

Trabalho II - Programação Orientada A Objetos II

Alunos: Arthur Carboni Linzing, Mateus de Oliveira Lopes e Yan Gabriel Reis Oliveira

Nº refatoração: 1	Classe: MenuController.java	Nome: Extract Class
Código antes de refatorar	Código após refatorar	
<pre> MenuController.java ... private int testeAtributo(int atributo) {     return (int)(Math.random() * 20) + 1 +     atributo; }  private int rodarAtributos() {     int menor = 7;     int atual;     int soma = 0;     for(int i = 0; i &lt; 4; i++)     {         atual = (int)(Math.random() * 6) + 1;         soma = soma + atual;         if(menor &gt; atual){             menor = atual;         }     }      return soma - menor; } ... </pre>	<pre> RolagemDados.java ... public class RolagemDados {     public int testeAtributo(int atributo)     {         return (int)(Math.random() * 20) + 1 +         atributo;     }      public int rodarAtributos()     {         int menor = 7;         int atual;         int soma = 0;         for(int i = 0; i &lt; 4; i++)         {             atual = (int)(Math.random() * 6) + 1;             soma = soma + atual;             if(menor &gt; atual){                 menor = atual;             }         }          return soma - menor;     } } </pre>	

Nº refatoração: 2	Classe: Main.java	Nome: Dead Code
Código antes de refatorar	Código após refatorar	
<pre> Main.java public class Main {     public static void main(String[] args)     throws SQLException{         Personagem personagem = new         Personagem(1,"Singus","The         GOAT","Imagem Legal",17,-1,3,4,7,2,-2);         Personagem personagem1 = new         Personagem(2,"Degniss","The         B","Imagem",12,-1,3,4,7,2,-2);         PersonagemDBDAO personagemDAO         = new PersonagemDBDAO(); </pre>	<pre> Main.java public class Main {     public static void main(String[] args){         /*2º Refatoração         Autor: Mateus de Oliveira Lopes         Refatoração geral para remover         códigos de teste não usados no main no         código final         Objetivo: Remover códigos não         usados         */     } } </pre>	

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

Nº refatoração: 3	Classe: PersonagemSistemaDBDAO.java e PersonagemSistemaDAO.java	Nome: Dead Code
Código antes de refatorar		Código após refatorar
PersonagemSistemaDAO.java ... public interface PersonagemSistemaDAO { void insere(PersonagemSistema var1) throws SQLException; void atualizar(PersonagemSistema varNova, PersonagemSistema varVelha) throws SQLException; void remover(PersonagemSistema var1) throws SQLException; PersonagemSistema buscaPorCodigo(int personagemId, int classId, int sistemId, int racId) throws SQLException; List<PersonagemSistema> listar() throws SQLException; }  PersonagemSistemaDBDAO.java ... public void remover(PersonagemSistema personagemSistema) throws SQLException {...}  public PersonagemSistema buscaPorCodigo(int personagemId, int classId, int sistemId, int racId) throws SQLException {...}  public List<PersonagemSistema> listar() throws SQLException {...}		PersonagemSistemaDAO.java ... public interface PersonagemSistemaDAO { void insere(PersonagemSistema var1) throws SQLException; void atualizar(PersonagemSistema varNova, PersonagemSistema varVelha) throws SQLException; void remover(PersonagemSistema var1) throws SQLException; List<PersonagemSistema> listar() throws SQLException; }  PersonagemSistemaDBDAO.java ... public void remover(PersonagemSistema personagemSistema) throws SQLException {...}  /*3º Refatoração Autor: Mateus de Oliveira Lopes Refatoração geral para remover a função busca por código que nunca é utilizada por PersonagemSistemaDBDAO Objetivo: Remover funções não usadas */  public List<PersonagemSistema> listar() throws SQLException {...}

Nº refatoração: 4	Classe: PersonagemSistemaDBDAO.java	Nome: Private Variable to Local Variable
Código antes de refatorar		Código após refatorar
PersonagemSistemaDBDAO.java ... private ResultSet result; ... public List<PersonagemSistema> listar() throws SQLException { open(); sql = "SELECT * FROM personagem_sistema"; statement = connection.prepareStatement(sql); result = statement.executeQuery(); ...		PersonagemSistemaDBDAO.java ... public List<PersonagemSistema> listar() throws SQLException { open(); sql = "SELECT * FROM personagem_sistema"; statement = connection.prepareStatement(sql); ResultSet result = statement.executeQuery(); ...

Nº refatoração: 5	Classe: MenuController.java	Nome: Extract Method
Código antes de refatorar		Código após refatorar
MenuController.java  public static void exibirAlerta(String titulo, String mensagem) { Alert alerta = new Alert(Alert.AlertType.ERROR); alerta.setTitle(titulo); alerta.setHeaderText(null); alerta.setContentText(mensagem); alerta.showAndWait(); }  //(para chamar em MenuController.java exibirAlerta("Erro de Banco de Dados", "Erro ao atualizar personagem no banco de dados.");		AlertUtil.java  public static void exibirAlerta(String titulo, String mensagem) { Alert alerta = new Alert(Alert.AlertType.ERROR); alerta.setTitle(titulo); alerta.setHeaderText(null); alerta.setContentText(mensagem); alerta.showAndWait(); } }  //(para chamar em MenuController.java AlertUtil.exibirAlerta("Erro de Banco de Dados", "Erro ao atualizar personagem no banco de dados.");

Nº refatoração: 6	Classe: MenuController.java	Nome: Tornando método estático
Código antes de refatorar		Código após refatorar

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

Nº refatoração: 7	Classe:MenusController.java	Nome: Extract Method
Código antes de refatorar	Código após refatorar	
MenusController.java  <pre>public static void exibirResultadoTeste(String atributo, int resultado) {     Alert alerta = new Alert(Alert.AlertType.INFORMATION);     alerta.setTitle("Resultado do Teste");     alerta.setHeaderText(null);     alerta.setContentText("O resultado do teste de " + atributo + " foi: " + resultado);     alerta.showAndWait(); }  exibirResultadoTeste(</pre>	ResultadoTesteView.java  <pre>public class ResultadoTesteView {     public static void exibirResultadoTeste(String atributo, int resultado) {         Alert alerta = new Alert(Alert.AlertType.INFORMATION);         alerta.setTitle("Resultado do Teste");         alerta.setHeaderText(null);         alerta.setContentText("O resultado do teste de " + atributo + " foi: " + resultado);         alerta.showAndWait();     } }  ResultadoTesteView.exibirResultadoTeste(</pre>	

Nº refatoração: 8	Classe: MenuController	Nome: Retirar repetição
Código antes de refatorar	Código após refatorar	
<pre>Button btnForcaAleatorio = new Button("🎲"); btnForcaAleatorio.setOnAction(e -&gt; campoForca.setText(String.valueOf(dado.ro darAtributos()))); grid.add(btnForcaAleatorio, 2, 1);</pre>	<pre>for (int i = 0; i &lt; atributos.length; i++) {     Label label = new Label(atributos[i] + ":");     label.setStyle("-fx-text-fill: #D3D3D3;");     TextField campo = new TextField();</pre>	

...	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: MenuController	Nome: Extract Method
Código antes de refatorar	Código após refatorar	
MenusController.java  private Button criarBotaoAleatorio(TextField campo) { Button btnAleatorio = new Button("🎲"); btnAleatorio.setOnAction(e -> campo.setText(String.valueOf(dado.rodarAtributos()))); return btnAleatorio; }	defAtributos.java  private Button criarBotaoAleatorio(TextField campo) { RolagemDados dado2 = new RolagemDados(); Button btnAleatorio = new Button("🎲"); btnAleatorio.setOnAction(e -> campo.setText(String.valueOf(dado2.rodarAtributos()))); return btnAleatorio; }	

Nº refatoração: 10	Classe: MenuController	Nome: Extract Method
Código antes de refatorar	Código após refatorar	
MenusController.java private void exibirTelaDefinirAtributos(int personagemId, String nome, String classe, String raca, String info, String caminhoImagem, Stage telaAnterior) { String[] atributos = {"Força", "Destreza", "Constituição", "Inteligência", "Sabedoria", "Carisma"}; TextField[] campos = new TextField[atributos.length]; telaAnterior.close(); Stage stageAtributos = new Stage(); stageAtributos.setTitle("Definir	defAtributos.java public void exibirTelaDefinirAtributos(int personagemId, String nome, String classe, String raca, String info, String caminhoImagem, Stage telaAnterior) { String[] atributos = {"Força", "Destreza", "Constituição", "Inteligência", "Sabedoria", "Carisma"}; TextField[] campos = new TextField[atributos.length]; telaAnterior.close(); Stage stageAtributos = new Stage();	

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

```

#D3D3D3;");
        TextField campoInteligencia = new
TextField();

        Label lblSabedoria = new
Label("Sabedoria:");
        lblSabedoria.setStyle("-fx-text-fill:
#D3D3D3;");
        TextField campoSabedoria = new
TextField();

        Label lblCarisma = new
Label("Carisma:");
        lblCarisma.setStyle("-fx-text-fill:
#D3D3D3;");
        TextField campoCarisma = new
TextField();

        for (int i = 0; i < atributos.length; i++)
{
            Label label = new
Label(atributos[i] + ":");
            label.setStyle("-fx-text-fill:
#D3D3D3;");
            TextField campo = new
TextField();
            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);
        }

        Button btnSalvar = new

```

```

        lblInteligencia.setStyle("-fx-text-fill:
#D3D3D3;");
        TextField campoInteligencia = new
TextField();

        Label lblSabedoria = new
Label("Sabedoria:");
        lblSabedoria.setStyle("-fx-text-fill:
#D3D3D3;");
        TextField campoSabedoria = new
TextField();

        Label lblCarisma = new
Label("Carisma:");
        lblCarisma.setStyle("-fx-text-fill:
#D3D3D3;");
        TextField campoCarisma = new
TextField();

        for (int i = 0; i < atributos.length; i++)
{
            Label label = new
Label(atributos[i] + ":");
            label.setStyle("-fx-text-fill:
#D3D3D3;");
            TextField campo = new
TextField();
            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);
        }

```

```

Button("Salvar Personagem");
    btnSalvar.setFont(Font.font("Arial",
16));
    btnSalvar.setTextFill(Color.WHITE);

btnSalvar.setStyle("-fx-background-color:
#007ACC; -fx-background-radius: 8;");
    btnSalvar.setOnAction(e -> {
        try {
            if
(campoForca.getText().isEmpty() ||
campoDestreza.getText().isEmpty() ||

campoInteligencia.getText().isEmpty() ||
campoConstituicao.getText().isEmpty() ||
campoSabedoria.getText().isEmpty() ||
campoCarisma.getText().isEmpty()) {
                throw new
IllegalArgumentException("Todos os
campos de atributos devem ser
preenchidos.");
            }
            if
(Integer.parseInt(campoForca.getText()) >
20 ||
Integer.parseInt(campoDestreza.getText())
> 20 ||
Integer.parseInt(campoInteligencia.getText()
) > 20 ||
Integer.parseInt(campoConstituicao.getText
()) > 20 ||
Integer.parseInt(campoSabedoria.getText())
> 20 ||
Integer.parseInt(campoCarisma.getText()) >
20) {
                throw new
IllegalArgumentException("Os atributos não
podem ser maiores que 20.");
            }

```

```

Button btnSalvar = new
Button("Salvar Personagem");
    btnSalvar.setFont(Font.font("Arial",
16));
    btnSalvar.setTextFill(Color.WHITE);

btnSalvar.setStyle("-fx-background-color:
#007ACC; -fx-background-radius: 8;");
    btnSalvar.setOnAction(e -> {
        try {
            if
(campoForca.getText().isEmpty() ||
campoDestreza.getText().isEmpty() ||

campoInteligencia.getText().isEmpty() ||
campoConstituicao.getText().isEmpty() ||
campoSabedoria.getText().isEmpty() ||
campoCarisma.getText().isEmpty()) {
                throw new
IllegalArgumentException("Todos os
campos de atributos devem ser
preenchidos.");
            }
            if
(Integer.parseInt(campoForca.getText()) >
20 ||
Integer.parseInt(campoDestreza.getText())
> 20 ||
Integer.parseInt(campoInteligencia.getText()
) > 20 ||
Integer.parseInt(campoConstituicao.getText
()) > 20 ||
Integer.parseInt(campoSabedoria.getText())
> 20 ||
Integer.parseInt(campoCarisma.getText()) >
20) {
                throw new
IllegalArgumentException("Os atributos não
podem ser maiores que 20.");
            }

```



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

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

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

Nº refatoração: 11	Classe: MenuController	Nome: Extract Method
Código antes de refatorar		Código após refatorar
<pre> MenuController.java private void exibirTelaEditarPersonagens() { </pre>		<pre> exibirEditar.java public void exibirTelaEditarPersonagens() {     Stage stageEditar = new </pre>

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

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

<pre> ComboBox&lt;String&gt; campoClasse = new ComboBox&lt;&gt;(); ComboBox&lt;String&gt; campoRaca = new ComboBox&lt;&gt;();  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 campolInfo = new TextField(personagemSistema.getPersonag em().getDescricao());  TextField campolmagem = new TextField(personagemSistema.getPersonag em().getUrlImg());  TextField campoForca = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getForca()));  TextField campoDestreza = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getDestreza()));  TextField campolInteligencia = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getInteligencia()));  TextField campoConstituicao = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getConstituicao()));  TextField campoSabedoria = </pre>	<pre> campoClasse = new ComboBox&lt;&gt;(); ComboBox&lt;String&gt; campoRaca = new ComboBox&lt;&gt;();  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 campolInfo = new TextField(personagemSistema.getPersonag em().getDescricao());  TextField campolmagem = new TextField(personagemSistema.getPersonag em().getUrlImg());  TextField campoForca = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getForca()));  TextField campoDestreza = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getDestreza()));  TextField campolInteligencia = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getInteligencia()));  TextField campoConstituicao = new TextField(String.valueOf(personagemSiste ma.getPersonagem().getConstituicao()));  TextField campoSabedoria = new </pre>
---	--

<pre> new TextField(String.valueOf(personagemSistema.getPersonagem().getSabedoria()));         TextField campoCarisma = new TextField(String.valueOf(personagemSistema.getPersonagem().getCarisma()));          Button btnSelecionarImagem = new Button("Selecionar Imagem");  btnSelecionarImagem.setFont(Font.font("Arial", 14));  btnSelecionarImagem.setTextFill(Color.WHITE);  btnSelecionarImagem.setStyle("-fx-background-color: #007ACC; -fx-background-radius: 8;");  btnSelecionarImagem.setOnAction(ev -&gt; {         FileChooser fileChooser = new FileChooser();  fileChooser.setTitle("Selecionar Imagem");  fileChooser.getExtensionFilters().addAll(         new FileChooser.ExtensionFilter("Imagens",         "*.png", "*.jpg", "*.jpeg", "*.gif")         );         File selectedFile = fileChooser.showOpenDialog(stageEditarPersonagem);  campoImagem.setText(selectedFile.getAbsolutePath()); </pre>	<pre> TextField(String.valueOf(personagemSistema.getPersonagem().getSabedoria()));         TextField campoCarisma = new TextField(String.valueOf(personagemSistema.getPersonagem().getCarisma()));          Button btnSelecionarImagem = new Button("Selecionar Imagem");  btnSelecionarImagem.setFont(Font.font("Arial", 14));  btnSelecionarImagem.setTextFill(Color.WHITE);  btnSelecionarImagem.setStyle("-fx-background-color: #007ACC; -fx-background-radius: 8;");  btnSelecionarImagem.setOnAction(ev -&gt; {         FileChooser fileChooser = new FileChooser();  fileChooser.setTitle("Selecionar Imagem");  fileChooser.getExtensionFilters().addAll(         new FileChooser.ExtensionFilter("Imagens",         "*.png", "*.jpg", "*.jpeg", "*.gif")         );         File selectedFile = fileChooser.showOpenDialog(stageEditarPersonagem);  campoImagem.setText(selectedFile.getAbsolutePath());         }); </pre>
--	--

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



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

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

```

de Formato", "Os atributos devem ser
números inteiros.");
        } catch (SQLException ex) {
            AlertUtil.exibirAlerta("Erro
de Banco de Dados", "Erro ao atualizar
personagem no banco de dados.");
        }
    });

    Label nomeLabel = new
Label("Nome:");

nomeLabel.setStyle("-fx-text-fill: white;");
grid.add(nomeLabel, 0, 1);

    Label classeLabel = new
Label("Classe:");

classeLabel.setStyle("-fx-text-fill: white;");
grid.add(classeLabel, 0, 2);

    Label racaLabel = new
Label("Raça:");

racaLabel.setStyle("-fx-text-fill: white;");
grid.add(racaLabel, 0, 3);

    Label descricaoLabel = new
Label("Descrição:");

descricaoLabel.setStyle("-fx-text-fill:
white;");
grid.add(descricaoLabel, 0,
4);

    Label caminholImagemLabel
= new Label("Caminho da Imagem:");

caminholImagemLabel.setStyle("-fx-text-fill:

```

```

números inteiros.");
        } catch (SQLException ex) {
            AlertUtil.exibirAlerta("Erro
de Banco de Dados", "Erro ao atualizar
personagem no banco de dados.");
        }
    });

    Label nomeLabel = new
Label("Nome:");

nomeLabel.setStyle("-fx-text-fill: white;");
grid.add(nomeLabel, 0, 1);

    Label classeLabel = new
Label("Classe:");

classeLabel.setStyle("-fx-text-fill: white;");
grid.add(classeLabel, 0, 2);

    Label racaLabel = new
Label("Raça:");

racaLabel.setStyle("-fx-text-fill: white;");
grid.add(racaLabel, 0, 3);

    Label descricaoLabel = new
Label("Descrição:");

descricaoLabel.setStyle("-fx-text-fill:
white;");
grid.add(descricaoLabel, 0,
4);

    Label caminholImagemLabel
= new Label("Caminho da Imagem:");

caminholImagemLabel.setStyle("-fx-text-fill:
white;");

```

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

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

<pre> Scene cena = new Scene(vbox, 400, 300); stageEditar.setScene(cena); stageEditar.show(); } </pre>	<pre> Scene cena = new Scene(vbox, 400, 300); stageEditar.setScene(cena); stageEditar.show(); } </pre>
--	--

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

<p>Label lblNome = new Label("Nome:");     lblNome.setStyle("-fx-text-fill: #D3D3D3;");     TextField campoNome = new TextField();</p> <p>Label lblClasse = new Label("Classe:");     lblClasse.setStyle("-fx-text-fill: #D3D3D3;");     ComboBox&lt;String&gt; campoClasse = new ComboBox&lt;&gt;();</p> <p>campoClasse.getItems().addAll("Guerreiro", "Mago", "Arqueiro", "Ladino");</p> <p>Label lblRaca = new Label("Raça:");     lblRaca.setStyle("-fx-text-fill: #D3D3D3;");     ComboBox&lt;String&gt; campoRaca = new ComboBox&lt;&gt;();</p> <p>campoRaca.getItems().addAll("Humano", "Elfo", "Anão", "Orc");</p> <p>Label lblDescricao = new Label("Descrição:");     lblDescricao.setStyle("-fx-text-fill: #D3D3D3;");     TextField campoInfo = new TextField();</p> <p>Label lblImagem = new Label("Caminho da Imagem:");     lblImagem.setStyle("-fx-text-fill: #D3D3D3;");     TextField campoImagem = new TextField();</p>	<p>Label lblNome = new Label("Nome:");     lblNome.setStyle("-fx-text-fill: #D3D3D3;");     TextField campoNome = new TextField();</p> <p>Label lblClasse = new Label("Classe:");     lblClasse.setStyle("-fx-text-fill: #D3D3D3;");     ComboBox&lt;String&gt; campoClasse = new ComboBox&lt;&gt;();</p> <p>campoClasse.getItems().addAll("Guerreiro", "Mago", "Arqueiro", "Ladino");</p> <p>Label lblRaca = new Label("Raça:");     lblRaca.setStyle("-fx-text-fill: #D3D3D3;");     ComboBox&lt;String&gt; campoRaca = new ComboBox&lt;&gt;();</p> <p>campoRaca.getItems().addAll("Humano", "Elfo", "Anão", "Orc");</p> <p>Label lblDescricao = new Label("Descrição:");     lblDescricao.setStyle("-fx-text-fill: #D3D3D3;");     TextField campoInfo = new TextField();</p> <p>Label lblImagem = new Label("Caminho da Imagem:");     lblImagem.setStyle("-fx-text-fill: #D3D3D3;");     TextField campoImagem = new TextField();</p>
--	--

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



```
btnProximo.setFont(Font.font("Arial", 16));
```

```
btnProximo.setTextFill(Color.WHITE);
```

```
btnProximo.setStyle("-fx-background-color:  
#007ACC; -fx-background-radius: 8;");
```

```
    btnProximo.setOnAction(e ->  
exibirTelaDefinirAtributos(
```

```
Integer.parseInt(campoPersonagemId.getTe  
xt()),
```

```
        campoNome.getText(),  
        campoClasse.getValue(),  
        campoRaca.getValue(),  
        campoInfo.getText(),  
        campolmagem.getText(),  
        stageCaracteristicas  
    ));
```

```
    grid.add(titulo, 0, 0);  
    grid.add(lblPersonagemId, 0, 1);  
    grid.add(campoPersonagemId, 1,  
1);
```

```
    grid.add(lblNome, 0, 2);  
    grid.add(campoNome, 1, 2);  
    grid.add(lblClasse, 0, 3);  
    grid.add(campoClasse, 1, 3);  
    grid.add(lblRaca, 0, 4);  
    grid.add(campoRaca, 1, 4);  
    grid.add(lblDescricao, 0, 5);  
    grid.add(campoInfo, 1, 5);  
    grid.add(lblImagem, 0, 6);  
    grid.add(campolmagem, 1, 6);  
    grid.add(btnProximo, 1, 7);
```

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

```
defAtributos();
```

```
    Button btnProximo = new  
Button("Próximo: Definir Atributos");
```

```
btnProximo.setFont(Font.font("Arial", 16));
```

```
btnProximo.setTextFill(Color.WHITE);
```

```
btnProximo.setStyle("-fx-background-color:  
#007ACC; -fx-background-radius: 8;");
```

```
    btnProximo.setOnAction(e ->  
exibirTelaDefinirAtributos.exibirTelaDefinirAt  
ributos(
```

```
Integer.parseInt(campoPersonagemId.getTe  
xt()),
```

```
        campoNome.getText(),  
        campoClasse.getValue(),  
        campoRaca.getValue(),  
        campoInfo.getText(),  
        campolmagem.getText(),  
        stageCaracteristicas  
    ));
```

```
    grid.add(titulo, 0, 0);  
    grid.add(lblPersonagemId, 0, 1);  
    grid.add(campoPersonagemId, 1,  
1);
```

```
    grid.add(lblNome, 0, 2);  
    grid.add(campoNome, 1, 2);  
    grid.add(lblClasse, 0, 3);  
    grid.add(campoClasse, 1, 3);  
    grid.add(lblRaca, 0, 4);  
    grid.add(campoRaca, 1, 4);  
    grid.add(lblDescricao, 0, 5);  
    grid.add(campoInfo, 1, 5);  
    grid.add(lblImagem, 0, 6);  
    grid.add(campolmagem, 1, 6);  
    grid.add(btnProximo, 1, 7);
```

<pre>stageCaracteristicas.setScene(cena);     stageCaracteristicas.show(); }</pre>	<pre>Scene cena = new Scene(grid, 400, 350);  stageCaracteristicas.setScene(cena);     stageCaracteristicas.show(); }</pre>
--	---