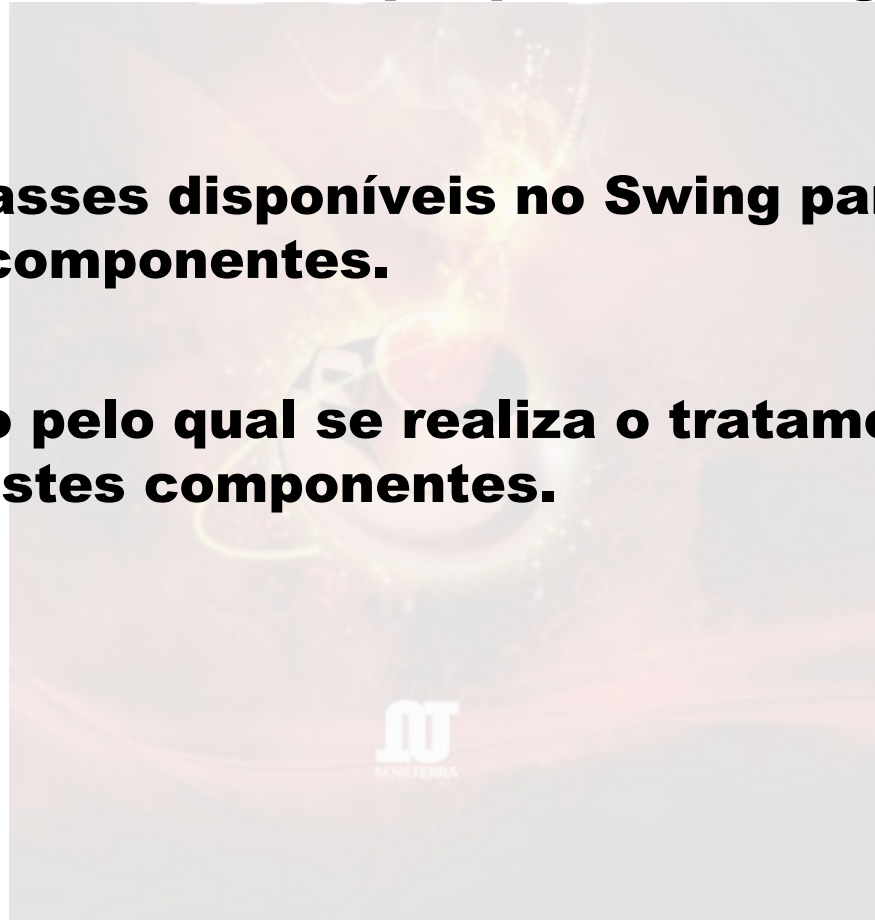


Capítulo 26

Entradas de Texto

Objetivos do Capítulo

- Apresentar três diferentes tipos de componentes utilizados para a captação de textos que podem ser agregados a uma janela.**
- Explorar as classes disponíveis no Swing para a criação deste tipo de componentes.**
- Indicar o modo pelo qual se realiza o tratamento de eventos específicos destes componentes.**



Campos de Texto

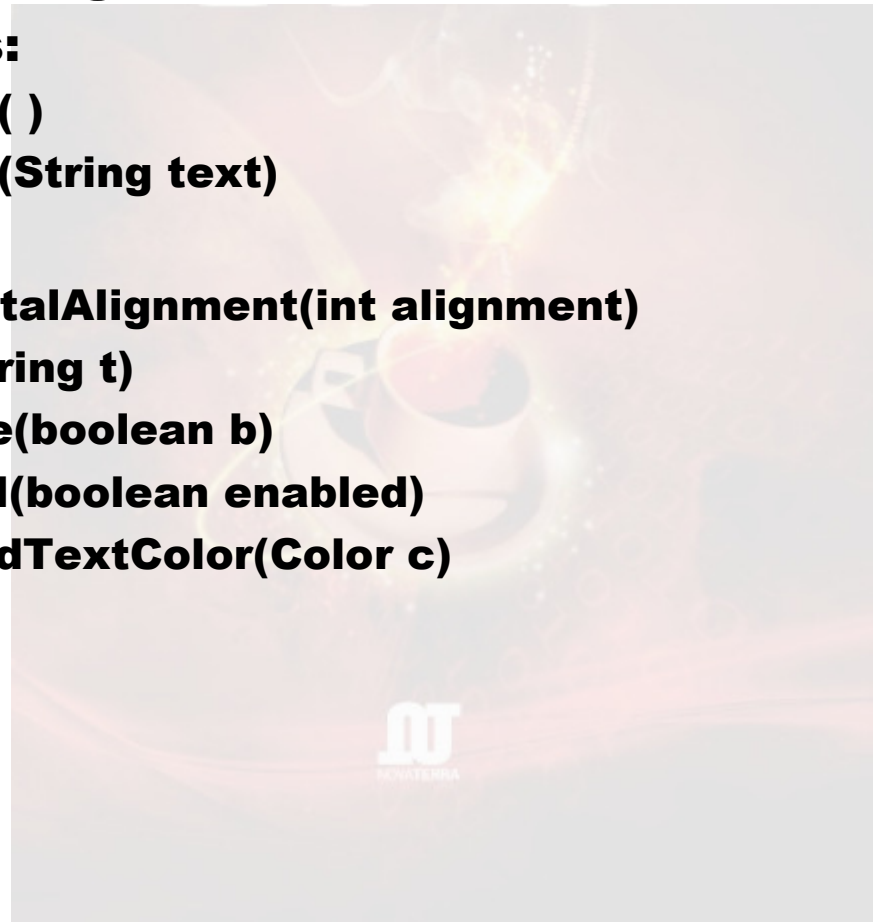
❑ **javax.swing.JTextField extends JTextComponent implements SwingConstants ...**

➤ **Construtores:**

- ❖ **JTextField()**
- ❖ **JTextField(String text)**

➤ **Métodos:**

- ❖ **setHorizontalAlignment(int alignment)**
- ❖ **setText(String t)**
- ❖ **setEditable(boolean b)**
- ❖ **setEnabled(boolean enabled)**
- ❖ **setDisabledTextColor(Color c)**



Campos de Texto

❑ Código 26.1 – CampoTexto.java

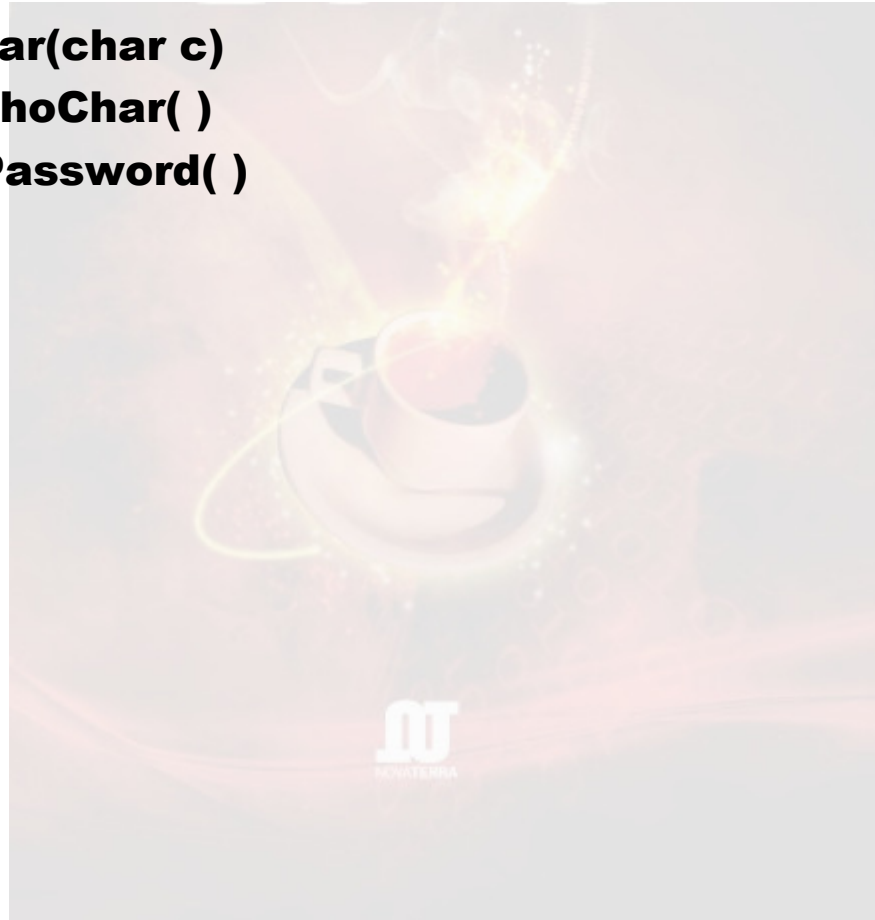


Campos de Senha

❑ **javax.swing.JPasswordField extends JTextField ...**

➤ **Métodos:**

- ❖ **setEchoChar(char c)**
- ❖ **char getEchoChar()**
- ❖ **char[] getPassword()**



Campos de Senha

❑ Código 26.2 – CampoSenha.java



Áreas de Texto

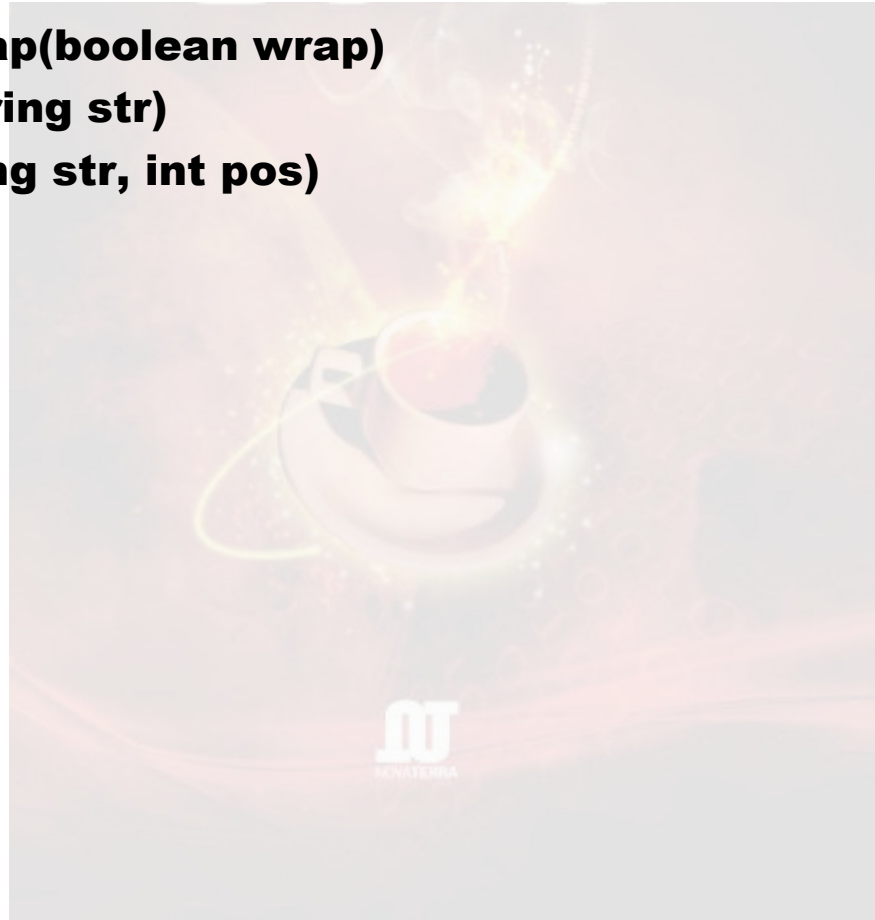
❑ **javax.swing.JTextArea extends JTextComponent ...**

➤ **Métodos:**

❖ **setLineWrap(boolean wrap)**

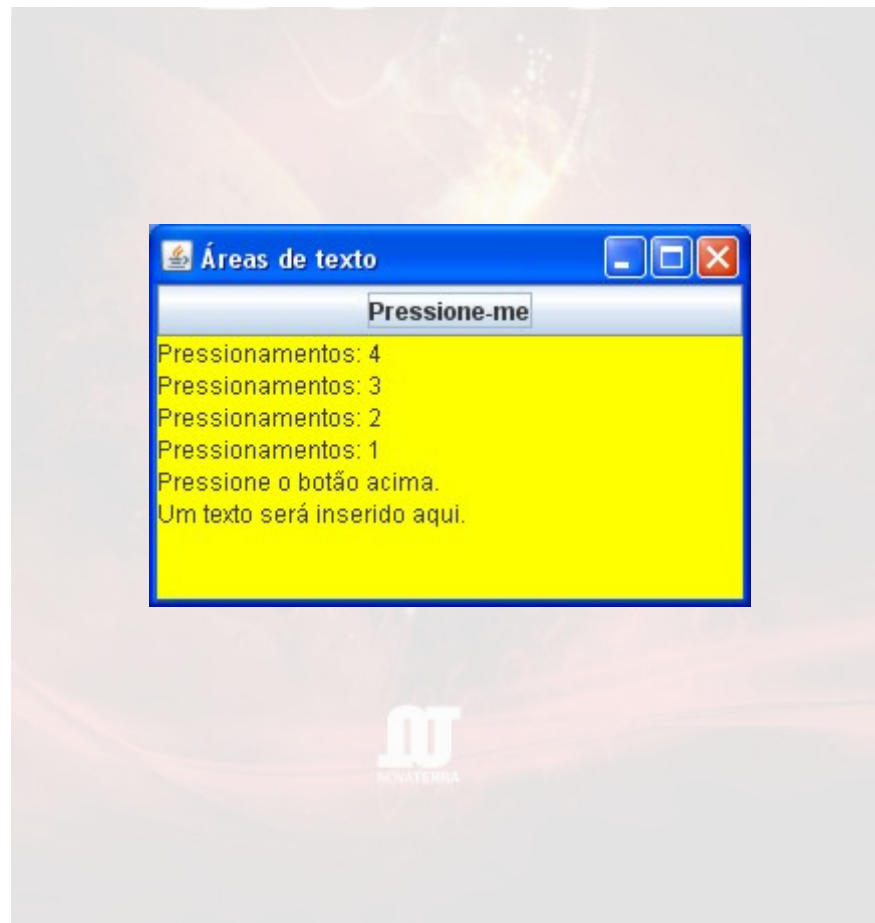
❖ **append(String str)**

❖ **insert(String str, int pos)**



Áreas de Texto

❑ Código 26.3 – AreaTexto.java



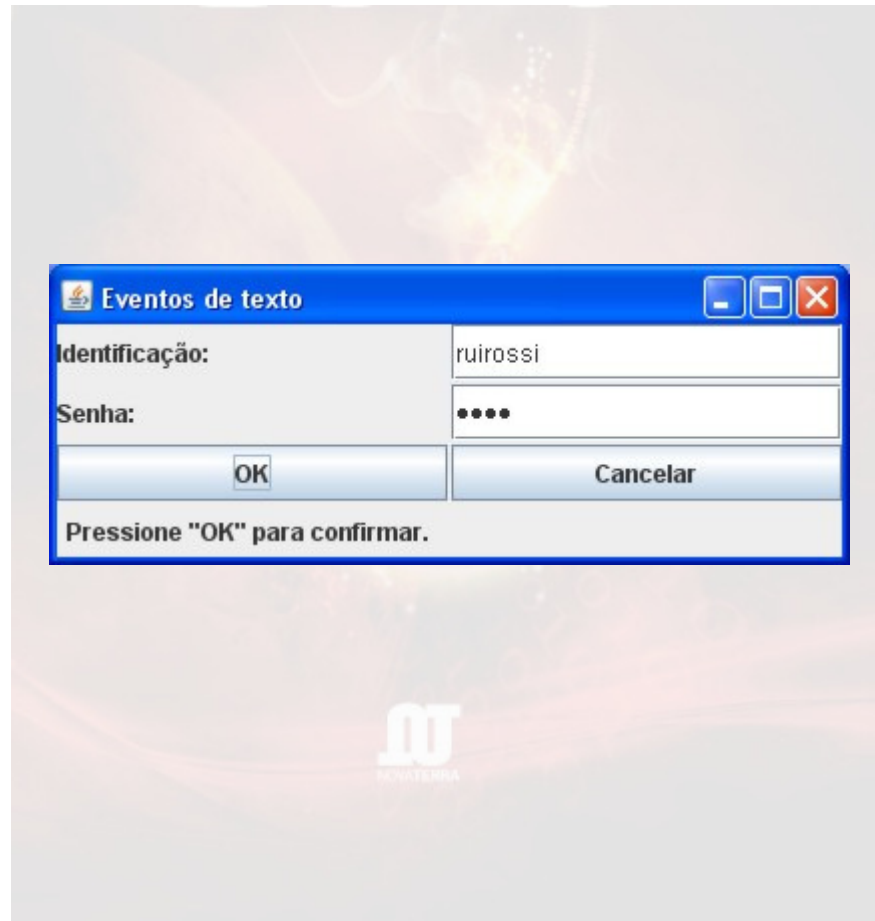
Eventos de Texto

□ **javax.swing.text.DocumentEvent**

- **Gatilho:** disparado quando o documento subjacente é modificado
- **Interface:** **javax.swing.text.DocumentListener**
- **Métodos:**
 - ❖ **changedUpdate(DocumentEvent e):** alteração de atributo
 - ❖ **insertUpdate(DocumentEvent e):** inclusão de caractere
 - ❖ **removeUpdate(DocumentEvent e):** exclusão de caractere
- **Arquitetura Swing:**
 - ❖ **Visualização:** aparência do componente (**javax.swing.JTextField**)
 - ❖ **Modelo:** dados do componente (**javax.swing.text.Document**)

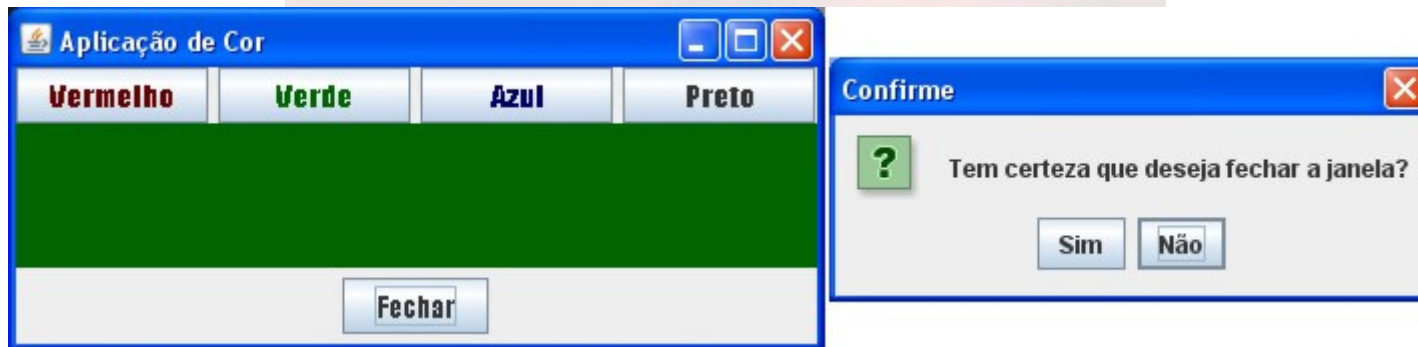
Eventos de Texto

❑ Código 26.4 – EventoDocumento.java



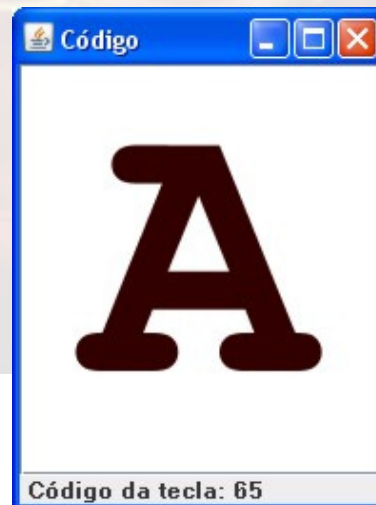
Exercício 1

- ❑ **Crie uma nova janela, chamada AplicaCor, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela deve ter quatro botões dispostos na região superior e um botão disposto na região inferior.**
 - **Os primeiros servirão para aplicar uma cor ao fundo da janela e o último servirá para fechá-la.**
 - **Defina a posição, o tamanho e as demais configurações destes botões de acordo com o modelo.**
 - **Quando o botão “Fechar” for pressionado, uma mensagem de confirmação deve ser exibida.**
 - ❖ **Se a opção “Sim” for selecionada, a janela deve ser fechada.**
 - ❖ **Caso contrário, ela deve permanecer aberta.**



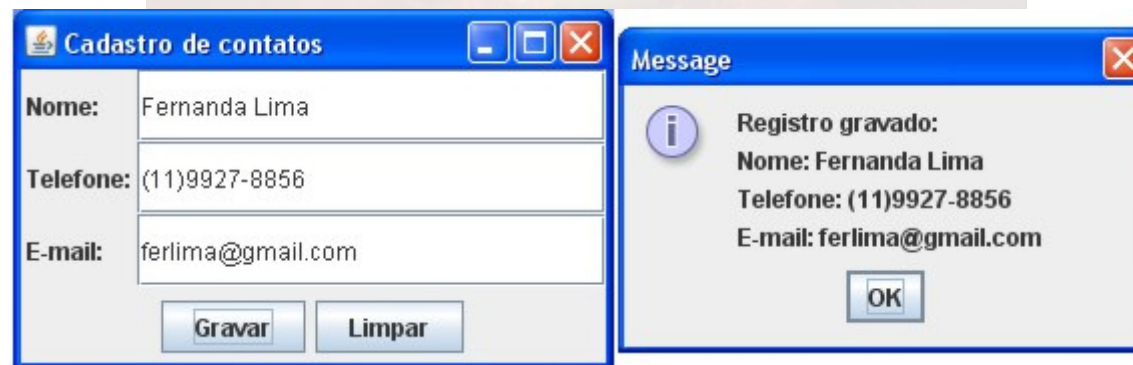
Exercício 2

- ❑ **Crie uma nova janela, chamada CódigoTecla, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela deve ter apenas um campo de texto e um rótulo.**
 - **Inicialmente, o campo de texto deve estar vazio e deve ser disposto no centro da janela.**
 - **O rótulo deve ser disposto em sua região inferior.**
 - **Sempre que uma tecla for pressionada dentro do campo de texto, o caractere deve ser exibido no próprio campo e o código correspondente deve ser apresentado no rótulo.**
 - **Se a tecla ESC for pressionada, a janela deve ser fechada.**



Exercício 3

- ❑ **Crie uma nova janela, chamada CadastroContato, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela será apenas uma simulação de cadastro, pois não gravará realmente nenhum dado.**
 - **Sempre que o botão “Gravar” for pressionado, uma mensagem deve ser exibida com o conteúdo dos três campos.**
 - **Quando esta mensagem for fechada, o texto destes campos deve ser apagado e o foco deve retornar ao primeiro campo.**
 - **Se o botão “Limpar” for pressionado, os textos de todos os campos devem ser apagados.**



Exercício 4

- ❑ **Crie uma nova janela, chamada CadastroFuncionario, de acordo com o modelo apresentado na figura abaixo.**
- ❑ **Esta janela será apenas uma simulação de cadastro de funcionários.**
 - **Sempre que o botão “Gravar” for pressionado, ela deve exibir uma mensagem com os dados informados, deve limpar todos os campos e retornar o foco para o primeiro campo.**
 - **O pressionamento do botão “Limpar” deve limpar os campos.**
 - **Observe que você deve formatar o salário e a data de nascimento para exibi-los.**
 - **Além disso, deve permitir que o salário seja informado utilizando a vírgula como separador de casas decimais.**

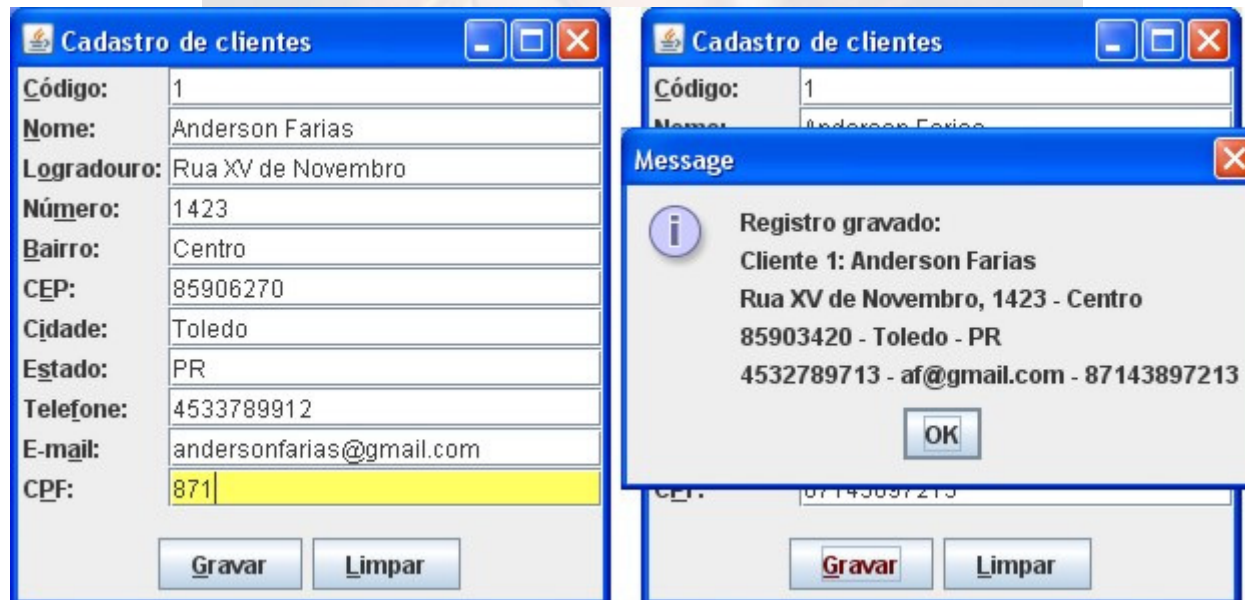


Exercício 4

- ❑ **Se um dado inválido for informado e o botão “Gravar” for pressionado, deve ser exibida uma mensagem de erro que indique qual é o dado que provocou o erro e o motivo.**
- ❑ **As regras para a validação dos dados são as seguintes:**
 - **Matrícula: deve ser um número inteiro maior que zero.**
 - **Nome: deve ter de 5 a 50 caracteres.**
 - **Salário: deve ser um valor maior que zero.**
 - **Nascimento: o funcionário deve ter pelo menos 14 anos completos.**
- ❑ **Além da classe CadastroFuncionario, também é sugerido que você crie outra classe como parte da solução deste exercício.**
 - **Sugere-se que seja criada uma classe, chamada Funcionario, com atributos que representem os dados cadastrais dos funcionários, com métodos de leitura e de escrita e com o método toString().**
 - **A validação destes dados pode ser realizada pelos métodos de escrita desta classe.**

Exercício 5

- ❑ Crie uma nova janela, chamada **CadastroCliente**, de acordo com o modelo apresentado na figura abaixo.
- ❑ Esta janela servirá como uma simulação de cadastro de clientes.
 - Sempre que o botão “Gravar” for pressionado, ela deve exibir uma mensagem com os dados informados, deve limpar todos os campos e retornar o foco para o primeiro campo.
 - O pressionamento do botão “Limpar” deve limpar todos os campos.



Exercício 5

- ❑ **Se um dado inválido for informado e o botão “Gravar” for pressionado, deve ser exibida uma mensagem de erro que indique qual é o dado que provocou o erro e o motivo.**
 - **Defina você mesmo as regras que serão utilizadas para validar cada um dos dados.**
- ❑ **Além da classe CadastroCliente, também é sugerido que você crie outra classe como parte da solução deste exercício.**
 - **Sugere-se que seja criada uma classe, chamada Cliente, com atributos que representem os dados cadastrais dos clientes, com métodos de leitura e de escrita e com o método toString().**
 - **A validação destes dados pode ser realizada pelos métodos de escrita desta classe.**
- ❑ **Defina uma tecla de atalho para cada um dos dois botões.**
- ❑ **Também defina atalhos para os campos através dos rótulos que se encontram ao lado deles.**
 - **Utilize os métodos setDisplayedMnemonic() e setLabelFor() dos rótulos.**
 - **O método setDisplayedMnemonic() exige apenas um caractere como argumento e ele define a tecla de atalho para um rótulo.**
 - **O método setLabelFor() define o componente que receberá o foco quando a tecla de atalho do rótulo for pressionada.**

Contato

Com o autor:

Rui Rossi dos Santos

E-mail: livros@ruirossi.pro.br

Web Site: <http://www.ruirossi.pro.br>

Com a editora:

Editora NovaTerra

Telefone: (21) 2218-5314

Web Site: <http://www.editoranovatterra.com.br>

