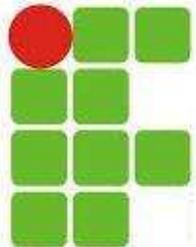

Programação Orientada a Objetos

Professor: Diego Oliveira



**Conteúdo 20:
Interface Gráfica 3**

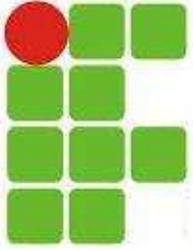




Manipulando Arquivos

- Aprendemos como trabalhar com vários elementos de interface gráfica
- Porém as informações até agora não eram salvas ou ficavam em um arquivo fixo
- E se quiséssemos gravar em um arquivo personalizado?
- E se desejássemos abrir um arquivo específico com dados de um formulário?
- Vamos aprender isso nesta aula!





Abrir Arquivo

- Para abrir arquivos utilizamos o JFileChooser
- Esta classe permitirá a escolher um arquivo específico do sistema de arquivos do computador
- Para abrir um arquivo utilizamos o método `showOpenDialog()`
- Depois de aberto o arquivo, podemos ler seu conteúdo ou alterá-lo





Abrir Arquivo

- Vamos analisar o código abaixo:

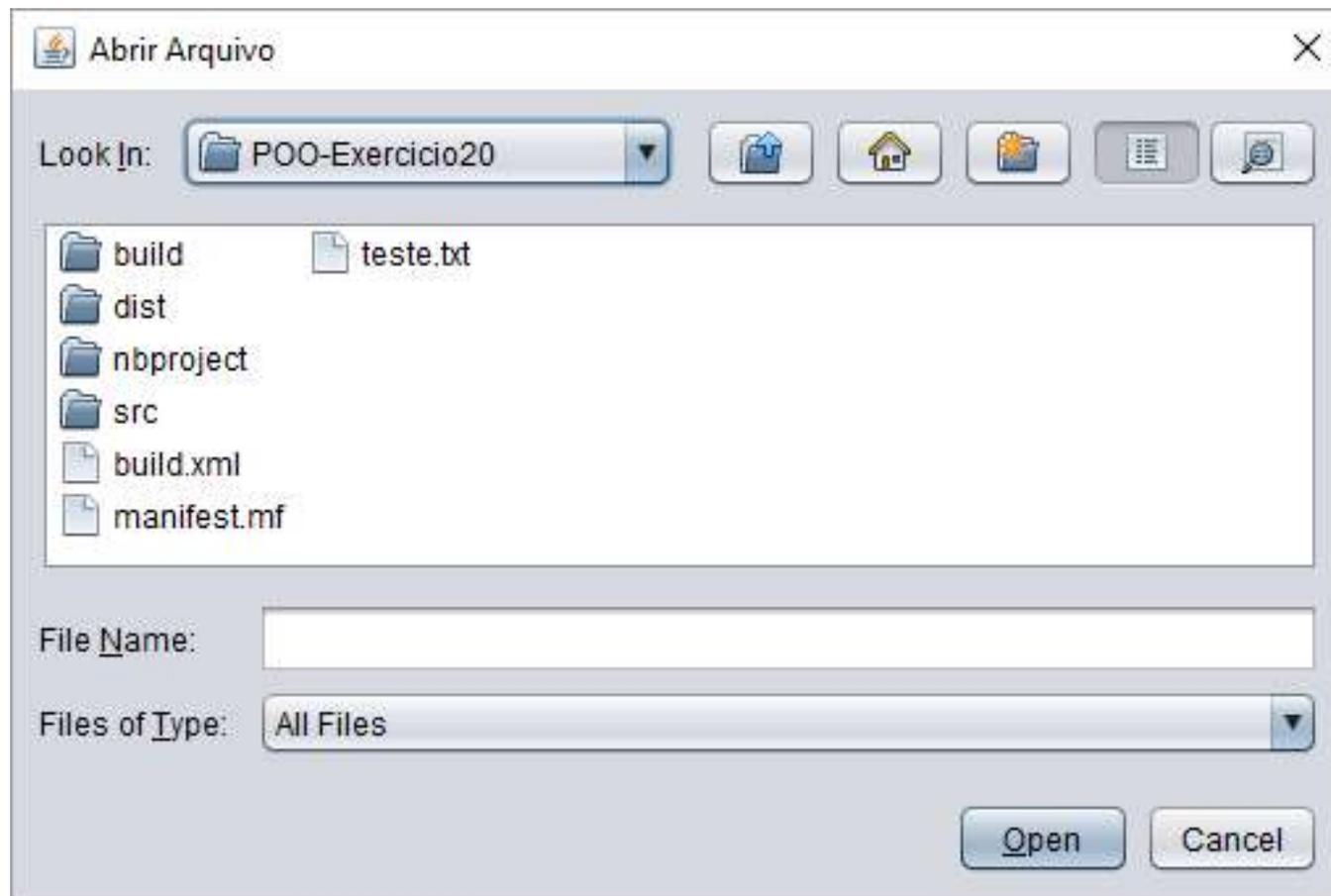
```
public static String abrirArquivo() {  
    JFileChooser abrir = new JFileChooser();  
    abrir.setDialogTitle("Abrir Arquivo");  
    abrir.setCurrentDirectory(new File("."));  
    int result = abrir.showOpenDialog(null);  
    File arquivo = null;  
    if (result == JFileChooser.APPROVE_OPTION) {  
        arquivo = abrir.getSelectedFile();  
    }  
    return arquivo.getAbsolutePath();  
}
```





Abrir Archivo

- Resultado:





Salvar Arquivo

- Para abrir arquivos utilizamos a mesma classe `JFileChooser`
- Porém utilizaremos um método diferente, o `showSaveDialog()`
- Note que para salvar um arquivo precisamos colocar um conteúdo dentro
- Para exemplificar esta aula, utilizaremos um dado de texto, ou seja, uma `String`
- Já vimos como salvar `Strings` em arquivos!





Salvar Arquivo

- Vamos analisar o código abaixo:

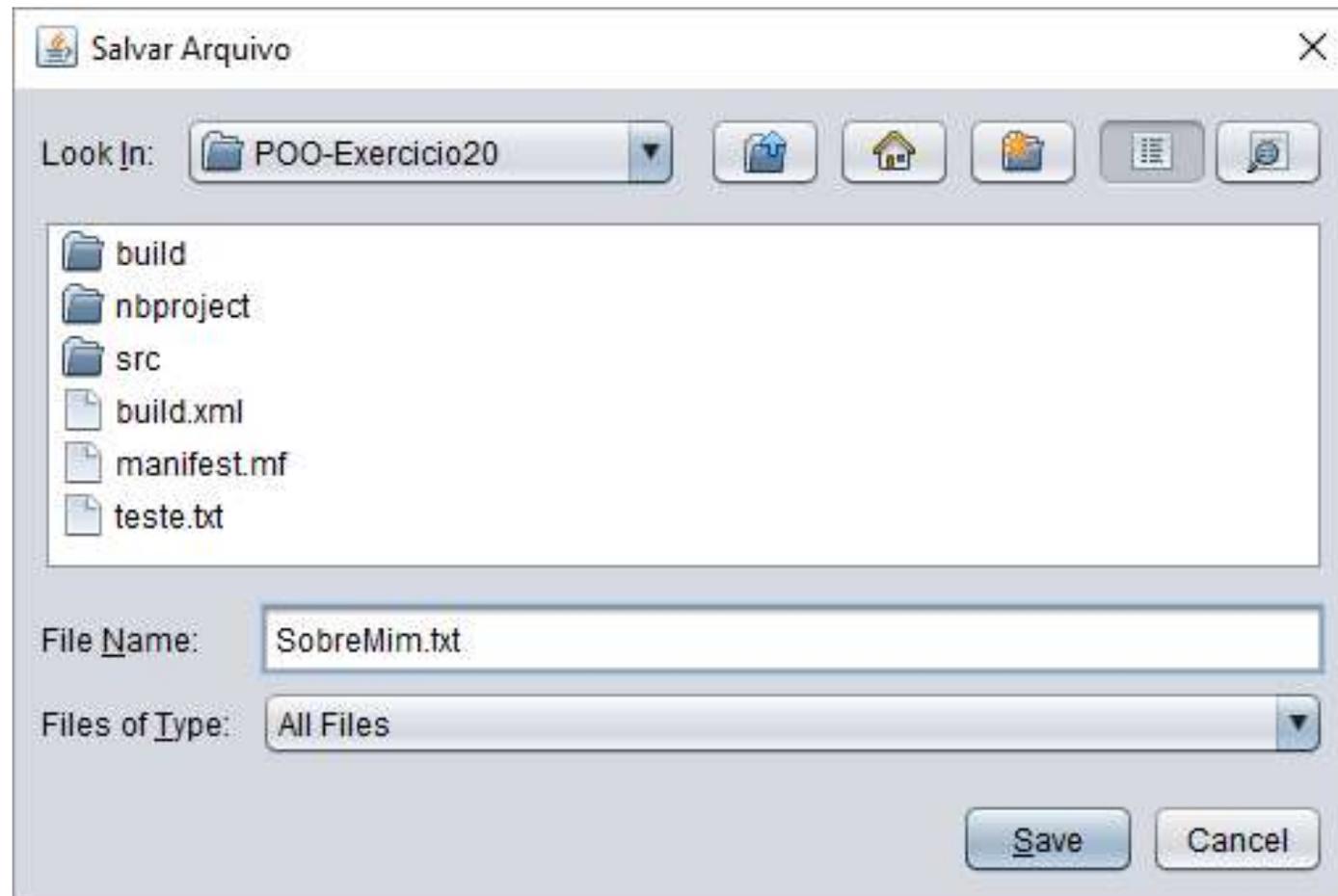
```
public static void salvarArquivo(String str)
    throws FileNotFoundException {
    JFileChooser salvar = new JFileChooser();
    salvar.setDialogTitle("Salvar Arquivo");
    salvar.setCurrentDirectory(new File("."));
    int select = salvar.showSaveDialog(null);
    if (select == JFileChooser.APPROVE_OPTION) {
        File arquivo2 = salvar.getSelectedFile();
        PrintWriter pw = new PrintWriter(arquivo2);
        pw.write(str);
        pw.close();
    }
}
```





Salvar Arquivo

- O resultado:





Filtros

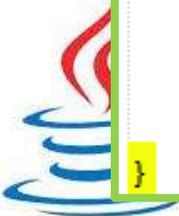
- Um JFileChooser pode abrir ou salvar qualquer tipo de arquivo
- Para evitar erros de usuário, podemos travar o tipo de arquivo que será aberto ou salvo
- Para isso utilizamos filtros que indicam as extensões de arquivo que queremos
- Desta maneira arquivos de outro tipo não aparecerão para serem abertos e no momento de salvar já se coloca a extensão





Filtros

```
public void lerArquivoTXT() {  
    JFileChooser abrir = new JFileChooser();  
    abrir.setDialogTitle("Abrir Arquivo TXT");  
    abrir.setFileFilter(new javax.swing.filechooser.FileFilter() {  
        @Override  
        public boolean accept(File f) {  
            return (f.getName().endsWith(".txt") || f.isDirectory());  
        }  
  
        @Override  
        public String getDescription() {  
            return "TXT (.txt)";  
        }  
    });  
    abrir.setCurrentDirectory(new File("."));  
    int result = abrir.showOpenDialog(null);  
    if (result == JFileChooser.APPROVE_OPTION) {  
        musica = abrir.getSelectedFile();  
    }  
}
```





Filtros

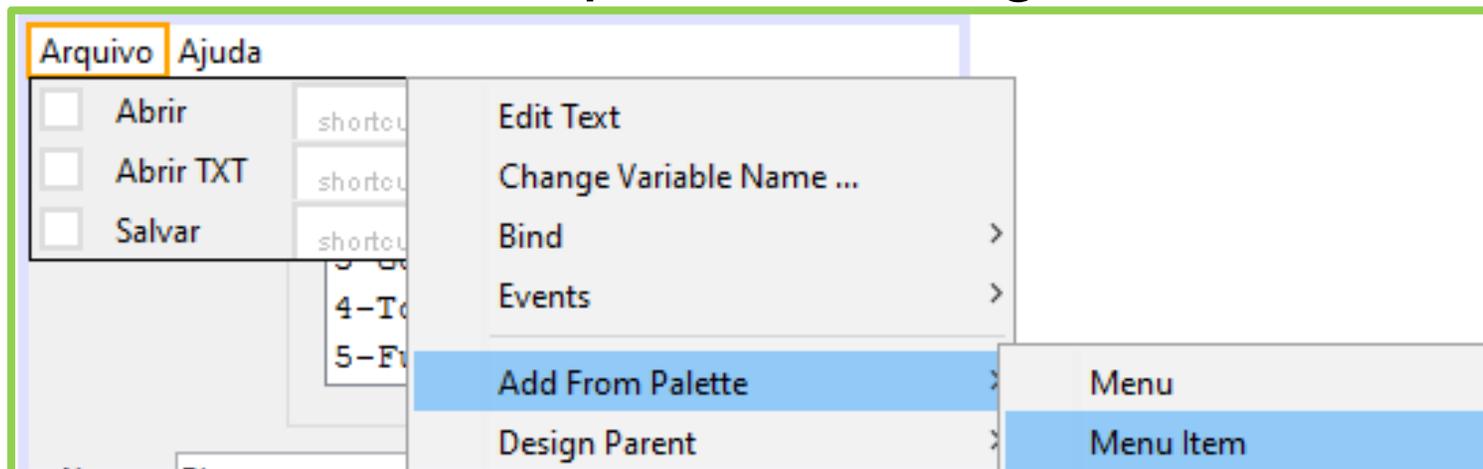
- Observe que já vem travado o tipo de arquivo:





JMenu e JMenuItem

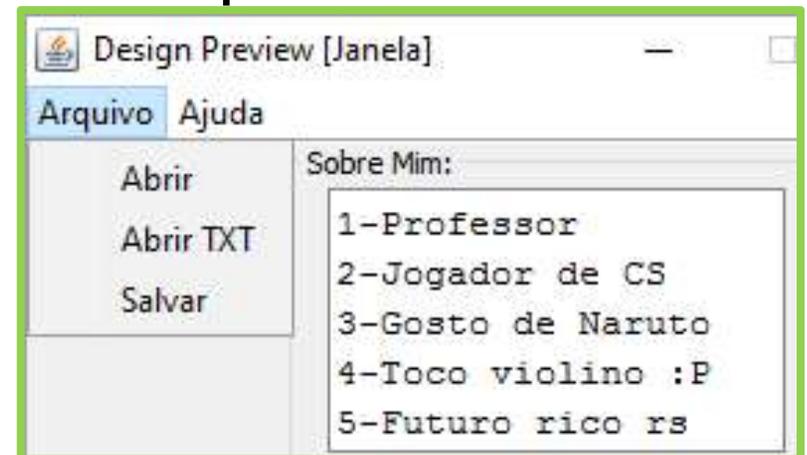
- Para colocar Menu em uma janela Java, arraste um JMenu para sua janela na aba DESIGN
- Para adicionar itens no Menu, clique com o botão direito -> Add From Palette -> Menu Item
- Para colocar eventos, clique 2x sobre Menu Item
- Para renomear, aperte F2 e digite o texto

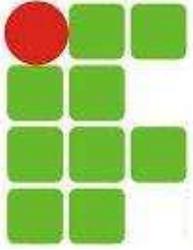




Exercício

- Utilize o **JFileChooser** e os conhecimentos da aula de Tratamento de Arquivo para:
 - 1: Abrir um arquivo TXT e mostrar seu conteúdo em um JTextArea
 - 2: Salvar um arquivo TXT com o conteúdo de um JTextArea
- Lembre-se de utilizar o filtro TXT para abertura e salvamento do arquivo
- Faça conforme o exemplo:





Perguntas?

