Trabalho II - Programação Orientada A Objetos II Alunos: Arthur Carboni Linzing, Mateus de Oliveira Lopes e Yan Gabriel Reis Oliveira

Nº refatoração: 1	Classe: MenusController.java Nome: Extract Cla		Nome: Extract Class
Código antes de refato	orar	Código após refato	rar
MenusController.java		RolagemDados.jav	ra
private int testeAtribute { return (int)(Math.rane atribute; } private int rodarAtribute { int menor = 7; int atual; int soma = 0; for(int i = 0; i < 4; i++ { atual = (int)(Math.rane atual) { soma = soma + atribute atual; if(menor > atual) { menor = atual; } } return soma - menor }	dom() * 20) + 1 + os() random() * 6) + 1; rual;	atributo; } public int rodarAt { int menor = 7; int atual; int soma = 0; for(int i = 0; i < {	ributo(int atributo) th.random() * 20) + 1 + ributos() 4; i++) Math.random() * 6) + 1; na + atual; tual){ atual;

Nº refatoração: 2	Classe: Main.java		Nome: Dead Code
Código antes de refato	orar	Código após refato	rar
Main.java public class Main { public static void ma throws SQLException{ Personagem pers Personagem(1,"Singu GOAT","Imagem Lega Personagem pers Personagem(2,"Degni B","Imagem",12,-1,3,4 PersonagemDBD = new PersonagemDBD	onagem = new s","The l",17,-1,3,4,7,2,-2); onagem1 = new s","The ,7,2,-2); AO personagemDAO	/*2° Refatoraçã Autor: Mateus Refatoração g códigos de teste ná código final	main(String[] args){ ão de Oliveira Lopes jeral para remover ão usados no main no nover códigos não

System.out.println("INSERE PERSONAGENS"); personagemDAO.insere(personagem);	}
---	---

Nº refatoração: 3	Classe: PersonagemSistemaDBDAO.java Nome: Dead Code e PersonagemSistemaDAO.java		Nome: Dead Code
Código antes de refa	itorar	Código após refato	rar
void insere(Person throws SQLException void atualizar(Personage throws SQLException void remover(Personage throws SQLException PersonagemSistem personagemId, int claint racald) throws SQLException personagemId, int claint racald	onagemSistemaDAO { agemSistema var1) n; onagemSistema mSistema varVelha) n; onagemSistema var1) n; onagemSistema var1) n; na buscaPorCodigo(int asseld, int sistemald,	void insere(Perso throws SQLExcepti void atualizar(Per varNova, Personag throws SQLExcepti void remover(Per throws SQLExcepti	rsonagemSistemaDAO { onagemSistema var1) ion; rsonagemSistema gemSistema varVelha) ion; rsonagemSistema var1) ion; rsonagemSistema var1) ion; nSistema> listar() throws
	PersonagemSistema) throws SQLException istema personagemId, int Id, int racald) throws	personagemSistem {} /*3° Refatoração Autor: Mateus de C Refatoração geral p busca por código q PersonagemSistem Objetivo: Remover */	para remover a função ue nunca é utilizada por naDBDAO funções não usadas agemSistema> listar()

Nº refatoração: 4	Classe: PersonagemSistemaDBDAO.java		Nome: Private Variable to Local Variable
Código antes de refa	atorar	Código após refato	rar
PersonagemSistems private ResultSet re public List <personal ;="" open();="" sql="SELECT * F personagem_sistem statement = connection.prepares result = statement</td><td>sult; gemSistema> listar() on { FROM na" sqlexceptic="" statement(sql);<="" td="" throws=""><td>PersonagemSisten public List<persona ;="" estatement(sql);<="" open();="" sql="SELECT * personagem_sisten statement = connection.prepare ResultSet result = statement.executed</td><td>agemSistema> listar() ion { FROM ma" sqlexcept="" td="" throws=""></persona></td></personal>	PersonagemSisten public List <persona ;="" estatement(sql);<="" open();="" sql="SELECT * personagem_sisten statement = connection.prepare ResultSet result = statement.executed</td><td>agemSistema> listar() ion { FROM ma" sqlexcept="" td="" throws=""></persona>		

Nº refatoração: 5	Classe: MenusContro	ller.java	Nome: Extract Method	
Código antes de refato	orar	Código após refato	rar	
MenusController.java		AlertUtil.java		
public static void exibit String mensagem) { Alert alerta = new Alert(Alert.AlertType.E alerta.setTitle(titu alerta.setContent alerta.showAndW } //(para chamar em Me exibirAlerta("Erro de B "Erro ao atualizar pers dados.");	Paragrama (Armonical Controller.java (Armonical Controller.java (Armonical Controller.)	String mensagem) Alert alerta = r Alert(Alert.AlertTyp alerta.setTitle(alerta.setHead alerta.setCont alerta.showAn } } //(para chamar em AlertUtil.exibirAlerta	MenusController.java a("Erro de Banco de tualizar personagem no	

Nº refatoração: 6	,		Nome: Tornando método estático
Código antes de refato	orar Código após refato		rar

```
MenusController.java
                                                  MenusController.java
public void exibirResultadoTeste(String
                                                  public static void
atributo, int resultado) {
                                                  exibirResultadoTeste(String atributo, int
     Alert alerta = new
                                                  resultado) {
Alert(Alert.AlertType.INFORMATION);
                                                       Alert alerta = new
     alerta.setTitle("Resultado do Teste");
                                                  Alert(Alert.AlertType.INFORMATION);
     alerta.setHeaderText(null);
                                                       alerta.setTitle("Resultado do Teste");
alerta.setContentText("O resultado do
teste de " + atributo + " foi: " + resultado);
                                                       alerta.setHeaderText(null);
                                                       alerta.setContentText("O resultado do
                                                  teste de " + atributo + " foi: " + resultado);
     alerta.showAndWait();
                                                       alerta.showAndWait();
                                                    }
```

Nº refatoração: 7	Classe:MenusControl	classe:MenusController.java Nome: Extract M		
Código antes de refato	orar	Código após refato	rar	
MenusController.java		ResultadoTesteVie	w.java	
alerta.setHeader	VINFORMATION); esultado do Teste"); Text(null); Text("O resultado do " foi: " + resultado); Vait();	resultado) { Alert alerta = r Alert(Alert.AlertTyp alerta.setTitle(alerta.setHead alerta.setCont teste de " + atributo alerta.showAr } }	te(String atributo, int new e.INFORMATION); ("Resultado do Teste"); derText(null); centText("O resultado do o + " foi: " + resultado);	

Nº refatoração: 8	Classe: MenusController		Nome: Retirar repetição
Código antes de refatorar		Código após refatorar	
Button btnForcaAleato Button("""); btnForcaAleatorio.set(campoForca.setText(S darAtributos()))); grid.add(btnForcaAlea	OnAction(e -> String.valueOf(dado.ro	#D3D3D3;");	el = new

```
...

Campos[i] = campo;

Button btnAleatorio =
criarBotaoAleatorio(campo);
grid.add(label, 0, i + 1);
grid.add(campo, 1, i + 1);
grid.add(btnAleatorio, 2, i + 1);
}
```

Nº refatoração: 9	Classe: MenusController		Nome: Extract Method	
Código antes de refato	orar	Código após refato	rar	
MenusController.java		defAtributos.java	defAtributos.java	
private Button criarBot campo) {	torio = new etOnAction(e -> valueOf(dado.rodarAt	campo) { RolagemDa RolagemDados(); Button btnA Button("""); btnAleatoric	BotaoAleatorio(TextField ados dado2 = new aleatorio = new b.setOnAction(e -> ng.valueOf(dado2.rodarA aleatorio;	

Nº refatoração: 10	Classe:MenusControll	ller Nome: Extract Metho	
Código antes de refato	Código antes de refatorar		rar
MenusController.java private void exibirTelaDefinirAtributos(int		defAtributos.java public void exibirTelaDefinirAtributos(int	
personagemld, String	nome, String classe,	personagemld, Stri	ng nome, String classe,
String raca, String info	, String	String raca, String	info, String
caminholmagem, Stag	e telaAnterior) {	caminholmagem, S	Stage telaAnterior) {
String[] atributos = {"Força",			
"Destreza", "Constituiç	ão", "Inteligência",	String[] atrib	outos = {"Força",
"Sabedoria", "Carisma	"};	"Destreza", "Consti	tuição", "Inteligência",
TextField[] cam	npos = new	"Sabedoria", "Caris	sma"};
TextField[atributos.leng	gth];	TextField[] campos = new	
telaAnterior.close();		TextField[atributos.length];	
Stage stageAtributos = new Stage();		telaAnterior.close();	
stageAtributos.	setTitle("Definir	Stage stage	eAtributos = new Stage();

```
Atributos");
                                                       stageAtributos.setTitle("Definir
                                               Atributos");
       GridPane grid = new GridPane();
                                                       GridPane grid = new GridPane();
       grid.setPadding(new Insets(20));
                                                       grid.setPadding(new Insets(20));
       grid.setHgap(10);
       grid.setVgap(10);
                                                       grid.setHgap(10);
       grid.setStyle("-fx-background-color:
                                                       grid.setVgap(10);
#2B2B2B;");
                                                       grid.setStyle("-fx-background-color:
                                               #2B2B2B;");
       Label titulo = new Label("Definir
Atributos do Personagem");
                                                       Label titulo = new Label("Definir
       titulo.setFont(Font.font("Arial", 18));
                                               Atributos do Personagem");
       titulo.setTextFill(Color.WHITE);
                                                       titulo.setFont(Font.font("Arial", 18));
       GridPane.setColumnSpan(titulo, 2);
                                                       titulo.setTextFill(Color.WHITE);
                                                       GridPane.setColumnSpan(titulo, 2);
       Label lblForca = new
Label("Força:");
                                                       Label lblForca = new
       lblForca.setStyle("-fx-text-fill:
                                               Label("Força:");
#D3D3D3;");
                                                       lblForca.setStyle("-fx-text-fill:
       TextField campoForca = new
                                               #D3D3D3;");
TextField();
                                                       TextField campoForca = new
                                               TextField();
       Label IbIDestreza = new
Label("Destreza:");
                                                       Label lblDestreza = new
       lblDestreza.setStyle("-fx-text-fill:
                                               Label("Destreza:");
#D3D3D3;");
                                                       lblDestreza.setStyle("-fx-text-fill:
       TextField campoDestreza = new
                                               #D3D3D3;");
TextField();
                                                       TextField campoDestreza = new
                                               TextField();
       Label lblConstituicao = new
Label("Constituição:");
                                                       Label lblConstituicao = new
       lblConstituicao.setStyle("-fx-text-fill:
                                               Label("Constituição:");
#D3D3D3;");
                                                       lblConstituicao.setStyle("-fx-text-fill:
       TextField campoConstituicao = new
                                               #D3D3D3;");
TextField();
                                                       TextField campoConstituicao = new
                                               TextField();
       Label IblInteligencia = new
Label("Inteligência:");
                                                       Label IblInteligencia = new
       lblInteligencia.setStyle("-fx-text-fill:
                                               Label("Inteligência:");
```

```
#D3D3D3;");
                                                       lblInteligencia.setStyle("-fx-text-fill:
       TextField campoInteligencia = new
                                                #D3D3D3;");
TextField();
                                                       TextField campoInteligencia = new
                                                TextField();
       Label IblSabedoria = new
Label("Sabedoria:");
                                                       Label IblSabedoria = new
       lblSabedoria.setStyle("-fx-text-fill:
                                                Label("Sabedoria:");
                                                       lblSabedoria.setStyle("-fx-text-fill:
#D3D3D3;");
       TextField campoSabedoria = new
                                                #D3D3D3;");
                                                       TextField campoSabedoria = new
TextField();
                                                TextField();
       Label lblCarisma = new
Label("Carisma:");
                                                       Label lblCarisma = new
       lblCarisma.setStyle("-fx-text-fill:
                                                Label("Carisma:");
                                                       lblCarisma.setStyle("-fx-text-fill:
#D3D3D3;");
       TextField campoCarisma = new
                                                #D3D3D3;");
TextField();
                                                       TextField campoCarisma = new
                                                TextField();
       for (int i = 0; i < atributos.length; <math>i++)
{
                                                       for (int i = 0; i < atributos.length; i++)
          Label label = new
                                                {
Label(atributos[i] + ":");
                                                          Label label = new
          label.setStyle("-fx-text-fill:
                                                Label(atributos[i] + ":");
#D3D3D3;");
                                                          label.setStyle("-fx-text-fill:
          TextField campo = new
                                                #D3D3D3;");
                                                          TextField campo = new
TextField();
          campos[i] = campo;
                                                TextField();
                                                          campos[i] = campo;
          Button btnAleatorio =
criarBotaoAleatorio(campo);
                                                          Button btnAleatorio =
          grid.add(label, 0, i + 1);
                                                criarBotaoAleatorio(campo);
          grid.add(campo, 1, i + 1);
                                                          grid.add(label, 0, i + 1);
          grid.add(btnAleatorio, 2, i + 1);
                                                          grid.add(campo, 1, i + 1);
       }
                                                          grid.add(btnAleatorio, 2, i + 1);
                                                       }
       Button btnSalvar = new
```

```
Button("Salvar Personagem");
                                                     Button btnSalvar = new
       btnSalvar.setFont(Font.font("Arial",
                                              Button("Salvar Personagem");
                                                     btnSalvar.setFont(Font.font("Arial",
16));
       btnSalvar.setTextFill(Color.WHITE);
                                              16));
                                                     btnSalvar.setTextFill(Color.WHITE);
btnSalvar.setStyle("-fx-background-color:
#007ACC; -fx-background-radius: 8;");
                                              btnSalvar.setStyle("-fx-background-color:
       btnSalvar.setOnAction(e -> {
                                              #007ACC; -fx-background-radius: 8;");
         try {
                                                     btnSalvar.setOnAction(e -> {
                                                       try {
(campoForca.getText().isEmpty() ||
                                                          if
campoDestreza.getText().isEmpty() ||
                                              (campoForca.getText().isEmpty() ||
                                              campoDestreza.getText().isEmpty() ||
campoInteligencia.getText().isEmpty() ||
campoConstituicao.getText().isEmpty() ||
                                              campoInteligencia.getText().isEmpty() ||
campoSabedoria.getText().isEmpty() ||
                                              campoConstituicao.getText().isEmpty() ||
campoCarisma.getText().isEmpty()) {
                                              campoSabedoria.getText().isEmpty() ||
              throw new
                                              campoCarisma.getText().isEmpty()) {
IllegalArgumentException("Todos os
                                                            throw new
                                              IllegalArgumentException("Todos os
campos de atributos devem ser
preenchidos.");
                                              campos de atributos devem ser
            }
                                              preenchidos.");
            if
(Integer.parseInt(campoForca.getText()) >
                                                          if
                                              (Integer.parseInt(campoForca.getText()) >
20 ||
Integer.parseInt(campoDestreza.getText())
                                              20 ||
> 20 ||
                                             Integer.parseInt(campoDestreza.getText())
Integer.parseInt(campoInteligencia.getText()
                                             > 20 ||
) > 20 ||
                                             Integer.parseInt(campoInteligencia.getText()
Integer.parseInt(campoConstituicao.getText
                                             ) > 20 ||
                                             Integer.parseInt(campoConstituicao.getText
()) > 20 ||
Integer.parseInt(campoSabedoria.getText())
                                             ()) > 20 ||
> 20 ||
                                             Integer.parseInt(campoSabedoria.getText())
Integer.parseInt(campoCarisma.getText()) >
                                              > 20 ||
20) {
                                              Integer.parseInt(campoCarisma.getText()) >
              throw new
                                              20) {
IllegalArgumentException("Os atributos não
                                                             throw new
podem ser maiores que 20.");
                                              IllegalArgumentException("Os atributos não
                                              podem ser maiores que 20.");
            }
```

```
}
           Personagem personagem =
new Personagem(personagemId, nome,
                                                      Personagem personagem =
                                           new Personagem(personagemId, nome,
info, caminholmagem, 1,
                                           info, caminholmagem, 1,
Integer.parseInt(campoForca.getText()),
Integer.parseInt(campoDestreza.getText()),
                                           Integer.parseInt(campoForca.getText()),
                                           Integer.parseInt(campoDestreza.getText()),
Integer.parseInt(campoInteligencia.getText()
                                           Integer.parseInt(campoInteligencia.getText()
Integer.parseInt(campoConstituicao.getText
                                           Integer.parseInt(campoConstituicao.getText
()),
Integer.parseInt(campoSabedoria.getText())
                                           Integer.parseInt(campoSabedoria.getText())
Integer.parseInt(campoCarisma.getText()));
                                           Integer.parseInt(campoCarisma.getText()));
personagens.add(personagem);
           PersonagemDBDAO
                                           //personagens.add(personagem);
personagemDB = new
                                                      PersonagemDBDAO
PersonagemDBDAO();
                                           personagemDB = new
           SistemaDBDAO sistemaDB =
                                           PersonagemDBDAO();
new SistemaDBDAO();
                                                      SistemaDBDAO sistemaDB =
                                           new SistemaDBDAO();
           Sistema sistema;
           try {
                                                       Sistema sistema:
              sistema =
                                                      try {
sistemaDB.buscaPorCodigo(1);
                                                         sistema =
           } catch (SQLException ex) {
                                           sistemaDB.buscaPorCodigo(1);
             AlertUtil.exibirAlerta("Erro de
                                                      } catch (SQLException ex) {
Banco de Dados", "Erro ao buscar sistema
                                                         AlertUtil.exibirAlerta("Erro de
no banco de dados.");
                                           Banco de Dados", "Erro ao buscar sistema
                                           no banco de dados.");
             return;
           }
                                                         return;
           ClasseDBDAO classeDB =
                                                       ClasseDBDAO classeDB =
new ClasseDBDAO();
           Classe classet;
                                           new ClasseDBDAO();
                                                      Classe classet;
           try {
              classet =
                                                      try {
classeDB.buscaPorNome(classe);
                                                         classet =
           } catch (SQLException ex) {
                                           classeDB.buscaPorNome(classe);
```

```
AlertUtil.exibirAlerta("Erro de
                                                       } catch (SQLException ex) {
Banco de Dados", "Erro ao buscar classe
                                                         AlertUtil.exibirAlerta("Erro de
                                           Banco de Dados", "Erro ao buscar classe
no banco de dados.");
              return;
                                           no banco de dados.");
           }
                                                         return;
                                                       }
           RacaDBDAO racaDB = new
RacaDBDAO():
                                                       RacaDBDAO racaDB = new
                                           RacaDBDAO();
           Raca racat;
                                                       Raca racat;
           try {
              racat =
                                                       try {
racaDB.buscaPorNome(raca);
                                                         racat =
           } catch (SQLException ex) {
                                           racaDB.buscaPorNome(raca);
              AlertUtil.exibirAlerta("Erro de
                                                       } catch (SQLException ex) {
Banco de Dados", "Erro ao buscar raça no
                                                         AlertUtil.exibirAlerta("Erro de
                                           Banco de Dados", "Erro ao buscar raça no
banco de dados.");
                                           banco de dados.");
              return:
                                                         return:
           PersonagemSistemaDBDAO
                                                       PersonagemSistemaDBDAO
personagemSistemaDB = new
PersonagemSistemaDBDAO();
                                           personagemSistemaDB = new
                                           PersonagemSistemaDBDAO();
           PersonagemSistema
                                                       PersonagemSistema
personagemSistema = new
PersonagemSistema(personagem, sistema,
                                           personagemSistema = new
racat, classet);
                                           PersonagemSistema(personagem, sistema,
                                           racat, classet);
           try {
                                                       try {
personagemDB.insere(personagem);
                                           personagemDB.insere(personagem);
personagemSistemaDB.insere(personagem
Sistema);
                                           personagemSistemaDB.insere(personagem
           } catch (SQLException ex) {
                                           Sistema);
              AlertUtil.exibirAlerta("Erro de
                                                       } catch (SQLException ex) {
Banco de Dados", "Erro ao salvar
                                                         AlertUtil.exibirAlerta("Erro de
personagem no banco de dados.");
                                           Banco de Dados", "Erro ao salvar
                                           personagem no banco de dados.");
           stageAtributos.close();
         } catch (NumberFormatException
                                                       stageAtributos.close();
```

```
ex) {
                                                         } catch (NumberFormatException
            AlertUtil.exibirAlerta("Erro de
                                               ex) {
Formato", "Os atributos devem ser números
                                                           AlertUtil.exibirAlerta("Erro de
inteiros.");
                                               Formato", "Os atributos devem ser números
          } catch (IllegalArgumentException
                                               inteiros.");
ex) {
                                                         } catch (IllegalArgumentException
            AlertUtil.exibirAlerta("Campos
                                               ex) {
Vazios", ex.getMessage());
                                                           AlertUtil.exibirAlerta("Campos
                                               Vazios", ex.getMessage());
          }
       });
                                                         }
                                                      });
       grid.add(titulo, 0, 0);
       grid.add(lblForca, 0, 1);
                                                      grid.add(titulo, 0, 0);
       grid.add(campoForca, 1, 1);
                                                      grid.add(lblForca, 0, 1);
       grid.add(lblDestreza, 0, 2);
                                                      grid.add(campoForca, 1, 1);
       grid.add(campoDestreza, 1, 2);
                                                      grid.add(lblDestreza, 0, 2);
       grid.add(lblConstituicao, 0, 3);
                                                      grid.add(campoDestreza, 1, 2);
       grid.add(campoConstituicao, 1, 3);
                                                      grid.add(lblConstituicao, 0, 3);
       grid.add(lblInteligencia, 0, 4);
                                                      grid.add(campoConstituicao, 1, 3);
       grid.add(campoInteligencia, 1, 4);
                                                      grid.add(lblInteligencia, 0, 4);
       grid.add(lblSabedoria, 0, 5);
                                                      grid.add(campoInteligencia, 1, 4);
       grid.add(campoSabedoria, 1, 5);
                                                      grid.add(lblSabedoria, 0, 5);
       grid.add(lblCarisma, 0, 6);
                                                      grid.add(campoSabedoria, 1, 5);
       grid.add(campoCarisma, 1, 6);
                                                      grid.add(lblCarisma, 0, 6);
       grid.add(btnSalvar, 1, 7);
                                                      grid.add(campoCarisma, 1, 6);
                                                      grid.add(btnSalvar, 1, 7);
       Scene cena = new Scene(grid, 400,
300);
                                                      Scene cena = new Scene(grid, 400,
                                               300);
       stageAtributos.setScene(cena);
                                                      stageAtributos.setScene(cena);
       stageAtributos.show();
     }
                                                      stageAtributos.show();
                                                    }
```

Nº refatoração: 11	Classe:MenusController		Nome: Extract Method
Código antes de refato	rar Código após refato		rar
MenusController.java private void exibirTela	EditarPersonagens()	'	laEditarPersonagens() { stageEditar = new

```
Stage stageEditar = new
                                           Stage();
Stage();
                                                       stageEditar.setTitle("Editar
           stageEditar.setTitle("Editar
                                           Personagens");
Personagens");
                                                      VBox vbox = new VBox();
           VBox vbox = new VBox():
                                                      vbox.setSpacing(10);
           vbox.setSpacing(10);
                                                      vbox.setPadding(new
           vbox.setPadding(new
                                           Insets(15));
Insets(15));
                                           vbox.setStyle("-fx-background-color:
vbox.setStyle("-fx-background-color:
                                           #2B2B2B;");
#2B2B2B;");
                                                      PersonagemSistemaDBDAO
           PersonagemSistemaDBDAO
                                           personagemSistemaDB = new
personagemSistemaDB = new
                                           PersonagemSistemaDBDAO();
PersonagemSistemaDBDAO();
                                                      List<PersonagemSistema>
           List<PersonagemSistema>
                                           personagensSistema;
personagensSistema;
                                                      try {
           try {
                                                      personagensSistema =
                                           personagemSistemaDB.listar();
           personagensSistema =
personagemSistemaDB.listar();
                                                      } catch (SQLException ex) {
                                                         AlertUtil.exibirAlerta("Erro de
           } catch (SQLException ex) {
              AlertUtil.exibirAlerta("Erro de
                                           Banco de Dados", "Erro ao buscar
Banco de Dados", "Erro ao buscar
                                           personagens no banco de dados.");
personagens no banco de dados.");
                                                      return;
           return;
                                                      }
           }
                                                      for (PersonagemSistema
                                           personagemSistema:
           for (PersonagemSistema
personagemSistema:
                                           personagensSistema) {
                                                       Button personagemButton =
personagensSistema) {
           Button personagemButton =
                                           new
                                           Button(personagemSistema.getPersonage
new
                                           m().getNome() + " - " +
Button(personagemSistema.getPersonage
m().getNome() + " - " +
                                           personagemSistema.getClasse().getNome
                                           Classe());
personagemSistema.getClasse().getNome
Classe());
                                           personagemButton.setFont(Font.font("Arial"
personagemButton.setFont(Font.font("Arial"
                                           , 16));
```

```
, 16));
                                            personagemButton.setTextFill(Color.WHITE
personagemButton.setTextFill(Color.WHITE
                                            );
);
                                            personagemButton.setStyle("-fx-backgroun
                                            d-color: #007ACC; -fx-background-radius:
personagemButton.setStyle("-fx-backgroun
d-color: #007ACC; -fx-background-radius:
                                            8;");
8;");
                                            personagemButton.setOnAction(e -> {
personagemButton.setOnAction(e -> {
                                                           Stage
              Stage
                                            stageEditarPersonagem = new Stage();
stageEditarPersonagem = new Stage();
                                            stageEditarPersonagem.setTitle("Editar
stageEditarPersonagem.setTitle("Editar
                                            Personagem");
Personagem");
                                                           GridPane grid = new
              GridPane grid = new
                                            GridPane();
GridPane();
                                                           grid.setPadding(new
              grid.setPadding(new
                                            Insets(20));
Insets(20));
                                                           grid.setHgap(10);
              grid.setHgap(10);
                                                           grid.setVgap(10);
              grid.setVgap(10);
                                            grid.setStyle("-fx-background-color:
grid.setStyle("-fx-background-color:
                                            #2B2B2B;");
#2B2B2B;");
                                                           Label titulo = new
              Label titulo = new
                                            Label("Editar Personagem");
Label("Editar Personagem");
                                            titulo.setFont(Font.font("Arial", 18));
titulo.setFont(Font.font("Arial", 18));
                                            titulo.setTextFill(Color.WHITE);
titulo.setTextFill(Color.WHITE);
                                            GridPane.setColumnSpan(titulo, 2);
GridPane.setColumnSpan(titulo, 2);
                                                           TextField campoNome =
              TextField campoNome =
                                            TextField(personagemSistema.getPersonag
new
TextField(personagemSistema.getPersonag
                                            em().getNome());
em().getNome());
                                                           ComboBox<String>
```

campoClasse.setValue(personagemSistem
a.getClasse().getNomeClasse());

TextField campoInfo = new TextField(personagemSistema.getPersonag em().getDescricao());

TextField campolmagem =

new

TextField(personagemSistema.getPersonagem().getUrlImg());

TextField campoForca = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getForca()));

TextField campoDestreza =

new

TextField(String.valueOf(personagemSiste ma.getPersonagem().getDestreza()));

TextField campoInteligencia

= new

TextField(String.valueOf(personagemSiste ma.getPersonagem().getInteligencia()));

TextField campoConstituicao

= new

TextField(String.valueOf(personagemSiste ma.getPersonagem().getConstituicao()));

TextField campoSabedoria =

campoRaca.getItems().addAll("Humano", "Elfo", "Anão", "Orc");

campoClasse.getItems().addAll("Guerreiro",
"Mago", "Arqueiro", "Ladino");

campoRaca.setValue(personagemSistema.getRaca().getNomeRaca());

campoClasse.setValue(personagemSistem a.getClasse().getNomeClasse());

TextField campoInfo = new TextField(personagemSistema.getPersonag em().getDescricao());

TextField campolmagem =

new

TextField(personagemSistema.getPersonagem().getUrlImg());

TextField campoForca = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getForca()));

TextField campoDestreza =

new

TextField(String.valueOf(personagemSiste ma.getPersonagem().getDestreza()));

TextField campoInteligencia

= new

TextField(String.valueOf(personagemSiste ma.getPersonagem().getInteligencia()));

TextField campoConstituicao

= new

TextField(String.valueOf(personagemSiste ma.getPersonagem().getConstituicao()));

TextField campoSabedoria =

new

new TextField(String.valueOf(personagemSiste TextField(String.valueOf(personagemSiste ma.getPersonagem().getSabedoria())); ma.getPersonagem().getSabedoria())); TextField campoCarisma = TextField campoCarisma = new new TextField(String.valueOf(personagemSiste TextField(String.valueOf(personagemSiste ma.getPersonagem().getCarisma())); ma.getPersonagem().getCarisma())); Button Button btnSelecionarImagem = new btnSelecionarImagem = new Button("Selecionar Imagem"); Button("Selecionar Imagem"); btnSelecionarImagem.setFont(Font.font("Ar ial", 14)); btnSelecionarImagem.setFont(Font.font("Ar ial", 14)); btnSelecionarImagem.setTextFill(Color.WHI btnSelecionarImagem.setTextFill(Color.WHI TE); TE); btnSelecionarImagem.setStyle("-fx-backgro btnSelecionarImagem.setStyle("-fx-backgro und-color: #007ACC; und-color: #007ACC; -fx-background-radius: 8;"); -fx-background-radius: 8;"); btnSelecionarImagem.setOnAction(ev -> { FileChooser fileChooser = btnSelecionarImagem.setOnAction(ev -> { FileChooser fileChooser = new FileChooser(): new FileChooser(); fileChooser.setTitle("Selecionar Imagem"); fileChooser.setTitle("Selecionar Imagem"); fileChooser.getExtensionFilters().addAll(fileChooser.getExtensionFilters().addAll(new FileChooser.ExtensionFilter("Imagens", new "*.png", "*.jpg", "*.jpeg", "*.gif") FileChooser.ExtensionFilter("Imagens", "*.png", "*.jpg", "*.jpeg", "*.gif") File selectedFile = File selectedFile = fileChooser.showOpenDialog(stageEditarPe fileChooser.showOpenDialog(stageEditarPe rsonagem); rsonagem); campolmagem.setText(selectedFile.getAbs campolmagem.setText(selectedFile.getAbs olutePath()); olutePath()); **})**;

```
});
                                              grid.add(btnSelecionarlmagem, 2, 5);
grid.add(btnSelecionarlmagem, 2, 5);
                                                             Button btnSalvar = new
                                              Button("Salvar Alterações");
               Button btnSalvar = new
Button("Salvar Alterações");
                                              btnSalvar.setFont(Font.font("Arial", 16));
btnSalvar.setFont(Font.font("Arial", 16));
                                              btnSalvar.setTextFill(Color.WHITE);
btnSalvar.setTextFill(Color.WHITE);
                                              btnSalvar.setStyle("-fx-background-color:
btnSalvar.setStyle("-fx-background-color:
                                              #007ACC; -fx-background-radius: 8;");
#007ACC; -fx-background-radius: 8;");
                                                             btnSalvar.setOnAction(ev ->
              btnSalvar.setOnAction(ev ->
                                              {
{
                                                             try {
              try {
                                              (campoForca.getText().isEmpty() ||
(campoForca.getText().isEmpty() ||
                                              campoDestreza.getText().isEmpty() ||
campoDestreza.getText().isEmpty() ||
                                              campoInteligencia.getText().isEmpty() ||
campoInteligencia.getText().isEmpty() ||
                                              campoConstituicao.getText().isEmpty() ||
campoConstituicao.getText().isEmpty() ||
                                              campoSabedoria.getText().isEmpty() ||
campoSabedoria.getText().isEmpty() ||
                                              campoCarisma.getText().isEmpty()) {
campoCarisma.getText().isEmpty()) {
                                                               throw new
                                              IllegalArgumentException("Todos os
                 throw new
IllegalArgumentException("Todos os
                                              campos de atributos devem ser
campos de atributos devem ser
                                              preenchidos.");
preenchidos.");
                 }
                                              (Integer.parseInt(campoForca.getText()) >
(Integer.parseInt(campoForca.getText()) >
                                              20 ||
20 ||
                                              Integer.parseInt(campoDestreza.getText())
Integer.parseInt(campoDestreza.getText())
                                              > 20 ||
                                              Integer.parseInt(campoInteligencia.getText()
> 20 ||
Integer.parseInt(campoInteligencia.getText()
                                             ) > 20 ||
) > 20 ||
                                              Integer.parseInt(campoConstituicao.getText
Integer.parseInt(campoConstituicao.getText
                                              ()) > 20 ||
()) > 20 ||
                                              Integer.parseInt(campoSabedoria.getText())
Integer.parseInt(campoSabedoria.getText())
                                              > 20 ||
```

```
> 20 ||
                                            Integer.parseInt(campoCarisma.getText()) >
Integer.parseInt(campoCarisma.getText()) >
                                            20) {
20) {
                                            AlertUtil.exibirAlerta("Valores inválidos", "Os
AlertUtil.exibirAlerta("Valores inválidos", "Os
                                            atributos devem não podem ser maiores
atributos devem não podem ser maiores
                                            que 20.");
que 20.");
                                                             throw new
                throw new
                                            IllegalArgumentException("Os atributos não
IllegalArgumentException("Os atributos não
                                            podem ser maiores que 20.");
podem ser maiores que 20.");
                }
                                                             Personagem
                                            personagemNovo = new Personagem(
                Personagem
personagemNovo = new Personagem(
                                            personagemSistema.getPersonagem().getP
personagemSistema.getPersonagem().getP
                                            ersonagemId(),
                                                             campoNome.getText(),
ersonagemld(),
                campoNome.getText(),
                                                             campoInfo.getText(),
                campoInfo.getText(),
                                                             campolmagem.getText(),
                campolmagem.getText(),
                                            personagemSistema.getPersonagem().get
personagemSistema.getPersonagem().get
                                            Nivel(),
Nivel(),
                                            Integer.parseInt(campoForca.getText()),
Integer.parseInt(campoForca.getText()),
                                            Integer.parseInt(campoDestreza.getText()),
Integer.parseInt(campoDestreza.getText()),
                                            Integer.parseInt(campoInteligencia.getText()
Integer.parseInt(campoInteligencia.getText()
                                            ),
),
                                            Integer.parseInt(campoConstituicao.getText
Integer.parseInt(campoConstituicao.getText
                                            ()),
()),
                                            Integer.parseInt(campoSabedoria.getText())
Integer.parseInt(campoSabedoria.getText())
                                            Integer.parseInt(campoCarisma.getText())
Integer.parseInt(campoCarisma.getText())
                                                             );
                );
```

```
PersonagemDBDAO
               PersonagemDBDAO
                                         personagemDB = new
personagemDB = new
                                         PersonagemDBDAO();
PersonagemDBDAO();
                                         personagemDB.atualizar(personagemNovo
personagemDB.atualizar(personagemNovo
);
                                                         ClasseDBDAO classeDB
               ClasseDBDAO classeDB
                                         = new ClasseDBDAO();
= new ClasseDBDAO();
                                                         RacaDBDAO racaDB =
               RacaDBDAO racaDB =
                                         new RacaDBDAO():
new RacaDBDAO();
                                                         Classe classeNova =
               Classe classeNova =
                                         classeDB.buscaPorNome(campoClasse.get
classeDB.buscaPorNome(campoClasse.get
                                         Value());
                                                         Raca racaNova =
Value());
               Raca racaNova =
                                         racaDB.buscaPorNome(campoRaca.getVal
racaDB.buscaPorNome(campoRaca.getVal
                                         ue());
ue());
                                                         PersonagemSistema
               PersonagemSistema
                                         personagemSistemaNovo = new
personagemSistemaNovo = new
                                         PersonagemSistema(
PersonagemSistema(
                                                         personagemNovo,
               personagemNovo,
                                         personagemSistema.getSistema(),
personagemSistema.getSistema(),
                                                         racaNova,
               racaNova.
                                                         classeNova
               classeNova
                                                         );
               );
                                         personagemSistemaDB.atualizar(personag
personagemSistemaDB.atualizar(personag
                                         emSistema, personagemSistemaNovo);
emSistema, personagemSistemaNovo);
                                         stageEditarPersonagem.close();
stageEditarPersonagem.close();
                                                         stageEditar.close();
               stageEditar.close();
                                         exibirTelaEditarPersonagens();
exibirTelaEditarPersonagens();
                                                      } catch
                                         (NumberFormatException ex) {
             } catch
(NumberFormatException ex) {
                                                         AlertUtil.exibirAlerta("Erro
               AlertUtil.exibirAlerta("Erro
                                         de Formato", "Os atributos devem ser
```

```
de Formato", "Os atributos devem ser
                                              números inteiros.");
números inteiros.");
                                                             } catch (SQLException ex) {
               } catch (SQLException ex) {
                                                                AlertUtil.exibirAlerta("Erro
                 AlertUtil.exibirAlerta("Erro
                                              de Banco de Dados", "Erro ao atualizar
de Banco de Dados", "Erro ao atualizar
                                              personagem no banco de dados.");
personagem no banco de dados.");
                                                             });
               });
                                                              Label nomeLabel = new
               Label nomeLabel = new
                                              Label("Nome:");
Label("Nome:");
                                              nomeLabel.setStyle("-fx-text-fill: white;");
nomeLabel.setStyle("-fx-text-fill: white;");
                                                              grid.add(nomeLabel, 0, 1);
               grid.add(nomeLabel, 0, 1);
                                                              Label classeLabel = new
               Label classeLabel = new
                                              Label("Classe:");
Label("Classe:");
                                              classeLabel.setStyle("-fx-text-fill: white;");
classeLabel.setStyle("-fx-text-fill: white;");
                                                              grid.add(classeLabel, 0, 2);
               grid.add(classeLabel, 0, 2);
                                                              Label racaLabel = new
               Label racaLabel = new
                                              Label("Raça:");
Label("Raça:");
                                              racaLabel.setStyle("-fx-text-fill: white;");
racaLabel.setStyle("-fx-text-fill: white;");
                                                              grid.add(racaLabel, 0, 3);
               grid.add(racaLabel, 0, 3);
                                                              Label descricaoLabel = new
               Label descricaoLabel = new
                                              Label("Descrição:");
Label("Descrição:");
                                              descricaoLabel.setStyle("-fx-text-fill:
descricaoLabel.setStyle("-fx-text-fill:
                                              white;");
white;");
                                                              grid.add(descricaoLabel, 0,
               grid.add(descricaoLabel, 0,
                                              4);
4);
                                                              Label caminholmagemLabel
               Label caminholmagemLabel
                                              = new Label("Caminho da Imagem:");
= new Label("Caminho da Imagem:");
                                              caminholmagemLabel.setStyle("-fx-text-fill:
caminholmagemLabel.setStyle("-fx-text-fill:
                                              white;");
```

```
white;");
                                                grid.add(caminholmagemLabel, 0, 5);
grid.add(caminholmagemLabel, 0, 5);
                                                                Label forcaLabel = new
               Label forcaLabel = new
                                                Label("Força:");
Label("Força:");
                                                forcaLabel.setStyle("-fx-text-fill: white;");
forcaLabel.setStyle("-fx-text-fill: white;");
                                                                grid.add(forcaLabel, 0, 6);
               grid.add(forcaLabel, 0, 6);
                                                                Label destrezaLabel = new
               Label destrezaLabel = new
                                                Label("Destreza:");
Label("Destreza:");
                                                destrezaLabel.setStyle("-fx-text-fill: white;");
destrezaLabel.setStyle("-fx-text-fill: white;");
                                                                grid.add(destrezaLabel, 0,
               grid.add(destrezaLabel, 0,
                                                7);
7);
                                                                Label constituicaoLabel =
               Label constituicaoLabel =
                                                new Label("Constituição:");
new Label("Constituição:");
                                                constituicaoLabel.setStyle("-fx-text-fill:
constituicaoLabel.setStyle("-fx-text-fill:
                                                white;");
white;");
                                                                grid.add(constituicaoLabel,
               grid.add(constituicaoLabel,
                                                0, 8);
0, 8);
                                                                Label inteligenciaLabel =
               Label inteligenciaLabel =
                                                new Label("Inteligência:");
new Label("Inteligência:");
                                                inteligenciaLabel.setStyle("-fx-text-fill:
inteligenciaLabel.setStyle("-fx-text-fill:
                                                white;");
white;");
                                                                grid.add(inteligenciaLabel, 0,
                                                9);
               grid.add(inteligenciaLabel, 0,
9);
                                                                Label sabedoriaLabel = new
                                                Label("Sabedoria:");
               Label sabedoriaLabel = new
Label("Sabedoria:");
                                                sabedoriaLabel.setStyle("-fx-text-fill:
                                                white;");
sabedoriaLabel.setStyle("-fx-text-fill:
white;");
                                                                grid.add(sabedoriaLabel, 0,
                                                10);
               grid.add(sabedoriaLabel, 0,
```

```
10);
                                                            Label carismaLabel = new
              Label carismaLabel = new
                                             Label("Carisma:");
Label("Carisma:");
                                             carismaLabel.setStyle("-fx-text-fill: white;");
carismaLabel.setStyle("-fx-text-fill: white;");
                                                            grid.add(carismaLabel, 0,
              grid.add(carismaLabel, 0,
                                             11);
11);
                                                            grid.add(campoNome, 1, 1);
              grid.add(campoNome, 1, 1);
                                                            grid.add(campoClasse, 1, 2);
              grid.add(campoClasse, 1, 2);
                                                            grid.add(campoRaca, 1, 3);
              grid.add(campoRaca, 1, 3);
                                                            grid.add(campoInfo, 1, 4);
              grid.add(campoInfo, 1, 4);
                                                            grid.add(campolmagem, 1,
              grid.add(campolmagem, 1,
                                             5);
5);
                                                            grid.add(campoForca, 1, 6);
              grid.add(campoForca, 1, 6);
                                                            grid.add(campoDestreza, 1,
              grid.add(campoDestreza, 1,
                                             7);
7);
                                                            grid.add(campoConstituicao,
              grid.add(campoConstituicao,
                                             1, 8);
1, 8);
                                                            grid.add(campoInteligencia,
              grid.add(campoInteligencia,
                                             1, 9);
1, 9);
                                                            grid.add(campoSabedoria,
              grid.add(campoSabedoria,
                                             1, 10);
1, 10);
                                                            grid.add(campoCarisma, 1,
              grid.add(campoCarisma, 1,
                                             11);
11);
                                                            grid.add(btnSalvar, 1, 12);
              grid.add(btnSalvar, 1, 12);
                                                            Scene cena = new
              Scene cena = new
                                             Scene(grid, 400, 500);
Scene(grid, 400, 500);
                                             stageEditarPersonagem.setScene(cena);
stageEditarPersonagem.setScene(cena);
                                             stageEditarPersonagem.show();
stageEditarPersonagem.show();
                                                         });
            });
                                             vbox.getChildren().add(personagemButton);
vbox.getChildren().add(personagemButton);
                                                         }
            }
```

```
Scene cena = new
Scene(vbox, 400, 300);
StageEditar.setScene(cena);
stageEditar.show();
}

Scene cena = new
Scene(vbox, 400, 300);
stageEditar.setScene(cena);
stageEditar.show();
}
```

```
Nº refatoração: 12
                       Classe: MenusController
                                                                  Nome: Extract Method
Código antes de refatorar
                                              Código após refatorar
MenusController.java
                                              exibirDefinir.java
                                              public void exibirTelaDefinirCaracteristicas()
private void
exibirTelaDefinirCaracteristicas() {
       Stage stageCaracteristicas = new
                                                     Stage stageCaracteristicas = new
Stage();
                                              Stage();
       stageCaracteristicas.setTitle("Definir
                                                     stageCaracteristicas.setTitle("Definir
Características");
                                              Características");
       GridPane grid = new GridPane();
                                                     GridPane grid = new GridPane();
       grid.setPadding(new Insets(20));
                                                     grid.setPadding(new Insets(20));
       grid.setHgap(10);
                                                     grid.setHgap(10);
       grid.setVgap(10);
                                                     grid.setVgap(10);
       grid.setStyle("-fx-background-color:
                                                     grid.setStyle("-fx-background-color:
                                              #2B2B2B;");
#2B2B2B;");
       Label titulo = new Label("Definir
                                                     Label titulo = new Label("Definir
Características do Personagem");
                                              Características do Personagem");
       titulo.setFont(Font.font("Arial", 18));
                                                     titulo.setFont(Font.font("Arial", 18));
       titulo.setTextFill(Color.WHITE);
                                                     titulo.setTextFill(Color.WHITE);
       GridPane.setColumnSpan(titulo, 2);
                                                     GridPane.setColumnSpan(titulo, 2);
       Label lblPersonagemId = new
                                                     Label lblPersonagemId = new
Label("ID do Personagem:");
                                              Label("ID do Personagem:");
lblPersonagemId.setStyle("-fx-text-fill:
                                              lblPersonagemId.setStyle("-fx-text-fill:
#D3D3D3;");
                                              #D3D3D3;");
       TextField campoPersonagemId =
                                                     TextField campoPersonagemId =
new TextField();
                                              new TextField();
```

```
Label IblNome = new
                                                    Label IblNome = new
                                             Label("Nome:");
Label("Nome:");
       lblNome.setStyle("-fx-text-fill:
                                                    lblNome.setStyle("-fx-text-fill:
#D3D3D3;");
                                             #D3D3D3;");
       TextField campoNome = new
                                                    TextField campoNome = new
TextField():
                                             TextField():
       Label lblClasse = new
                                                    Label lblClasse = new
                                             Label("Classe:");
Label("Classe:");
       lblClasse.setStyle("-fx-text-fill:
                                                    lblClasse.setStyle("-fx-text-fill:
#D3D3D3;");
                                             #D3D3D3;");
       ComboBox<String> campoClasse =
                                                    ComboBox<String> campoClasse =
new ComboBox<>();
                                             new ComboBox<>();
campoClasse.getItems().addAll("Guerreiro",
                                             campoClasse.getItems().addAll("Guerreiro",
"Mago", "Arqueiro", "Ladino");
                                             "Mago", "Arqueiro", "Ladino");
       Label lblRaca = new Label("Raça:");
                                                    Label lblRaca = new Label("Raça:");
       lblRaca.setStyle("-fx-text-fill:
                                                    lblRaca.setStyle("-fx-text-fill:
#D3D3D3;");
                                             #D3D3D3;");
       ComboBox<String> campoRaca =
                                                    ComboBox<String> campoRaca =
new ComboBox<>();
                                             new ComboBox<>();
campoRaca.getItems().addAll("Humano",
                                             campoRaca.getItems().addAll("Humano",
"Elfo", "Anão", "Orc");
                                             "Elfo", "Anão", "Orc");
       Label IblDescricao = new
                                                    Label IblDescricao = new
Label("Descrição:");
                                             Label("Descrição:");
       lblDescricao.setStyle("-fx-text-fill:
                                                    lblDescricao.setStyle("-fx-text-fill:
#D3D3D3;");
                                             #D3D3D3;");
       TextField campoInfo = new
                                                    TextField campoInfo = new
TextField();
                                             TextField();
       Label lbllmagem = new
                                                    Label lbllmagem = new
Label("Caminho da Imagem:");
                                             Label("Caminho da Imagem:");
       lbllmagem.setStyle("-fx-text-fill:
                                                    lbllmagem.setStyle("-fx-text-fill:
#D3D3D3;");
                                             #D3D3D3;");
       TextField campolmagem = new
                                                    TextField campolmagem = new
                                             TextField();
TextField();
```

```
Button btnSelecionarImagem = new
                                                     Button btnSelecionarImagem = new
Button("Selecionar Imagem");
                                              Button("Selecionar Imagem");
btnSelecionarImagem.setFont(Font.font("Ar
                                              btnSelecionarImagem.setFont(Font.font("Ar
ial", 14));
                                              ial", 14));
btnSelecionarImagem.setTextFill(Color.WHI
                                              btnSelecionarImagem.setTextFill(Color.WHI
TE);
                                              TE);
btnSelecionarImagem.setStyle("-fx-backgro
                                              btnSelecionarImagem.setStyle("-fx-backgro
und-color: #007ACC;
                                              und-color: #007ACC;
-fx-background-radius: 8;");
                                              -fx-background-radius: 8;");
btnSelecionarImagem.setOnAction(e -> {
                                              btnSelecionarImagem.setOnAction(e -> {
         FileChooser fileChooser = new
                                                       FileChooser fileChooser = new
FileChooser();
                                              FileChooser();
         fileChooser.setTitle("Selecionar
                                                       fileChooser.setTitle("Selecionar
Imagem");
                                              Imagem");
fileChooser.getExtensionFilters().addAll(
                                              fileChooser.getExtensionFilters().addAll(
            new
                                                          new
FileChooser.ExtensionFilter("Imagens",
                                              FileChooser.ExtensionFilter("Imagens",
"*.png", "*.jpg", "*.jpeg", "*.gif")
                                              "*.png", "*.jpg", "*.jpeg", "*.gif")
         );
                                                       );
         File selectedFile =
                                                       File selectedFile =
fileChooser.showOpenDialog(stageCaracter
                                             fileChooser.showOpenDialog(stageCaracter
isticas);
                                              isticas);
         if (selectedFile != null) {
                                                       if (selectedFile != null) {
campolmagem.setText(selectedFile.getAbs
                                              campolmagem.setText(selectedFile.getAbs
olutePath());
                                              olutePath());
         }
                                                       }
       });
                                                     });
       grid.add(btnSelecionarImagem, 2,
                                                     grid.add(btnSelecionarlmagem, 2,
6);
                                              6);
                                                     defAtributos
       Button btnProximo = new
Button("Próximo: Definir Atributos");
                                              exibirTelaDefinirAtributos = new
```

```
defAtributos();
btnProximo.setFont(Font.font("Arial", 16));
                                                     Button btnProximo = new
                                              Button("Próximo: Definir Atributos");
btnProximo.setTextFill(Color.WHITE);
                                              btnProximo.setFont(Font.font("Arial", 16));
btnProximo.setStyle("-fx-background-color:
#007ACC; -fx-background-radius: 8;");
                                              btnProximo.setTextFill(Color.WHITE);
       btnProximo.setOnAction(e ->
exibirTelaDefinirAtributos(
                                              btnProximo.setStyle("-fx-background-color:
                                              #007ACC; -fx-background-radius: 8;");
Integer.parseInt(campoPersonagemId.getTe
                                                     btnProximo.setOnAction(e ->
                                              exibirTelaDefinirAtributos.exibirTelaDefinirAt
xt()),
          campoNome.getText(),
                                              ributos(
          campoClasse.getValue(),
                                              Integer.parseInt(campoPersonagemId.getTe
          campoRaca.getValue(),
          campoInfo.getText(),
                                              xt()),
          campolmagem.getText(),
                                                        campoNome.getText(),
          stageCaracteristicas
                                                        campoClasse.getValue(),
       ));
                                                        campoRaca.getValue(),
                                                        campoInfo.getText(),
       grid.add(titulo, 0, 0);
                                                       campolmagem.getText(),
       grid.add(lblPersonagemId, 0, 1);
                                                       stageCaracteristicas
       grid.add(campoPersonagemId, 1,
                                                     ));
1);
       grid.add(lblNome, 0, 2);
                                                     grid.add(titulo, 0, 0);
                                                     grid.add(lblPersonagemId, 0, 1);
       grid.add(campoNome, 1, 2);
       grid.add(lblClasse, 0, 3);
                                                     grid.add(campoPersonagemId, 1,
       grid.add(campoClasse, 1, 3);
                                              1);
       grid.add(lblRaca, 0, 4);
                                                     grid.add(lblNome, 0, 2);
       grid.add(campoRaca, 1, 4);
                                                     grid.add(campoNome, 1, 2);
       grid.add(lblDescricao, 0, 5);
                                                     grid.add(lblClasse, 0, 3);
       grid.add(campoInfo, 1, 5);
                                                     grid.add(campoClasse, 1, 3);
       grid.add(lbllmagem, 0, 6);
                                                     grid.add(lblRaca, 0, 4);
       grid.add(campolmagem, 1, 6);
                                                     grid.add(campoRaca, 1, 4);
       grid.add(btnProximo, 1, 7);
                                                     grid.add(lblDescricao, 0, 5);
                                                     grid.add(campoInfo, 1, 5);
       Scene cena = new Scene(grid, 400,
                                                     grid.add(lbllmagem, 0, 6);
350);
                                                     grid.add(campolmagem, 1, 6);
                                                     grid.add(btnProximo, 1, 7);
```

```
stageCaracteristicas.setScene(cena);
stageCaracteristicas.show();
}

Scene cena = new Scene(grid, 400, 350);

stageCaracteristicas.setScene(cena);
stageCaracteristicas.setScene(cena);
stageCaracteristicas.show();
}
```