

# Programação orientada ao objeto II

# ROTEIRO DE AULA PRÁTICA

**NOME DA DISCIPLINA:** Programação orientada ao objeto II.

## Unidade 1

## Seção 1.2

### OBJETIVOS

#### Definição dos objetivos da aula prática:

Desenvolver o conhecimento e a habilidade para a construção de uma interface gráfica que trata os eventos gerados pelo usuário.

### INFRAESTRUTURA

#### Instalações:

Laboratório de computação

#### Materiais de consumo:

Descrição	Quantidade de materiais por procedimento/atividade
Computadores	Máximo dois alunos por computador

#### Software:

Sim ( X ) Não ( )

Em caso afirmativo, qual? Eclipse OXYGEN .2 e Java SDK 9\_.

Pago ( ) Não Pago ( X )

Tipo de Licença: OpenSource Eclipse Public 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

#### 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:

Criar uma interface gráfica para realizar *login* (usuário e senha) de um sistema. Quando o usuário terminar de preencher o campo de usuário e senha e clicar no botão enviar, o sistema deverá guardar os dados em variáveis.

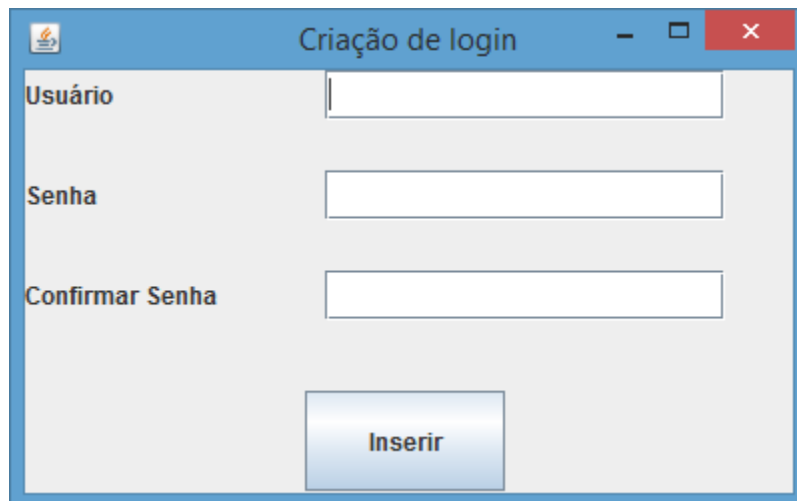
#### Procedimentos para a realização da atividade:

Para criar a interface gráfica siga os seguintes passos:

1. Criar uma relação de especialização com a classe que representa sua tela, exemplo:
  - `public class TelaLogin extends JFrame;`
2. Declarar como atributos os elementos que serão adicionados a tela, exemplo:
  - `private JPasswordField txtSenha;`
3. Definir a forma de alocação dos elementos gráficos na tela, nesse caso deve ser utilizado as posições absolutas e definir os tamanhos iniciais da janela:
  - `setSize(400,500);`
  - `setTitle("Sistema de cadastro");`
4. No construtor, instanciar, configurar e posicionar os itens na tela:
  - `ctn.setLayout(null);`
  - `lblNome.setBounds(0, 0, 100, 25);`
5. No construtor, criar e definir quais interfaces tratarão os eventos.

A Figura 1 é um exemplo de como deve ficar a interface:

Figura 1 – Tela login



Fonte: Captura de tela da IDE Eclipse.

Alguns usuários estão reclamando que o sistema precisa enviar os dados para o servidor com o objetivo de verificar se as regras de criação de *login* foram respeitadas. Para diminuir o fluxo de dados na rede, a verificação pode ser feita no próprio computador que o sistema está rodando. Crie uma interface de *login* (usuário e senha) e implemente uma verificação no campo “usuário” para a seguinte regra: “O usuário precisa ter no mínimo três caracteres e a primeira letra deve, obrigatoriamente, ser maiúscula”. Caso essa regra não seja satisfeita o usuário deverá ser informado.

O código abaixo apresenta uma possível implementação, consulte-a caso tenha dúvida:

```
import java.awt.Container;
import java.awt.event.*;
import javax.swing.*;

public class LoginSistemaFoco extends JFrame{

    private JLabel lblUsuario;
    private JTextField txtUsuario;

    private JLabel lblSenha1;
    private JPasswordField txtSenha1;

    private JLabel lblSenha2;
    private JPasswordField txtSenha2;

    private JButton btnInserir;
    private Container ctn;

    public LoginSistemaFoco()
    {
        setSize(400,250);
        setTitle("Criação de login");
        ctn = getContentPane();
        ctn.setLayout(null );
    }
}
```

```

lblUsuario = new JLabel("Usuário");
txtUsuario = new JTextField();

lblSenha1 = new JLabel("Senha");
txtSenha1 = new JPasswordField();

lblSenha2 = new JLabel("Confirmar Senha");
txtSenha2 = new JPasswordField();

btnInserir = new JButton("Inserir");

lblUsuario.setBounds(0, 0, 100, 25);
txtUsuario.setBounds(150,0,200,25);

lblSenha1.setBounds(0,50,100,25);
txtSenha1.setBounds(150,50,200,25);

lblSenha2.setBounds(0,100,100,25);
txtSenha2.setBounds(150,100,200,25);

btnInserir.setBounds(140, 160, 100, 50);

ctn.add(lblUsuario);
ctn.add(txtUsuario);
ctn.add(lblSenha1);
ctn.add(txtSenha1);
ctn.add(lblSenha2);
ctn.add(txtSenha2);
ctn.add(btnInserir);

setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

//tratamento do clique no botão
btnInserir.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        trataEventoBotaoinserir();
    }
});
//tratamento para a perda de foco do campo txtUsuario
txtUsuario.addFocusListener(new FocusListener() {
    @Override
    public void focusLost(FocusEvent e) {
        trataFocoUsuario();
    }
    @Override
    public void focusGained(FocusEvent e) {
    }
});
}

```

```

    public void trataFocoUsuario()
    {
        String usuario = txtUsuario.getText();
        //Verifica se o comprimento é menor que 3 ou se a primeira letra é
        diferente de maiúscula
        if(usuario.length() < 3 || Character.isUpperCase(usuario.charAt(0)) !=
        true)
        {
            JOptionPane.showMessageDialog(this,
            "O usuário deve ter mais que 3 caracteres e
            começar em caixa alta.");
        }
    }
    public void trataEventoBotaoInserir()
    {
        String usuario = txtUsuario.getText();
        String senha1 = new String(txtSenha1.getPassword());
        String senha2 = new String(txtSenha2.getPassword());
        JOptionPane.showMessageDialog(this, "Senha cadastrada para o
        usuário: " + usuario);
    }
    public static void main(String[] args) {
        LoginSistemaFoco tela = new LoginSistemaFoco();
    }
}

```

### Checklist:

Para verificar se a tarefa foi concluída com sucesso, os seguintes itens devem ser contemplados:

1. Existem todos os campos relacionados aos itens pedidos.
2. Todos os componentes utilizados são relacionados ao tipo de dado pedido;
3. Todos os componentes foram configurados corretamente;
4. Interface e código estão organizados, com comentários e os nomes das variáveis tem
5. Nomes relativos à sua função na tela.
6. Os eventos foram tratados por métodos declarados fora da interface implementada pela classe anônima.

## RESULTADOS

### Resultados da aula prática:

Ao final da aula prática deverá ser entregue um projeto que contenha uma *interface* gráfica para login e que responda a um evento gerado pelo usuário.