Sistma Gerenciador de FTP
Exercício Computacional I

Rafael Gonçalves de Oliveria Vianal[1]

[1]Sistemas de Informação – Universidade Federal do Mato Grosso do Sul (UFMS)
Caixa Postal 79400-000 – Coxim – MS – Brazil

rafael.viana@aluno.ufms.br

Resumo. Este relatório descreve como foi construido um sistema gerenciador de FTP, utilizando JavaFx como Graphical User Interface e o Apache Commons Net 3.6, como biblioteca de conexão FTP.

1. JavaFx

Foi escolhido o JavaFx para criar uma interface gráfica onde o usuário terá um melhor desempenho, ao utilizar o sistema. Para criar um sistema elegante foi utilizado uma biblioteca com novos elementos CSS, a bibliteca utilizada para essa finalidade foi JFoenix, essa bibliteca é open sorce e pode ser baixada no github https://github.com/jfoenixadmin/JFoenix Para icones foi utilizado a biblioteca fontawesomefx-8.9

2. Apache Commons Net 3.6

Para melhor desempenho nas conexões ftp, foi utilizada a biblioteca de conexão FTP da Apache Commons, onde a mesma se encontra na versão 3.6 current.

3. Problematica

U

4. Sections and Paragraphs

Section titles must be in boldface, 13pt, flush left. There should be an extra 12 pt of space before each title. Section numbering is optional. The first paragraph of each section should not be indented, while the first lines of subsequent paragraphs should be indented by 1.27 cm.

4.1. Subsections

The subsection titles must be in boldface, 12pt, flush left.

5. Figures and Captions

Figure and table captions should be centered if less than one line (Figure 1), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 2. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do

Figura 1. A typical figure

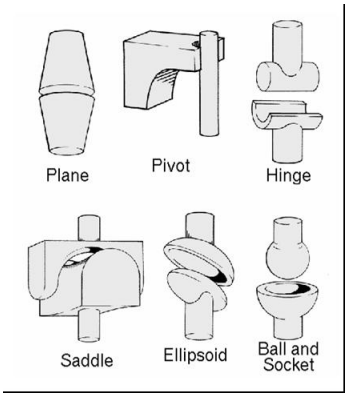

Figura 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 5.

not use more decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

Tabela 1. Variables to be considered on the evaluation of interaction techniques

	Value 1	Value 2
Case 1	1.0 ± 0.1	$1.75\times10^{-5} \pm 5\times10^{-7}$
Case 2	0.003(1)	100.0

6. Images

All images and illustrations should be in black-and-white, or gray tones, excepting for the papers that will be electronically available (on CD-ROMs, internet, etc.). The image resolution on paper should be about 600 dpi for black-and-white images, and

150-300 dpi for grayscale images. Do not include images with excessive resolution, as they may take hours to print, without any visible difference in the result.

7. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

Referências

Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, New Trends in Animation and Visualization. John Wiley & Sons ltd.

Knuth, D. E. (1984). The TEX Book. Addison-Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, Advances in Computer Science, pages 555–566. Publishing Press.

8. Anexos

A estrutura do código foi dividida em 3 pastas sendo elas Cliente, Icons e Socket.

8.1. A pasta cliente é responsável pela parte de Interface Gráfica do Usuário.

8.2. A pasta Icons armazena icons utilizados na pasta cliente.

8.3. A pasta socket e responsável por toda comunicação da Lib da Apache com o código.

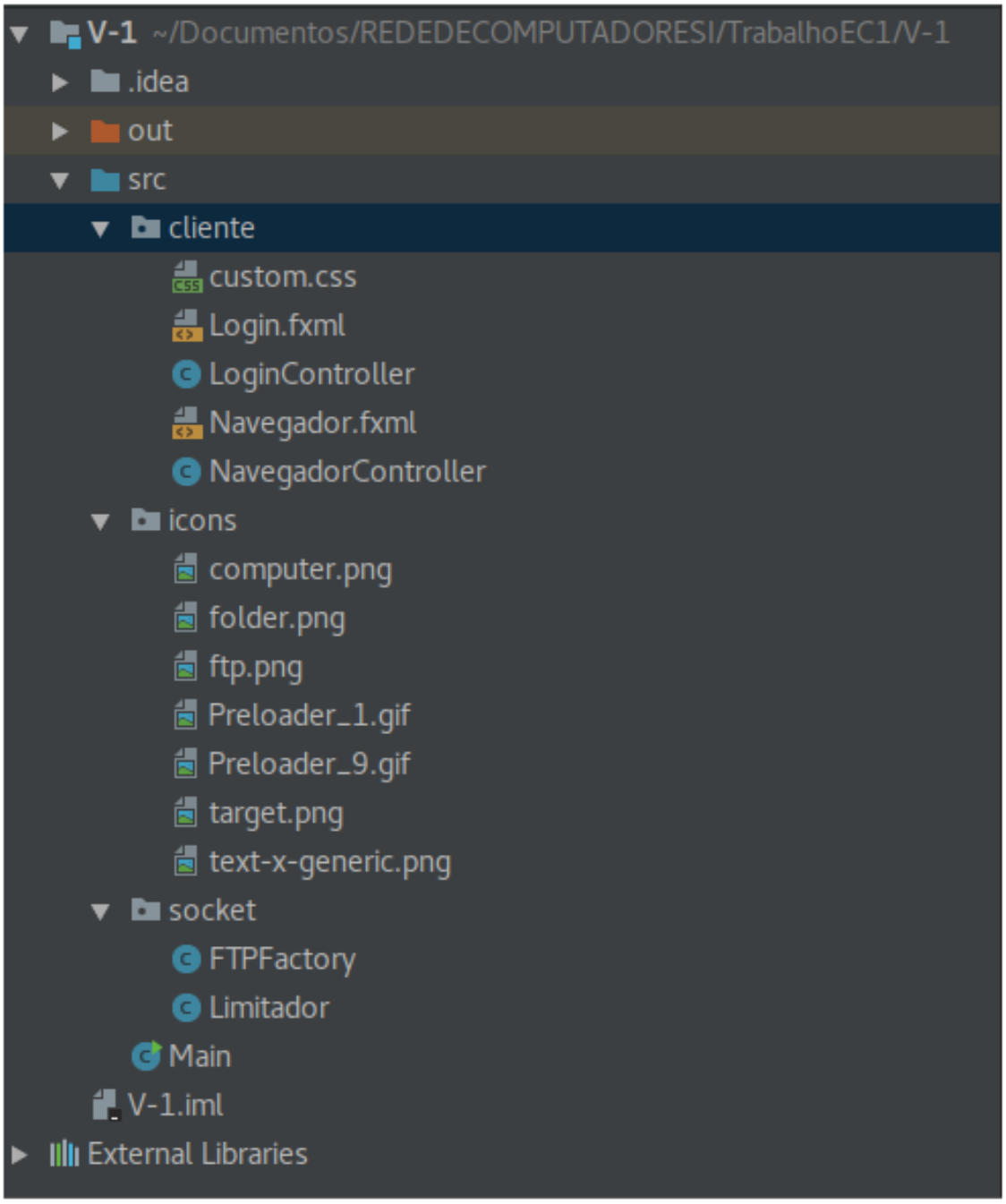

Figura 3. Imagem da Tela de Login.

Primeiramente a Class Main e invocada, chamando a Scene do Login.fxml.

```
1
2  public class Main  extends Application{
3
4  @Override
5      public void start(Stage stage) throws Exception {
6      Parent root = FXMLLoader.load(getClass().getResource("/
           cliente/Login.fxml"));
7      Scene scene = new Scene(root);
8      stage.setScene(scene);
9      stage.show();
10 }
11
12
13 public static void main(String[] args) {
14     launch(args);
15
16 }
17
18 }
```

Figura 4. Imagem da Tela de Login.

Com a tela de login aberta o usuário entra com as informações login , senha,

endereço do host, port do host mostrado no código abaixo.

```
1
2  private void login(ActionEvent event) {
3      btnLogin.setVisible(false);
4      imgProgress.setVisible(true);
5
6      PauseTransition pauseTransition = new PauseTransition();
7      pauseTransition.setDuration(Duration.seconds(3));
8      pauseTransition.setOnFinished(ev -> {
9
10 try {
11
12     int reply = FTPFactory.getInstance().FTPConecta(txtHostName.
           getText(),Integer.parseInt(txtHostPort.getText()),this.
           txtUsername.getText(),this.txtPassword.getText());
13     System.out.println("Igual:" + reply);
14
15     if (reply == 230) {
16
17         btnLogin.getScene().getWindow().hide();
18         completeLogin();
19
20     } else {
21
22         imgProgress.setVisible(false);
23         btnLogin.setVisible(true);
24         JOptionPane.showMessageDialog(null,"Erro Senha ou
               Usuário incorreto !!", "Erro ao Logar",JOptionPane.
               ERROR_MESSAGE);
25     }
26
27 } catch (IOException ex) {
28     Logger.getLogger(LoginController.class.getName()).log(Level.
           SEVERE, null, ex);
29 } catch (Exception ex) {
30     Logger.getLogger(LoginController.class.getName()).log(Level.
           SEVERE, null, ex);
31 }
32
33 });
34   pauseTransition.play();
35 }
36
37 private void completeLogin() throws IOException {
38
39     imgProgress.setVisible(false);
40     Stage dashboardStage = new Stage();
41     dashboardStage.setTitle("");
42     Parent root = FXMLLoader.load(getClass().getResource("
```

```
        Navegador.fxml"));
43      Scene scene = new Scene(root);
44      dashboardStage.setScene(scene);
45      dashboardStage.show();
46
47  }
```

Após validar os dados de usuário a tela de navegação de documentos e aberta, essa tela nada mais é do que um conjunto de botões em uma Grid e um TreeView do JavaFX.

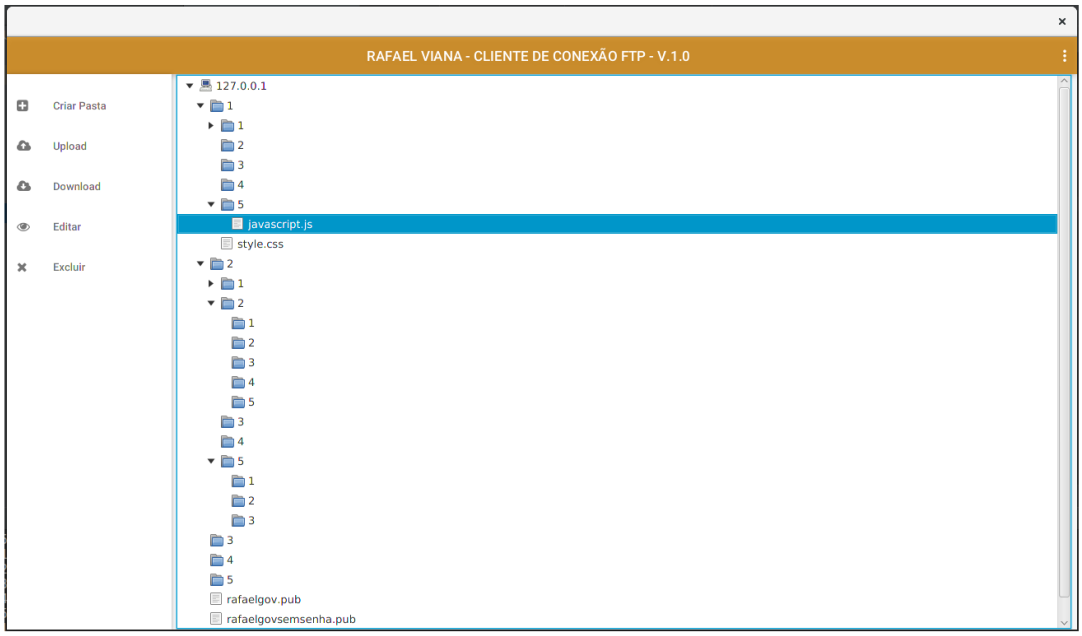

Figura 5. Imagem da Tela de Navegação.

Para poder realizar a comunicação entre as classes e o FTPClient da apache, foi criado uma classe chamada de FTPFactory onde cria uma getInstance de FTP-Client, podendo ser chamada de qualquer classe sem ter que que ser instanciada novamente, oque iria ocasionar a perda da conexão FTP.

```
1   public class FTPFactory {
2
3       private final FTPClient ftp;
4       private TreeItem<FTPFile> file;
5       private FTPFactory() {
6           this.ftp = new FTPClient();
7       }
8
9       public static FTPFactory getInstance() {
10
11          return FTPFactoryHolder.INSTANCE;
12      }
```

```
13
14
15      /**
16      * Classe privada que armazena a única instância de
            FTPFactory.
17      */
18      private static class FTPFactoryHolder {
19
20          private static final FTPFactory INSTANCE = new
                FTPFactory();
21      }
22
23
24
25      public FTPClient getFTP(){
26          return this.ftp;
27      }
28
29
30      public boolean Excluir(FTPFile file){
31          try {
32              if(file.isDirectory()){
33                  System.out.println(file.getLink());
34                  return     ftp.removeDirectory(file.getLink());
35              }else{
36                  System.out.println(file.getLink());
37                  return    ftp.deleteFile(file.getLink());
38              }
39          } catch (IOException e) {
40              e.printStackTrace();
41
42          }
43          return false;
44      }
45
46
47      public int FTPConecta(String host,int port, String user,
            String pwd) throws Exception {
48          int reply;
49          ftp.connect(host,port);
50          reply = ftp.getReplyCode();
51          if (!FTPReply.isPositiveCompletion(reply)) {
52              ftp.disconnect();
53              throw new Exception("Exception in connecting to FTP
                    Server");
54          }
55          ftp.login(user, pwd);
56          reply = ftp.getReplyCode();
57          ftp.setFileType(FTPClient.BINARY_FILE_TYPE);
```

```
58          ftp.enterLocalPassiveMode();
59          ftp.setAutodetectUTF8(true);
60          return reply;
61      }
62
63
64
65
66      public void disconnect() {
67          if (this.ftp.isConnected()) {
68              try {
69                  this.ftp.logout();
70                  this.ftp.disconnect();
71              } catch (IOException f) {
72
73              }
74          }
75      }
76
77  }
```