

POO – Módulo 9

Arquivos

Módulo ministrado pelo
Prof. Edson Emílio Scalabrin
PUCPR

Material produzido pelo
prof. Alcides Calsavara (BCC/PUCPR)

Conceitos

- 1. Arquivo txt**
- 2. Tabela de dados**
- 3. Arquivo csv**
- 4. Conversão de string para dados numéricos e boolean**

ARQUIVO TXT

LEITURA DE DADOS DE UM ARQUIVO TXT

FileReader

https://www.tutorialspoint.com/java/java_filereader_class.htm

```
class LeitorTXT {  
    public static void main(String[] args) {  
        try {  
            FileReader arquivo =  
                new FileReader("dados.txt");  
            BufferedReader buffer =  
                new BufferedReader(arquivo);  
            String str;  
            while ((str = buffer.readLine()) != null) {  
                System.out.println(str);  
            }  
        }  
        catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Crie um arquivo dados.txt e teste o programa acima.

**ESCRITA DE DADOS
EM UM ARQUIVO TXT**

FileWriter

[https://www.tutorialspoint.com/java/
java_filewriter_class.htm](https://www.tutorialspoint.com/java/java_filewriter_class.htm)

TABELA DE DADOS

Tabela de Dados

Cotação do real:

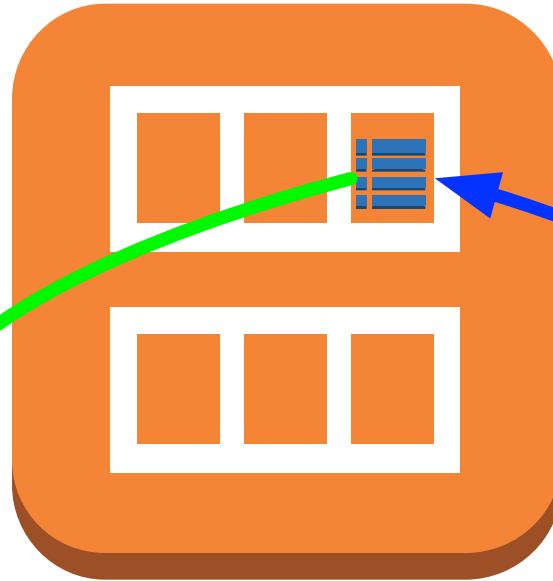
moeda	valor (R\$)
dolar	5.39
euro	6.38
libra	6.96
bitcoin	59638.56

cabeçalho
(header)

linhas (registros)

colunas

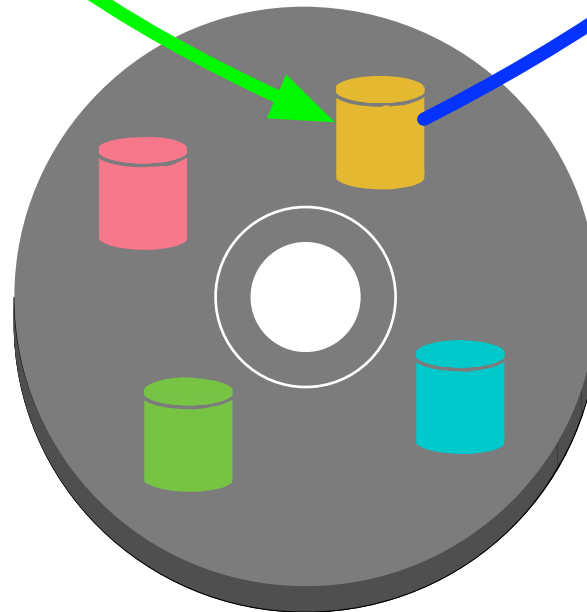
MEMÓRIA



**TABELA DE DADOS
PRESENTE NA
MEMÓRIA É
GRAVADA EM
UM ARQUIVO**

**TABELA DE DADOS
É LIDA DE UM ARQUIVO
E ARMAZENADA NA
MEMÓRIA**

DISCO



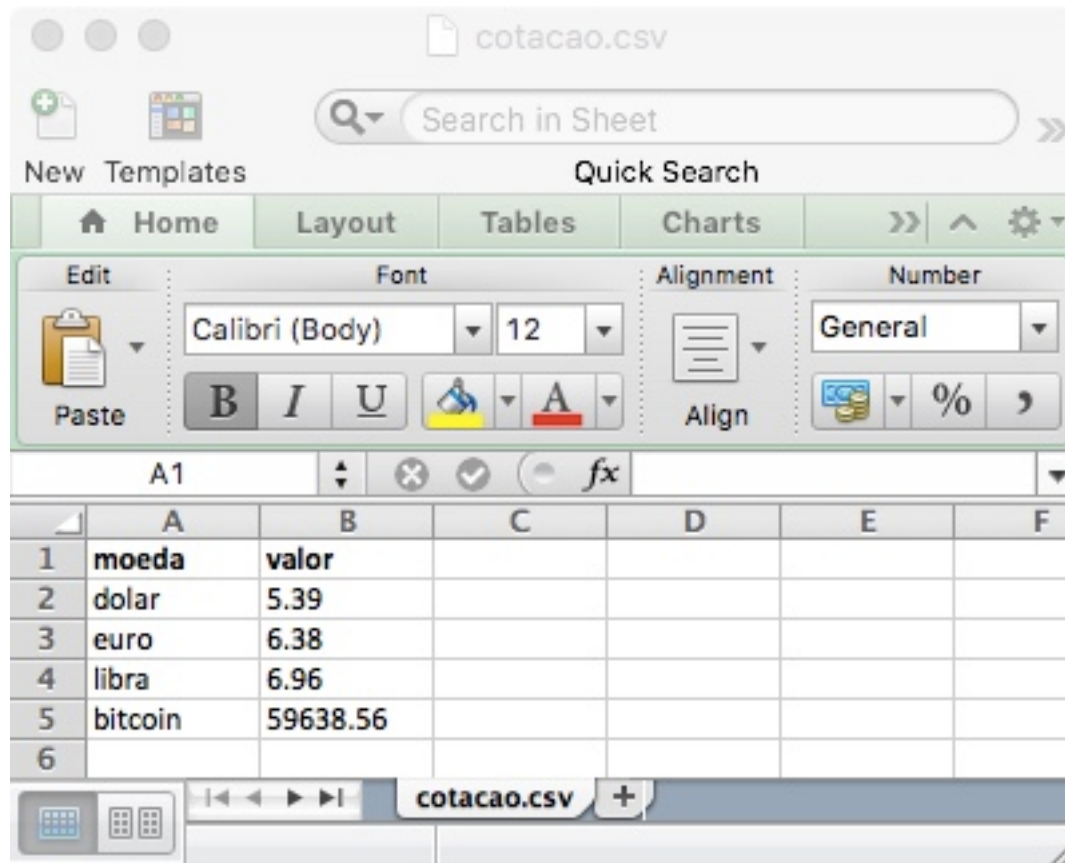
**Locais de armazenamento
de uma tabela de dados**

ARQUIVO CSV

comma-separated values

Arquivo csv

Arquivo csv no Excel (tabela de dados)



	A	B	C	D	E	F
1	moeda	valor				
2	dolar	5.39				
3	euro	6.38				
4	libra	6.96				
5	bitcoin	59638.56				
6						

Arquivo csv no editor de texto



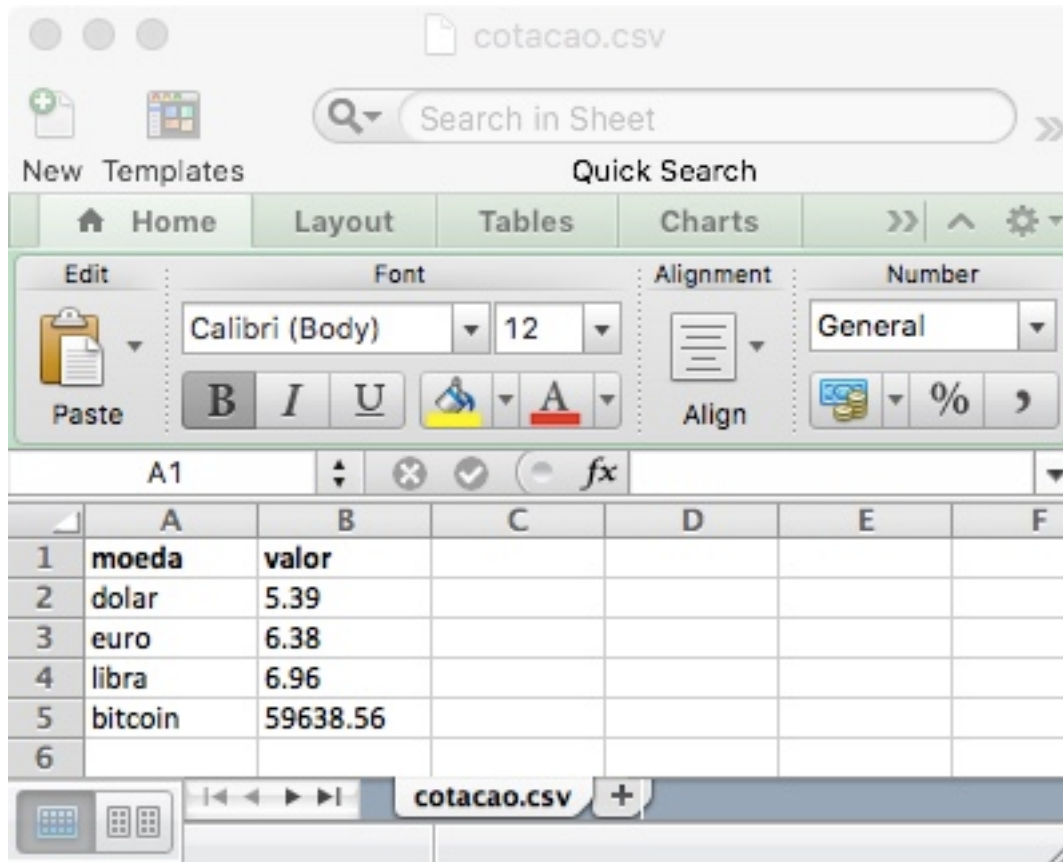
```
moeda;valor
dolar;5.39
euro;6.38
libra;6.96
bitcoin;59638.56
```

valores na mesma linha são
separados por **vírgula** ou
por **ponto-e-vírgula**

Arquivo csv

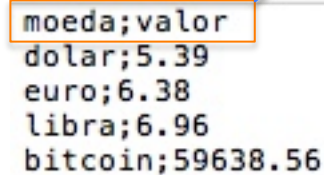
a primeira linha do arquivo contém o cabeçalho da tabela

Arquivo csv no Excel (tabela de dados)



	A	B	C	D	E	F
1	moeda	valor				
2	dolar	5.39				
3	euro	6.38				
4	libra	6.96				
5	bitcoin	59638.56				
6						

Arquivo csv no editor de texto



```
moeda;valor
dolar;5.39
euro;6.38
libra;6.96
bitcoin;59638.56
```

dados na mesma linha são separados por **vírgula** ou por **ponto-e-vírgula**

Exemplo

```
nome,idade,email  
Gustavo,28,gustavo@dicasdeprogramcao.com.br  
Joao,35,joao@dicasdejava.com.br  
Maria,23,maria@dicasdeprogramacao.com.br  
Ana,25,ana@dicasdejava.com.br
```

Gere o correspondente arquivo com o nome **pessoas.csv**:

1. Usando um editor de texto
2. Usando uma ferramenta de planilha (por exemplo, Excel)

LEITURA DE UM ARQUIVO CSV EM TABELA DE STRINGS USANDO BUFFEREDREADER

```
import java.io.*;
import java.util.*;

public class LeitorBuffer{
    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";

        List<List<String>> tabela = new ArrayList<>();
    }
}
```



```
import java.io.*;
import java.util.*;

public class LeitorBuffer{
    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";

        List<List<String>> tabela = new ArrayList<>();

        try
        {
            FileReader arquivo = new FileReader(NOME_ARQUIVO);
            BufferedReader br = new BufferedReader(arquivo);

        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

```
import java.io.*;
import java.util.*;

public class LeitorBuffer {
    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";
        List<List<String>> tabela = new ArrayList<>();
        try
        {
            FileReader arquivo = new FileReader(NOME_ARQUIVO);
            BufferedReader buffer = new BufferedReader(arquivo);
            String cabecalho = buffer.readLine();
            while (buffer.ready())
            {
                String linha = buffer.readLine();
                String[] tokens = linha.split(SEPARADOR);
                List<String> registro = Arrays.asList(tokens);
                tabela.add(registro);
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        for (List<String> registro: tabela)
        {
            for (String campo: registro)
                System.out.print(campo + " ");
            System.out.println();
        }
    }
}
```

LEITURA DE UM ARQUIVO CSV EM TABELA DE STRINGS USANDO SCANNER

```
import java.io.*;
import java.util.*;

public class LeitorScanner {
    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";
        List<List<String>> tabela = new ArrayList<>();
        try
        {
            File arquivo = new File(NOME_ARQUIVO);
            Scanner scanner_arquivo = new Scanner(arquivo);
            String cabecalho = scanner_arquivo.nextLine();
            while (scanner_arquivo.hasNextLine())
            {
                String linha = scanner_arquivo.nextLine();
                Scanner scanner_linha = new Scanner(linha);
                scanner_linha.useDelimiter(SEPARADOR);
                List<String> registro = new ArrayList<>();
                while (scanner_linha.hasNext())
                {
                    String campo = scanner_linha.next();
                    registro.add(campo);
                }
                tabela.add(registro);
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        for (List<String> registro: tabela)
        {
            for (String campo: registro)
                System.out.print(campo + " ");
            System.out.println();
        }
    }
}
```

CONVERSÃO DE STRING PARA DADOS NUMÉRICOS E BOOLEAN

Classes e métodos da Biblioteca de Java

1. Integer

- parseInt

2. Float

- parseFloat

3. Double

- parseDouble

4. Boolean

- parseBoolean

Exemplo de código

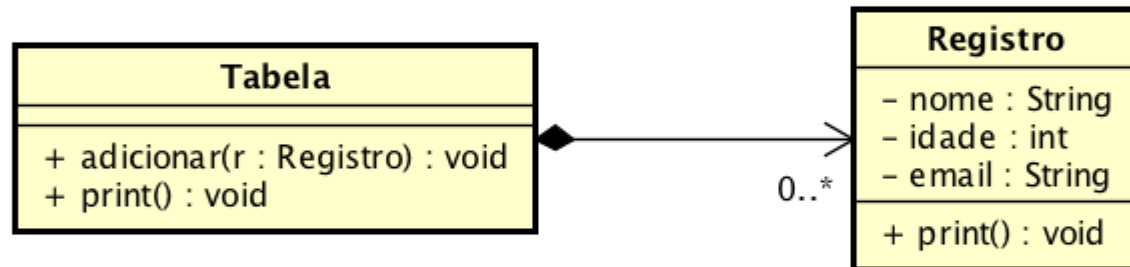
```
String texto_double = "3.49";  
double d = Double.parseDouble(texto_double);  
System.out.println(d);
```

```
String texto_int = "3";  
int k = Integer.parseInt(texto_int);  
System.out.println(k);
```

```
String texto_float = "3.49";  
float f = Float.parseFloat(texto_float);  
System.out.println(f);
```

```
String texto_boolean = "true";  
boolean b = Boolean.parseBoolean(texto_boolean);  
System.out.println(b);
```

LEITURA DE UM ARQUIVO CSV EM TABELA DE OBJETOS



```
public class Registro {
    private String nome;
    private int idade;
    private String email;

    public Registro(String nome, int idade, String email)
    {
        this.nome = nome;
        this.idade = idade;
        this.email = email;
    }

    public void print()
    {
        System.out.println(nome + " " + idade + " " + email);
    }
}
```

```
import java.util.*;

public class Tabela {
    private List<Registro> registros;
    public Tabela()
    {
        registros = new ArrayList<Registro>();
    }

    public void adicionar(Registro r)
    {
        registros.add(r);
    }

    public void print()
    {
        for (Registro r: registros) r.print();
    }
}
```

```
import java.util.*;
import java.io.*;

public class LeitorPOO {

    public static void main(String[] args)
    {
        String NOME_ARQUIVO = "pessoas.csv";
        String SEPARADOR = ",";
        Tabela tabela = new Tabela();
        try
        {
            File arquivo = new File(NOME_ARQUIVO);
            Scanner scanner_arquivo = new Scanner(arquivo);
            String cabecalho = scanner_arquivo.nextLine();
            while (scanner_arquivo.hasNextLine())
            {
                String linha = scanner_arquivo.nextLine();
                Scanner scanner_linha = new Scanner(linha);
                scanner_linha.useDelimiter(SEPARADOR);
                String nome = scanner_linha.next();
                int idade = scanner_linha.nextInt();
                String email = scanner_linha.next();
                Registro registro = new Registro(nome, idade, email);
                tabela.adicionar(registro);
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        tabela.print();
    }
}
```

Persistência de Objetos Java

GRAVAÇÃO E RESTAURAÇÃO

Gravação de objetos

```
public class Main {  
  
    public static void main(String [] args){  
  
        Empresa emp = new Empresa("Alpha");  
  
        emp.inserir(new Pessoa("Ana", "Pires", 21));  
        emp.inserir(new Pessoa("Marie", "Dupond", 23));  
        emp.inserir(new Pessoa("Claudia", "Silva", 35));  
        emp.inserir(new Pessoa("Pedro", "Predeira", 20));  
        emp.inserir(new Pessoa("Emilio", "Midio", 25));  
  
        String path = "/Users/scalabrin/NetBeansProjects/"  
            + "P00/src/main/java/com/poo/modulo9/";  
  
        Gravador.gravar(emp, path + "empresa.dat");  
  
        emp.imprimir();  
  
        Empresa empRead = (Empresa)Leitor.restaurar(path + "empresa.dat");  
  
        empRead.imprimir();  
    }  
}
```

Classe e Interface

(preparação para gravação de objetos)

```
public class Pessoa implements Serializable {  
    private String nome, sobrenome;  
    private int idade;  
  
    public Pessoa(String nome, String sobrenome, int idade) {  
        this.nome = nome;  
        this.sobrenome = sobrenome;  
        this.idade = idade;  
    }  
  
    @Override  
    public String toString() {  
        return "Pessoa{" + "nome=" + nome  
            + ", sobrenome=" + sobrenome  
            + ", idade=" + idade + '}';  
    }  
}
```

Classe e Interface

(preparação para gravação de objetos)

```
public class Empresa implements Serializable {  
    private String razaoSocial;  
    private ArrayList<Pessoa> RH;  
  
    public Empresa(String razaoSocial) {  
        this.razaoSocial = razaoSocial;  
        this.RH = new ArrayList();  
    }  
  
    public void inserir(Pessoa p){ this.RH.add(p); }  
  
    public void imprimir(){  
        for (Pessoa p : this.RH){  
            System.out.println(p.toString());  
        }  
    }  
}
```

Gravador de Objetos

```
public class Gravador {  
    public static void gravar(Object objeto, String caminho) {  
        try {  
            FileOutputStream arquivo = new FileOutputStream(caminho);  
            ObjectOutputStream stream = new ObjectOutputStream(arquivo);  
  
            stream.writeObject(objeto);  
  
            stream.close();  
  
        } catch (Exception exc) {  
            exc.printStackTrace();  
        }  
    }  
}
```


Leitor/Restaurador de Objetos

```
public class Leitor {  
    public static Object restaurar(String caminho) {  
        Object objeto = null;  
  
        try {  
            FileInputStream arquivo = new FileInputStream(caminho);  
            ObjectInputStream stream = new ObjectInputStream(arquivo);  
  
            objeto = stream.readObject();  
  
            stream.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        return objeto;  
    }  
}
```

ATIVIDADE

DIGITAR E TESTAR O EXEMPLO DE PESSOA E
EMPRESA