



## **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: .....	3
Desarrollo: .....	4
Requerimientos: .....	4
solución a los requerimientos: .....	4
Conclusión: .....	8
Referencias: .....	9

## 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:

```
1 #include <iostream>
2 #include <fstream>
3 #include <conio.h>
4 #include <windows.h>
5 #include <time.h>
6 #include <stdlib.h>
7
8 #define ARRIBA 'u'
9 #define ABAJO 's'
10 #define ENTER 13
11
12 using namespace std;
13
14 // Mapas
15
16
17 // Variables
18
19 int x=5, y=5, nums, aux=0;
20 char z='0';
21 char m;
22 int op;
23 int move;
24 const int limite = 20;
25 int np;
26 char Nombre_Archivo[30] = "DatosJugador.dat";
27 char Nombre_Archivo2[30] = "StatsJugador.dat";
28 char c, cadena[80];
29 int hpP=100, hpE=150;
30 int op;
31 fstream archivo;
32
33 // Estructuras
34
35 struct player{
36     int id;
37     char name[50];
38     int dif;
39     bool gmode;
40 };
41
42 struct stats{
43     int atk;
44     int def;
45     int dex;
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```

1 #include <iostream>
2 #include <fstream>
3 #include <conio.h>
4 #include <windows.h>
5 #include <time.h>
6 #include <stdlib.h>
7
8 #define ARRIBA 'w'
9 #define ABAJO 's'
10 #define ENTER 13
11
12 using namespace std;
13
14 // Mapas
15
16
17 // Variables
18
19
20 int x=5, y=5, nums, aux=0;
21 char z='@';
22 char m;
23 int op;
24 int move;
25 const int limite = 20;
26 int np;
27 char Nombre_Archivo[30] = "DatosJugador.dat";
28 char Nombre_Archivo2[30] = "StatsJugador.dat";
29 char c, cadena[80];
30 int hpP=100, hpE=150;
31 int op;
32 fstream archivo;
33
34 // Estructuras
35
36 struct player{
37     int id;
38     char name[50];
39     int dif;
40     bool gmode;
41 };
42
43 struct stats{
44     int atk;
45     int def;
46     int dex;
47     int agi;

```

```

1 #include <iostream>
2 #include <fstream>
3 #include <conio.h>
4 #include <windows.h>
5 #include <time.h>
6 #include <stdlib.h>
7
8 #define ARRIBA 'w'
9 #define ABAJO 's'
10 #define ENTER 13
11
12 using namespace std;
13
14 // Mapas
15
16
17 // Variables
18
19
20 int x=5, y=5, nums, aux=0;
21 char z='@';
22 char m;
23 int op;
24 int move;
25 const int limite = 20;
26 int np;
27 char Nombre_Archivo[30] = "DatosJugador.dat";
28 char Nombre_Archivo2[30] = "StatsJugador.dat";
29 char c, cadena[80];
30 int hpP=100, hpE=150;
31 int op;
32 fstream archivo;
33
34 // Estructuras
35
36 struct player{
37     int id;
38     char name[50];
39     int dif;
40     bool gmode;
41 };
42
43 struct stats{
44     int atk;
45     int def;
46     int dex;
47     int agi;

```

```

1 #include <iostream>
2 #include <fstream>
3 #include <conio.h>
4 #include <windows.h>
5 #include <time.h>
6 #include <stdlib.h>
7
8 #define ARRIBA 'w'
9 #define ABAJO 's'
10 #define ENTER 13
11
12 using namespace std;
13
14 // Mapas
15
16
17 // Variables
18
19
20 int x=5, y=5, nums, aux=0;
21 char z='@';
22 char m;
23 int op;
24 int move;
25 const int limite = 20;
26 int np;
27 char Nombre_Archivo[30] = "DatosJugador.dat";
28 char Nombre_Archivo2[30] = "StatsJugador.dat";
29 char c, cadena[80];
30 int hpP=100, hpE=150;
31 int op;
32 fstream archivo;
33
34 // Estructuras
35
36 struct player{
37     int id;
38     char name[50];
39     int dif;
40     bool gmode;
41 };
42
43 struct stats{
44     int atk;
45     int def;
46     int dex;
47     int agi;

```

```

1 #include <iostream>
2 #include <fstream>
3 #include <conio.h>
4 #include <windows.h>
5 #include <time.h>
6 #include <stdlib.h>
7
8 #define ARRIBA 'w'
9 #define ABAJO 's'
10 #define ENTER 13
11
12 using namespace std;
13
14 // Mapas
15
16
17 // Variables
18
19
20 int x=5, y=5, nums, aux=0;
21 char z='@';
22 char m;
23 int op;
24 int move;
25 const int limite = 20;
26 int np;
27 char Nombre_Archivo[30] = "DatosJugador.dat";
28 char Nombre_Archivo2[30] = "StatsJugador.dat";
29 char c, cadena[80];
30 int hpP=100, hpE=150;
31 int op;
32 fstream archivo;
33
34 // Estructuras
35
36 struct player{
37     int id;
38     char name[50];
39     int dif;
40     bool gmode;
41 };
42
43 struct stats{
44     int atk;
45     int def;
46     int dex;
47     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 read0(){
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-2,z);
86             aux++;
87             cout << aux;
88             combat();
89         }
90         else if(m=='a'){
91             system("cls");
92             gotoxy(x-2,y,z);
93             aux++;
94             cout << aux;
95             combat();
96         }
97         else if(m=='s'){
98             system("cls");

```

```

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-2,z);
86             aux++;
87             cout << aux;
88             combat();
89         }
90         else if(m=='a'){
91             system("cls");
92             gotoxy(x-2,y,z);
93             aux++;
94             cout << aux;
95             combat();
96         }
97         else if(m=='s'){
98             system("cls");

```

```

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");

```

```

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");

```

```

462
463 void battle(){
464     //bool vici=false;
465     int run;
466     bool repite = true;
467     int opcion;
468     const char *titulo = "Let the battle begin";
469     const char *opciones[] = {"Ataque fuerte", "Ataque rapido", "Ataque magico", "Muir"};
470     int n = 4; // numero de opciones
471     do{
472         opcion = menu(titulo, opciones, n);
473         switch (opcion) {
474             case 1:
475                 atkk();
476                 damage();
477                 if(hpE<=0){
478                     cout << "Ganaste";
479                     system("pause");
480                     aux=0;
481                     hpP=100;
482                     hpE=150;
483                     movement();
484                 }
485                 break;
486             case 2:
487                 dexx();
488                 damage();
489                 if(hpE<=0){
490                     cout << "Ganaste";
491                     system("pause");
492                     aux=0;
493                     hpP=100;
494                     hpE=150;
495                     movement();
496                 }
497                 break;
498             case 3:
499                 arcc();
500                 damage();
501                 if(hpE<=0){
502                     cout << "Ganaste";
503                     system("pause");
504                     aux=0;
505                     hpP=100;
506                     hpE=150;
507                     movement();
508                 }
509                 break;
510             case 4:
511                 srand(time(NULL));
512                 run=1+rand()%(31-1);

```

```

495         hpE=150;
496         movement();
497     }
498     break;
499     case 3:
500         arcc();
501         damage();
502         if(hpE<=0){
503             cout << "Ganaste";
504             system("pause");
505             aux=0;
506             hpP=100;
507             hpE=150;
508             movement();
509         }
510         break;
511     case 4:
512         srand(time(NULL));
513         run=1+rand()%(31-1);
514         if(run>=10){
515             cout << "Has logrado escapar...";
516             hpP=100;
517             hpE=150;
518             system("pause");
519             movement();
520         }
521         else{
522             cout << "No has logrado escapar...";
523             system("pause");
524         }
525         break;
526     }
527 } while(repite);
528 }
529
530 void atkk(){
531     int dmg;
532     srand(time(NULL));
533     dmg=1+rand()%(31-1);
534     cout << "Haz realizado: " << dmg << "de danio";
535     system("pause");
536     hpE=hpE-dmg;
537 }
538
539 void dexx(){
540     int dmg;
541     srand(time(NULL));
542     dmg=1+rand()%(31-1);
543     cout << "Haz realizado: " << dmg << "de danio";
544     system("pause");
545     hpE=hpE-dmg;
546 }
547
548

```

```

496         hpE=150;
497         movement();
498     }
499     break;
500     case 3:
501         arcc();
502         damage();
503         if(hpE<=0){
504             cout << "Ganaste";
505             system("pause");
506             aux=0;
507             hpE=100;
508             hpE=150;
509             movement();
510         }
511         break;
512     case 4:
513         srand(time(NULL));
514         run=1+rand()%(31-1);
515         if(run>=10){
516             cout << "Has logrado escapar...";
517             hpE=100;
518             hpE=150;
519             system("pause");
520             movement();
521         }else{
522             cout << "No has logrado escapar...";
523             system("pause");
524         }
525         break;
526     }
527 } while(repite);
528 }
529
530 void atkk(){
531     int dmg;
532     srand(time(NULL));
533     dmg=1+rand()%(51-1);
534     cout << "Haz realizado: " << dmg << "de danio";
535     system("pause");
536     hpE=hpE-dmg;
537 }
538
539 void dexx(){
540     int dmg;
541     srand(time(NULL));
542     dmg=1+rand()%(51-1);
543     cout << "Haz realizado: " << dmg << "de danio";
544     system("pause");
545     hpE=hpE-dmg;
546 }
547
548 void combat(){
549     if(aux==nums){
550         enemy();
551     }
552 }
553
554 void status(){
555     ifstream archivo;
556     cout << "Se mostrara a continuacion toda nuestra base de datos : " << endl;
557     archivo.open(nombre_Archivo2, ios::binary|ios::in);
558     if (!archivo){
559         cerr << "No se pudo abrir el archivo " << endl;
560     }
561     else{
562         stats y;
563         for(int i=0; i<5; i++){
564             archivo.read(reinterpret_cast<char*>(&y),sizeof(stats));
565             cout << "Estadisticas " << endl;
566             cout << "Ataque : " << y.atk << endl;
567             cout << "Defensa : " << y.def << endl;
568             cout << "Destreza : " << y.dex << endl;
569             cout << "Agilidad : " << y.agi << endl;
570             cout << "Arcana : " << y.arc << endl;
571         }
572         archivo.close();
573     }
574 }
575
576 void history(){
577     archivo.open("lore.txt",ios::in);
578     archivo.getline(cadena,80);
579     while (!archivo.eof()){
580         cout<< cadena <<endl;
581         archivo.getline(cadena,80);
582     }
583     archivo.close();
584 }
585
586 void game(){
587     int opc;
588     do{
589         system("cls");
590         cout << "1. Nuevos datos\n";
591         cout << "2. Nueva Partida\n";
592         cout << "3. Cargar Partida\n";
593         cout << "4.->"; cin >> opc;
594         switch(opc){
595             case 1:
596                 cout << "Precaucion... si ya existe el archivo se borrara"<< endl;
597                 cout << "Generando desde cero nuestra base de datos (cascaron)"<< endl;
598                 gen();
599                 gen2();
600                 break;
601             case 2:
602                 break;
603             case 3:
604                 break;
605             case 4:
606                 break;
607         }
608     }while(opc!=0);
609 }

```

## 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.>