

ARELLANO GRANADOS ANGEL MARIANO
218123444



ARELLANO GRANADOS ANGEL MARIANO

218123444

DIVISIÓN DE ELECTRÓNICA Y
COMPUTACIÓN.

ESTRUCTURAS DE DATOS I

22A D03

DAVID ALEJANDRO GÓMEZ ANAYA

REPORTE ACT. 1

1. Planteamiento del problema.

Un lugar de juegos, requiere un sistema que registra a los jugadores del lugar y acumula sus puntuaciones. El sistema permite dar de alta y eliminar jugadores (hasta 100 jugadores), y sumar y restar puntos a los usuarios.

Las características de los jugadores son: Nombre (30 caracteres), puntos acumulados y un id único que es generado aleatoriamente. Para la eliminación, suma de puntos y decremento de puntos, se requiere de realizar búsquedas, éstas se pueden realizar con el id, que asegura que habrá una sola coincidencia con los jugadores registrados, o con el nombre, que podría haber más de un jugador con el mismo nombre. Si la búsqueda se realiza por nombre, el sistema deberá mostrar la información del jugador y preguntar si ese es el jugador buscado, si sí, continuar con la solicitud (mostrar, eliminar, sumar puntos o restar puntos), si no, entonces debe de buscar otra coincidencia con el nombre y repetir el proceso.

Además de mostrar las coincidencias cuando se realiza una búsqueda, el sistema debe tener la capacidad de encontrar y mostrar al jugador que lleva más puntos acumulados y también mostrar todos los jugadores.

2. Objetivos:

1. Uso de tipos de dato estructurados.
2. Manipulación de datos estructurados en un arreglo: Inserción, Arreglo (agregar y eliminar elementos), búsquedas secuenciales, generación aleatoria de números.

3. Marco teórico.

1. Tipos de datos estructurados.

Las estructuras de datos en C++ se pueden entender como un tipo de dato compuesto (no complejo). Las estructuras de datos permiten almacenar de manera ordenada una serie de valores dados en una misma variable.

1. Arreglos.

Un arreglo (matriz) es una colección ordenada de datos (tanto primitivos u objetos dependiendo del lenguaje). Los arreglos (matrices) se emplean para almacenar múltiples valores en una sola variable, frente a las variables que sólo pueden almacenar un valor (por cada variable).

1. Inserción.

El algoritmo de ordenamiento por inserción es un algoritmo de fácil aplicación que permite el ordenamiento de una lista.

Su funcionamiento consiste en el recorrido por la lista seleccionando en cada iteración un valor como clave y compararlo con el resto insertándolo en el lugar correspondiente.

2. Eliminación.

Si se quiere borrar el dato que hay en una cierta posición, los que estaban a continuación deberán desplazarse "hacia la izquierda" para que no queden huecos. Como en el caso anterior, habrá que actualizar el contador, pero ahora para indicar que queda una posición libre más.

3. Búsqueda secuencial.

Consiste en ir comparando el elemento que se busca con cada elemento del arreglo hasta cuando se encuentra.

2. Registros.

El tipo de datos registro es el que sirve para guardar información de distinto tipo en una estructura única.

4. Desarrollo.

Inicie el proyecto en el IDE codeblocks con el lenguaje de programación de C++, primero agregue mis datos e importe las librerías que considere necesarias para el programa.

Ver Ilustración 1 Librerías

Tras las librerías inicie el arreglo principal del programa, que contenía los 3 datos que el programa usaría para guardar sus datos nombre, ID y puntos, así como declarar la lista de hasta 100 jugadores.

Ver Ilustración 2 Arreglo

Ahora ya me dispuse a iniciar el main que después de iniciar algunas variables que se necesitarían después inicié un ciclo repetitivo do-while para colocar en el un menú que no se detuviera hasta seleccionar la opción adecuada, pero antes coloqué unos cout para mostrar el menú en pantallas y recibir la respuesta con un cin y guardarla en la variable opc.

Ver Ilustración 3 Menú 1

Una vez sabiendo la opción elegida por el usuario la estructura switch se encargaría de ejecutar una acción o acciones dependiendo del número de opc:

Si opc era igual a 1 llamaría a la función Crearjugador() y regresaría al menu.

Sí opc era 2 entraba a otro menú que preguntaría si quiera efectuar una búsqueda por ID o por nombre donde se endentaría otra estructura switch

con una variable llamada opc2 que si era 1 preguntaría el ID a buscar y mandaría el entero ingresado como parámetro en la función `BuscarJugadorID(id)` que retornaría la posición del jugador buscado en el arreglo como entero para inmediatamente mandarlo como parámetro en la función `AccederJugador(pos)`; si opc2 era 2 pasaría exactamente lo mismo con la diferencia de que se preguntaría un char llamado nombre que se enviaría como parámetro a `BuscarJugadorNombre(name)`; Tras terminar los procesos de las funciones regresa al manu principal.

Si opc es 3 llama a la función `MejorPuntaje()` y regresa al menú principal.

Si opc es igual a 4 se llama a la función `MostrarJugadores()` y regresa al menú principal.

Y por último si opc es igual a 5 se sale del menú y acaba el programa tras mostrar un mensaje.

Ver Ilustración 4 Menú 2

Después de definir el main podemos iniciar con las funciones en sí, inicie con las más importante la de crear jugador, pues para probar las demás tenía que llenar algunas posiciones de la lista, primero declare la variable global contador igualada a cero que representaba la cantidad de jugadores que ya había registrado y por lo tanto la posición siguiente a llenar en la lista, en esta función primero inicie unas variables y cree una bandera, antes de todo confirmaba si no se habían registrado ya los 100 jugadores si sí continuaba, sino regresaba al menú principal con un error, después conforme el contador preguntaba el nombre del jugador y lo guardaba en el arreglo en la parte del nombre, declaraba sus puntos en cero y generaba un ID aleatorio e irrepitable a través de un método que revisaba todos los ID previamente usados para asegurarse que el generado no se había repetido; al final del proceso mostraba el nombre y el ID como mensaje de confirmación, y regresa al menú principal también aumenta el contador principal.

Ver Ilustración 5 Crear Un Jugador

Para la función de acceder a un jugador primero se tiene que saber a que jugador se quiere acceder, por ello el programa tiene dos métodos de búsqueda por ID y por nombre, el de ID es muy simple es una función con paso de parámetro que retorna un entero que va a ser la posición del jugador que se desea acceder a sus datos; la función recibirá un entero que almacenara en la variable id y con ayuda de una bandera y un ciclo repetitivo for compara el id recibido con todos los id de la lista si encuentra coincidencias la bandera se hace verdadera y retorna la posición del id, sino la bandera se mantiene falsa y retorna una posición falsa para regresar el menú principal y muestra un error.

Ver Ilustración 6 Búsqueda Por ID

También tenemos la búsqueda por nombre que es más complicada pues puede haber más de un jugador con el mismo nombre, ambas inician igual un ciclo for una bandera y esta vez la cadena recibida, ahora con ayuda de la función strcmp compara la cadena recibida con todas la de la lista, si encuentra un resultado muestra los datos del jugador y un sub menú que pregunta si ese es el jugador deseado si sí lo es la bandera es verdadera y se retorna la posición del jugador, sino el ciclo continúa mostrando más coincidencias, si ya no quedan coincidencias manda error y regresa al manu principal, igual si no encontró ningún resultado.

Ver Ilustración 7 Búsqueda Por Nombre

Ya con la posición del jugador deseado la función de acceder a jugador puede funcionar, en esta primero se declaran sus variables locales y se verifica que la posición es menor a la máxima (esto como apoyo para cuando las búsquedas fallaran no se entrara a este menú), después se muestra le sub menú y se pregunta la opc deseada y se guarda en la variable local opc.

Ver Ilustración 8 Acceder A Jugador Menú

Dependiendo de la variable opc se ejecutará una acción u otra con el switch:

Si es 1 preguntara el número de puntos que desea sumar y los sumara el puntaje del jugador.

Si es 2 preguntara los puntos que se quiere restar y si estos son mayores a los puntos actuales del jugador regresara al sub menú, sino los restara del puntaje del jugador.

Si es 3 se eliminará el registro del jugador y todos los registros posteriores se recorrerán a la posición anterior, también resta en uno el contador principal.

Por último si es 4 regresara al menú principal.

Ilustración 9 Acceder A Jugador Menú 2

La función Mejor Jugador muestra al jugador o jugadores que tengan el mayor puntaje de la lista, pues puede haber mas de un jugador con el mismo puntaje, la función en un ciclo for compara todos los puntajes de la lista y almacene el mayor, y en otro ciclo for diferente muestra todos los jugadores con el puntaje máximo.

Ver Ilustración 10 Mejor Puntaje

Por último la función mostrar todos los jugadores simplemente con un ciclo for imprime todos los datos de todos los jugadore en la lista.

Ver Ilustración 11 Mostrar Jugadores

ARELLANO GRANADOS ANGEL MARIANO
218123444

5. Pruebas y resultados.

Podemos ver al inicio que si muestra el menú principal y poder registrar jugadores, donde también nos muestra su id asignado con la funcion 1 de nuestro menú.

```
"C:\Users\Usuario\Documents\Trabajos 3 CUCEI\5. Estructurad De Datos
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
1
ingrese el nombre del jugador:
aaa
el jugador aaa tiene el Id: 42

1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
1
ingrese el nombre del jugador:
bbb
el jugador bbb tiene el Id: 68

1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
1
ingrese el nombre del jugador:
ccc
el jugador ccc tiene el Id: 35
```

ARELLANO GRANADOS ANGEL MARIANO
218123444

Cree jugadores con el nombre de las letras del abecedario 3 veces como aaa, bbb, ccc, ect. Hasta la letra j, es decir 10 jugadores para esta prueba, para confirmar que los datos se guardaron con éxito usamos la opción 4 de mostrar jugadores y apreciamos los 10 registros con id's diferentes y puntos en 0.

```
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
```

Seleccione una opcion:

4

```
Jugador 1      Nombre: aaa
ID: 42
Puntos: 0
```

```
-----
Jugador 2      Nombre: bbb
ID: 68
Puntos: 0
```

```
-----
Jugador 3      Nombre: ccc
ID: 35
Puntos: 0
```

```
-----
Jugador 4      Nombre: ddd
ID: 1
Puntos: 0
```

```
-----
Jugador 5      Nombre: eee
ID: 70
Puntos: 0
```

```
-----
Jugador 6      Nombre: fff
ID: 25
Puntos: 0
```

```
-----
Jugador 7      Nombre: ggg
ID: 79
Puntos: 0
```

```
-----
Jugador 8      Nombre: hhh
ID: 59
Puntos: 0
```

```
-----
Jugador 9      Nombre: iii
ID: 63
Puntos: 0
```

```
-----
Jugador 10     Nombre: jjj
ID: 65
Puntos: 0
-----
```


ARELLANO GRANADOS ANGEL MARIANO
218123444

Para ver si funciona la búsqueda por ID accedemos a la opción 2 y vemos que funciona el sub menú de tipo de búsqueda, accedemos a la búsqueda por ID y colocamos un ID falso que sabemos que no esta en la lista y vemos que si marca el error y nos regresa al menú principal, repetimos pero ahora con un Id verdadero y vemos que si accede al sub menú del jugador, mostrando que jugador estamos editando.

```
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
2
1) Buscar por ID
2) Buscar por Nombre
1
Ingresa un ID:
80
ERROR
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
2
1) Buscar por ID
2) Buscar por Nombre
1
Ingresa un ID:
63
Jugador 9 Nombre: iii
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
```

ARELLANO GRANADOS ANGEL MARIANO
218123444

Aquí vemos que funcionan las opciones de sumar y restar puntos así como el de salir pues nos regresa al menú principal.

```
Jugador 9 Nombre: iii
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
1
Puntaje actual: 0
Cuantos puntos desea sumar?
50
Jugador 9 Nombre: iii
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
2
Puntaje actual: 50
Cuantos puntos desea restar?
25
Jugador 9 Nombre: iii
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
1
Puntaje actual: 25
Cuantos puntos desea sumar?
1
Jugador 9 Nombre: iii
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
4
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:

```

ARELLANO GRANADOS ANGEL MARIANO
218123444

Ahora vemos que si es capaz de eliminar registros y volver a mostrarlos de manera correcta sin que se repita el último de ellos (en vez de jjj puse 111).

```
Ingresa un ID:
59
Jugador 8 Nombre: hhh
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
3
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
4
```

```
Jugador 1      Nombre: aaa
ID: 42
Puntos: 0
-----
Jugador 2      Nombre: bbb
ID: 68
Puntos: 0
-----
Jugador 3      Nombre: ccc
ID: 35
Puntos: 0
-----
Jugador 4      Nombre: ddd
ID: 1
Puntos: 0
-----
Jugador 5      Nombre: eee
ID: 70
Puntos: 0
-----
Jugador 6      Nombre: fff
ID: 25
Puntos: 0
-----
Jugador 7      Nombre: ggg
ID: 79
Puntos: 0
-----
Jugador 8      Nombre: iii
ID: 63
Puntos: 0
-----
Jugador 9      Nombre: 111
ID: 65
Puntos: 0
-----
```

ARELLANO GRANADOS ANGEL MARIANO
218123444

También tenemos que ver si funciona la búsqueda por nombre, por lo que a mi lista agregue 3 registros mas de jugadores que se llamaran bbb y apreciar que si muestra las 4 coincidencias y que si continuamos regresa al menu principal.

```
-----
Jugador 10      Nombre: bbb
ID: 6
Puntos: 0
-----
Jugador 11      Nombre: bbb
ID: 46
Puntos: 0
-----
Jugador 12      Nombre: bbb
ID: 82
Puntos: 0
-----
```

```
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
```

Seleccione una opcion:

2

1) Buscar por ID

2) Buscar por Nombre

2

Ingresa un Nombre:

bbb

Jugador 2 bbb

ID: 68

Puntos: 0

continuar buscando?

1) Acceder

2) Continuar

2

Jugador 10 bbb

ID: 6

Puntos: 0

continuar buscando?

1) Acceder

2) Continuar

2

Jugador 11 bbb

ID: 46

Puntos: 0

continuar buscando?

1) Acceder

2) Continuar

2

Jugador 12 bbb

ID: 82

Puntos: 0

continuar buscando?

1) Acceder

2) Continuar

2

No hay mas coincidencias!

1) Crear jugador

2) Acceder a un jugador

3) Jugador con mayor puntaje

4) Mostrar todos los jugadores

5) Salir

Seleccione una opcion:

ARELLANO GRANADOS ANGEL MARIANO

218123444

Ahora por último vamos a sumarle a 3 jugadores diferentes 100 puntos accediendo a ellos por diferentes métodos y veremos que en la opción de mejor puntuación aparecen los 3 jugadores.

```
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
```

Seleccione una opcion:

2

```
1) Buscar por ID
2) Buscar por Nombre
```

2

Ingresar un Nombre:

aaa

Jugador 1 aaa

ID: 42

Puntos: 0

continuar buscando?

```
1) Acceder
2) Continuar
```

1

Jugador 1 Nombre: aaa

```
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
```

Seleccione una opcion:

1

Puntaje actual: 0

Cuantos puntos desea sumar?

100

Jugador 1 Nombre: aaa

```
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
```

Seleccione una opcion:

4

ARELLANO GRANADOS ANGEL MARIANO
218123444

```
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
2
1) Buscar por ID
2) Buscar por Nombre
1
Ingresa un ID:
70
Jugador 5 Nombre: eee
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
1
Puntaje actual: 0
Cuantos puntos desea sumar?
100
Jugador 5 Nombre: eee
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
4
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
2
1) Buscar por ID
2) Buscar por Nombre
1
Ingresa un ID:
35
Jugador 3 Nombre: ccc
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
Seleccione una opcion:
1
Puntaje actual: 0
Cuantos puntos desea sumar?
100
Jugador 3 Nombre: ccc
1) Sumar Puntos
2) Restar Puntos
3) Eliminar
4) Salir
```

ARELLANO GRANADOS ANGEL MARIANO
218123444

```
Seleccione una opcion:
4
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
3
El Jugador / Jugadores como mejor puntaje es:
Jugador 1      aaa
ID: 42
Puntos: 100
-----
Jugador 3      ccc
ID: 35
Puntos: 100
-----
Jugador 5      eee
ID: 70
Puntos: 100
-----
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
```

Y al final con la opción 5 salimos del programa.

```
1) Crear jugador
2) Acceder a un jugador
3) Jugador con mayor puntaje
4) Mostrar todos los jugadores
5) Salir
Seleccione una opcion:
5
Gracias por usar.

Process returned 0 (0x0)   execution time : 678.866 s
Press any key to continue.
```

6. Conclusiones.

Con esta actividad podemos ver las aplicaciones reales de los datos estructurados, así como las maneras en las que podemos alterar o cambiar estas estructuras, todo implementado en un programa con bases reales que podría ser un trabajo o reto que se nos presente en nuestro futuro como programadores.

7. Apéndice(s).

Ilustración 1. Librerías.

```
1  //ARELLANO GRANADOS ANGEL MARIANO 218123444
2  #include <iostream>
3  #include <cstring>
4  #include <string.h>
5  using namespace std;
6
```

Ilustración 2. Arreglo.

```
7  struct Player{
8      int id;
9      char name[30];
10     int score;
11 }p[100];
12
```

Ilustración 3. Menú 1.

```
162 int main() {
163     char name[30];
164     int opc=0, opc2=0, id;
165     do{
166         //menu
167         cout<<"1) Crear jugador"<<endl;
168         cout<<"2) Acceder a un jugador"<<endl;
169         cout<<"3) Jugador con mayor puntaje"<<endl;
170         cout<<"4) Mostrar todos los jugadores"<<endl;
171         cout<<"5) Salir"<<endl;
172
173         cout<<"Seleccione una opcion:"<<endl;
174         cin>>opc;
175         switch(opc) {
```


Ilustración 4. Menú 2.

```
176     case 1:{
177         CrearJugador();
178     }break;
179     case 2:{
180         cout<<"1) Buscar por ID"<<endl;
181         cout<<"2) Buscar por Nombre"<<endl;
182         cin>>opc2;
183         switch(opc2){
184             case 1:{
185                 cout<<"Ingresa un ID:"<<endl;
186                 cin>>id;
187                 AccederJugador(BuscarJugadorID(id));
188             }break;
189             case 2:{
190                 cout<<"Ingresa un Nombre:"<<endl;
191                 cin>>name;
192                 AccederJugador(BuscarJugadorNombre(name));
193             }break;
194             default:{
195                 cout<<"Esa opcion no existe!"<<endl;
196             }break;
197         }
198     }break;
199     case 3:{
200         MejorPuntaje();
201     }break;
202     case 4:{
203         MostrarJugadores();
204     }break;
205     case 5:{
206         cout<<"Gracias por usar."<<endl;
207         exit(EXIT_SUCCESS);
208     }break;
209     default:{
210         cout<<"Esa opcion no existe!"<<endl;
211     }break;
212 }
213 }while(opc!=5);
214 return 0;
215 }
```

Ilustración 5. Crear Un jugador.

```
13     int cont=0;
14     void CrearJugador() {
15         int x=0;
16         bool unic;
17         if (cont<100){
18             cout<<"ingrese el nombre del jugador: "<<endl;
19             cin>>p[cont].name;
20             p[cont].score=0;
21             do{
22                 x=1+rand()%100;
23                 bool unic=true;
24                 for(int i=0;i<cont;i++){
25                     if (p[i].id==x){
26                         unic=false;
27                         break;
28                     }
29                 }
30                 unic=false;
31             }while(unic=false);
32             p[cont].id=x;
33             cout<<"el jugador "<<p[cont].name<<" tiene el Id: "<<p[cont].id<<endl<<endl;
34             cont++;
35         }
36         if (cont>=100)
37             cout<<"La lista esta llena"<<endl;
38     }
```

Ilustración 6. Búsqueda Por ID.

```
40 int BuscarJugadorID(int id) {
41     int i, pos;
42     bool aux=false;
43     for(i=0; i<cont; i++) {
44         if (id==p[i].id) {
45             pos=i;
46             aux=true;
47         }
48     }
49     if (aux==true)
50         return pos;
51     if(aux==false) {
52         cout<<"ERROR"<<endl;
53         return 101;
54     }
55 }
```

Ilustración 7. Búsqueda Por Nombre.

```
103 int BuscarJugadorNombre(char name[30]) {
104     int i, pos, opc;
105     bool aux=false;
106     for(i=0; i<cont; i++) {
107         if ((strcmp (name, p[i].name)) == 0) {
108             cout<<"Jugador " <<i+1<<"\t"<<p[i].name<<endl;
109             cout<<"ID: " <<p[i].id<<endl;
110             cout<<"Puntos: " <<p[i].score<<endl;
111             cout<<"-----" <<endl;
112             cout<<"continuar buscando?\n 1) Acceder\n 2) Continuar"<<endl;
113             cin>>opc;
114             if(opc==1) {
115                 pos=i;
116                 aux=true;
117             }
118         }
119     }
120     if(opc==2) {
121         cout<<"No hay mas coincidencias!"<<endl;
122         return 101;
123     }
124     if (aux==true)
125         return pos;
126     if(aux==false) {
127         cout<<"ERROR"<<endl;
128         return 101;
129     }
130 }
```

Ilustración 8. Acceder A jugador Menú.

```
57 void AccederJugador(int pos) {
58     int opc, sum, res, i;
59     if(pos<100) {
60         do{
61             cout<<"Jugador " <<pos+1<<" Nombre: " <<p[pos].name<<endl;
62             cout<<"1) Sumar Puntos"<<endl;
63             cout<<"2) Restar Puntos"<<endl;
64             cout<<"3) Eliminar"<<endl;
65             cout<<"4) Salir"<<endl;
66
67             cout<<"Seleccione una opcion:"<<endl;
68             cin>>opc;
69             switch(opc) {
```

Ilustración 9. Acceder A jugador Menú 2.

```
69      switch(opc){
70      case 1:{
71          cout<<"Puntaje actual: "<<p[pos].score<<endl;
72          cout<<"Cuantos puntos desea sumar?"<<endl;
73          cin>>sum;
74          p[pos].score+=sum;
75          }break;
76      case 2:{
77          cout<<"Puntaje actual: "<<p[pos].score<<endl;
78          cout<<"Cuantos puntos desea restar?"<<endl;
79          cin>>res;
80          if(res<=p[pos].score)
81              p[pos].score-=res;
82          else
83              cout<<"ERROR"<<endl;
84          }break;
85      case 3:{
86          for(i=pos;i<cont;i++){
87              p[i]=p[i+1];
88          }
89          cont--;
90          opc=4;
91          }break;
92      case 4:{
93          break;
94          }break;
95      default:{
96          cout<<"Esa opcion no existe!"<<endl;
97          }break;
98      }
99  }while(opc!=4);
```

Ilustración 10. Mejor Puntaje.

```
132 void MejorPuntaje(){
133     int i,aux=0;
134     aux=p[0].score;
135     for (i=1;i<cont;i++){
136         if(p[i].score>aux)
137             aux=p[i].score;
138     }
139     cout<<"El Jugador / Jugadores como mejor puntaje es: "<<endl;
140     for(i=0;i<cont;i++){
141         if(aux==p[i].score){
142             cout<<"Jugador "<<i+1<<"\t"<<p[i].name<<endl;
143             cout<<"ID: "<<p[i].id<<endl;
144             cout<<"Puntos: "<<p[i].score<<endl;
145             cout<<"-----"<<endl;
146         }
147     }
148 }
149
150
```

Ilustración 11. Mostrar Jugadores.

```
151 void MostrarJugadores(){
152     for(int i=0;i<cont;i++){
153         cout<<"Jugador "<<i+1<<"\tNombre: "<<p[i].name<<endl;
154         cout<<"ID: "<<p[i].id<<endl;
155         cout<<"Puntos: "<<p[i].score<<endl;
156         cout<<"-----"<<endl;
157     }
158 }
159
```