

**Resumo.** Este trabalho tem como intenção desenvolver um sistema que faça manipulação de arquivos binário. O software em função, consiste em criar um arquivo de times de futebol, com opções de você poder criar, alterar, excluir, ler e imprimir(CRUDS). Este sistema faz operações de CRUD Create (criar), Read (ler), Update (atualizar) e Delete (excluir) – as iniciais em inglês formam CRUD – são as quatro funções básicas que geralmente os sistemas que manipulam banco de dados e arquivos devem poder executar. Como a maioria dos sistemas têm a necessidade de realizar operações de CRUD, para agilizar o desenvolvimento dos sistemas, os desenvolvedores podem fazer o uso de um Framework de desenvolvimento.

## 1. Introdução

O trabalho foi desenvolvido em **Java** uma linguagem de programação orientada a objetos, foi feito um sistema que gerencia times de futebol onde são armazenados em um arquivo binário, que foi o tipo de arquivo utilizado. Foi realizado a criação dos CRUDs completo interagido com os arquivos de dados e implementação da solução.

## 2. Desenvolvimento



**Figure 1. Estrutura de arquivos.**

Foram utilizado 3 arquivos no nosso projeto, onde a seguir explicamos cada um.

### 2.1. Club-fut.db

Esse é o arquivo em binário criado onde foi feito a leitura e escrita através do nosso programa em java, e para auxiliar a leitura do arquivo em binário utilizamos o hex editor, uma extensão do vscode que permite ler os arquivos binários em hexadecimal.

### 2.2. Clube-futebol.java

Esse foi o arquivo criado a partir do código em java onde foi iniciado a classe e a estrutura do clube de futebol, suas variáveis/atributos e onde também foi criado e utilizado os métodos de construtor e métodos para passar os atributos do objeto para o array de bytes do main, de array de bytes para as variáveis de atributos do objeto.

### 2.3. Cruds-main.java

Esse arquivo é onde foi criado o corpo do projeto, que por meio dela ocorre a interação do usuário com o sistema e o sistema com o arquivo. Diante disso é criado ids, nomes de times, cidade, cnpj, diversas informações a serem armazenadas nos arquivos.db (Captura informações do clube inserido pelo usuário).

### 3. Informações inserida pelo usuario

Apresenta uma forma de utilizar as classes `RandomAccessFile` e `DataOutputStream` na criação e gravação de dados baseada em bytes para um arquivo binário. Os métodos `writeUTF()`, `writeChar()`, `writeInt()` e `writeDouble()` da classe `DataOutputStream` são aplicados na gravação de bytes que representam valores do tipo literal, caractere, inteiro e real, respectivamente

```
04
    writer.writeUTF(nome);
    System.out.println("Digite o nome do time:");
    String nome = sc.nextLine();
    System.out.println("Digite o CPF do time:");
    String CPF = sc.nextLine();
    System.out.println("Digite a cidade do time:");
    String cidade = sc.nextLine();

    Clube_futebol time = new Clube_futebol(id, nome, CPF, cidade);
    time.write(obj(time));

    System.out.println("\n-Adicionar outro time [y/n]->");
    System.out.println("Assinale: ");
    char op = sc.next().charAt(0);
    while(op != 'y' && op != 'n')
        op = sc.next().charAt(0);

    RandomAccessFile arq;
    try {
        //Abre o tipo byte declarado
        id = -1;
        arq = new RandomAccessFile("Trabalho_AED_Manipula_arq/Dados/Club_fut.db", "rw"); //Criação/abertura do arquivo .db, para leitura e escrita
        arq.seek(0);
        arq.writeInt(id); //Para o ponteiro do arquivo para a posição indicada
        //Para cada clube que o usuário quiser criar, escreva no RandomAccessFile
        for (int repet = 0; repet < time.write.size(); repet++) {
            arq.writeUTF(nome);
            b = time.write.get(repet).getBytes();
            arq.write(b.length); //Transforma e escreve os atributos do time.get(posição) no array de tipo byte b
            arq.write(b);
        }
        arq.close();
    } catch (IOException e) {
        System.out.println("ERRO!");
    }

    return "....Criado com sucesso....";
}
```

**Figure 2.** Essa figura esta um pouco do código onde foi desenvolvido o Create (Criação do arquivo)

Utilizamos métodos para criar o time no arquivo.db, listamos para armazenar o que vai ser escrito no arquivo, o sistema irá capturar informações pelo usuário, logo em seguida cada clube vai ser criado a sua identificação (Id clube). Ele cria um time de decisão do usuário, ou seja, o usuário vai poder criar um time dentro do arquivo.

### 4. Leitura do arquivo

Read realiza a leitura do arquivo de acordo com o qual time o usuário deseja ler do arquivo.

A leitura e escrita de arquivos por um programa pode ser muito útil, seja para servir como arquivo de configurações e evitar a necessidade de modificação frequente do código fonte do próprio programa para cada cenário ou simplesmente para servir para entrada e saída de dados.

```
public static Clube_futebol read(int id) throws IOException {
    RandomAccessFile arq = new RandomAccessFile("Trabalho_AED_Manipula_arq/Dados/Club_fut.db", "rw");
    arq.seek(4);
    //pular cabeçalho

    byte lapide;
    byte[] b;
    int tam;
    Clube_futebol objeto;
    while (arq.getFilePointer() < arq.length()) {
        lapide = arq.readByte();
        tam = arq.readInt();
        b = new byte[tam];
        arq.read(b);
        if (lapide != '*') {
            objeto = new Clube_futebol();
            objeto.fromByteArray(b);
            // System.out.println(objeto.getId());
            if (objeto.getId() == id) {
                arq.close();
                return objeto;
            }
        }
    }
    arq.close();
    return null;
}
```

**Figure 3.** Essa figura esta um pouco do código onde foi desenvolvido o Read (leitura do arquivo).

## 5. Atualiza o arquivo

É um método criado para realizar a atualização dos times existente no arquivo de acordo com a escolha de qual time o usuário deseja atualizar, ou seja atualiza os times do clube.

O comando UPDATE tem por finalidade a atualização dos dados que você inseriu na tabela. Se por algum motivo você quer modificar um campo, atualizar o valor ou acrescentar novos dados, é o UPDATE que você vai utilizar.

```
public static boolean Times_Update(Clube_futebol novoObjeto) throws IOException {
    RandomAccessFile arq = new RandomAccessFile("Trabalho_AED_Manipula_arq/Dados/Club_fut.db", "rw");
    arq.seek(4);
    // pos = 4;
    long pos;
    byte lapide;
    byte[] b;
    byte[] novoB;
    int tam;
    Clube_futebol objeto;
    while (arq.getFilePointer() < arq.length()) {
        pos = arq.getFilePointer();
        lapide = arq.readByte();
        tam = arq.readInt();
        b = new byte[tam];
        arq.read(b);
        if (lapide != '*') {
            objeto = new Clube_futebol();
            objeto.fromByteArray(b);
            if (objeto.getId() == novoObjeto.getId()) {
                novoB = novoObjeto.toByteArray();
                if (novoB.length < tam) {
                    arq.seek(pos + 5);
                    arq.write(novoB);
                } else {
                    arq.seek(pos);
                    arq.writeByte('*');
                    arq.seek(arq.length());
                    arq.writeByte(' ');
                    arq.writeInt(novoB.length);
                    arq.write(novoB);
                }
                arq.close();
                return true;
            }
        }
    }
    arq.close();
    return false;
}
```

Figure 4. Essa figura esta um pouco do código onde foi desenvolvido o update (Atualiza o arquivo).

## 6. Deleta o time que esta no arquivo

É um método criado para realizar a exclusão dos times existente no arquivo.

Por fim temos o comando DELETE que simplesmente exclui o dado especificado. Muito cuidado, pois, uma vez excluído não tem Ctrl + z para voltar no tempo e desfazer a cagadinha.

```
public static boolean Times_Delete(int id) throws IOException {
    RandomAccessFile arq = new RandomAccessFile("Trabalho_AED_Manipula_arq/Dados/Club_fut.db", "rw");
    arq.seek(4);
    // pos = 4;
    long pos;
    byte lapide;
    byte[] b;
    Clube_futebol objeto;
    while (arq.getFilePointer() < arq.length()) {
        pos = arq.getFilePointer();
        lapide = arq.readByte();
        tam = arq.readInt();
        b = new byte[tam];
        arq.read(b);
        if (lapide != '*') {
            objeto = new Clube_futebol();
            objeto.fromByteArray(b);
            if (objeto.getId() == id) {
                arq.seek(pos);
                arq.writeByte('*');
                arq.close();
                return true;
            }
        }
    }
    arq.close();
    return false;
}
```

Figure 5. Essa figura esta um pouco do código onde foi desenvolvido o delete (Deleta o arquivo).

## 7. Testes e Resultados:

### Arquivo criado apartir do programa

```
00000000 00 00 00 07 20 00 00 00 2E 00 00 00 01 00 08 43 . . . . . C
00000010 72 75 7A 65 69 72 6F 00 06 33 38 34 37 33 38 00 r u z e i r o . . 3 8 4 7 3 8 .
00000020 0E 42 65 6C 6F 20 48 6F 72 69 7A 6F 6E 74 65 00 . B e l o   H o r i z o n t e .
00000030 00 00 00 00 00 00 00 20 00 00 00 2E 00 00 00 02 . . . . .
00000040 00 08 46 6C 61 6D 65 6E 67 6F 00 06 33 38 34 33 . . F l a m e n g o . . 3 8 4 3
00000050 37 34 00 0E 52 69 6F 20 64 65 20 4A 61 6E 65 69 7 4 . . R i o   d e   J a n e i
00000060 72 6F 00 00 00 00 00 00 00 2A 00 00 00 33 00 r o . . . . . * . . . 3 .
00000070 00 00 03 00 0B 41 74 6C 65 74 69 63 6F 20 4D 47 . . . . . A t l e t i c o   M G
00000080 00 08 33 32 39 34 38 39 33 34 00 0E 42 65 6C 6F . . 3 2 9 4 8 9 3 4 . . B e l o
00000090 20 48 6F 72 69 7A 6F 6E 74 65 00 00 00 00 00 H o r i z o n t e . . . . .
000000A0 00 00 20 00 00 00 2C 00 00 00 04 00 0A 53 C3 86 . . . . . , . . . . . S Ã .
000000B0 6F 20 50 61 75 6C 6F 00 06 33 39 34 38 39 33 00 o   P a u l o . . 3 9 4 8 9 3 .
000000C0 0A 53 C3 86 6F 20 50 61 75 6C 6F 00 00 00 00 . S Ã . o   P a u l o . . . . .
000000D0 00 00 00 20 00 00 00 21 00 00 00 05 00 05 42 61 . . . . . ! . . . . . B a
000000E0 68 69 61 00 05 39 33 38 34 33 00 05 42 61 69 68 h i a . . 9 3 8 4 3 . . B a i h
000000F0 61 00 00 00 00 00 00 00 2A 00 00 00 2C 00 00 a . . . . . * . . . , .
00000100 00 06 00 0C 42 61 6E 63 65 6C 6F 6E 61 20 46 43 . . . . . B a n c e l o n a   F C
00000110 00 05 33 39 34 38 33 00 09 42 61 72 63 65 6C 6F . . 3 9 4 8 3 . . B a r c e l o
00000120 6E 61 00 00 00 00 00 00 20 00 00 00 2C 00 n a . . . . . , .
00000130 00 00 06 00 0C 42 61 6E 63 65 6C 6F 6E 61 20 46 . . . . . B a n c e l o n a   F
00000140 43 00 05 33 39 34 38 33 00 09 42 61 72 63 65 6C C . . 3 9 4 8 3 . . B a r c e l
00000150 6F 6E 61 00 00 00 01 00 00 00 03 2A 00 00 00 33 o n a . . . . . * . . . 3
```

**Criando o time dentro do arquivo.**

```
|1->Criar Clube|
|2->Ver o(s) clube(s)|
|3->Deletar||
|4->Update||
|5->Cadastrar Partida|
|0->Sair|
O que deseja fazer: 1
Digite o nome do time:
rio
Digite o CNPJ do time:
343453
Digite a cidade do time:
rio
|1->Adicionar outro time|
|0->Sair|
Escolha: 0
----Criado com sucesso----
|1->Criar Clube|
|2->Ver o(s) clube(s)|
|3->Deletar||
|4->Update||
|5->Cadastrar Partida|
|0->Sair|
O que deseja fazer: |
```

## Lendo todos os times criando.

```
[1->Criar Clube|
[2->Ver o(s) clube(s)|
[3->Deletar||
[4->Update||
[5->Cadastrar Partida|
[0->Sair|
O que deseja fazer: 2
Você deseja ler todos os times ou apenas um time:
[1->Todos|
[2->Apenas 1 time|
[0->Sair|
1
ID: 1
NOME: Cruzeiro
CNPJ: 384738
CIDADE: Belo Horizonte
Partidas Jogadas: 0
Pontos: 0
ID: 2
NOME: Flamengo
CNPJ: 384374
CIDADE: Rio de Janeiro
Partidas Jogadas: 0
Pontos: 0
ID: 4
NOME: São Paulo
CNPJ: 394893
CIDADE: São Paulo
Partidas Jogadas: 0
Pontos: 0
ID: 5
NOME: Bahia
CNPJ: 93843
CIDADE: Baiha
Partidas Jogadas: 0
Pontos: 0
ID: 6
NOME: Bancelona FC
CNPJ: 39483
CIDADE: Barcelona
Partidas Jogadas: 1
Pontos: 3
ID: 7
NOME: America MG
CNPJ: 3948439
CIDADE: Belo Horizonte
Partidas Jogadas: 1
Pontos: 3
ID: 3
NOME: Atletico MG
CNPJ: 32948934
CIDADE: Belo Horizonte
Partidas Jogadas: 2
Pontos: 0
ID: 8
NOME: rio
CNPJ: 343453
CIDADE: rio
Partidas Jogadas: 0
Pontos: 0
```

### Deletando os times criados.

```
|1->Criar Clube|
|2->Ver o(s) clube(s)|
|3->Deletar||
|4->Update||
|5->Cadastrar Partida|
|0->Sair|
O que deseja fazer: 3
Qual time você deseja deletar:5
Clube deletado
|1->Criar Clube|
|2->Ver o(s) clube(s)|
|3->Deletar||
|4->Update||
|5->Cadastrar Partida|
|0->Sair|
O que deseja fazer: 
```

### Atualizando os times criados.

```
[1->Criar Clube|
[2->Ver o(s) clube(s)|
[3->Deletar||
[4->Update||
[5->Cadastrar Partida|
[0->Sair|
O que deseja fazer: 4
Qual o id do time você deseja atualizar:1

Informe o nome:iueir

Informe o cnpj:549845

Informe a cidade:ireiirue
ID: 1
NOME: iueir
CNPJ: 549845
CIDADE: ireiirue
Partidas Jogadas: 0
Pontos: 0

Atualizado com sucesso!!
[1->Criar Clube|
[2->Ver o(s) clube(s)|
[3->Deletar||
[4->Update||
[5->Cadastrar Partida|
[0->Sair|
```



## Cadastrando os times criando no arquivo.

```
|1->Criar Clube|
|2->Ver o(s) clube(s)|
|3->Deletar||
|4->Update||
|5->Cadastrar Partida|
|0->Sair|
0 que deseja fazer: 5
Informe o id do primeiro time:1
Quantos gols o esse time fez:3
Informe o id do segundo time:4
Quantos gols esse time fez:4
ID: 1
NOME: iueir
CNPJ: 549845
CIDADE: ireiirue
Partidas Jogadas: 1
Pontos: 0
ID: 4
NOME: São Paulo
CNPJ: 394893
CIDADE: São Paulo
Partidas Jogadas: 1
Pontos: 3
|1->Criar Clube|
|2->Ver o(s) clube(s)|
|3->Deletar||
|4->Update||
|5->Cadastrar Partida|
|0->Sair|
0 que deseja fazer: 
```

## **8. Conclusão**

Praticamente todos que trabalham com desenvolvimento, de uma forma ou de outra, acabam tendo que manipular arquivos, sejam eles de texto, planilhas ou gerar relatórios. A seguir será visto como manipular arquivos com Java, bem como escrever e ler arquivos no formato de texto (txt). A manipulação de arquivos em Java acontece de forma simples e rápida, pois a linguagem dispõe de classes que executam praticamente todas as operações necessárias para tanto. Como foi possível ver ao longo do artigo, é bem simples a realização da leitura e escrita de arquivos quando se está programando em Java e as possibilidades são imensas cabendo somente ao programador ou analista o que fazer.

## **9. References**

<https://www.lncc.br/~rogerio/poo/08-Arquivos-Java/Escrever%20e%20Ler%20arquivos%20com%20Java.pdf>  
<https://www.devmedia.com.br/leitura-e-escrita-de-arquivos-de-texto-em-java/25529>