

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Paulo Kiyoshi Oyama Filho

Prática 03 - Uso de APIs BD e JUnit

1. Código

Entity

Livros Entity

```
package model;

import javax.persistence.*;

@Entity
@Table(name = "Livros")
@NamedQueries({
    @NamedQuery(name = "Livros.findByDesc",
        query = "SELECT l FROM Livros b WHERE l.title =
: title")
})
public class Livros {

    public Livros(String author, String title, int year) {
        this.author = author;
        this.title = title;
        this.year = year;
    }

    @Id
    @GeneratedValue
    private int id;

    @Column
    public String author;

    @Column
    public String title;

    @Column
    public int year;

    public String getAuthor() {
```

```

        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getTitle() {
        return title;
    }

    this.title = title;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public int getYear() {
    return year;
}

public void setYear(int year) {
    this.year = year;
}

@Override
public String toString() {
    return "Livros{" +
        "id=" + id +
        ", author='" + author + '\'' +
        ", title='" + title + '\'' +
        ", year=" + year +
        '}';
}
}

```

Revista Entity

```
package model;

import javax.persistence.*;

@Entity
@Table(name = "Revistas")
@NamedQueries({
    @NamedQuery(name = "Revista.findByDesc",
        query = "SELECT l FROM Revista b WHERE l.volume = :title")
})
public class Revista {

    @Id
    @GeneratedValue
    private int id;

    @Column
    public String title;

    @Column
    public int year;

    @Column
    public String org;

    @Column
    public int volume;

    @Column
    public int number;

    public Revista(String title, int year, String org, int volume,
int number) {
        this.title = title;
        this.year = year;
        this.org = org;
        this.volume = volume;
        this.number = number;
    }
}
```

```

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public int getYear() {
    return year;
}

public void setYear(int year) {
    this.year = year;
}

public String getOrg() {
    return org;
}

public void setOrg(String org) {
    this.org = org;
}

public int getVolume() {
    return volume;
}

public void setVolume(int volume) {
    this.volume = volume;
}

public int getNumber() {
    return number;
}

public void setNumber(int number) {
    this.number = number;
}
}

```

```
/* -----*/
```

Repository

Livros Repository

```
package repository;

import model.Livros;

import java.util.List;
import java.util.Optional;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class LivroRepository {
    protected EntityManager entityManager;

    private LivroRepository() {
        entityManager = getEntityManager();
    }

    private EntityManager getEntityManager() {
        EntityManagerFactory factory =
Persistence.createEntityManagerFactory("BibliotecaHibernate");
        if (entityManager == null) {
            entityManager = factory.createEntityManager();
        }

        return entityManager;
    }

    public Optional<Livros> save(Livros livro) {
        entityManager.getTransaction().begin();
        entityManager.persist(livro);
        entityManager.getTransaction().commit();
    }
}
```

```

        return Optional.of(livro);
    }

    public List<Livros> findAll() {
        return entityManager.createQuery("FROM " +
            Livros.class.getName()).getResultList();
    }
}

```

Revista Repository

```

package repository;

import model.Revista;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import java.util.List;
import java.util.Optional;

public class RevistaRepository {

    protected EntityManager entityManager;

    private RevistaRepository() {
        entityManager = getEntityManager();
    }

    private EntityManager getEntityManager() {
        EntityManagerFactory factory =
Persistence.createEntityManagerFactory("BibliotecaHibernate");
        if (entityManager == null) {
            entityManager = factory.createEntityManager();
        }
    }
}

```

```

        return entityManager;
    }

    public Optional<Revista> save(Revista revista) {
        entityManager.getTransaction().begin();
        entityManager.persist(revista);
        entityManager.getTransaction().commit();
        return Optional.of(revista);
    }

    public List<Revista> findAll() {
        return entityManager.createQuery("FROM " +
            Revista.class.getName()).getResultList();
    }
}

```

/* -----*/

Controller

Livros Controller

```

package controller;

import model.BaseType;
import model.Listagem;
import model.Livros;
import repository.LivroRepository;
import view.LivroForm;
import java.util.List;

public class LivrosController {

    public Listagem list = new Listagem();

    LivroRepository livroRepository = New LivroRepository();

    public void include(String title, int year, String author) {

```



```

        try{
            Livros l = new Livros(author,title,year);
            livroRepository.save(l);
        }catch (Exception e){
            System.out.println("Something occur during save...");
            System.out.println(e.getMessage());
        }
    }

    public String toList() {

        String textArea = "";

        try {
            for(Livros l : livroRepository.findAll()){
                textArea = textArea.concat(l.toString());
            }
        } catch (Exception e){
            System.out.println("Something occur during toList...");
            System.out.println(e.getMessage());
        }
        return textArea;
    }

    public boolean isFieldsNotNull(String title, int year, String
author) {
        try {
            if (title.isEmpty() || year == 0 || author.isEmpty()) {
                throw new Exception("Alguns campos estão nulos!!");
            } else {
                return true;
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
            return false;
        }
    }
}

```

```
}
```

Revista Controller

```
package controller;

import model.Listagem;
import model.Livros;
import model.Revista;
import model.BaseType;
import repository.LivroRepository;
import repository.RevistaRepository;

public class RevistaController {

    public Listagem list = new Listagem();

    RevistaRepository revistaRepository = New RevistaRepository();

    public void include(String org, int volume, int number, String
title, int year) {

        try{
            Revista l = new Revista(org, volume, title, number,
year);

            revistaRepository.save(l);

        }catch (Exception e){
            System.out.println("Something occur during save
Revista...");
            System.out.println(e.getMessage());
        }

    }

    public String toList() {
```

```

        String textArea = "";

        try {
            for(Revista r : revistaRepository.findAll()){
                textArea = textArea.concat(r.toString());
            }
        } catch (Exception e){
            System.out.println("Something occur during toList
Revista ...");
            System.out.println(e.getMessage());
        }

        return textArea;
    }

    public boolean isFieldsNotNull(String org, int volume, int
number, String title, int year) {
        try {
            if (title.isEmpty() || year == 0 || number == 0 ||
volume == 0 || org.isEmpty()) {
                throw new Exception("Alguns campos estão nulos!!");
            } else {
                return true;
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
            return false;
        }
    }
}
}

```

/* -----*/

View

Livros View

```

package view;

import biblioteca.controller.LivrosController;

```

```

import biblioteca.model.BaseType;
import biblioteca.model.Livros;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;

public class LivroForm
    extends JFrame {

    final protected Container c;
    final public JLabel typeLabel, authorLabel, titleLabel,
yearLabel;
    final public JTextField titleTextField, yearTextField,
authorTextField;
    final public JButton btnAdd, btnType, btnList;
    final LivrosController controller = new LivrosController();

    public LivroForm(String title, String geral) {
        setBounds(300, 90, 900, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setResizable(false);

        c = getContentPane();
        c.setLayout(null);

        typeLabel = new JLabel(title);
        typeLabel.setFont(new Font("Arial", Font.PLAIN, 30));
        typeLabel.setSize(300, 30);
        typeLabel.setLocation(220, 30);
        c.add(typeLabel);

        //FIRST ROW
        titleLabel = new JLabel("Titulo");
        titleLabel.setFont(new Font("Arial", Font.PLAIN, 20));
        titleLabel.setSize(100, 20);
        titleLabel.setLocation(100, 100);
        c.add(titleLabel);

        titleTextField = new JTextField();
        titleTextField.setFont(new Font("Arial", Font.PLAIN, 15));
        titleTextField.setSize(190, 20);
        titleTextField.setLocation(200, 100);
    }

```

```

c.add(titleTextField);

//SECOND ROW
authorLabel = new JLabel(geral);
authorLabel.setFont(new Font("Arial", Font.PLAIN, 20));
authorLabel.setSize(100, 20);
authorLabel.setLocation(100, 150);
c.add(authorLabel);

authorTextField = new JTextField();
authorTextField.setFont(new Font("Arial", Font.PLAIN, 15));
authorTextField.setSize(190, 20);
authorTextField.setLocation(200, 150);
c.add(authorTextField);

//THIRD ROW
yearLabel = new JLabel("Ano");
yearLabel.setFont(new Font("Arial", Font.PLAIN, 20));
yearLabel.setSize(100, 20);
yearLabel.setLocation(100, 200);
c.add(yearLabel);

yearTextField = new JTextField();
yearTextField.setFont(new Font("Arial", Font.PLAIN, 15));
yearTextField.setSize(50, 20);
yearTextField.setLocation(140, 200);
c.add(yearTextField);

//FORTH ROW
btnAdd = new JButton("Incluir");
btnAdd.setFont(new Font("Arial", Font.PLAIN, 15));
btnAdd.setSize(100, 20);
btnAdd.setLocation(100, 250);
btnAdd.addActionListener((ActionEvent e) -> {
    int year = Integer.parseInt(yearTextField.getText());
    controller.isFieldsNotNull(titleTextField.getText(),
year, authorTextField.getText());

    controller.include(titleTextField.getText(), year,
authorTextField.getText());
});
c.add(btnAdd);

```

```

        btnType = new JButton("Revistas");
        btnType.setFont(new Font("Arial", Font.PLAIN, 15));
        btnType.setSize(100, 20);
        btnType.setLocation(220, 250);
        btnType.addActionListener((ActionEvent e) -> {
            setVisible(false);
            RevistaForm re = new RevistaForm("Revista", "Org.");
        });
        c.add(btnType);

        btnList = new JButton("Listagem");
        btnList.setFont(new Font("Arial", Font.PLAIN, 15));
        btnList.setSize(100, 20);
        btnList.setLocation(340, 250);
        btnList.addActionListener((ActionEvent e) -> {
            setVisible(false);
            ListagemView li = new ListagemView(controller.toList());
        });
        c.add(btnList);

        setVisible(true);
    }

    public static void main(String[] args) {

        LivroForm f = new LivroForm("Titulo", "Autor");

    }
}

```

Revista View

```

package view;

import biblioteca.controller.RevistaController;
import biblioteca.model.BaseType;
import biblioteca.model.Livros;
import biblioteca.model.Revista;
import javax.swing.*;

```

```

import java.awt.*;
import java.awt.event.ActionEvent;

public class RevistaForm extends JFrame {

    final protected Container c;
    final public JLabel typeLabel, geralLabel, titleLabel,
yearLabel;
    final public JTextField titleTextField, yearTextField,
geralTextField;
    final public JButton btnAdd, btnType, btnList;
    final private JLabel numberLabel, volumeLabel;
    final private JTextField numberTextField, volumeTextField;
    final RevistaController controller = new RevistaController();

    public RevistaForm(String title, String geral) {

        setBounds(300, 90, 900, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setResizable(false);

        c = getContentPane();
        c.setLayout(null);

        typeLabel = new JLabel(title);
        typeLabel.setFont(new Font("Arial", Font.PLAIN, 30));
        typeLabel.setSize(300, 30);
        typeLabel.setLocation(220, 30);
        c.add(typeLabel);

        //FIRST ROW
        titleLabel = new JLabel("Titulo");
        titleLabel.setFont(new Font("Arial", Font.PLAIN, 20));
        titleLabel.setSize(100, 20);
        titleLabel.setLocation(100, 100);
        c.add(titleLabel);

        titleTextField = new JTextField();
        titleTextField.setFont(new Font("Arial", Font.PLAIN, 15));
        titleTextField.setSize(190, 20);
        titleTextField.setLocation(200, 100);
        c.add(titleTextField);
    }
}

```

```

//SECOND ROW
geralLabel = new JLabel(geral);
geralLabel.setFont(new Font("Arial", Font.PLAIN, 20));
geralLabel.setSize(100, 20);
geralLabel.setLocation(100, 150);
c.add(geralLabel);

geralTextField = new JTextField();
geralTextField.setFont(new Font("Arial", Font.PLAIN, 15));
geralTextField.setSize(190, 20);
geralTextField.setLocation(200, 150);
c.add(geralTextField);

//THIRD ROW
yearLabel = new JLabel("Ano");
yearLabel.setFont(new Font("Arial", Font.PLAIN, 20));
yearLabel.setSize(100, 20);
yearLabel.setLocation(100, 200);
c.add(yearLabel);

yearTextField = new JTextField();
yearTextField.setFont(new Font("Arial", Font.PLAIN, 15));
yearTextField.setSize(50, 20);
yearTextField.setLocation(140, 200);
c.add(yearTextField);

numberLabel = new JLabel("Nro.");
numberLabel.setFont(new Font("Arial", Font.PLAIN, 20));
numberLabel.setSize(100, 20);
numberLabel.setLocation(220, 200);
c.add(numberLabel);

numberTextField = new JTextField();
numberTextField.setFont(new Font("Arial", Font.PLAIN, 15));
numberTextField.setSize(50, 20);
numberTextField.setLocation(260, 200);
c.add(numberTextField);

volumeLabel = new JLabel("Vol.");
volumeLabel.setFont(new Font("Arial", Font.PLAIN, 20));
volumeLabel.setSize(100, 20);

```



```

        volumeLabel.setLocation(340, 200);
        c.add(volumeLabel);

        volumeTextField = new JTextField();
        volumeTextField.setFont(new Font("Arial", Font.PLAIN, 15));
        volumeTextField.setSize(50, 20);
        volumeTextField.setLocation(380, 200);
        c.add(volumeTextField);

        //FORTH ROW
        btnAdd = new JButton("Incluir");
        btnAdd.setFont(new Font("Arial", Font.PLAIN, 15));
        btnAdd.setSize(100, 20);
        btnAdd.setLocation(100, 250);
        btnAdd.addActionListener((ActionEvent e) -> {
            int year = Integer.parseInt(yearTextField.getText());
            int volume =
Integer.parseInt(volumeTextField.getText());
            int number =
Integer.parseInt(numberTextField.getText());

            controller.isFieldsNotNull(geralTextField.getText(),
volume, number, titleTextField.getText(), year);

            controller.include(geralTextField.getText(), volume,
number, titleTextField.getText(), year);
        });
        c.add(btnAdd);

        btnType = new JButton("Livros");
        btnType.setFont(new Font("Arial", Font.PLAIN, 15));
        btnType.setSize(100, 20);
        btnType.setLocation(220, 250);
        btnType.addActionListener((ActionEvent e) -> {
            setVisible(false);
            LivroForm re = new LivroForm("Livros", "Autor");
        });
        c.add(btnType);

        btnList = new JButton("Listagem");
        btnList.setFont(new Font("Arial", Font.PLAIN, 15));
        btnList.setSize(100, 20);

```

```

        btnList.setLocation(340, 250);
        btnList.addActionListener((ActionEvent e) -> {
            setVisible(false);
            ListagemView li = new ListagemView(controller.toList());
        });
        c.add(btnList);

        setVisible(true);
    }

    public static void main(String[] args) {

        RevistaForm f = new RevistaForm("Revista", "Org.");
    }
}

```

ListagemView

```

package view;

import java.awt.Container;
import java.awt.Font;
import javax.swing.JFrame;
import static javax.swing.JFrame.EXIT_ON_CLOSE;
import javax.swing.JLabel;
import javax.swing.JTextArea;

public class ListagemView extends JFrame {

    final protected Container c;
    final private JLabel typeLabel;
    final private JTextArea listTextArea;

    public ListagemView(String textArea) {
        setBounds(300, 90, 600, 800);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setResizable(false);

        c = getContentPane();
    }
}

```

```

        c.setLayout(null);

        typeLabel = new JLabel("Listagem");
        typeLabel.setFont(new Font("Arial", Font.PLAIN, 30));
        typeLabel.setSize(150, 100);
        typeLabel.setLocation(225, 10);
        c.add(typeLabel);

        listTextArea = new JTextArea();
        listTextArea.setFont(new Font("Arial", Font.PLAIN, 15));
        listTextArea.setSize(400, 500);
        listTextArea.setLocation(100, 200);
        listTextArea.setLineWrap(true);
        listTextArea.setText(textArea);
        listTextArea.setEditable(false);
        c.add(listTextArea);

        setVisible(true);
    }
}

```

/* -----*/

Teste

```

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertNotNull;

import java.util.Optional;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

import org.junit.Test;

import entity.Livro;
import repository.LivrosController;

```

```

public class PooTest
{
    final RevistaController controller = new RevistaController();
    @Test
    public void verificaInclude()
    {

        Livro l = new Livro("Teste", "teste", "2012");

        //Insere no BD
        controller.include(l);

        //Recebe os livros do BD
        String listLivros = controller.toList();

        //Verifica se o livro foi incluído
        assertTrue(listLivros.contains(l.getTitle()));
    }
}

```

/* -----*/

Comentário:

O código acima tem de diferente da implementação do anterior a criação da camada repository, cuja qual cuida das chamadas com o Banco de Dados, e o pacote Entity que por sua vez é mais uma modelagem de uma tabela no BD do que propriamente um Objeto, além disso foram feitas modificações do Controller para que ele pudesse receber uma lista do Banco de Dados ao invés de guardar tudo em um vetor e se perder ao fechar o programa. Além disso, na parte de View não foi feita modificação o me impressionou porque isso mostra que a camada View foi modelada de uma forma correta a conseguir separar propriamente a manutenção das funções para o que está sendo mostrado.

Por fim, o teste do JUnit foi feito voltado para fazer um assert que um determinado objeto foi realmente inserido no Banco de Dados. Somente não achei muito bom o uso do repositório online, e implementaria um mockado ao invés de um que propriamente depende da internet, tornando o teste mais isolado da aplicação possível.