

**UNIVERSIDAD UNIAGUSTINIANA**

**DESARROLLO DE SOFTWARE**

**Parcial estructuras**

**JUAN GARZON  
BRYAM VALBUENA**

**2220221046  
2220221047**

**19/11/2023**

## INFORME

### Codigo:

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <cstdlib>
#include <ctime>
#include <algorithm>
#include <numeric> // Necesario para accumulate
#include <direct.h>

using namespace std;

struct Equipo {
    string nombre;
    vector<string> jugadores;
    vector<int> puntos; // Cambiado a un vector para almacenar los puntos de cada
partido
    int golesFavor;
    int golesContra;

    Equipo() : puntos(), golesFavor(0), golesContra(0) {}
};

struct Partido {
    Equipo* equipoLocal;
    Equipo* equipoVisitante;
    int golesLocal;
    int golesVisitante;

    Partido(Equipo* local, Equipo* visitante) : equipoLocal(local),
equipoVisitante(visitante), golesLocal(0), golesVisitante(0) {}
};
```

```

vector<Equipo> leerEquiposDesdeArchivo(const char* nombreArchivo) {
    vector<Equipo> equipos;
    ifstream archivo(nombreArchivo);

    if (archivo.is_open()) {
        string linea;

        while (getline(archivo, linea)) {
            Equipo equipo;
            size_t pos = 0;
            string token;

            while ((pos = linea.find(',')) != string::npos) {
                token = linea.substr(0, pos);
                equipo.jugadores.push_back(token);
                linea.erase(0, pos + 1);
            }

            equipo.nombre = linea;
            equipos.push_back(equipo);
        }

        archivo.close();
    } else {
        cout << "No se pudo abrir el archivo." << endl;
    }

    return equipos;
}

void mostrarGrupos(const vector<Equipo>& equipos) {
    size_t totalEquipos = equipos.size();

    if (totalEquipos % 4 != 0) {
        cout << "No hay suficientes equipos para formar grupos de 4." << endl;
    }
}

```

```

        return;
    }

    size_t totalGrupos = totalEquipos / 4;

    // Crear grupos
    for (size_t i = 0; i < totalGrupos; ++i) {
        cout << "Grupo " << static_cast<char>('A' + i) << ":" << endl;

        for (size_t j = i * 4; j < (i + 1) * 4; ++j) {
            cout << equipos[j].nombre << endl;
        }

        cout << endl;
    }
}

void mostrarTablaPorGrupo(const vector<Equipo>& equipos, char grupo) {
    cout << "Tabla de Posiciones Grupo " << grupo << ":" << endl;
    cout << "Equipo\tPuntos\tGoles a favor\tGoles en contra" << endl;

    // Crear un vector de índices ordenados por puntos
    vector<size_t> indices(equipos.size());
    iota(indices.begin(), indices.end(), 0);
    sort(indices.begin(), indices.end(), [&equipos](size_t i, size_t j) {
        // Comparar por puntos y luego por diferencia de goles
        int puntosI = accumulate(equipos[i].puntos.begin(),
            equipos[i].puntos.end(), 0);
        int puntosJ = accumulate(equipos[j].puntos.begin(),
            equipos[j].puntos.end(), 0);

        if (puntosI != puntosJ) {
            return puntosI > puntosJ; // Ordenar por puntos de mayor a menor
        } else {
            // Si hay empate en puntos, ordenar por diferencia de goles
            int difGolesI = equipos[i].golesFavor - equipos[i].golesContra;
            int difGolesJ = equipos[j].golesFavor - equipos[j].golesContra;

```

```

        return difGolesI > difGolesJ; // Ordenar por diferencia de goles de
mayor a menor
    }
});

```

```

    for (size_t i : indices) {
        // Mostrar solo equipos del grupo correspondiente
        if (equipos[i].nombre[6] == grupo) {
            int totalPuntos = accumulate(equipos[i].puntos.begin(),
equipos[i].puntos.end(), 0);
            cout << equipos[i].nombre << "\t" << totalPuntos << "\t" <<
equipos[i].golesFavor << "\t\t" << equipos[i].golesContra << endl;
        }
    }
}

```

```

    cout << "-----" << endl;
}

```

```

void jugarPartidosEnGrupo(vector<Equipo>& equipos, size_t inicioGrupo) {
    size_t finGrupo = inicioGrupo + 4;

```

```

    // Crear partidos para cada equipo dentro del grupo
    for (size_t i = inicioGrupo; i < finGrupo; ++i) {
        for (size_t j = i + 1; j < finGrupo; ++j) {
            Partido partido(&equipos[i], &equipos[j]);
            // Simular partidos
            partido.golesLocal = rand() % 5;
            partido.golesVisitante = rand() % 5;

```

```

        // Actualizar estadísticas de equipos basadas en el resultado del
partido
        partido.equipoLocal->golesFavor += partido.golesLocal;
        partido.equipoLocal->golesContra += partido.golesVisitante;
        partido.equipoVisitante->golesFavor += partido.golesVisitante;
        partido.equipoVisitante->golesContra += partido.golesLocal;

```

```

        if (partido.golesLocal > partido.golesVisitante) {
            // Ganador: equipo local
            partido.equipoLocal->puntos.push_back(3);

```

```

    } else if (partido.golesLocal < partido.golesVisitante) {
        // Ganador: equipo visitante
        partido.equipoVisitante->puntos.push_back(3);
    } else {
        // Empate
        partido.equipoLocal->puntos.push_back(1);
        partido.equipoVisitante->puntos.push_back(1);
    }
}

cout << partido.equipoLocal->nombre << " vs " <<
partido.equipoVisitante->nombre << " - "
    << partido.golesLocal << " - " << partido.golesVisitante <<
endl;
}
}
}

void simularTorneo(vector<Equipo>& equipos) {
    size_t totalEquipos = equipos.size();

    if (totalEquipos % 4 != 0) {
        cout << "No hay suficientes equipos para formar grupos de 4." << endl;
        return;
    }

    // Mostrar los grupos antes de simular los partidos
    mostrarGrupos(equipos);

    // Simular partidos en cada grupo
    for (size_t grupo = 0; grupo < totalEquipos / 4; ++grupo) {
        char nombreGrupo = static_cast<char>('A' + grupo);
        cout << "Simulando partidos en Grupo " << nombreGrupo << ":" << endl;
        jugarPartidosEnGrupo(equipos, grupo * 4);
        // Calcular y mostrar la tabla después de todos los partidos en cada
        grupo
        mostrarTablaPorGrupo(equipos, nombreGrupo);
        cout << endl;
    }
}

```

```
void jugarRonda(vector<Equipo>& equiposRonda) {  
    size_t totalEquiposRonda = equiposRonda.size();
```

```
    if (totalEquiposRonda % 2 != 0) {  
        cout << "Número impar de equipos en la ronda. No se pueden simular los  
partidos." << endl;  
        return;  
    }  
}
```

```
// Vector para almacenar los equipos que avanzan a la siguiente ronda  
vector<Equipo> equiposSiguieteRonda;
```

```
// Crear partidos para la ronda  
for (size_t i = 0; i < totalEquiposRonda; i += 2) {  
    Partido partido(&equiposRonda[i], &equiposRonda[i + 1]);  
    // Simular partidos  
    partido.golesLocal = rand() % 5;  
    partido.golesVisitante = rand() % 5;
```

```
    // Imprimir información de los partidos de la ronda  
    cout << partido.equipoLocal->nombre << " vs " << partido.equipoVisitante->nombre << " - "  
        << partido.golesLocal << " - " << partido.golesVisitante << endl;
```

```
    // Determinar el equipo ganador y agregarlo a equiposSiguieteRonda  
    if (partido.golesLocal > partido.golesVisitante) {  
        equiposSiguieteRonda.push_back(*partido.equipoLocal);  
    } else {  
        equiposSiguieteRonda.push_back(*partido.equipoVisitante);  
    }  
}
```

```
cout << endl;
```

```
// Actualizar los equipos para la siguiente ronda  
equiposRonda = equiposSiguieteRonda;
```

```

}

int main(int argc, char* argv[]) {
    if (argc != 2) {
        cout << "Uso: " << argv[0] << " <nombre_del_archivo>" << endl;
        return 1;
    }

    string nombreArchivo = argv[1];

    size_t lastSlash = nombreArchivo.find_last_of("\\\\/");
    if (lastSlash != string::npos) {
        string directorio = nombreArchivo.substr(0, lastSlash);
        _chdir(directorio.c_str());
    }

    vector<Equipo> equipos = leerEquiposDesdeArchivo(nombreArchivo.c_str());

    // Inicializar la semilla para generar números aleatorios
    srand(static_cast<unsigned int>(time(0)));

    // Simular el torneo
    simularTorneo(equipos);

    cout << "-----
\n";

    // Simular los octavos de final
    jugarRonda(equipos);
    cout << "-----
\n";

    // Simular los cuartos de final
    jugarRonda(equipos);
    cout << "-----
\n";

```



```

// Simular las semifinales
jugarRonda(equipos);
cout << "-----
\n";

// Simular la final
jugarRonda(equipos);
cout << "-----
\n";

// El ganador del torneo estará en equipos[0]
cout << "Ganador del torneo: " << equipos[0].nombre << endl;

cout << "Presiona Enter para salir...";
getchar();

return 0;
}

```

**Grupos A-H:**

Grupo A:  
Brasil  
Alemania  
Argentina  
Francia

Grupo B:  
Italia  
Venezuela  
Portugal  
Países Bajos

Grupo C:  
Inglaterra  
Uruguay  
Belgica  
Colombia

Grupo D:  
Chile  
Mexico  
Croacia  
Suiza

Grupo E:  
Suecia  
Polonia  
Austria  
Dinamarca

Grupo F:  
Rusia  
Turquia  
Estados Unidos  
Japon

Grupo G:  
Australia  
Marruecos  
Nigeria  
Egipto

Grupo H:  
Ghana  
Costa de Marfil  
Camerun  
Senegal

**Simulacion de partidos:**

Simulando partidos en Grupo A:

Brasil vs Alemania - 1 - 2

Brasil vs Argentina - 4 - 1

Brasil vs Francia - 1 - 3

Alemania vs Argentina - 1 - 4

Alemania vs Francia - 3 - 1

Argentina vs Francia - 1 - 4

Tabla de Posiciones Grupo A:

Equipo	Puntos	Goles a favor	Goles en contra
--------	--------	---------------	-----------------

Simulando partidos en Grupo B:

Italia vs Venezuela - 3 - 3

Italia vs Portugal - 2 - 4

Italia vs Países Bajos - 2 - 1

Venezuela vs Portugal - 2 - 4

Venezuela vs Países Bajos - 3 - 4

Portugal vs Países Bajos - 2 - 0

Tabla de Posiciones Grupo B:

Equipo	Puntos	Goles a favor	Goles en contra
--------	--------	---------------	-----------------

Simulando partidos en Grupo C:

Inglaterra vs Uruguay - 1 - 2

Inglaterra vs Bélgica - 1 - 3

Inglaterra vs Colombia - 2 - 3

Uruguay vs Bélgica - 4 - 4

Uruguay vs Colombia - 0 - 0

Bélgica vs Colombia - 0 - 2

Tabla de Posiciones Grupo C:

Equipo	Puntos	Goles a favor	Goles en contra
--------	--------	---------------	-----------------

Simulando partidos en Grupo D:

Chile vs Mexico - 0 - 1

Chile vs Croacia - 0 - 4

Chile vs Suiza - 3 - 4

Mexico vs Croacia - 2 - 1

Mexico vs Suiza - 0 - 2

Croacia vs Suiza - 2 - 3

Tabla de Posiciones Grupo D:

Equipo	Puntos	Goles a favor	Goles en contra
--------	--------	---------------	-----------------

Simulando partidos en Grupo E:

Suecia vs Polonia - 3 - 1

Suecia vs Austria - 1 - 3

Suecia vs Dinamarca - 2 - 4

Polonia vs Austria - 4 - 4

Polonia vs Dinamarca - 0 - 2

Austria vs Dinamarca - 3 - 3

Tabla de Posiciones Grupo E:

Equipo	Puntos	Goles a favor	Goles en contra
--------	--------	---------------	-----------------

Simulando partidos en Grupo F:

Rusia vs Turquía - 4 - 1

Rusia vs Estados Unidos - 1 - 0

Rusia vs Japon - 3 - 3

Turquía vs Estados Unidos - 3 - 1

Turquía vs Japon - 1 - 3

Estados Unidos vs Japon - 4 - 1

Tabla de Posiciones Grupo F:

Equipo	Puntos	Goles a favor	Goles en contra
--------	--------	---------------	-----------------

```

Simulando partidos en Grupo G:
Australia vs Marruecos - 4 - 1
Australia vs Nigeria - 4 - 3
Australia vs Egipto - 3 - 3
Marruecos vs Nigeria - 3 - 3
Marruecos vs Egipto - 3 - 3
Nigeria vs Egipto - 3 - 1
Tabla de Posiciones Grupo G:
Equipo  Puntos  Goles a favor  Goles en contra
-----

```

```

Simulando partidos en Grupo H:
Ghana vs Costa de Marfil - 1 - 3
Ghana vs Camerun - 1 - 4
Ghana vs Senegal - 3 - 2
Costa de Marfil vs Camerun - 1 - 3
Costa de Marfil vs Senegal - 4 - 1
Camerun vs Senegal - 1 - 4
Tabla de Posiciones Grupo H:
Equipo  Puntos  Goles a favor  Goles en contra
-----

```

#### Fase de grupos:

```

-----
Brasil vs Alemania - 1 - 3
Argentina vs Francia - 1 - 3
Italia vs Venezuela - 3 - 3
Portugal vs Países Bajos - 0 - 0
Inglaterra vs Uruguay - 1 - 2
Bélgica vs Colombia - 4 - 4
Chile vs México - 0 - 1
Croacia vs Suiza - 0 - 2
Suecia vs Polonia - 2 - 2
Austria vs Dinamarca - 0 - 1
Rusia vs Turquía - 0 - 3
Estados Unidos vs Japón - 2 - 2
Australia vs Marruecos - 2 - 1
Nigeria vs Egipto - 4 - 3
Ghana vs Costa de Marfil - 0 - 1
Camerun vs Senegal - 1 - 2

```

#### Octavos de final:

```
Alemania vs Francia - 0 - 0
Venezuela vs Países Bajos - 4 - 0
Uruguay vs Colombia - 3 - 3
México vs Suiza - 2 - 4
Polonia vs Dinamarca - 0 - 4
Turquía vs Japón - 4 - 3
Australia vs Nigeria - 4 - 1
Costa de Marfil vs Senegal - 1 - 4
```

#### Cuartos de final:

```
Francia vs Venezuela - 1 - 3
Colombia vs Suiza - 0 - 3
Dinamarca vs Turquía - 2 - 4
Australia vs Senegal - 2 - 2
```

#### Semi final:

```
Venezuela vs Suiza - 1 - 1
Turquía vs Senegal - 0 - 1
```

#### Final:

```
Ganador del torneo: Suiza
Presiona Enter para salir...
```

