2º Trabalho de POOII Refatoração do Projeto do Trabalho I de POOII

Alunos: Ana Júlia Orlovski, Jotair Elio Kwiatkowski Junior, Júlia Dambrós
Link código do primeiro trabalho: https://github.com/Juliadambros/Held-Control.git
Link código com as refatorações e testes de unidade: https://github.com/Juliadambros/AppHeldControl2.git

1 - Ana Júlia Orlovski

Descrições das refatorações:

Refatoração 1: Na classe CadastroAnimalDBDAO tinha um método listaTodos()que não estava sendo usado durante a aplicação. Para não deixar métodos sem uso, optamos por retirá-lo e, também retirado da interface.

Refatoração 2: retirada do método showAlert pois ele não foi usado em nenhum momento com a aplicação rodando.

Refatoração 3: Refatoração foi realizada para melhorar a resiliência, segurança e manutenção do código ao tratar erros de acesso ao banco de dados de maneira mais robusta. Nos métodos que interagem com o banco de dados, como atualiza, foi introduzido um bloco try-catch-finally. Antes, os métodos apenas declaravam a exceção com throws SQLException, transferindo a responsabilidade de tratamento para a camada superior. Agora, a exceção é capturada e tratada no próprio método.Incluímos mensagens de erro no catch usando System.err.println, que exibe uma descrição clara do erro ao usuário ou no console de administração. Também foi incluído e.printStackTrace() para exibir a pilha completa do erro em cenários de depuração. O fechamento da conexão (close()) foi movido para dentro do finally para garantir que o recurso seja liberado independentemente de ocorrer ou não uma exceção.

Refatoração 4: A lógica de fechamento da janela atual e abertura de uma nova tela estava repetida em vários métodos (handleTelaInicial, handleCadastroAnimal, handleCadastroFazenda, etc.). Para evitar duplicação de código e melhorar a manutenção, esse processo foi extraído para um método único chamado

handleTela, que agora recebe os parâmetros de caminho da tela e título, simplificando os métodos individuais.

	Classe:	
Nº refatoração: 1	CadastroAnimalDBDAO	Nome: Refatoração Geral do código
Código antes de refatorar:		Código depois de refatorar:
package controle.animal.dao;		package controle.animal.dao;
import		import
-		controle.animal.model.Cadastr
al;		oAnimal;
<pre>import java.sql.*;</pre>		<pre>import java.sql.*;</pre>
<pre>import java.util.ArrayList;</pre>		<pre>import java.util.ArrayList;</pre>
<pre>import java.util.List;</pre>		<pre>import java.util.List;</pre>
public class C	adastroAnimalDBDAO	public class
implements Cad	astroAnimalDAO {	CadastroAnimalDBDAO
private Str		implements CadastroAnimalDAO
	nection connection;	{
_	paredStatement	private String sql;
<pre>statement; private ResultSet result;</pre>		<pre>private Connection connection;</pre>
private kes	uitset lesuit,	private PreparedStatement
private voi	d open() throws	statement;
SQLException {		private ResultSet result;
connect	ion =	
Conexao.getCon	nection(Conexao.url,	private void open() throws
Conexao.usuario, Conexao.senha);		SQLException {
}		connection =
~~i+-	d aloga() through	Conexao.getConnection(Conexao
<pre>private void close() throws SQLException {</pre>		.url, Conexao.usuario, Conexao.senha);
if (connection != null &&		}
!connection.isClosed()) {		
con	nection.close();	private void close()
}		throws SQLException {
		-

```
@Override
   public void
insere(CadastroAnimal animal)
throws SQLException {
       open();
CadastroAnimal (animalId,
especie, sexo, alimentacao,
dataChegada) VALUES (?, ?, ?, ?,
       statement =
connection.prepareStatement(sql);
       statement.setInt(1,
animal.getAnimalId());
       statement.setString(2,
animal.getFazendaId());
       statement.setString(3,
animal.getRaca());
       statement.setFloat(4,
                                    ql);
animal.getPeso());
       statement.setString(5,
animal.getRebanho());
       statement.setString(6,
animal.getProcedencia());
       statement.setString(7,
animal.getEspecie());
       statement.setString(8,
animal.getSexo());
       statement.setString(9,
animal.getAlimentacao());
       statement.setString(10,
animal.getEspecificacao());
       statement.setDate(11,
Date.valueOf(animal.getDataNascime
nto()));
```

```
if (connection != null
&& !connection.isClosed()) {
connection.close();
   @Override
   public void
insere(CadastroAnimal animal)
throws SQLException {
       open();
CadastroAnimal (animalId,
rebanho, " +
especie, sexo, alimentacao,
especificacao,
dataNascimento, dataChegada)
VALUES (?, ?, ?, ?, ?, ?, ?,
connection.prepareStatement(s
       statement.setInt(1,
animal.getAnimalId());
       statement.setString(2,
animal.getFazendaId());
       statement.setString(3,
animal.getRaca());
       statement.setFloat(4,
animal.getPeso());
       statement.setString(5,
animal.getRebanho());
       statement.setString(6,
animal.getProcedencia());
       statement.setString(7,
animal.getEspecie());
       statement.setString(8,
animal.getSexo());
```

```
statement.setDate(12,
                                           statement.setString(9,
Date.valueOf(animal.getDataChegada animal.getAlimentacao());
()));
                                    statement.setString(10,
       statement.executeUpdate();
                                    animal.getEspecificacao());
       close();
                                           statement.setDate(11,
                                    Date.valueOf(animal.getDataNa
   @Override
                                    scimento()));
   public void
                                           statement.setDate(12,
atualiza(CadastroAnimal animal)
                                    Date.valueOf(animal.getDataCh
throws SQLException {
                                    egada()));
       open();
       sql = "UPDATE
                                    statement.executeUpdate();
CadastroAnimal SET fazendaId=?,
                                           close();
raca=?, peso=?, rebanho=?, " +
                                       @Override
especie=?, sexo=?, alimentacao=?,
especificacao=?, dataNascimento=?,
                                       public void
dataChegada=? " +
                                    atualiza(CadastroAnimal
               "WHERE animalId=?";
                                    animal) throws SQLException {
                                           open();
                                           sql = "UPDATE
connection.prepareStatement(sql);
       statement.setString(1,
                                    CadastroAnimal SET
animal.getFazendaId());
       statement.setString(2,
animal.getRaca());
       statement.setFloat(3,
                                    "procedencia=?, especie=?,
                                    sexo=?, alimentacao=?,
animal.getPeso());
       statement.setString (4,
                                    especificacao=?,
animal.getRebanho());
                                    dataNascimento=?,
                                    dataChegada=? " +
       statement.setString(5,
animal.getProcedencia());
       statement.setString(6,
                                    animalId=?";
animal.getEspecie());
       statement.setString(7,
                                    connection.prepareStatement(s
animal.getSexo());
                                    ql);
       statement.setString(8,
                                           statement.setString(1,
animal.getAlimentacao());
                                    animal.getFazendaId());
       statement.setString(9,
                                           statement.setString(2,
animal.getEspecificacao());
                                    animal.getRaca());
                                           statement.setFloat (3,
       statement.setDate(10,
```

```
Date.valueOf(animal.getDataNascime animal.getPeso());
nto()));
                                           statement.setString(4,
       statement.setDate(11,
                                    animal.getRebanho());
Date.valueOf(animal.getDataChegada
                                           statement.setString(5,
()));
                                    animal.getProcedencia());
                                           statement.setString(6,
       statement.setInt(12,
animal.getAnimalId());
                                    animal.getEspecie());
                                           statement.setString(7,
       statement.executeUpdate();
                                    animal.getSexo());
                                           statement.setString(8,
       close();
                                    animal.getAlimentacao());
                                           statement.setString(9,
   @Override
                                    animal.getEspecificacao());
   public void
                                           statement.setDate(10,
remove(CadastroAnimal animal)
                                    Date.valueOf(animal.getDataNa
throws SQLException {
                                    scimento()));
                                           statement.setDate(11,
       open();
       sql = "DELETE FROM
                                    Date.valueOf(animal.getDataCh
CadastroAnimal WHERE animalId=?";
                                    egada()));
       statement =
                                           statement.setInt(12,
                                    animal.getAnimalId());
connection.prepareStatement(sql);
       statement.setInt(1,
animal.getAnimalId());
       statement.executeUpdate();
                                    statement.executeUpdate();
                                           close();
       close();
   @Override
                                       @Override
   public CadastroAnimal
                                       public void
buscaPorId(int animalId) throws
                                    remove(CadastroAnimal animal)
SQLException {
                                    throws SQLException {
       open();
                                           open();
       sql = "SELECT * FROM
                                           sql = "DELETE FROM
CadastroAnimal WHERE animalId=?";
                                    CadastroAnimal WHERE
                                    animalId=?";
connection.prepareStatement(sql);
       statement.setInt(1,
                                    connection.prepareStatement(s
animalId);
                                    ql);
                                           statement.setInt(1,
                                    animal.getAnimalId());
statement.executeQuery();
```

```
CadastroAnimal animal =
                                    statement.executeUpdate();
null;
                                           close();
       if (result.next()) {
           animal = new
                                       @Override
CadastroAnimal();
                                       public CadastroAnimal
animal.setAnimalId(result.getInt("
                                    buscaPorId(int animalId)
animalId"));
                                    throws SQLException {
                                           open();
animal.setFazendaId(result.getStri
nq("fazendaId"));
                                    CadastroAnimal WHERE
                                    animalId=?";
animal.setRaca(result.getString("r
                                           statement =
aca"));
                                    connection.prepareStatement(s
                                    ql);
animal.setPeso(result.getFloat("pe
                                           statement.setInt(1,
so"));
                                    animalId);
                                           result =
animal.setRebanho(result.getString statement.executeQuery();
("rebanho"));
                                           CadastroAnimal animal
animal.setProcedencia(result.getSt = null;
ring("procedencia"));
                                           if (result.next()) {
                                               animal = new
animal.setEspecie(result.getString CadastroAnimal();
("especie"));
                                    animal.setAnimalId(result.get
animal.setSexo(result.getString("s Int("animalId"));
exo"));
                                    animal.setFazendaId(result.ge
animal.setAlimentacao(result.getSt tString("fazendaId"));
ring("alimentacao"));
                                    animal.setRaca(result.getStri
animal.setEspecificacao(result.get ng("raca"));
String("especificacao"));
                                    animal.setPeso(result.getFloa
animal.setDataNascimento(result.ge t("peso"));
tDate("dataNascimento").toLocalDat
e());
                                    animal.setRebanho(result.getS
                                    tring("rebanho"));
animal.setDataChegada(result.getDa
```

```
te("dataChegada").toLocalDate());
       close();
       return animal;
   @Override
   public List<CadastroAnimal>
listaTodos() throws SQLException -
       open();
       sql = "SELECT * FROM
CadastroAnimal";
       statement =
connection.prepareStatement(sql);
       result =
statement.executeQuery();
       List<CadastroAnimal>
animais = new ArrayList<>();
       while (result.next()) {
new CadastroAnimal();
animal.setAnimalId(result.getInt("
animalId"));
animal.setFazendaId(result.getStri
nq("fazendaId"));
animal.setRaca(result.getString("r
aca"));
animal.setPeso(result.getFloat("pe
so"));
animal.setRebanho(result.getString
("rebanho"));
animal.setProcedencia(result.getSt
ring("procedencia"));
```

```
animal.setProcedencia(result.
                        getString("procedencia"));
                        animal.setEspecie(result.getS
                        tring("especie"));
                        animal.setSexo(result.getStri
                        ng("sexo"));
                        animal.setAlimentacao(result.
                        getString("alimentacao"));
                        animal.setEspecificacao(resul
                         t.getString("especificacao"))
                         animal.setDataNascimento(resu
                         lt.getDate("dataNascimento").
                        toLocalDate());
CadastroAnimal animal = animal.setDataChegada(result.
                        qetDate("dataCheqada").toLoca
                        lDate());
                                close();
                                return animal;
                            } // foi retirado o método
                         da interface também
```

```
animal.setEspecie(result.getString
("especie"));
animal.setSexo(result.getString("s
exo"));
animal.setAlimentacao(result.getSt
ring("alimentacao"));
animal.setEspecificacao(result.get
String("especificacao"));
animal.setDataNascimento(result.ge
tDate("dataNascimento").toLocalDat
e());
animal.setDataChegada(result.getDa
te("dataChegada").toLocalDate());
           animais.add(animal);
       close();
       return animais;
               Classe:TelaInicialControll
Nº refatoração: 2 | er
                                     Nome: Refatoração Geral do código
Código antes de refatorar:
                                     Código depois de refatorar:
package
controle.animal.controller;
                                     controle.animal.controller;
import javafx.event.ActionEvent;
                                     import
import javafx.fxml.FXML;
                                     javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
                                     import javafx.fxml.FXML;
import javafx.scene.Node;
                                     import
import javafx.scene.Parent;
                                     javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
                                     import javafx.scene.Node;
import javafx.scene.control.Alert;
                                     import javafx.scene.Parent;
                                     import javafx.scene.Scene;
import
javafx.scene.control.Button;
                                     import
import javafx.scene.image.Image;
                                      javafx.scene.control.Alert;
```

```
import
                                    javafx.scene.control.Button;
javafx.scene.image.ImageView;
import javafx.stage.Stage;
                                    import
                                    javafx.scene.image.Image;
import java.io.IOException;
                                    import
                                    javafx.scene.image.ImageView;
public class TelaInicialController import javafx.stage.Stage;
                                    import java.io.IOException;
   @FXMI.
                                    public class
   private Button
                                    TelaInicialController {
btnCadastrarFazenda;
   @FXML
                                       @FXML
   private Button
                                       private Button
btnCadastroAnimal;
                                    btnCadastrarFazenda;
   @FXML
                                       @FXML
   private Button
                                       private Button
btnPesquisaAnimal;
                                    btnCadastroAnimal;
   @FXMT.
                                       @FXMT.
   private Button btnTelaInicial;
                                       private Button
                                    btnPesquisaAnimal;
   @FXML
   private ImageView imagem;
                                       @FXML
                                       private Button
                                    btnTelaInicial;
   @FXML
   private ImageView imagem1;
                                       @FXML
   @FXML
                                       private ImageView imagem;
   public void initialize() {
       Image img = new
                                       @FXML
Image(getClass().getResourceAsStre
                                       private ImageView imagem1;
am("/controle/animal/imagem/Logo.j
peg"));
                                       @FXML
                                       public void initialize() {
       imagem.setImage(img);
                                            Image img = new
                                    Image(getClass().getResourceA
                                    sStream("/controle/animal/ima
                                     gem/Logo.jpeg"));
   @FXML
```

```
private void
handleTelaInicial(ActionEvent
event) {
       ((Stage) ((Node)
event.getSource()).getScene().getW
indow()).close();
abrirNovaTela("/controle/animal/vi
ew/TelaInicial.fxml", "Tela
Inicial");
   @FXML
   private void
handleCadastroAnimal(ActionEvent
event) {
       ((Stage) ((Node)
event.getSource()).getScene().getW
indow()).close();
abrirNovaTela("/controle/animal/vi
ew/CadastroAnimal.fxml", "Cadastro
de Animal");
   @FXML
   private void
handleCadastroFazenda(ActionEvent
event) {
       ((Stage) ((Node)
event.getSource()).getScene().getW
indow()).close();
abrirNovaTela("/controle/animal/vi vent event) {
ew/CadastroFazenda.fxml",
"Cadastro de Fazenda");
   @FXML
   private void
nandlePesquisaAnimal(ActionEvent
```

```
imagem.setImage(img);
   @FXML
   private void
handleTelaInicial(ActionEvent
event) {
       ((Stage) ((Node)
event.getSource()).getScene()
.getWindow()).close();
abrirNovaTela("/controle/anim
al/view/TelaInicial.fxml",
"Tela Inicial");
   @FXML
handleCadastroAnimal(ActionEv
ent event) {
      ((Stage) ((Node)
event.getSource()).getScene()
.qetWindow()).close();
abrirNovaTela("/controle/anim
al/view/CadastroAnimal.fxml",
"Cadastro de Animal");
   @FXML
   private void
handleCadastroFazenda(ActionE
       ((Stage) ((Node)
event.getSource()).getScene()
.getWindow()).close();
abrirNovaTela("/controle/anim
al/view/CadastroFazenda.fxml"
  "Cadastro de Fazenda");
```

```
event) {
       ((Stage) ((Node)
event.getSource()).getScene().getW
                                       @FXML
indow()).close();
                                       private void
                                    handlePesquisaAnimal(ActionEv
abrirNovaTela("/controle/animal/vi
                                   ent event) {
ew/PesquisaAnimal.fxml", "Pesquisa
                                           ((Stage) ((Node)
de Animal");
                                    event.getSource()).getScene()
                                    .getWindow()).close();
   private void
                                    abrirNovaTela("/controle/anim
abrirNovaTela(String caminhoFXML,
String titulo) {
                                    "Pesquisa de Animal");
           FXMLLoader loader = new
FXMLLoader(getClass().getResource(
                                       private void
                                    abrirNovaTela(String
caminhoFXML));
                                    caminhoFXML, String titulo) {
           Parent root =
loader.load();
           Stage newStage = new
                                               FXMLLoader loader
Stage();
           Scene newScene = new
                                    FXMLLoader(getClass().getReso
                                    urce(caminhoFXML));
Scene (root);
                                               Parent root =
                                    loader.load();
newStage.setScene(newScene);
                                               Stage newStage =
                                    new Stage();
                                               Scene newScene =
newStage.setTitle(titulo);
                                    new Scene(root);
newStage.setResizable(false);
           newStage.show();
       } catch (IOException e) {
                                    newStage.setScene(newScene);
           e.printStackTrace();
           showErrorAlert("Erro ao newStage.setTitle(titulo);
abrir a tela: " + titulo, e);
                                    newStage.setResizable(false);
                                               newStage.show();
                                           } catch (IOException
   private void
                                    e) {
showErrorAlert(String message,
                                    e.printStackTrace();
Exception e) {
```

```
Alert alert = new
Alert(Alert.AlertType.ERROR);
                                     showErrorAlert("Erro ao abrir
       alert.setTitle("Erro");
                                     a tela: " + titulo, e);
       alert.setHeaderText(null);
alert.setContentText(message);
       alert.setResizable(true);
                                        private void
       alert.setHeight(150);
                                     showErrorAlert(String
       alert.setWidth(300);
                                     message, Exception e) {
       alert.showAndWait();
                                            Alert alert = new
                                     Alert(Alert.AlertType.ERROR);
                                     alert.setTitle("Erro");
   private void showAlert(String
title, String message) {
       Alert alert = new
                                     alert.setHeaderText(null);
Alert(Alert.AlertType.INFORMATION)
                                     alert.setContentText(message)
       alert.setTitle(title);
       alert.setHeaderText(null);
                                     alert.setResizable(true);
alert.setContentText(message);
                                            alert.setHeight(150);
       alert.showAndWait();
                                            alert.setWidth(300);
                                            alert.showAndWait();
                                     Nome:Tratamento de Exceções e
                                     Gerenciamento de Recursos
               Classe:
Nº refatoração: 3 | CadastroFazendaDBDAO
Código antes de refatorar:
                                     Código depois de refatorar:
@Override
                                     @Override
public void
                                     public void
atualiza(CadastroFazenda fazenda)
                                     atualiza(CadastroFazenda
                                     fazenda) throws SQLException
throws SQLException {
   open();
   sql = "UPDATE CadastroFazenda
```

```
SET nome=?, ramo=?, telefone=?,
email=?, cep=?, rua=?, nCasa=?,
bairro=?, cidade=?, estado=? " +
           "WHERE cnpj=?";
   statement =
connection.prepareStatement(sql);
   statement.setString(1,
fazenda.getNome());
   statement.setString(2,
fazenda.getRamo());
   statement.setString(3,
fazenda.getTelefone());
   statement.setString(4,
fazenda.getEmail());
   statement.setString (5,
fazenda.getCep());
   statement.setString(6,
fazenda.getRua());
   statement.setInt(7,
fazenda.getnCasa());
   statement.setString(8,
fazenda.getBairro());
   statement.setString(9,
fazenda.getCidade());
   statement.setString(10,
fazenda.getEstado());
   statement.setString(11,
fazenda.getCnpj());
   statement.executeUpdate();
   close();
```

```
if (fazenda.getNome()
== null ||
fazenda.getNome().isEmpty())
JOptionPane.showMessageDialog
pode ser vazio.", "Erro de
JOptionPane. WARNING MESSAGE);
           return;
       open();
       sql = "UPDATE
CadastroFazenda SET nome=?,
bairro=?, cidade=?, estado=?
cnpj=?";
connection.prepareStatement(s
ql);
       statement.setString(1,
fazenda.getNome());
       statement.setString(2,
fazenda.getRamo());
       statement.setString(3,
fazenda.getTelefone());
       statement.setString(4,
fazenda.getEmail());
       statement.setString(5,
fazenda.getCep());
       statement.setString(6,
fazenda.getRua());
       statement.setInt(7,
fazenda.getnCasa());
       statement.setString(8,
fazenda.getBairro());
       statement.setString(9,
```

```
fazenda.getCidade());
                                          statement.setString(10,
                                          fazenda.getEstado());
                                          statement.setString(11,
                                          fazenda.getCnpj());
                                          statement.executeUpdate();
                                                 close();
                                             } catch (SQLException e) {
                                          System.err.println("Erro ao
                                          e.getMessage());
                                                  e.printStackTrace();
                                                       close();
                                                 } catch (SQLException
                                          e) {
                                          System.err.println("Erro ao
                                          e.getMessage());
                 Classe:
                 TelaInicialController.java
Nº refatoração: 4
                                          Nome: Extract Method
Código antes de refatorar:
                                          Código depois de refatorar:
@FXML
                                          private void handleTela(ActionEvent
private void handleTelaInicial(ActionEvent event) event, String caminhoFXML, String titulo)
 ((Stage) ((Node)
                                           ((Stage) ((Node)
event.getSource()).getScene().getWindow()).close event.getSource()).getScene().getWindow(
                                          )).close();
```

```
abrirNovaTela(caminhoFXML, titulo);
abrirNovaTela("/controle/animal/view/TelaInici
al.fxml", "Tela Inicial");
                                                @FXML
                                                private void handleTelaInicial(ActionEvent
                                                event) {
                                                  handleTela(event,
@FXML
                                                 /controle/animal/view/TelaInicial.fxml",
private void handleCadastroAnimal(ActionEvent
                                                 Tela Inicial");
event) {
 ((Stage) ((Node)
event.getSource()).getScene().getWindow()).close
                                                @FXML
();
                                                private void
abrirNovaTela("/controle/animal/view/Cadastr
                                                handleCadastroAnimal(ActionEvent event)
oAnimal.fxml", "Cadastro de Animal");
                                                  handleTela(event,
                                                 /controle/animal/view/CadastroAnimal.
                                                fxml", "Cadastro de Animal");
@FXML
private void
handleCadastroFazenda(ActionEvent event) {
 ((Stage) ((Node)
                                                 @FXML
event.getSource()).getScene().getWindow()).close private void
                                                handleCadastroFazenda(ActionEvent
();
                                                event) {
                                                  handleTela(event,
abrirNovaTela("/controle/animal/view/Cadastr
oFazenda.fxml", "Cadastro de Fazenda");
                                                "/controle/animal/view/CadastroFazend
                                                a.fxml", "Cadastro de Fazenda");
@FXML
private void handlePesquisaAnimal(ActionEvent
                                                @FXML
event) {
                                                private void
 ((Stage) ((Node)
                                                handlePesquisaAnimal(ActionEvent event)
event.getSource()).getScene().getWindow()).close
                                                  handleTela(event,
();
                                                 /controle/animal/view/PesquisaAnimal.
abrirNovaTela("/controle/animal/view/Pesquis
                                                fxml", "Pesquisa de Animal");
aAnimal.fxml", "Pesquisa de Animal");
```

2 - Jotair Elio Kwiatkowski Junior

Descrição das refatorações:

Refatoração 1: Na classe CadastroFazendaDBDAO tinha um método listaTodos() que não estava sendo usado durante a aplicação. Para não deixar métodos sem uso, optamos por retirá-lo e, também retirado da interface.

Refatoração 2: A lógica de fechamento da janela atual e abertura de uma nova tela estava repetida em vários métodos (handleTelaInicial, handleCadastroAnimal, handleCadastroFazenda, etc.). Para evitar duplicação de código e melhorar a manutenção, esse processo foi extraído de um método único chamado handleTela, que agora recebe os parâmetros de caminho da tela e título, simplificando os métodos individuais.

Refatoração 3: Na classe CadastroFazendaController quando cadastrarmos o nome da fazenda e caso fosse vazio não gerava um alerta para o usuário, cometendo um erro. Agora com essa nova formulação gera um alerta e o usuário é obrigado a cadastrar a fazenda diferente de vazio. Dessa forma, foi necessário passar a variável nos métodos handleCadastrar, validarCadastro e CadastroFazenda.

Refatoração 4: A refatoração foi realizada para melhorar a resiliência, segurança e manutenção do código ao tratar erros de acesso ao banco de dados de maneira mais robusta. Nos métodos que interagem com o banco de dados, como insere, foi introduzido um bloco try-catch-finally. Antes, os métodos apenas declaravam a exceção com throws SQLException, transferindo a responsabilidade de tratamento para a camada superior. Agora, a exceção é capturada e tratada no próprio método.Incluímos mensagens de erro no catch usando System.err.println, que exibe uma descrição clara do erro ao usuário ou no console de administração. Também foi incluído e.printStackTrace() para exibir a pilha completa do erro em cenários de depuração. O fechamento da conexão (close()) foi movido para dentro do finally para garantir que o recurso seja liberado independentemente de ocorrer ou não uma exceção.

	Classe:	
Nº refatoração: 1	CadastroFazendaDBDAO	Nome: Refatoração Geral do código

Código antes de refatorar: package controle.animal.dao; import controle.animal.model.CadastroFaz import enda; import java.sql.*; import java.util.ArrayList; import java.util.List; public class CadastroFazendaDBDAO implements CadastroFazendaDAO { private String sql; private static Connection connection; private PreparedStatement statement; private ResultSet result; private void open() throws SQLException { Conexao.getConnection(Conexao.url SQLException { Conexao.usuario, Conexao.senha); private void close() throws SQLException { if (connection != null && !connection.isClosed()) { connection.close(); @Override insere(CadastroFazenda fazenda) throws SQLException { open();

Código depois de refatorar:

```
package controle.animal.dao;
controle.animal.model.CadastroF
azenda;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
public class
CadastroFazendaDBDAO implements
CadastroFazendaDAO {
   private String sql;
   private static Connection
connection;
   private PreparedStatement
statement;
   private ResultSet result;
   private void open() throws
       connection =
Conexao.getConnection(Conexao.u
rl, Conexao.usuario,
Conexao.senha);
   private void close() throws
SQLException {
       if (connection != null
&& !connection.isClosed()) {
           connection.close();
   @Override
   public void
 nsere (CadastroFazenda fazenda)
```

```
sql = "INSERT INTO
                                   throws SQLException {
                                          open();
ramo, telefone, email, cep, rua,
                                          sql = "INSERT INTO
nCasa, bairro, cidade, estado) "
                                   CadastroFazenda (nome, cnpj,
                                   ramo, telefone, email, cep,
                                   rua, nCasa, bairro, cidade,
                                   estado) " +
?, ?, ?, ?, ?, ?, ?, ?)";
connection.prepareStatement(sql); ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
       statement.setString(1,
fazenda.getNome());
                                   connection.prepareStatement(sql
       statement.setString(2,
                                   );
fazenda.getCnpj());
                                          statement.setString(1,
       statement.setString(3,
                                   fazenda.getNome());
                                          statement.setString(2,
fazenda.getRamo());
       statement.setString(4,
                                   fazenda.getCnpj());
                                          statement.setString(3,
fazenda.getTelefone());
       statement.setString(5,
                                   fazenda.getRamo());
                                          statement.setString(4,
fazenda.getEmail());
       statement.setString(6,
                                   fazenda.getTelefone());
fazenda.getCep());
                                          statement.setString(5,
       statement.setString(7,
                                   fazenda.getEmail());
                                          statement.setString(6,
fazenda.getRua());
       statement.setInt(8,
                                   fazenda.getCep());
fazenda.getnCasa());
                                          statement.setString(7,
       statement.setString(9,
                                   fazenda.getRua());
                                          statement.setInt(8,
fazenda.getBairro());
       statement.setString(10,
                                   fazenda.getnCasa());
fazenda.getCidade());
                                          statement.setString(9,
       statement.setString(11,
                                   fazenda.getBairro());
fazenda.getEstado());
                                          statement.setString(10,
       statement.executeUpdate();
                                  fazenda.getCidade());
       close();
                                          statement.setString(11,
                                   fazenda.getEstado());
   @Override
                                   statement.executeUpdate();
   public void
                                          close();
atualiza(CadastroFazenda fazenda)
throws SQLException {
                                      @Override
       open();
                                      public void
```

```
CadastroFazenda SET nome=?,
                                   atualiza(CadastroFazenda
                                   fazenda) throws SQLException {
                                          open();
                                          sql = "UPDATE
                                   CadastroFazenda SET nome=?,
       statement =
connection.prepareStatement(sql);
                                   cep=?, rua=?, nCasa=?,
                                   bairro=?, cidade=?, estado=? "
       statement.setString(1,
fazenda.getNome());
       statement.setString(2,
fazenda.getRamo());
       statement.setString(3,
                                   connection.prepareStatement(sql
fazenda.getTelefone());
                                          statement.setString(1,
       statement.setString(4,
fazenda.getEmail());
                                   fazenda.getNome());
                                          statement.setString(2,
       statement.setString(5,
fazenda.getCep());
                                   fazenda.getRamo());
       statement.setString(6,
                                          statement.setString(3,
fazenda.getRua());
                                   fazenda.getTelefone());
       statement.setInt(7,
                                          statement.setString(4,
fazenda.getnCasa());
                                   fazenda.getEmail());
       statement.setString(8,
                                          statement.setString(5,
fazenda.getBairro());
                                   fazenda.getCep());
                                          statement.setString(6,
       statement.setString(9,
fazenda.getCidade());
                                   fazenda.getRua());
       statement.setString(10,
                                          statement.setInt(7,
fazenda.getEstado());
                                   fazenda.getnCasa());
       statement.setString(11,
                                          statement.setString(8,
fazenda.getCnpj());
                                   fazenda.getBairro());
       statement.executeUpdate();
                                          statement.setString(9,
       close();
                                   fazenda.getCidade());
                                          statement.setString(10,
                                   fazenda.getEstado());
   @Override
                                          statement.setString(11,
   public void
                                   fazenda.getCnpj());
remove (CadastroFazenda fazenda)
throws SQLException {
                                   statement.executeUpdate();
       open();
                                          close();
                                      @Override
```

```
connection.prepareStatement(sql);
                                      public void
                                   remove(CadastroFazenda fazenda)
       statement.setString(1,
fazenda.getCnpj());
                                   throws SQLException {
       statement.executeUpdate();
                                          open();
                                          sql = "DELETE FROM
       close();
                                          statement =
   @Override
                                   connection.prepareStatement(sql
   public CadastroFazenda
buscaPorCNPJ(String cnpj) throws
                                          statement.setString(1,
SQLException {
                                   fazenda.getCnpj());
       open();
       sql = "SELECT * FROM
                                   statement.executeUpdate();
                                          close();
connection.prepareStatement(sql);
       statement.setString(1,
                                      @Override
                                      public CadastroFazenda
cnpj);
                                   buscaPorCNPJ(String cnpj)
       result =
statement.executeQuery();
                                   throws SQLException {
                                          open();
                                          sql = "SELECT * FROM
       CadastroFazenda fazenda =
                                   CadastroFazenda WHERE cnpj=?";
null;
       if (result.next()) {
                                          statement =
           fazenda = new
                                   connection.prepareStatement(sql
CadastroFazenda();
                                          statement.setString(1,
fazenda.setNome(result.getString(|cnpj);
"nome"));
                                          result =
                                   statement.executeQuery();
fazenda.setCnpj(result.getString(
"cnpj"));
                                          CadastroFazenda fazenda
                                   = null;
fazenda.setRamo(result.getString(
                                          if (result.next()) {
"ramo"));
                                              fazenda = new
                                   CadastroFazenda();
fazenda.setTelefone(result.getStr
ing("telefone"));
                                   fazenda.setNome(result.getStrin
                                   g("nome"));
fazenda.setEmail(result.getString
 ("email"));
                                   fazenda.setCnpj(result.getStrin
```

```
g("cnpj"));
fazenda.setCep(result.getString('
cep"));
                                   fazenda.setRamo(result.getStrin
                                   q("ramo"));
fazenda.setRua(result.getString("
                                   fazenda.setTelefone(result.getS
rua"));
                                   tring("telefone"));
fazenda.setnCasa(result.getInt("n
Casa"));
                                   fazenda.setEmail(result.getStri
                                   nq("email"));
fazenda.setBairro(result.getStrin
q("bairro"));
                                   fazenda.setCep(result.getString
                                   ("cep"));
fazenda.setCidade(result.getStrin
q("cidade"));
                                   fazenda.setRua(result.getString
                                   ("rua"));
fazenda.setEstado(result.getStrin
q("estado"));
                                   fazenda.setnCasa(result.getInt(
                                   "nCasa"));
       close();
                                   fazenda.setBairro(result.getStr
       return fazenda;
                                   ing("bairro"));
   @Override
                                   fazenda.setCidade(result.getStr
   public List<CadastroFazenda>
                                   ing("cidade"));
listaTodos() throws SQLException
                                   fazenda.setEstado(result.getStr
                                   ing("estado"));
       open();
       sql = "SELECT * FROM
CadastroFazenda";
                                          close();
       statement =
                                          return fazenda;
connection.prepareStatement(sql);
                                      public static boolean
       result =
statement.executeQuery();
                                   cnpjExists(String cnpj) throws
                                   SQLException {
       List<CadastroFazenda>
                                          String query = "SELECT
fazendas = new ArrayList<>();
                                   COUNT(*) FROM fazendas WHERE
       while (result.next()) {
           CadastroFazenda
                                          try (PreparedStatement
fazenda = new CadastroFazenda();
                                   statement =
                                   connection.prepareStatement (que
```

```
fazenda.setNome(result.getString( ry)) {
"nome"));
fazenda.setCnpj(result.getString(
"cnpj"));
fazenda.setRamo(result.getString( (resultSet.next()) {
"ramo"));
fazenda.setTelefone(result.getStr
ing("telefone"));
fazenda.setEmail(result.getString
("email"));
fazenda.setCep(result.getString(" interface também
cep"));
fazenda.setRua(result.getString("
rua"));
fazenda.setnCasa(result.getInt("n
Casa"));
fazenda.setBairro(result.getStrin
g("bairro"));
fazenda.setCidade(result.getStrin
q("cidade"));
fazenda.setEstado(result.getStrin
g("estado"));
           fazendas.add(fazenda);
       close();
       return fazendas;
   public static boolean
cnpjExists(String cnpj) throws
SQLException {
```

```
statement.setString(1, cnpj);
           ResultSet resultSet
= statement.executeQuery();
               return
resultSet.getInt(1) > 0;
       return false;
 // foi retirado o método da
```

```
String query = "SELECT
cnpj= ?";
        try (PreparedStatement
statement =
connection.prepareStatement (query
) ) {
             statement.setString(1,
cnpj);
              ResultSet resultSet =
statement.executeQuery();
              if (resultSet.next())
                   return
resultSet.getInt(1) > 0;
         return false;
                  Classe:
                  PesquisaAnimalControll
Nº refatoração: 2 |er.java
                                           Nome: Extract Method
Código antes de refatorar:
                                           Código depois de refatorar:
@FXML
                                           private void handleTela(ActionEvent event,
                                           String caminhoFXML, String titulo) {
private void handleTelaInicial(ActionEvent
event) {
                                            ((Stage) ((Node)
                                           event.getSource()).getScene().getWindow()).
 ((Stage) ((Node)
event.getSource()).getScene().getWindow()).clo
                                           close();
                                            abrirNovaTela(caminhoFXML, titulo);
se();
abrirNovaTela("/controle/animal/view/TelaIni @FXML
cial.fxml", "Tela Inicial");
                                           private void handleTelaInicial(ActionEvent
                                           event) {
                                            handleTela(event,
                                           "/controle/animal/view/TelaInicial.fxml",
@FXML
                                            Tela Inicial");
orivate void
```

1 11 0 1

```
event) {
       String cnpjInput =
cnpj.getText().replaceAll("[^0-9]
", "");
  if (!isValidCNPJ(cnpjInput)) {
           showErrorAlert("CNPJ
inválido: O CNPJ deve ter 14
dígitos.", null);
           return;
(!isValidTelefone(fone.getText())
) {
showErrorAlert("Telefone
inválido: O telefone deve ter
entre 10 e 11 dígitos.", null);
           return;
       CadastroFazendaDAO
cadastroFazendaDAO = new
CadastroFazendaDBDAO();
(controle.animal.dao.CadastroFaze
ndaDAO.cnpjExists(cnpjInput)) {
           showErrorAlert("CNPJ
inválido: Este CNPJ já está
cadastrado!", null);
           return;
       if
(!isValidCep(cep.getText())) {
           showErrorAlert("CEP
inválido: O CEP deve ter 8
dígitos.", null);
           return;
```

```
(!isValidEmail(email.getText()))
           showErrorAlert("E-mail
inválido: Por favor, insira um
e-mail válido.", null);
           return;
Não existia código para tratar esse erro.
               Classe:
               CadastroFazendaDBDAO Nome: Tratamento de Exceções e
Nº refatoração: 4
                                    Gerenciamento de Recursos
Código antes de refatorar:
                                    Código depois de refatorar:
@Override
                                    @Override
public void
                                    public void
insere(CadastroFazenda fazenda)
                                    insere(CadastroFazenda fazenda)
throws SQLException {
                                    throws SQLException {
   open();
   sql = "INSERT INTO
                                            if (fazenda.getNome() ==
                                    null ||
ramo, telefone, email, cep, rua,
                                    fazenda.getNome().isEmpty()) {
nCasa, bairro, cidade, estado) "
                                    JOptionPane.showMessageDialog(n
                                    ull, "O nome da fazenda não
                                    Validação",
   statement =
connection.prepareStatement(sql);
                                    JOptionPane. WARNING MESSAGE);
   statement.setString(1,
                                               return; //
fazenda.getNome());
   statement.setString(2,
fazenda.getCnpj());
   statement.setString(3,
                                            open();
fazenda.getRamo());
   statement.setString(4,
                                    ramo, telefone, email, cep,
fazenda.getTelefone());
   statement.setString(5,
```

```
fazenda.getEmail());
    statement.setString(6,
fazenda.getCep());
    statement.setString(7,
fazenda.getRua());
    statement.setInt(8,
fazenda.getnCasa());
    statement.setString(9,
fazenda.getBairro());
    statement.setString(10,
fazenda.getCidade());
    statement.setString(11,
fazenda.getEstado());
    statement.executeUpdate();
    close();
}
```

```
?, ?, ?, ?, ?, ?, ?, ?, ?)";
connection.prepareStatement(sql
       statement.setString(1,
fazenda.getNome());
       statement.setString(2,
fazenda.getCnpj());
       statement.setString(3,
fazenda.getRamo());
       statement.setString(4,
fazenda.getTelefone());
       statement.setString (5,
fazenda.getEmail());
       statement.setString(6,
fazenda.getCep());
       statement.setString(7,
fazenda.getRua());
       statement.setInt(8,
fazenda.getnCasa());
       statement.setString(9,
fazenda.getBairro());
       statement.setString(10,
fazenda.getCidade());
       statement.setString(11,
fazenda.getEstado());
statement.executeUpdate();
   } catch (SQLException e) {
       System.err.println("Erro
ao acessar o banco: " +
e.getMessage());
       e.printStackTrace();
           close();
       } catch (SQLException e)
```

```
System.err.println("Erro ao
fechar a conexão: " +
e.getMessage());
}
}
```

3 - Júlia Dambrós

Descrições das refatorações:

Refatoração 1: A validação de dados (CNPJ, telefone, CEP, e-mail, e a verificação da existência do CNPJ) estava repetida tanto no método handleCadastrar quanto no handleEditar. Para evitar a duplicação de código e melhorar a manutenção, a validação foi extraída para um único método, validarCadastro.

Refatoração 2: A criação do objeto CadastroFazenda estava duplicada tanto no handleCadastrar quanto no handleEditar. Para evitar duplicação e melhorar a clareza, o processo de criação do objeto foi extraído para um método separado, criarFazenda.

Refatoração 3: A lógica de fechamento da janela atual e abertura de uma nova tela estava repetida em vários métodos (handleTelaInicial, handleCadastroAnimal, handleCadastroFazenda, etc.). Para evitar duplicação de código e melhorar a manutenção, esse processo foi extraído para um método único chamado handleTela, que agora recebe os parâmetros de caminho da tela e título, simplificando os métodos individuais.

Refatoração 4: Remover variável que não está sendo utilizada.

Nº refatoração: 1	Classe:	Nome: Extract Method
	l .	

CadastroFazendaController.java

Código antes de refatorar:

Código depois de refatorar:

```
@FXML
private void handleCadastrar(ActionEvent
event) {
      String cnpjInput =
cnpj.getText().replaceAll("[^0-9]", "");
       if (!isValidCNPJ(cnpjInput)) {
          showErrorAlert("CNPJ inválido:
O CNPJ deve ter 14 dígitos.", null);
(!isValidTelefone(fone.getText())) {
           showErrorAlert("Telefone
inválido: O telefone deve ter entre 10 e
11 dígitos.", null);
       CadastroFazendaDAO
cadastroFazendaDAO = new
CadastroFazendaDBDAO();
(controle.animal.dao.CadastroFazendaDAO.c
npjExists(cnpjInput)) {
           showErrorAlert("CNPJ inválido:
       if (!isValidCep(cep.getText())) {
           showErrorAlert("CEP inválido:
O CEP deve ter 8 dígitos.", null);
           return;
(!isValidEmail(email.getText())) {
           showErrorAlert("E-mail
```

```
private boolean validarCadastro(String
cnpj, String telefone, String cep,
String email) throws SQLException {
  if (!isValidCNPJ(cnpj)) {
       showErrorAlert("CNPJ inválido: 0
CNPJ deve ter 14 dígitos.", null);
      return false;
  if (!isValidTelefone(telefone)) {
       showErrorAlert("Telefone
 11 dígitos.", null);
       return false;
(CadastroFazendaDAO.cnpjExists(cnpj)) {
       showErrorAlert("CNPJ inválido:
Este CNPJ já está cadastrado!", null);
       return false;
  if (!isValidCep(cep)) {
       showErrorAlert("CEP inválido: 0
CEP deve ter 8 dígitos.", null);
      return false;
  if (!isValidEmail(email)) {
      showErrorAlert("E-mail inválido:
Por favor, insira um e-mail válido.",
null);
      return false;
   return true;
```

Nos códigos:

```
@FXML
private void
handleCadastrar(ActionEvent event) {
   try {
     if
   (!validarCadastro(cnpj.getText().replac
```

```
eAll("[^0-9]", ""), fone.getText(),
inválido: Por favor, insira um e-mail
válido.", null);
                                           cep.getText(), email.getText())) {
                                                      return;
           return;
@FXML
                                           @FXML
private void handleEditar(ActionEvent
                                           private void handleEditar(ActionEvent
event) {
                                           event) {
       String cnpjInput =
                                                  if
cnpj.getText().replaceAll("[^0-9]", "");
                                           (!validarCadastro(cnpj.getText().replac
                                           eAll("[^0-9]", ""), fone.getText(),
       if (!isValidCNPJ(cnpjInput)) {
                                           cep.getText(), email.getText())) {
           showErrorAlert("CNPJ inválido:
                                                      return;
O CNPJ deve ter 14 dígitos.", null);
           return;
(!isValidTelefone(fone.getText())) {
          showErrorAlert("Telefone
           return;
       CadastroFazendaDAO
cadastroFazendaDAO = new
CadastroFazendaDBDAO();
(CadastroFazendaDAO.cnpjExists(cnpjInput))
           showErrorAlert("CNPJ inválido:
Este CNPJ já está cadastrado!", null);
       if (!isValidCep(cep.getText())) {
           showErrorAlert("CEP inválido:
O CEP deve ter 8 dígitos.", null);
```

```
(!isValidEmail(email.getText())) {
              showErrorAlert("E-mail
válido.", null);
                   Classe:
Nº refatoração: 2 | CadastroFazendaController.java
                                                      Nome: Extract Method
                                                       Código depois de refatorar:
Código antes de refatorar:
@FXML
                                                       private CadastroFazenda criarFazenda() {
private void handleCadastrar(ActionEvent event) {
                                                        CadastroFazenda fazenda = new
                                                       CadastroFazenda();
 try {
                                                        fazenda.setCnpj(cnpj.getText());
                                                        fazenda.setNome(nomeFazenda.getText());
    CadastroFazenda fazenda = new
                                                        fazenda.setRamo(ramo.getText());
CadastroFazenda():
                                                        fazenda.setTelefone(fone.getText());
    fazenda.setCnpj(cnpj.getText());
                                                        fazenda.setEmail(email.getText());
   fazenda.setNome(nomeFazenda.getText());
                                                        fazenda.setCep(cep.getText());
   fazenda.setRamo(ramo.getText());
                                                        fazenda.setRua(rua.getText());
   fazenda.setTelefone(fone.getText());
   fazenda.setEmail(email.getText());
                                                       fazenda.setnCasa(Integer.parseInt(nCasa.getText()));
                                                        fazenda.setBairro(bairro.getText());
   fazenda.setCep(cep.getText());
    fazenda.setRua(rua.getText());
                                                        fazenda.setCidade(cidade.getText());
                                                        fazenda.setEstado(choiceBoxEstado.getValue());
fazenda.setnCasa(Integer.parseInt(nCasa.getText()));
                                                        return fazenda;
    fazenda.setBairro(bairro.getText());
   fazenda.setCidade(cidade.getText());
    fazenda.setEstado(choiceBoxEstado.getValue());
                                                      Nos códigos:
    cadastroFazendaDAO.insere(fazenda);
                                                       @FXML
                                                       private void handleCadastrar(ActionEvent event) {
    showAlert("Cadastro de Fazenda", "Fazenda
                                                        try {
cadastrada com sucesso!");
 } catch (SQLException e) {
    showErrorAlert("Erro ao cadastrar fazenda: " +
                                                          CadastroFazenda fazenda = criarFazenda();
```

```
e.getMessage(), e);
 } catch (NumberFormatException e) {
   showErrorAlert("Por favor, insira valores
numéricos válidos!", e);
@FXML
private void handleEditar(ActionEvent event) {
 try {
   CadastroFazenda fazenda = new
CadastroFazenda();
   fazenda.setCnpj(cnpj.getText());
   fazenda.setNome(nomeFazenda.getText());
   fazenda.setRamo(ramo.getText());
   fazenda.setTelefone(fone.getText());
   fazenda.setEmail(email.getText());
   fazenda.setCep(cep.getText());
   fazenda.setRua(rua.getText());
fazenda.setnCasa(Integer.parseInt(nCasa.getText()));
   fazenda.setBairro(bairro.getText());
   fazenda.setCidade(cidade.getText());
   fazenda.setEstado(choiceBoxEstado.getValue());
   cadastroFazendaDAO.atualiza(fazenda);
   showAlert("Edição de Fazenda", "Fazenda
atualizada com sucesso!");
 } catch (SQLException e) {
   showErrorAlert("Erro ao editar fazenda: " +
e.getMessage(), e);
 } catch (NumberFormatException e) {
   showErrorAlert("Por favor, insira valores
numéricos válidos!", e);
```

```
CadastroFazendaDAO cadastroFazendaDAO =
new CadastroFazendaDBDAO();
   cadastroFazendaDAO.insere(fazenda);
   showAlert("Cadastro de Fazenda", "Fazenda
cadastrada com sucesso!");
 } catch (SQLException e) {
   showErrorAlert("Erro ao cadastrar fazenda: " +
e.getMessage(), e);
 } catch (NumberFormatException e) {
   showErrorAlert("Por favor, insira valores
numéricos válidos!", e);
@FXML
private void handleEditar(ActionEvent event) {
 try {
    CadastroFazenda fazenda = criarFazenda();
    CadastroFazendaDAO cadastroFazendaDAO =
new CadastroFazendaDBDAO();
   cadastroFazendaDAO.atualiza(fazenda);
   showAlert("Edição de Fazenda", "Fazenda
atualizada com sucesso!");
 } catch (SQLException e) {
   showErrorAlert("Erro ao editar fazenda: " +
e.getMessage(), e);
 } catch (NumberFormatException e) {
   showErrorAlert("Por favor, insira valores
numéricos válidos!", e);
```

	Classe: CadastroFazendaController.java	
Nº refatoração: 3	CadastroAnimalController.java	Nome: Extract Method
Código antes de refatorar:		Código depois de refatorar:
<pre>@FXML private void handleTelaInicial(ActionEvent event) { ((Stage) ((Node) event.getSource()).getScene().getWindow()).close(); abrirNovaTela("/controle/animal/view/TelaInicial.fxml</pre>		<pre>private void handleTela(ActionEvent event, String caminhoFXML, String titulo) { ((Stage) ((Node) event.getSource()).getScene().getWindow()).close(); abrirNovaTela(caminhoFXML, titulo); }</pre>
", "Tela Inicial");		@FXML
} @FXML		<pre>private void handleTelaInicial(ActionEvent event) { handleTela(event,</pre>
<pre>private void handle({</pre>	CadastroAnimal(ActionEvent event)	"/controle/animal/view/TelaInicial.fxml", "Tela Inicial") ;
((Stage) ((Node) event.getSource()).getScene().getWindow()).close();		} @FXML
abrirNovaTela("/controle/animal/view/CadastroAnimal.fxml", "Cadastro de Animal"); }		<pre>private void handleCadastroAnimal(ActionEvent event) { handleTela(event,</pre>
@FXML private void handleCadastroFazenda(ActionEvent		"/controle/animal/view/CadastroAnimal.fxml", "Cadastro de Animal"); }
event) { ((Stage) ((Node)		@FXML
event.getSource()).getScene().getWindow()).close();		<pre>private void handleCadastroFazenda(ActionEvent event) {</pre>
abrirNovaTela("/controle/animal/view/CadastroFazen da.fxml", "Cadastro de Fazenda"); } @FXML		handleTela(event, "/controle/animal/view/CadastroFazenda.fxml", "Cadastro de Fazenda"); }
private void handlePesquisaAnimal(ActionEvent event)		<pre>@FXML private void handlePesquisaAnimal(ActionEvent event) {</pre>

event.getSource()).getScene().getWindow()).close(); abrirNovaTela("/controle/animal/view/PesquisaAnima l.fxml", "Pesquisa de Animal"); }	handleTela(event, "/controle/animal/view/PesquisaAnimal.fxml", "Pesquisa de Animal"); }
Nº refatoração: 4 Classe: TelaInicialController.java	Nome: Refatoração Geral
Código antes de refatorar:	Código depois de refatorar:
<pre>@FXML private ImageView imagem; @FXML private ImageView imagem1;</pre>	@FXML private ImageView imagem;