ROTEIRO AULA PRÁTICA

Unidade 3 | Seção 3

Programação orientada a objetos II

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: Programação orientada a objetos II.

Unidade 3

Seção 3.3

OBJETIVOS
Definição dos objetivos da aula prática:
Simular um cenário onde é possível a aplicação de uma metodologia ágil como o Scrum.

INFRAESTRUTURA	
Instalações:	
Laboratório de computação.	
Materiais de consumo:	
Descrição	Quantidade de materiais por procedimento/atividade
Computador	Máximo 2 alunos por computador
Software:	
Sim (X) Não ( ) Em caso afirmativo, qual? Eclipse OXYGEN 2, LibreOffice 6.0 e Java SDK 10. Pago ( ) Não Pago (X) Tipo de Licença: OpenSource Eclipse Public License, Apache License e Oracle Binary Code License .	
Descrição do software:	
O Eclipse OXYGEN .2 deve ser instalado com a opção: “Eclipse IDE for Java Developers” O	

java SDK deve ser instalado de forma padrão e o LibreOffice de maneira padrão em português Brasil.
Equipamento de Proteção Individual (EPI):
Não se aplica.

PROCEDIMENTOS PRÁTICOS

Neste momento você deve ajudar o professor por meio da descrição de todas as etapas que deverão ser realizadas para a execução dos procedimentos práticos. Considerando a carga horária da aula prática, você pode replicar a caixa de procedimento/atividade quantas vezes for necessário.

Procedimento/Atividade n.1

Atividade proposta:

Desenvolvimento de sistema de automação residencial por uma empresa que utiliza o sistema *Scrum*.

Procedimentos para a realização da atividade:

Imagine uma situação de uma empresa que está começando a utilizar o *Scrum* e já possui os elementos do software descritos como história de usuário:

1. Como usuário comum, eu preciso ter uma interface gráfica para acionar as lâmpadas de forma individual, pois é necessário um ajuste unitário de cada elemento de iluminação.
2. Como usuário comum, eu preciso ter uma opção de emergência que acione todos os elementos de iluminação da casa, pois consiste em um requisito de segurança do projeto.
3. Como administrador, eu preciso ter uma interface gráfica para configurar ou remover elementos de iluminação, pois o sistema deve ser configurável para cenário proposto.

Agora é necessário organizar o *sprint* visando que o cliente gostaria de ter uma versão básica o quanto antes focando no acionamento das lâmpadas. Dessa forma é necessário converter as histórias de usuário em tarefas. O aluno deve entregar o protótipo da interface gráfica, uma modelagem do banco de dados e simular 2 a 3 *daily meating* para alinhar o que deve estar na interface gráfica e no banco de dados.

Utilizando como base a história de usuário, o aluno deve dividir todos os elementos em tarefas visando uma atividade que o programador poderia implementar e testar, encaminhando para a entrega que o usuário pediu para criar o *sprint*:

- Tarefa A: Criar banco de dados para cadastro de elementos de iluminação;
- Tarefa B: Criar interface gráfica de controles para cadastro de elementos de iluminação;
- Tarefa C: Criar interface gráfica de controles para o acionamento e desligamento dos elementos de iluminação.

Como parte do desenvolvimento, os alunos devem simular 4 dias de desenvolvimento com os membros da equipe. No primeiro dia deve ser desenvolvido um modelo do banco de dados contendo:

1. Tabela itensIluminacao:
    a. id (int);
    b. nome (String);
    c. Localização (String);
    d. Estado (booleano).

    Deve ser efetuada a daily meeting e apresentar esse modelo de banco de dados, e iniciar o desenvolvimento da primeira interface gráfica (Figura 1). Figura 1: Protótipo da tela de cadastro

Fonte: elaborada pelo Autor.

Caso tenha dúvidas quanto a implementação da tela, o Quadro 1 apresenta o código fonte da tela de Cadastro.

Quadro 1: Código fonte da Interface de Cadastro.

```
package U3S3;

import java.awt.Container;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
```

```
import javax.swing.JTextField;

public class TelaCadastro extends JFrame {

	private JLabel lblId, lblNome, lblLocalizacao;
	private JButton btnAdd,btnRemover;
	private JComboBox jmbId;
	private JTextField txtNome, txtLocalizacao;
	private Container ct;

	public TelaCadastro()
	{
		super("Tela Cadastro");
		setSize(400,250);
		ct = getContentPane();

		ct.setLayout(null);

		lblId  = new JLabel("ID");
		jmbId = new JComboBox();

		lblNome = new JLabel("Nome");
		txtNome = new JTextField();

		lblLocalizacao = new JLabel("Localização");
		txtLocalizacao = new JTextField();

		btnAdd = new JButton("Adicionar/alterar");
		btnRemover = new JButton("Remover");

		lblId.setBounds(20,20,100,20);
		ct.add(lblId);

		jmbId.setBounds(150, 20, 200,20 );
		ct.add(jmbId);

		lblNome.setBounds(20, 40, 100, 20);
		txtNome.setBounds(150,40,200,20);
		ct.add(lblNome);
		ct.add(txtNome);

		lblLocalizacao.setBounds(20, 60, 100, 20);
		txtLocalizacao.setBounds(150,60,200,20);
		ct.add(lblLocalizacao);
		ct.add(txtLocalizacao);

		btnAdd.setBounds(20,100,150,50);
		ct.add(btnAdd);

		btnRemover.setBounds(200,100,150,50);
		ct.add(btnRemover);

		setVisible(true);
```

```
        }

        public static void main(String[] args) {

                TelaCadastro tc = new TelaCadastro();
        }
}
```

Fonte: elaborado pelo Autor.

2. Deve ser feita a *daily meeting* e apresenta a interface de cadastro que foi desenvolvida e se deve conversar de como será feita a interface de para controle dos dispositivos. O Quadro 2 apresenta o código da interface de acionamento dos dispositivos e a Figura 2 apresenta a imagem da tela.

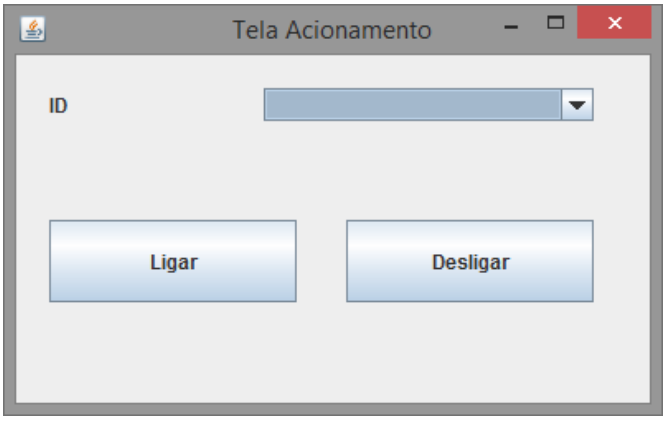
Quadro 2: Código da interface de acionamento.

```
package U3S3;

import java.awt.Container;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class TelaAcionamento extends JFrame {

        private JLabel lblId;
        private JButton btnLiga,btnDesliga;
        private JComboBox jmbId;
        private Container ct;

        public TelaAcionamento()
        {
                super("Tela Acionamento");
                setSize(400,250);
                ct = getContentPane();

                ct.setLayout(null);

                lblId  = new JLabel("ID");
                jmbId = new JComboBox();

                btnLiga = new JButton("Ligar");
                btnDesliga = new JButton("Desligar");

                lblId.setBounds(20,20,100,20);
                ct.add(lblId);

                jmbId.setBounds(150, 20, 200,20 );
                ct.add(jmbId);
```

```
                btnLiga.setBounds(20,100,150,50);
                ct.add(btnLiga);

                btnDesliga.setBounds(200,100,150,50);
                ct.add(btnDesliga);

                setVisible(true);


        }

        public static void main(String[] args) {

                TelaAcionamento ta = new TelaAcionamento();
        }
}
```

Fonte: elaborado pelo Autor.

Figura 2: Tela de acionamento.



Fonte: Elaborado pelo Autor.

Checklist:

Para verificar a tarefa se deve:

1. Verificar se todas as tarefas são derivadas das histórias de usuário;
2. Existem apenas tarefas que levam a entrega que o cliente pediu;
3. Verificar se as interfaces gráficas e banco de dados contemplam as descrições que os usuários colocam.

RESULTADOS

Resultados da aula prática:

Ao final da aula prática deverá ser entregue um relatório apresentado o escopo original, a

conversão para histórias de usuário, a derivação das tarefas que guiam para a entrega pedida pelo cliente, os protótipos das interfaces gráficas e uma modelagem básica do banco de dados contendo quais tabelas deverão ser utilizados.