

Federação Portuguesa de Futebol

AEDA 19/20

José David Rocha, up201806371
Telmo Botelho, up201806821
Tomás Mendes, up201806522

Introdução

- O trabalho que realizámos tem como objetivo apresentar uma base de dados relacionada com a federação portuguesa de futebol.

Solução

- Primeiramente, dividimos o trabalho nas suas diferentes classes, de forma a facilitar a implementação do código.
- Deste modo, definimos as seguintes classes: Seleção, Membro da Seleção, tendo duas classes derivadas (Futebolista e Equipa Técnica); Convocatória e Jogo.

Solução

- Ao longo do trabalho, decidimos criar o ficheiro Utils com várias funções necessárias para a implementação dos membros-função de diversas classes.
- Seguidamente, criamos os Menus e um ficheiro cpp com as estatísticas da seleção portuguesa.

Solução

- Quanto à segunda parte do trabalho, implementamos as novas estruturas de dados pedidas - a fila de prioridade, a árvore binária e a tabela de dispersão.

Solução

- Foram criados dois novos ficheiros – um *cpp* e um *header* - ambos referentes aos equipamentos (fila de prioridade).
- Além disso, foram adicionados dois ficheiros de texto: um para os fornecedores (fila de prioridade) e outro para os selecionadores (*BST*).
- Finalmente, tratamos da parte do registo de funcionários (tabela de dispersão).

Algoritmo de Pesquisa

- *Binary Search*

```
int BinarySearch(const vector<EquipaTecnica> &v, EquipaTecnica x) {  
    int left = 0;  
    int right = v.size() - 1;  
    while (left <= right) {  
        int mid = (left + right) / 2;  
        if (v.at(mid) < x) left = mid + 1;  
        else if (x < v.at(mid)) right = mid - 1;  
        else return mid;  
    }  
    return -1;  
}
```



Pesquisa funcionários
da seleção

Algoritmo de Ordenação

- *Sort*

```
bool maisgolos(JogadorFama f1, JogadorFama f2){  
    return f1.golos > f2.golos;  
}  
  
sort(v1.begin(), v1.end(), maisgolos);
```

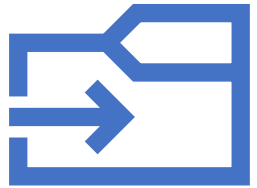
Implementado para obtermos
o jogador com mais golos
marcados pela seleção

Novos Operadores

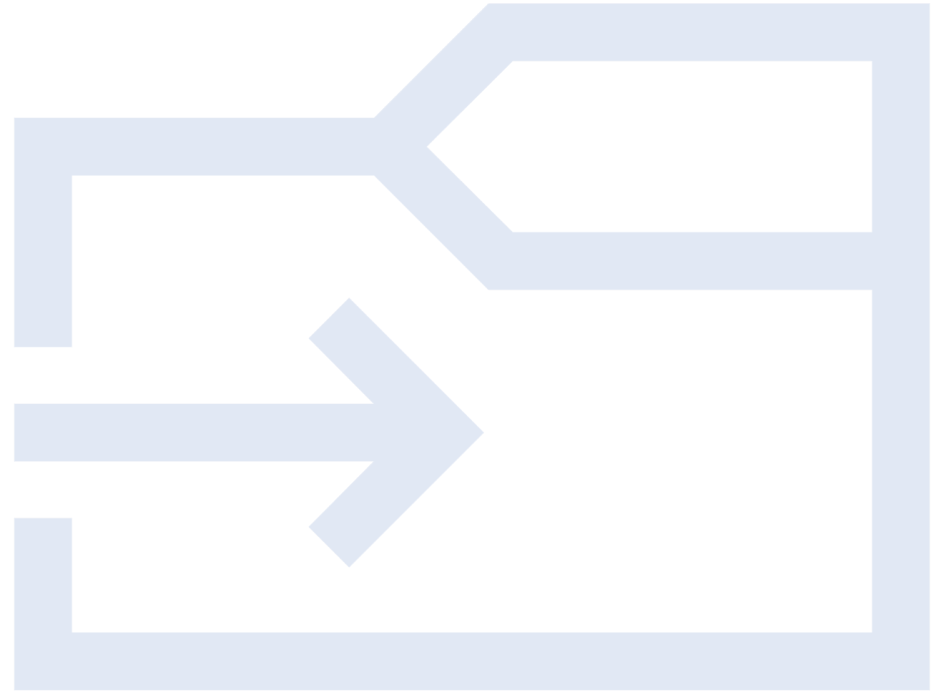
- Exemplos:

```
bool operator< (const Fornecedores& f) const;|
bool operator== (const Fornecedores& f) const;
friend ostream& operator<<(ostream& out, const Fornecedores& f);
```


```
bool operator < (const Seleccionadores &s) const;
friend ostream& operator<<(ostream& out, const Seleccionadores& s);
```



Estrutura dos Ficheiros



Estrutura dos Ficheiros

 Jogo.txt - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

28/11/2019 | Inglaterra | Inglaterra | Londres | Wembley Stadium

Cristiano Ronaldo dos Santos Aveiro | Bernardo Mota Veiga de Carvalho e Silva | Bruno Miguel Borges Fernandes | Nelson Cabral Semedo

05/02/1985 | 15/10/2000 | 08/09/1994 | 26/02/1983

Nelson Cabral Semedo | Cristiano Ronaldo dos Santos Aveiro | Bruno Miguel Borges Fernandes | Bernardo Mota Veiga de Carvalho e Silva

26/02/1983 | 05/02/1985 | 08/09/1994 | 15/10/2000

Tomas | David | Jose | Carlos

Pickford | Smalling | Vardy | Sterling |

0 | 3 | 1 | 0

0 | 1 | 1 | 2

34 | 100 | 56 | 90

23 | 95 | 45 | 90

5 | 0 | 0 | 5

4 | 0 | 0 | 2

4 | 0 | 0 | 2

90 | 90 | 90 | 90


0 | 0 | 0 | 0

1 | 0 | 0 | 0

Cristiano Ronaldo dos Santos Aveiro


05/02/1985

Estrutura dos Ficheiros

 fornecedores.txt - Bloco de notas


Ficheiro Editar Formatar Ver Ajuda

```
Adidas
4.83333
21
bola | Calcoes
10 | 10
:~::~:
Nike
3.7
100
Calcoes
15
```

 seleccionadores.txt - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

```
David Rocha | 09/05/2000 | 1000 | 17
1
Tomas Mendes | 09/05/2000 | 1000 | 17
1 | 2
Ronaldo | 09/05/2000 | 1000 | 17
1 | 2
```

 equipatecnica.txt - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

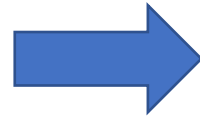
```
Fernando Santos | 10/10/1954 | Seleccionador Nacional | 1000
Ilidio Vale | 13/12/1957 | Treinador Nacional | 600
Joao Vieira Pinto | 19/08/1971 | Diretor | 2000
Emilio Peixe | 16/01/1973 | Treinador Adjunto | 1000
Fernanda Meireles | 06/07/1990 | Departamento Medico | 500
Bernardo Moreira | 09/05/1967 | Medico Principal | 1500
```

Tratamento das Exceções

Tratamento das Exceções

- Exceções:

- JogadorJaExistente



```
class JogadorJaExistente {  
public:  
    string nome, datanascimento;  
    JogadorJaExistente(string n, string dn) {  
        nome = n;  
        datanascimento = dn;  
    }  
};
```

- MembroTecnicoJaExistente



```
class MembroTecnicoJaExistente {  
public:  
    string nome, datanascimento;  
    MembroTecnicoJaExistente(string n, string dn) {  
        nome = n;  
        datanascimento = dn;  
    }  
};
```

Tratamento das Exceções

- Exceções:

- JogoJaExistente



```
class JogoJaExistente {  
public:  
    string data;  
    JogoJaExistente(string d) {  
        data = d;  
    }  
};
```

- ConvocatoriaJaExistente



```
class ConvocatoriaJaExistente {  
public:  
    unsigned int id;  
    ConvocatoriaJaExistente(unsigned int i) {  
        id = i;  
    }  
};
```

Tratamento das Exceções

- Exceções:

- SemFornecedor



```
class SemFornecedor {  
public:  
    string tipo;  
    SemFornecedor(string t) {  
        tipo = t;  
    }  
};
```

- FuncionarioNaoExistente



```
class FuncionarioNaoExistente {  
public:  
    string f;  
    FuncionarioNaoExistente(string f) {  
        this->f = f;  
    }  
};
```




Funcionalidades
Implementadas



CRUD

```
void JogadoresToFile(vector<Futebolista> vf);  
void EquipaTecnicoToFile(vector<EquipaTecnica> et);  
void ConvocatorioToFile(vector<Convocatoria> conv);  
void JogostoToFile(vector<Jogo> jogos);
```



Seleção (OK)

```
vector<Futebolista> ReadJogadores();  
void AdicionarJogador(vector<Futebolista>& jogadores) const;  
void RemoverJogador(vector<Futebolista>& jogadores) const;  
void InformacoesJogador(vector<Futebolista>& jogadores) const;  
void EditarJogador(vector<Futebolista>& jogadores) const;
```



Jogadores (OK)

```
vector<EquipaTecnica> ReadEquipaTecnica();  
void AdicionarTecnico(vector<EquipaTecnica>& equipatecnica) const;  
void RemoverTecnico(vector<EquipaTecnica>& equipatecnica) const;  
void InformacoesTecnico(vector<EquipaTecnica>& equipatecnica) const;  
void EditarTecnico(vector<EquipaTecnica>& equipatecnica) const;
```



Equipa Técnica (OK)

```
vector<Jogo> ReadJogo();  
void addJogo(vector<Jogo>& jogos, vector<Futebolista>& jogadores);  
Jogo removeJogo(vector<Jogo>& jogos);  
void infoJogo(vector<Jogo>& jogos);  
void editJogo(vector<Jogo>& jogos, vector<Futebolista>& jogadores);
```



Jogo (OK)

CRUD

```
//Outros metodos
vector<Convocatoria> ReadConvocatoria(vector<Jogo> jogo);
void InformacoesConvocatoria(vector<Convocatoria>& convocatoria) const;
void AdicionarConvocatoria(vector<Convocatoria>& convocatoria, vector<Futebolista> jogadores, vector<EquipaTecnica> equipatecnica, vector<Jogo> jogo) const;
void RemoverConvocatoria(vector<Convocatoria>& convocatoria) const;
void EditarConvocatoria(vector<Convocatoria>& convocatoria, vector<Jogo>& jogo) const;
void updateJogos(vector<Convocatoria>& convocatoria, Jogo& j);
```



Convocatória (OK)

Referente à segunda parte

- Alguns exemplos:

```
priority_queue<Fornecedores> ReadFornecedores();  
void InfoFornecedores(priority_queue<Fornecedores> fornecedores);  
void VenderEquipamentos(priority_queue<Fornecedores> &fornecedores);
```



Fila de prioridade (OK)

```
void FornecedorestToFile(priority_queue<Fornecedores> fornecedores);  
void SelecionadorestToFile(BST<Selecionadores> selecionadores);  
void updateSelecionadores(BST<Selecionadores> &selecionadores, Convocatoria c);  
HTFuncionariosRecord generateRecords(vector<EquipaTecnica> et);  
void InformacoesFuncionariosConvocatoria(vector<Convocatoria> c, HTFuncionariosRecord ht);  
void InformacoesFuncionariosConhecidos(HTFuncionariosRecord ht);  
void ContratarFuncionario(vector<Convocatoria> &c, HTFuncionariosRecord ht, vector<EquipaTecnica> &et);
```



Fila de prioridade (OK)
BST (OK)
Tabela de Dispersão (OK)

Referente à segunda parte

- Alguns exemplos:

```
struct FuncionariosRecordHash
{
    int operator() (const EquipaTecnica& cr) const
    {
        return 0 ;
    }

    bool operator() (const EquipaTecnica& cr1, const EquipaTecnica& cr2) const
    {
        return (cr1.getNome() == cr2.getNome() && cr1.getDataNascimento() == cr2.getDataNascimento());
    }
};

typedef unordered_set<EquipaTecnica, FuncionariosRecordHash , FuncionariosRecordHash > HTFuncionariosRecord;
```

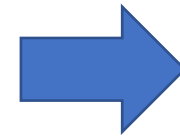


Tabela de Dispersão
(OK)

Referente à segunda parte

- Listagem:

```
void Selecao::InformacoesFuncionariosConhecidos(HTFuncionariosRecord ht) {  
    int x = 0, size = ht.size();  
    for (auto it = ht.begin(); it != ht.end(); it++){  
        cout << *it;  
        if (x < size - 1)  
            cout << endl << endl << "-----" << endl << endl;  
        x++;  
    }  
}
```



Listagem de todos os
funcionários que trabalharam
(ou trabalham) para a
seleção

Destaque

9 - Corredor da Fama



Nova opção colocada no menu

Destaque

CORREDOR DA FAMA

Melhor Marcador de Sempre: Cristiano Ronaldo dos Santos Aveiro (10 golos)

Jogador com mais Assistencias de Sempre: Bernardo Mota Veiga de Carvalho e Silva (6 assistencias)

Mais Internacional de Sempre: Bernardo Mota Veiga de Carvalho e Silva (3 internacionalizacoes)



Hall of fame com recordes e
os seus detentores

Dificuldades

- Na primeira fase do trabalho, deparamo-nos com duas principais dificuldades: as funções de *read* e as funções de *edit*, uma vez que havia ficheiros de texto que eram bastante extensos e, além disso, havia vários parâmetros editáveis.

Dificuldades

- Quanto à segunda parte do trabalho, o maior obstáculo esteve, claramente, na implementação da *BST*, visto que se tratava da estrutura não linear mais complexa.

Esforço e Contributo

- O grupo fez um bom trabalho coletivamente. Contudo, o David foi, sem dúvida, o elemento que se destacou mais pelo seu empenho durante as duas fases do projeto. Deste modo, concordamos que deve ser valorizado e ter uma nota final mais alta.