                 }
329                 if(strcmp(nombreB,nombreR)==0){
330                     cout << "-Jugador  " << j+1<<endl;
331                     cout << "   ID     : " << x.id << endl;
332                     cout << "   Nombre : " << x.name << endl;
333                     cout << "   Dificultad : " << x.dif << endl;
334                     cout << "   Modo dios : " << x.gmode << endl << endl;
335                     encontrado = true;
336                 }
337             }
338             archivo.close(); // cierra el archivo
339             if(encontrado == false){
340                 cout << "El jugador con ese nombre no esta " << endl;
```

```
52      player x;
53      stats y;
54  };
55
56  // Funciones
57
58  void combat();
59  void status();
60  int menu(const char titulo[], const char *opciones[], int n);
61  void credits();
62  void atkk();
63  void dexx();
64  void arcc();
65  void damage();
66  void enemy();
67  void history();
68  void game();
69
70  void gotoxy(int x, int y, char z){
71      HANDLE hcon;
72      hcon = GetStdHandle(STD_OUTPUT_HANDLE);
73      COORD dwPos;
74      dwPos.X = x;
75      dwPos.Y= y;
76      SetConsoleCursorPosition(hcon,dwPos);
77      cout << z;
78  }
79
80  void movement(){
81      do{
82          m = getch();
83          if(m=='w'){
84              system("cls");
85              gotoxy(x,y=y-2,z);
86              aux++;
87              cout << aux;
88              combat();
89          }
90          else if(m=='a'){
91              system("cls");
92              gotoxy(x=x-2,y,z);
93              aux++;
94              cout << aux;
95              combat();
96          }
97          else if(m=='s'){
```

```
52      player x;
53      stats y;
54  };
55
56  // Funciones
57
58  void combat();
59  void status();
60  int menu(const char titulo[], const char *opciones[], int n);
61  void credits();
62  void atkk();
63  void dexx();
64  void arcc();
65  void damage();
66  void enemy();
67  void history();
68  void game();
69
70  void gotoxy(int x, int y, char z){
71      HANDLE hcon;
72      hcon = GetStdHandle(STD_OUTPUT_HANDLE);
73      COORD dwPos;
74      dwPos.X = x;
75      dwPos.Y= y;
76      SetConsoleCursorPosition(hcon,dwPos);
77      cout << z;
78  }
79
80  void movement(){
81      do{
82          m = getch();
83          if(m=='w'){
84              system("cls");
85              gotoxy(x,y=y-2,z);
86              aux++;
87              cout << aux;
88              combat();
89          }
90          else if(m=='a'){
91              system("cls");
92              gotoxy(x=x-2,y,z);
93              aux++;
94              cout << aux;
95              combat();
96          }
97          else if(m=='s'){
98              system("cls");
```

```
    player x;
    stats y;
};

// Funciones

void combat();
void status();
int menu(const char titulo[], const char *opciones[], int n);
void credits();
void atkk();
void dexx();
void arcc();
void damage();
void enemy();
void history();
void game();

void gotoxy(int x, int y, char z){
    HANDLE hcon;
    hcon = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD dwPos;
    dwPos.X = x;
    dwPos.Y= y;
    SetConsoleCursorPosition(hcon,dwPos);
    cout << z;
}

void movement(){
    do{
        m = getch();
        if(m=='w'){
            system("cls");
            gotoxy(x,y=y-2,z);
            aux++;
            cout << aux;
            combat();
        }
        else if(m=='a'){
            system("cls");
            gotoxy(x=x-2,y,z);
            aux++;
            cout << aux;
            combat();
        }
        else if(m=='s'){
            system("cls");
```

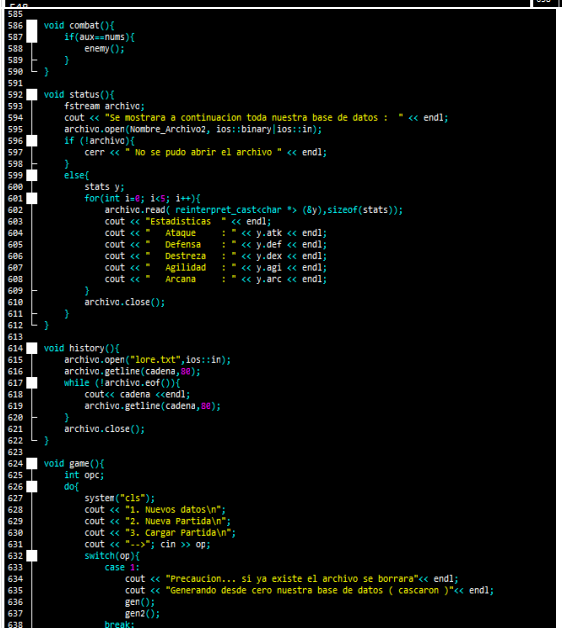
```
    player x;
    stats y;
};

// Funciones

void combat();
void status();
int menu(const char titulo[], const char *opciones[], int n);
void credits();
void atkk();
void dexx();
void arcc();
void damage();
void enemy();
void history();
void game();

void gotoxy(int x, int y, char z){
    HANDLE hcon;
    hcon = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD dwPos;
    dwPos.X = x;
    dwPos.Y= y;
    SetConsoleCursorPosition(hcon,dwPos);
    cout << z;
}

void movement(){
    do{
        m = getch();
        if(m=='w'){
            system("cls");
            gotoxy(x,y=y-2,z);
            aux++;
            cout << aux;
            combat();
        }
        else if(m=='a'){
            system("cls");
            gotoxy(x=x-2,y,z);
            aux++;
            cout << aux;
            combat();
        }
        else if(m=='s'){
            system("cls");
```

```

void battle(){
    //bool vic=false;
    int run;
    bool repite = true;
    int opcion;
    const char *titulo = "Let the Battle begin";
    const char *opciones[] = {"Ataque fuerte", "Ataque rapido", "Ataque magico", "Huir"};
    int n = 4;  // Numero de opciones
    do{
        opcion = menu(titulo, opciones, n);
        switch (opcion) {
            case 1:
                atkk();
                damage();
                if(hpE<=0){
                    cout << "Ganaste";
                    system("pause");
                    aux=0;
                    hpP=100;
                    hpE=150;
                    movement();
                }
            break;

            case 2:
                dexx();
                damage();
                if(hpE<=0){
                    cout << "Ganaste";
                    system("pause");
                    aux=0;
                    hpP=100;
                    hpE=150;
                    movement();
                }
            break;

            case 3:
                arcc();
                damage();
                if(hpE<=0){
                    cout << "Ganaste";
                    system("pause");
                    aux=0;
                    hpP=100;
                    hpE=150;
                    movement();
                }
            break;

            case 4:
                srand(time(NULL));
                run=1+rand()%(31-1);
```

```
                    hpE=150;
                    movement();
                }
            break;

            case 3:
                arcc();
                damage();
                if(hpE<=0){
                    cout << "Ganaste";
                    system("pause");
                    aux=0;
                    hpP=100;
                    hpE=150;
                    movement();
                }
            break;

            case 4:
                srand(time(NULL));
                run=1+rand()%(31-1);
                if(run>=10){
                    cout << "Has logrado escapar...";
                    hpP=100;
                    hpE=150;
                    system("pause");
                    movement();
                }else{
                    cout << "No has logrado escapar...";
                    system("pause");
                }
            break;
        }
    } while(repite);
}

void atkk(){
    int dmg;
    srand(time(NULL));
    dmg=1+rand()%(51-1);
    cout << "Haz realizado: " << dmg << "de danio";
    system("pause");
    hpE=hpE-dmg;
}

void dexx(){
    int dmg;
    srand(time(NULL));
    dmg=1+rand()%(51-1);
    cout << "Haz realizado: " << dmg << "de danio";
    system("pause");
    hpE=hpE-dmg;
}

```

```
                    hpE=150;
                    movement();
                }
            break;

            case 3:
                arcc();
                damage();
                if(hpE<=0){
                    cout << "Ganaste";
                    system("pause");
                    aux=0;
                    hpP=100;
                    hpE=150;
                    movement();
                }
            break;

            case 4:
                srand(time(NULL));
                run=1+rand()%(31-1);
                if(run>=10){
                    cout << "Has logrado escapar...";
                    hpP=100;
                    hpE=150;
                    system("pause");
                    movement();
                }else{
                    cout << "No has logrado escapar...";
                    system("pause");
                }
            break;
        }
    } while(repite);
}

void atkk(){
    int dmg;
    srand(time(NULL));
    dmg=1+rand()%(51-1);
    cout << "Haz realizado: " << dmg << "de danio";
    system("pause");
    hpE=hpE-dmg;
}

void dexx(){
    int dmg;
    srand(time(NULL));
    dmg=1+rand()%(51-1);
    cout << "Haz realizado: " << dmg << "de danio";
    system("pause");
    hpE=hpE-dmg;
}
```

```
void combat(){
    if(aux==nums){
        enemy();
    }
}

void status(){
    fstream archivo;
    cout << "Se mostrara a continuacion toda nuestra base de datos :  " << endl;
    archivo.open(Nombre_Archivo2, ios::binary|ios::in);
    if (!archivo){
        cerr << " No se pudo abrir el archivo " << endl;
    }
    else{
        stats y;
        for(int i=0; i<5; i++){
            archivo.read( reinterpret_cast<char *> (&y),sizeof(stats));
            cout << "Estadisticas  " << endl;
            cout << "   Ataque     : " << y.atk << endl;
            cout << "   Defensa    : " << y.def << endl;
            cout << "   Destreza   : " << y.dex << endl;
            cout << "   Agilidad   : " << y.agi << endl;
            cout << "   Arcana     : " << y.arc << endl;
        }
        archivo.close();
    }
}

void history(){
    archivo.open("lore.txt",ios::in);
    archivo.getline(cadena,80);
    while (!archivo.eof()){
        cout<< cadena <<endl;
        archivo.getline(cadena,80);
    }
    archivo.close();
}

void game(){
    int opc;
    do{
        system("cls");
        cout << "1. Nuevos datos\n";
        cout << "2. Nueva Partida\n";
        cout << "3. Cargar Partida\n";
        cout << "-->"; cin >> op;
        switch(op){
            case 1:
                cout << "Precaucion... si ya existe el archivo se borrara"<< endl;
                cout << "Generando desde cero nuestra base de datos ( cascaron )"<< endl;
                gen();
                gen2();
            break;
```



```
void combat(){
    if(aux==nums){
        enemy();
    }
}

void status(){
    fstream archivo;
    cout << "Se mostrara a continuacion toda nuestra base de datos :  " << endl;
    archivo.open(Nombre_Archivo2, ios::binary|ios::in);
    if (!archivo){
        cerr << " No se pudo abrir el archivo " << endl;
    }
    else{
        stats y;
        for(int i=0; i<5; i++){
            archivo.read( reinterpret_cast<char *> (&y),sizeof(stats));
            cout << "Estadisticas  " << endl;
            cout << "   Ataque     : " << y.atk << endl;
            cout << "   Defensa    : " << y.def << endl;
            cout << "   Destreza   : " << y.dex << endl;
            cout << "   Agilidad   : " << y.agi << endl;
            cout << "   Arcana     : " << y.arc << endl;
        }
        archivo.close();
    }
}

void history(){
    archivo.open("lore.txt",ios::in);
    archivo.getline(cadena,80);
    while (!archivo.eof()){
        cout<< cadena <<endl;
        archivo.getline(cadena,80);
    }
    archivo.close();
}

void game(){
    int opc;
    do{
        system("cls");
        cout << "1. Nuevos datos\n";
        cout << "2. Nueva Partida\n";
        cout << "3. Cargar Partida\n";
        cout << "-->"; cin >> op;
        switch(op){
            case 1:
                cout << "Precaucion... si ya existe el archivo se borrara"<< endl;
                cout << "Generando desde cero nuestra base de datos ( cascaron )"<< endl;
                gen();
                gen2();
            break;
```

Manual de uso:

Gotoxy() / Gotoxy2(): su principal función viene dada en poder ir moviendo tanto el personaje como las flechas que permiten se puedan ir moviendo.

void combat(); Funcion principal que permite entrar en modo combate

void status(); Funcion que permite ver las estadísticas

int menu(const char titulo[], const char *opciones[], int n); Funcion que forma el menú del juego

void credits(); Funcion que muestra los créditos del proyecto

void atkk(); Funcion que mide el ataque en base a fuerza

void dexx(); Funcion que mide el ataque en base a destreza

void arcc();Funcion que mide el ataque en base a magia

void damage(); Funcion que marca el daño recibido por el enemigo

void enemy(); Funcion que permite que el enemigo aparezca en pantalla

void history(); Funcion que muestra la historia del juego

void game(); Funcion que permite generar los datos de los usuarios para empezar a jugar

void battle(); Funcion que permite realizar acciones en el combate

char getch2 () y getch(); permiten guardar datos con solo presionar el botón

void hub(); hub principal del programa

void show(); permite ir mostrando el movimiento del jugador

void movement(); permite al jugador moverse

Las funciones gen, gen2, caps, caps2, readO, readO2, add, add2. Permiten realizar las operaciones binarias para ir registrando la los jugadores.

Conclusión:

Fernando Toledo Pacciotta:

En conclusión, “Rojito Adventure” es una aventura creada en c++ haciendo uso de lo aprendido en el curso más una investigación aparte que buscaba complementar el programa. Constando de bases como los “roguelite” se busco un estilo de juego simple pero que diera paso al análisis. Podríamos decir que es un proyecto con el que se busco poder poner a prueba todos los conocimientos adquiridos de una forma práctica y que a su vez busco poder entretener a otros.

Paris Ahmed Hernández Pérez:

Durante este proyecto estuvimos realizando líneas de código que utilizamos durante una gran parte del semestre en esta materia (programación), las utilizamos por 2 razones, las especificaciones del proyecto indicaba utilizar cierto

tipo de código y la segunda razón es porque facilitaba el trabajo, utilizamos también líneas de códigos ajenas a las vistas en clase, dado que permitían utilizar otras funciones que le dan más personalidad a nuestro proyecto, el cual es un videojuego; agregamos pequeños guiños en este, uno de ellos podría ser música (dependientemente de si funciona en el momento, esto lo realizamos utilizando un char y opciones que el mismo programa donde se está realizando el proyecto nos brinda), y eso es todo amigos, nos vemos en CP :c .

Referencias:

https://es.wikipedia.org/wiki/C%2B%2B

https://algoritmosyalgomas.com/menu-de-opciones-con-teclas-direccionales-en-c-diferentes-versiones/

https://profile.es/blog/que-son-los-paradigmas-de-programacion/#:~:text=%C2%BFQu%C3%A9%20es%20un%20paradigma%20de,resultados%20que%20necesitan%20los%20programadores.