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.

<table>
<tr><td>Nº refatoração: 1</td><td>Classe:
CadastroAnimalDBDAO</td><td>Nome: Refatoração Geral do código</td></tr>
<tr><td colspan="2">Código antes de refatorar:</td><td>Código depois de refatorar:</td></tr>
<tr><td colspan="2">

```
package controle.animal.dao;

import
controle.animal.model.CadastroAnim
al;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class CadastroAnimalDBDAO
implements CadastroAnimalDAO {
   private String sql;
   private Connection connection;
   private PreparedStatement
statement;
   private ResultSet result;

   private void open() throws
SQLException {
       connection =
Conexao.getConnection(Conexao.url,
Conexao.usuario, Conexao.senha);
   }

   private void close() throws
SQLException {
       if (connection != null &&
!connection.isClosed()) {
           connection.close();
       }
```

</td><td>

```
package controle.animal.dao;

import
controle.animal.model.Cadastr
oAnimal;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class
CadastroAnimalDBDAO
implements CadastroAnimalDAO
{
   private String sql;
   private Connection
connection;
   private PreparedStatement
statement;
   private ResultSet result;

   private void open() throws
SQLException {
       connection =
Conexao.getConnection(Conexao
.url, Conexao.usuario,
Conexao.senha);
   }

   private void close()
throws SQLException {
```

</td></tr>
</table>

```
    }

    @Override
    public void
insere(CadastroAnimal animal)
throws SQLException {
        open();
        sql = "INSERT INTO
CadastroAnimal (animalId,
fazendaId, raca, peso, rebanho, "
+
                "procedencia,
especie, sexo, alimentacao,
especificacao, dataNascimento,
dataChegada) VALUES (?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?)";
        statement =
connection.prepareStatement(sql);
        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.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();
        sql = "INSERT INTO
CadastroAnimal (animalId,
fazendaId, raca, peso,
rebanho, " +
                "procedencia,
especie, sexo, alimentacao,
especificacao,
dataNascimento, dataChegada)
VALUES (?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?)";
        statement =
connection.prepareStatement(s
ql);
        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,
Date.valueOf(animal.getDataChegada
()));
        statement.executeUpdate();
        close();
    }

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

```
        statement.setString(9,
animal.getAlimentacao());

statement.setString(10,
animal.getEspecificacao());
        statement.setDate(11,
Date.valueOf(animal.getDataNa
scimento()));
        statement.setDate(12,
Date.valueOf(animal.getDataCh
egada()));

statement.executeUpdate();
        close();
    }

    @Override
    public void
atualiza(CadastroAnimal
animal) throws SQLException {
        open();
        sql = "UPDATE
CadastroAnimal SET
fazendaId=?, raca=?, peso=?,
rebanho=?, " +

"procedencia=?, especie=?,
sexo=?, alimentacao=?,
especificacao=?,
dataNascimento=?,
dataChegada=? " +
                "WHERE
animalId=?";
        statement =
connection.prepareStatement(s
ql);
        statement.setString(1,
animal.getFazendaId());
        statement.setString(2,
animal.getRaca());
        statement.setFloat(3,
```

```
Date.valueOf(animal.getDataNascimento()));
        statement.setDate(11, Date.valueOf(animal.getDataChegada()));
        statement.setInt(12, animal.getAnimalId());

        statement.executeUpdate();
        close();
    }

    @Override
    public void remove(CadastroAnimal animal) throws SQLException {
        open();
        sql = "DELETE FROM CadastroAnimal WHERE animalId=?";
        statement = connection.prepareStatement(sql);
        statement.setInt(1, animal.getAnimalId());
        statement.executeUpdate();
        close();
    }

    @Override
    public CadastroAnimal buscaPorId(int animalId) throws SQLException {
        open();
        sql = "SELECT * FROM CadastroAnimal WHERE animalId=?";
        statement = connection.prepareStatement(sql);
        statement.setInt(1, animalId);
        result = statement.executeQuery();
```

```
animal.getPeso());
        statement.setString(4, animal.getRebanho());
        statement.setString(5, animal.getProcedencia());
        statement.setString(6, animal.getEspecie());
        statement.setString(7, animal.getSexo());
        statement.setString(8, animal.getAlimentacao());
        statement.setString(9, animal.getEspecificacao());
        statement.setDate(10, Date.valueOf(animal.getDataNascimento()));
        statement.setDate(11, Date.valueOf(animal.getDataChegada()));
        statement.setInt(12, animal.getAnimalId());


statement.executeUpdate();
        close();
    }

    @Override
    public void remove(CadastroAnimal animal) throws SQLException {
        open();
        sql = "DELETE FROM CadastroAnimal WHERE animalId=?";
        statement = connection.prepareStatement(sql);
        statement.setInt(1, animal.getAnimalId());
```

```
        CadastroAnimal animal =
null;
        if (result.next()) {
            animal = new
CadastroAnimal();

animal.setAnimalId(result.getInt("
animalId"));

animal.setFazendaId(result.getStri
ng("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.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
```

```
statement.executeUpdate();
        close();
    }

    @Override
    public CadastroAnimal
buscaPorId(int animalId)
throws SQLException {
        open();
        sql = "SELECT * FROM
CadastroAnimal WHERE
animalId=?";
        statement =
connection.prepareStatement(s
ql);
        statement.setInt(1,
animalId);
        result =
statement.executeQuery();

        CadastroAnimal animal
= null;
        if (result.next()) {
            animal = new
CadastroAnimal();

animal.setAnimalId(result.get
Int("animalId"));

animal.setFazendaId(result.ge
tString("fazendaId"));

animal.setRaca(result.getStri
ng("raca"));

animal.setPeso(result.getFloa
t("peso"));

animal.setRebanho(result.getS
tring("rebanho"));
```

```
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()) {
            CadastroAnimal animal =
new CadastroAnimal();

animal.setAnimalId(result.getInt("
animalId"));

animal.setFazendaId(result.getStri
ng("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());

animal.setDataChegada(result.
getDate("dataChegada").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;
    }
}
```

Nº refatoração: 2	Classe:TelaInicialController	Nome: Refatoração Geral do código

Código antes de refatorar:

```
package
controle.animal.controller;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import
javafx.scene.control.Button;
import javafx.scene.image.Image;
```

Código depois de refatorar:

```
package
controle.animal.controller;
import
javafx.event.ActionEvent;
import javafx.fxml.FXML;
import
javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import
javafx.scene.control.Alert;
```

```
import
javafx.scene.image.ImageView;
import javafx.stage.Stage;

import java.io.IOException;

public class TelaInicialController
{

   @FXML
   private Button
btnCadastrarFazenda;

   @FXML
   private Button
btnCadastroAnimal;

   @FXML
   private Button
btnPesquisaAnimal;

   @FXML
   private Button btnTelaInicial;

   @FXML
   private ImageView imagem;

   @FXML
   private ImageView imagem1;

   @FXML
   public void initialize() {
       Image img = new
Image(getClass().getResourceAsStre
am("/controle/animal/imagem/Logo.j
peg"));
       imagem.setImage(img);

   }

   @FXML
```

```
import
javafx.scene.control.Button;
import
javafx.scene.image.Image;
import
javafx.scene.image.ImageView;
import javafx.stage.Stage;

import java.io.IOException;

public class
TelaInicialController {

   @FXML
   private Button
btnCadastrarFazenda;

   @FXML
   private Button
btnCadastroAnimal;

   @FXML
   private Button
btnPesquisaAnimal;

   @FXML
   private Button
btnTelaInicial;

   @FXML
   private ImageView imagem;

   @FXML
   private ImageView imagem1;

   @FXML
   public void initialize() {
       Image img = new
Image(getClass().getResourceA
sStream("/controle/animal/ima
gem/Logo.jpeg"));
```

```
    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
ew/CadastroFazenda.fxml",
"Cadastro de Fazenda");
    }

    @FXML
    private void
handlePesquisaAnimal(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
    private void
handleCadastroAnimal(ActionEv
ent event) {
        ((Stage) ((Node)
event.getSource()).getScene()
.getWindow()).close();

abrirNovaTela("/controle/anim
al/view/CadastroAnimal.fxml",
"Cadastro de Animal");
    }

    @FXML
    private void
handleCadastroFazenda(ActionE
vent event) {
        ((Stage) ((Node)
event.getSource()).getScene()
.getWindow()).close();

abrirNovaTela("/controle/anim
al/view/CadastroFazenda.fxml"
, "Cadastro de Fazenda");
```

```
event) {
        ((Stage) ((Node)
event.getSource()).getScene().getW
indow()).close();

abrirNovaTela("/controle/animal/vi
ew/PesquisaAnimal.fxml", "Pesquisa
de Animal");
    }

    private void
abrirNovaTela(String caminhoFXML,
String titulo) {
        try {
            FXMLLoader loader = new
FXMLLoader(getClass().getResource(
caminhoFXML));
            Parent root =
loader.load();
            Stage newStage = new
Stage();
            Scene newScene = new
Scene(root);


newStage.setScene(newScene);

newStage.setTitle(titulo);

newStage.setResizable(false);
            newStage.show();
        } catch (IOException e) {
            e.printStackTrace();
            showErrorAlert("Erro ao
abrir a tela: " + titulo, e);
        }
    }

    private void
showErrorAlert(String message,
Exception e) {
    }

    @FXML
    private void
handlePesquisaAnimal(ActionEv
ent event) {
        ((Stage) ((Node)
event.getSource()).getScene()
.getWindow()).close();

abrirNovaTela("/controle/anim
al/view/PesquisaAnimal.fxml",
"Pesquisa de Animal");
    }

    private void
abrirNovaTela(String
caminhoFXML, String titulo) {
        try {
            FXMLLoader loader
= new
FXMLLoader(getClass().getReso
urce(caminhoFXML));
            Parent root =
loader.load();
            Stage newStage =
new Stage();
            Scene newScene =
new Scene(root);


newStage.setScene(newScene);

newStage.setTitle(titulo);

newStage.setResizable(false);
            newStage.show();
        } catch (IOException
e) {

e.printStackTrace();
```

```
        Alert alert = new
Alert(Alert.AlertType.ERROR);
        alert.setTitle("Erro");
        alert.setHeaderText(null);

alert.setContentText(message);
        alert.setResizable(true);
        alert.setHeight(150);
        alert.setWidth(300);
        alert.showAndWait();
    }

    private void showAlert(String
title, String message) {
        Alert alert = new
Alert(Alert.AlertType.INFORMATION)
;
        alert.setTitle(title);
        alert.setHeaderText(null);

alert.setContentText(message);
        alert.showAndWait();
    }



}
```

```
showErrorAlert("Erro ao abrir
a tela: " + titulo, e);
        }
    }

    private void
showErrorAlert(String
message, Exception e) {
        Alert alert = new
Alert(Alert.AlertType.ERROR);

alert.setTitle("Erro");

alert.setHeaderText(null);

alert.setContentText(message)
;

alert.setResizable(true);
        alert.setHeight(150);
        alert.setWidth(300);
        alert.showAndWait();
    }

}
```

Nº refatoração: 3	Classe: CadastroFazendaDBDAO	Nome:Tratamento de Exceções e Gerenciamento de Recursos

Código antes de refatorar:

```
@Override
public void
atualiza(CadastroFazenda fazenda)
throws SQLException {
    open();
    sql = "UPDATE CadastroFazenda
```

Código depois de refatorar:

```
@Override
public void
atualiza(CadastroFazenda
fazenda) throws SQLException
{
    try {
```

```
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
(null, "O nome da fazenda não
pode ser vazio.", "Erro de
Validação",
JOptionPane.WARNING_MESSAGE);
            return;
        }
        open();
        sql = "UPDATE
CadastroFazenda SET nome=?,
ramo=?, telefone=?, email=?,
cep=?, rua=?, nCasa=?,
bairro=?, cidade=?, estado=?
" +
                "WHERE
cnpj=?";
        statement =
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,
```

<table>
<tr><td colspan="2"></td><td>

```
fazenda.getCidade());

statement.setString(10,
fazenda.getEstado());

statement.setString(11,
fazenda.getCnpj());

statement.executeUpdate();
        close();
    } catch (SQLException e) {

System.err.println("Erro ao
acessar o banco: " +
e.getMessage());
        e.printStackTrace();
    } finally {
        try {
            close();
        } catch (SQLException
e) {

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

</td></tr>
<tr><td>Nº refatoração: 4</td><td>Classe: TelaInicialController.java</td><td>Nome: Extract Method</td></tr>
<tr><td colspan="2">Código antes de refatorar:</td><td>Código depois de refatorar:</td></tr>
<tr><td colspan="2">

```
@FXML
private void handleTelaInicial(ActionEvent event)
{
    ((Stage) ((Node)
event.getSource()).getScene().getWindow()).close
();
```

</td><td>

```
private void handleTela(ActionEvent
event, String caminhoFXML, String titulo)
{
    ((Stage) ((Node)
event.getSource()).getScene().getWindow(
)).close();
```

</td></tr>
</table>

```
abrirNovaTela("/controle/animal/view/TelaInicial.fxml", "Tela Inicial");
}

@FXML
private void handleCadastroAnimal(ActionEvent event) {
  ((Stage) ((Node) event.getSource()).getScene().getWindow()).close();

abrirNovaTela("/controle/animal/view/CadastroAnimal.fxml", "Cadastro de Animal");
}

@FXML
private void handleCadastroFazenda(ActionEvent event) {
  ((Stage) ((Node) event.getSource()).getScene().getWindow()).close();

abrirNovaTela("/controle/animal/view/CadastroFazenda.fxml", "Cadastro de Fazenda");
}

@FXML
private void handlePesquisaAnimal(ActionEvent event) {
  ((Stage) ((Node) event.getSource()).getScene().getWindow()).close();

abrirNovaTela("/controle/animal/view/PesquisaAnimal.fxml", "Pesquisa de Animal");
}
```

```
  abrirNovaTela(caminhoFXML, titulo);
}
@FXML
private void handleTelaInicial(ActionEvent event) {
  handleTela(event, "/controle/animal/view/TelaInicial.fxml", "Tela Inicial");
}

@FXML
private void handleCadastroAnimal(ActionEvent event) {
  handleTela(event, "/controle/animal/view/CadastroAnimal.fxml", "Cadastro de Animal");
}

@FXML
private void handleCadastroFazenda(ActionEvent event) {
  handleTela(event, "/controle/animal/view/CadastroFazenda.fxml", "Cadastro de Fazenda");
}

@FXML
private void handlePesquisaAnimal(ActionEvent event) {
  handleTela(event, "/controle/animal/view/PesquisaAnimal.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.

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

Código antes de refatorar:	Código depois de refatorar:

```java
package controle.animal.dao;
import
controle.animal.model.CadastroFaz
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 {
       connection =
Conexao.getConnection(Conexao.url
, Conexao.usuario,
Conexao.senha);
   }

   private void close() throws
SQLException {
       if (connection != null &&
!connection.isClosed()) {
           connection.close();
       }
   }

   @Override
   public void
insere(CadastroFazenda fazenda)
throws SQLException {
       open();
```

```java
package controle.animal.dao;

import
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
SQLException {
       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
insere(CadastroFazenda fazenda)
```

```
        sql = "INSERT INTO
CadastroFazenda (nome, cnpj,
ramo, telefone, email, cep, rua,
nCasa, bairro, cidade, estado) "
+
                "VALUES (?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?)";
        statement =
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();
        close();
    }

    @Override
    public void
atualiza(CadastroFazenda fazenda)
throws SQLException {
        open();
        sql = "UPDATE
```

```
throws SQLException {
        open();
        sql = "INSERT INTO
CadastroFazenda (nome, cnpj,
ramo, telefone, email, cep,
rua, nCasa, bairro, cidade,
estado) " +
                "VALUES (?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?)";
        statement =
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();
        close();
    }

    @Override
    public void
```

```
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();
    }

    @Override
    public void
remove(CadastroFazenda fazenda)
throws SQLException {
        open();
        sql = "DELETE FROM
CadastroFazenda WHERE cnpj=?";
        statement =
```

```
atualiza(CadastroFazenda
fazenda) 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();
    }

    @Override
```

```
connection.prepareStatement(sql);
        statement.setString(1,
fazenda.getCnpj());
        statement.executeUpdate();
        close();
    }

    @Override
    public CadastroFazenda
buscaPorCNPJ(String cnpj) throws
SQLException {
        open();
        sql = "SELECT * FROM
CadastroFazenda WHERE cnpj=?";
        statement =
connection.prepareStatement(sql);
        statement.setString(1,
cnpj);
        result =
statement.executeQuery();

        CadastroFazenda fazenda =
null;
        if (result.next()) {
            fazenda = new
CadastroFazenda();

fazenda.setNome(result.getString(
"nome"));

fazenda.setCnpj(result.getString(
"cnpj"));

fazenda.setRamo(result.getString(
"ramo"));

fazenda.setTelefone(result.getStr
ing("telefone"));

fazenda.setEmail(result.getString
("email"));
```

```
    public void
remove(CadastroFazenda fazenda)
throws SQLException {
        open();
        sql = "DELETE FROM
CadastroFazenda WHERE cnpj=?";
        statement =
connection.prepareStatement(sql
);
        statement.setString(1,
fazenda.getCnpj());

statement.executeUpdate();
        close();
    }

    @Override
    public CadastroFazenda
buscaPorCNPJ(String cnpj)
throws SQLException {
        open();
        sql = "SELECT * FROM
CadastroFazenda WHERE cnpj=?";
        statement =
connection.prepareStatement(sql
);
        statement.setString(1,
cnpj);
        result =
statement.executeQuery();

        CadastroFazenda fazenda
= null;
        if (result.next()) {
            fazenda = new
CadastroFazenda();

fazenda.setNome(result.getStrin
g("nome"));

fazenda.setCnpj(result.getStrin
```

```
fazenda.setCep(result.getString("
cep"));

fazenda.setRua(result.getString("
rua"));

fazenda.setnCasa(result.getInt("n
Casa"));

fazenda.setBairro(result.getStrin
g("bairro"));

fazenda.setCidade(result.getStrin
g("cidade"));

fazenda.setEstado(result.getStrin
g("estado"));
        }
        close();
        return fazenda;
    }

    @Override
    public List<CadastroFazenda>
listaTodos() throws SQLException
{
        open();
        sql = "SELECT * FROM
CadastroFazenda";
        statement =
connection.prepareStatement(sql);
        result =
statement.executeQuery();

        List<CadastroFazenda>
fazendas = new ArrayList<>();
        while (result.next()) {
            CadastroFazenda
fazenda = new CadastroFazenda();
```

```
g("cnpj"));

fazenda.setRamo(result.getStrin
g("ramo"));

fazenda.setTelefone(result.getS
tring("telefone"));

fazenda.setEmail(result.getStri
ng("email"));

fazenda.setCep(result.getString
("cep"));

fazenda.setRua(result.getString
("rua"));

fazenda.setnCasa(result.getInt(
"nCasa"));

fazenda.setBairro(result.getStr
ing("bairro"));

fazenda.setCidade(result.getStr
ing("cidade"));

fazenda.setEstado(result.getStr
ing("estado"));
        }
        close();
        return fazenda;
    }
    public static boolean
cnpjExists(String cnpj) throws
SQLException {
        String query = "SELECT
COUNT(*) FROM fazendas WHERE
cnpj = ?";
        try (PreparedStatement
statement =
connection.prepareStatement(que
```

```
fazenda.setNome(result.getString("nome"));

fazenda.setCnpj(result.getString("cnpj"));

fazenda.setRamo(result.getString("ramo"));

fazenda.setTelefone(result.getString("telefone"));

fazenda.setEmail(result.getString("email"));

fazenda.setCep(result.getString("cep"));

fazenda.setRua(result.getString("rua"));

fazenda.setnCasa(result.getInt("nCasa"));

fazenda.setBairro(result.getString("bairro"));

fazenda.setCidade(result.getString("cidade"));

fazenda.setEstado(result.getString("estado"));
            fazendas.add(fazenda);
        }
        close();
        return fazendas;
    }

    public static boolean cnpjExists(String cnpj) throws SQLException {
```

```
ry)) {

statement.setString(1, cnpj);
            ResultSet resultSet = statement.executeQuery();
            if (resultSet.next()) {
                return resultSet.getInt(1) > 0;
            }
        }
        return false;
    }

} // foi retirado o método da interface também
```

```
        String query = "SELECT
COUNT(*) FROM fazendas WHERE
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;
    }

}
```

Nº refatoração: 2	Classe: PesquisaAnimalController.java	Nome: Extract Method

Código antes de refatorar:	Código depois de refatorar:

`@FXML`
`private void handleTelaInicial(ActionEvent event) {`
`  ((Stage) ((Node) event.getSource()).getScene().getWindow()).close();`
 	
`abrirNovaTela("/controle/animal/view/TelaInicial.fxml", "Tela Inicial");`
`}`
 	
`@FXML`
`private void`
	`private void handleTela(ActionEvent event, String caminhoFXML, String titulo) {`
	`  ((Stage) ((Node) event.getSource()).getScene().getWindow()).close();`
	`  abrirNovaTela(caminhoFXML, titulo);`
	`}`
	`@FXML`
	`private void handleTelaInicial(ActionEvent event) {`
	`  handleTela(event, "/controle/animal/view/TelaInicial.fxml", "Tela Inicial");`

<table>
<tr><td>

```
handleCadastroAnimal(ActionEvent event) {
  ((Stage) ((Node)
event.getSource()).getScene().getWindow()).clo
se();

abrirNovaTela("/controle/animal/view/Cadast
roAnimal.fxml", "Cadastro de Animal");
}

@FXML
private void
handleCadastroFazenda(ActionEvent event) {
  ((Stage) ((Node)
event.getSource()).getScene().getWindow()).clo
se();

abrirNovaTela("/controle/animal/view/Cadast
roFazenda.fxml", "Cadastro de Fazenda");
}

@FXML
private void
handlePesquisaAnimal(ActionEvent event) {
  ((Stage) ((Node)
event.getSource()).getScene().getWindow()).clo
se();

abrirNovaTela("/controle/animal/view/Pesqui
saAnimal.fxml", "Pesquisa de Animal");
}
```

</td><td>

```
}

@FXML
private void
handleCadastroAnimal(ActionEvent event) {
  handleTela(event,
"/controle/animal/view/CadastroAnimal.fx
ml", "Cadastro de Animal");
}

@FXML
private void
handleCadastroFazenda(ActionEvent event)
{
  handleTela(event,
"/controle/animal/view/CadastroFazenda.
fxml", "Cadastro de Fazenda");
}

@FXML
private void
handlePesquisaAnimal(ActionEvent event) {
  handleTela(event,
"/controle/animal/view/PesquisaAnimal.fx
ml", "Pesquisa de Animal");
}
```

</td></tr>
<tr><td>Nº refatoração:3</td><td>Classe:CadastroFazenda Controller</td><td>Nome:Refatoração Geral</td></tr>
<tr><td colspan="2">Código antes de refatorar:</td><td>Código depois de refatorar:</td></tr>
<tr><td colspan="2">

```
@FXML
private void
handleCadastrar(ActionEvent
```

</td><td>

```
private boolean isValidNomeFazenda(String
nomeFazenda) {
  return nomeFazenda != null &&
!nomeFazenda.trim().isEmpty();
```

</td></tr>
</table>

```
event) {
   try {
       String cnpjInput =
cnpj.getText().replaceAll("[^0-9]
", "");

  if (!isValidCNPJ(cnpjInput)) {
           showErrorAlert("CNPJ
inválido: O CNPJ deve ter 14
dígitos.", null);
           return;
       }

       if
(!isValidTelefone(fone.getText())
) {

showErrorAlert("Telefone
inválido: O telefone deve ter
entre 10 e 11 dígitos.", null);
           return;
       }

       CadastroFazendaDAO
cadastroFazendaDAO = new
CadastroFazendaDBDAO();
       if
(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;
```

```
}
```

<table>
<tr><td colspan="2">

```
        }

        if
(!isValidEmail(email.getText()))
{
            showErrorAlert("E-mail
inválido: Por favor, insira um
e-mail válido.", null);
            return;
        }
```

Não existia código para tratar esse erro.

</td><td></td></tr>
<tr><td>Nº refatoração: 4</td><td>Classe:
CadastroFazendaDBDAO</td><td>Nome: Tratamento de Exceções e Gerenciamento de Recursos</td></tr>
<tr><td colspan="2">Código antes de refatorar:</td><td>Código depois de refatorar:</td></tr>
<tr><td colspan="2">

```
@Override
public void
insere(CadastroFazenda fazenda)
throws SQLException {
    open();
    sql = "INSERT INTO
CadastroFazenda (nome, cnpj,
ramo, telefone, email, cep, rua,
nCasa, bairro, cidade, estado) "
+
            "VALUES (?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?)";
    statement =
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,
```

</td><td>

```
@Override
public void
insere(CadastroFazenda fazenda)
throws SQLException {
    try {
        if (fazenda.getNome() ==
null ||
fazenda.getNome().isEmpty()) {

JOptionPane.showMessageDialog(n
ull, "O nome da fazenda não
pode ser vazio.", "Erro de
Validação",
JOptionPane.WARNING_MESSAGE);
            return;  //
Interrompe a execução se o nome
for vazio
        }
        open();
        sql = "INSERT INTO
CadastroFazenda (nome, cnpj,
ramo, telefone, email, cep,
rua, nCasa, bairro, cidade,
```

</td></tr>
</table>

```
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();
}
```

```
estado) " +
                "VALUES (?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?)";
       statement =
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();  //
Opcional para ver a pilha
completa
   } finally {
       try {
           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

	CadastroFazendaController.java	
Código antes de refatorar:		Código depois de refatorar:

```
@FXML
private void handleCadastrar(ActionEvent
event) {
   try {
       String cnpjInput =
cnpj.getText().replaceAll("[^0-9]", "");

       if (!isValidCNPJ(cnpjInput)) {
           showErrorAlert("CNPJ inválido:
O CNPJ deve ter 14 dígitos.", null);
           return;
       }

       if
(!isValidTelefone(fone.getText())) {
           showErrorAlert("Telefone
inválido: O telefone deve ter entre 10 e
11 dígitos.", null);
           return;
       }

       CadastroFazendaDAO
cadastroFazendaDAO = new
CadastroFazendaDBDAO();
       if
(controle.animal.dao.CadastroFazendaDAO.c
npjExists(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;
       }

       if
(!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: O
CNPJ deve ter 14 dígitos.", null);
       return false;
   }
   if (!isValidTelefone(telefone)) {
       showErrorAlert("Telefone
inválido: O telefone deve ter entre 10
e 11 dígitos.", null);
       return false;
   }
   if
(CadastroFazendaDAO.cnpjExists(cnpj)) {
       showErrorAlert("CNPJ inválido:
Este CNPJ já está cadastrado!", null);
       return false;
   }
   if (!isValidCep(cep)) {
       showErrorAlert("CEP inválido: O
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
```

```
inválido: Por favor, insira um e-mail
válido.", null);
            return;
        }
        ....
```

```
eAll("[^0-9]", ""), fone.getText(),
cep.getText(), email.getText())) {
            return;
        }
. . .
```

```
@FXML
private void handleEditar(ActionEvent
event) {
    try {
        String cnpjInput =
cnpj.getText().replaceAll("[^0-9]", "");

        if (!isValidCNPJ(cnpjInput)) {
            showErrorAlert("CNPJ inválido:
O CNPJ deve ter 14 dígitos.", null);
            return;
        }

        if
(!isValidTelefone(fone.getText())) {
            showErrorAlert("Telefone
inválido: O telefone deve ter entre 10 e
11 dígitos.", null);
            return;
        }

        CadastroFazendaDAO
cadastroFazendaDAO = new
CadastroFazendaDBDAO();
        if
(CadastroFazendaDAO.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;
        }
```

```
@FXML
private void handleEditar(ActionEvent
event) {
    try {
        if
(!validarCadastro(cnpj.getText().replac
eAll("[^0-9]", ""), fone.getText(),
cep.getText(), email.getText())) {
            return;
        }
. . .
```

```
        if
(!isValidEmail(email.getText())) {
            showErrorAlert("E-mail
inválido: Por favor, insira um e-mail
válido.", null);
            return;
        }
        ....
```

Nº refatoração: 2	Classe: CadastroFazendaController.java	Nome: Extract Method

Código antes de refatorar:

```
@FXML
private void handleCadastrar(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.insere(fazenda);

     showAlert("Cadastro de Fazenda", "Fazenda
cadastrada com sucesso!");
  } catch (SQLException e) {
     showErrorAlert("Erro ao cadastrar fazenda: " +
```

Código depois de refatorar:

```
private CadastroFazenda criarFazenda() {
  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());
  return fazenda;
}
```

Nos códigos:

```
@FXML
private void handleCadastrar(ActionEvent event) {
  try {
     . . .

     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);
  }
}
```

Nº refatoração: 3	Classe: CadastroFazendaController.java CadastroAnimalController.java	Nome: Extract Method
Código antes de refatorar:		Código depois de refatorar:

```
@FXML
private void handleTelaInicial(ActionEvent event) {
  ((Stage) ((Node)
event.getSource()).getScene().getWindow()).close();

abrirNovaTela("/controle/animal/view/TelaInicial.fxml
", "Tela Inicial");
}

@FXML
private void handleCadastroAnimal(ActionEvent event)
{
  ((Stage) ((Node)
event.getSource()).getScene().getWindow()).close();

abrirNovaTela("/controle/animal/view/CadastroAnim
al.fxml", "Cadastro de Animal");
}

@FXML
private void handleCadastroFazenda(ActionEvent
event) {
  ((Stage) ((Node)
event.getSource()).getScene().getWindow()).close();

abrirNovaTela("/controle/animal/view/CadastroFazen
da.fxml", "Cadastro de Fazenda");
}

@FXML
private void handlePesquisaAnimal(ActionEvent event)
{
  ((Stage) ((Node)
```

```
private void handleTela(ActionEvent event, String
caminhoFXML, String titulo) {
  ((Stage) ((Node)
event.getSource()).getScene().getWindow()).close();
  abrirNovaTela(caminhoFXML, titulo);
}

@FXML
private void handleTelaInicial(ActionEvent event) {
  handleTela(event,
"/controle/animal/view/TelaInicial.fxml", "Tela
Inicial");
}

@FXML
private void handleCadastroAnimal(ActionEvent
event) {
  handleTela(event,
"/controle/animal/view/CadastroAnimal.fxml",
"Cadastro de Animal");
}

@FXML
private void handleCadastroFazenda(ActionEvent
event) {
  handleTela(event,
"/controle/animal/view/CadastroFazenda.fxml",
"Cadastro de Fazenda");
}

@FXML
private void handlePesquisaAnimal(ActionEvent
event) {
```

<table>
<tr><td>

```
event.getSource()).getScene().getWindow()).close();

abrirNovaTela("/controle/animal/view/PesquisaAnimal.fxml", "Pesquisa de Animal");
}
```

</td><td>

```
  handleTela(event, "/controle/animal/view/PesquisaAnimal.fxml", "Pesquisa de Animal");
}
```

</td></tr>
<tr><td>Nº refatoração: 4</td><td>Classe: TelaInicialController.java</td><td>Nome: Refatoração Geral</td></tr>
<tr><td colspan="2">Código antes de refatorar:</td><td>Código depois de refatorar:</td></tr>
<tr><td colspan="2">

```
@FXML
private ImageView imagem;

@FXML
private ImageView imagem1;
```

</td><td>

```
@FXML
private ImageView imagem;
```

</td></tr>
</table>