

2013

Apostila Programação Visual com Java e IDE Netbeans



Robson de Lima Carvalho

Centro Paula Souza: ETEC's "Prof. Fausto
Mazzola" de Avaré e "Pref. José Esteves"
de Cerqueira César – contato:

robson.carvalho13@etec.sp.gov.br

Publicado: 15/04/2013

Sumário

Criando projetos e janelas	2
Conversão de tipos de dados.....	9
Inserindo figuras/ícones	12
Renomeando componentes e adicionando legendas	17
Menu, itens de menu e navegação entre as janelas.	19
Botão de Rádio e Grupo de Botões	22
Campo formatado e caixa de seleção	27
Obras Citadas	34

Sobre o autor

O professor Robson de Lima Carvalho atua na área acadêmica nas escolas técnica de “Prof. Fausto Mazzola” de Avaré e “Prefeito José Esteves” de Cerqueira César – São Paulo. No ano de 2012 ministrou curso de Desenvolvedor de Sistemas pela IFSP de Avaré

Formação acadêmica:

- Bacharel em Sistema de Informação formado pelo Centro Universitário Central Paulista (UNICEP) São Carlos-SP;
- Especialista em Desenvolvimento de Sistema Web pela faculdade Anhanguera de Bauru-SP;
- Licenciado em informática pelo Programa Especial de Formação Pedagógica de Docentes para as Disciplinas do Currículo da Educação Profissional de Nível Médio.

Disciplinas Ministradas:

- Lógica de Programação (C e Javascript);
- Desenvolvimento de Software I e II (Com linguagem Java);
- Técnicas de Linguagem de Banco de Dados I e II;
- Planejamento e Desenvolvimento de TCC;
- Estrutura de dados;
- Entre outras.

Criando projetos e janelas

Criando um novo projeto: Para criar um novo projeto no Netbeans clique na aba Arquivo->Novo Projeto, Figura 1- Criando projeto.

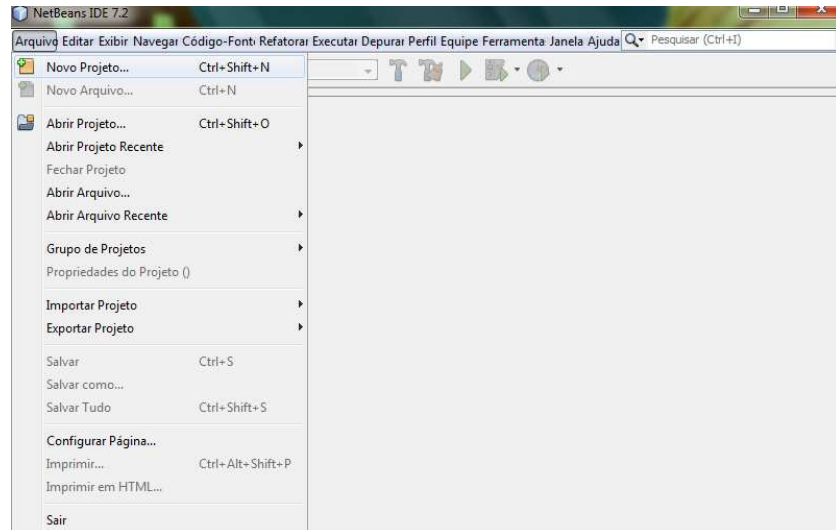


Figura 1- Criando projeto

Em seguida aparecerá a tela para escolha do tipo de projeto, Figura 2 - Escolha de Projeto, escolha Categoria->Java e Projetos->Aplicação Java para desenvolver sistemas para Desktop as outras categorias servem para outros tipos de projetos que nestas aulas não serão trabalhadas.

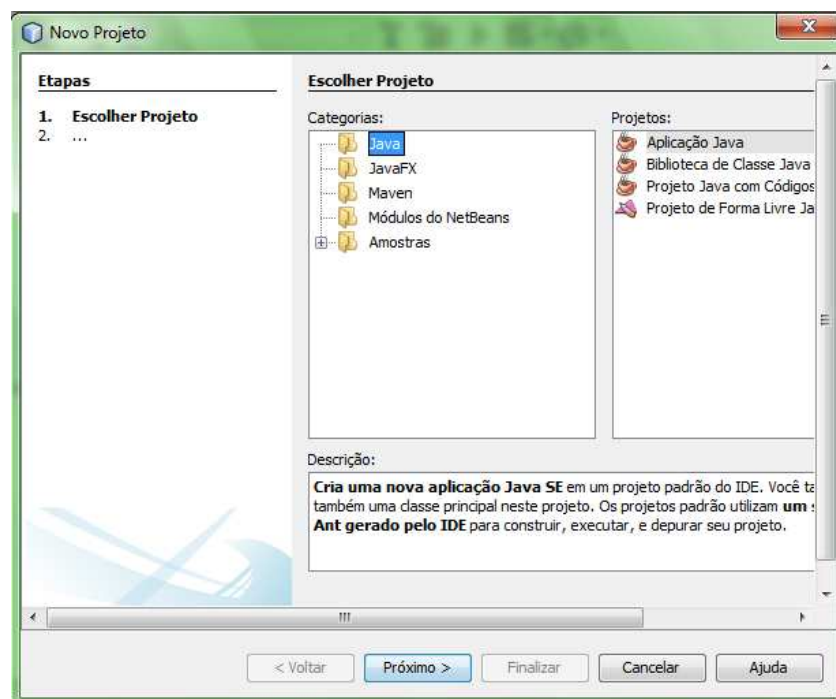


Figura 2 - Escolha de Projeto

Próximo passo é definir um nome para o projeto **Figura 3 - Atribuir nome ao projeto**, neste caso devemos colocar um nome sem acentuação e espaços em branco, normalmente a primeira letra do projeto é maiúscula.

Obs. No início não vamos deixar selecionar a opção para “Criar Classe Principal”.

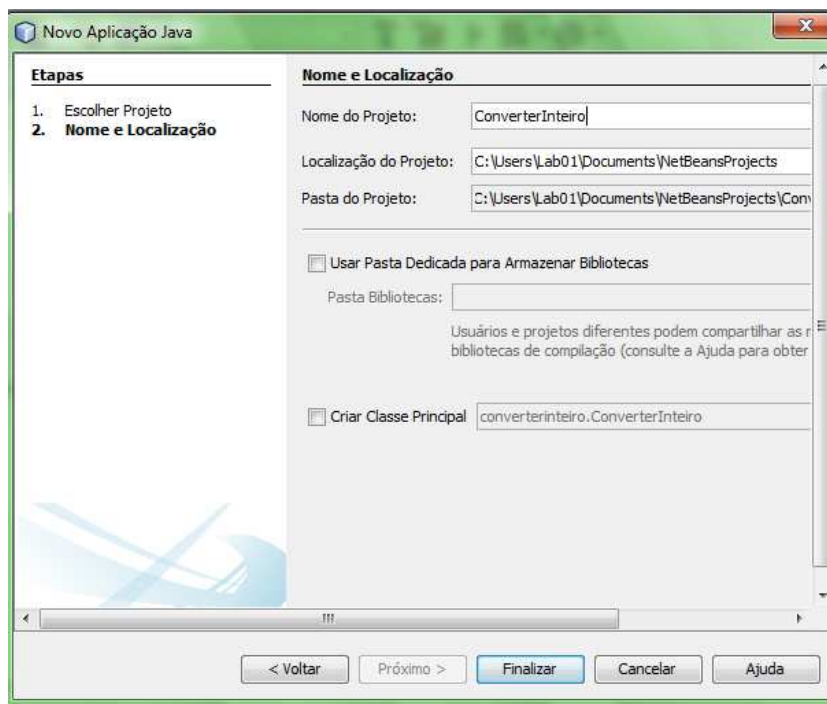


Figura 3 - Atribuir nome ao projeto

Criação de uma Janela (JFrame): Clique com botão direito em cima da xícara (Projeto criado) e selecione Novo->FormJFrame, **Figura 4 - Criação de Janelas:**

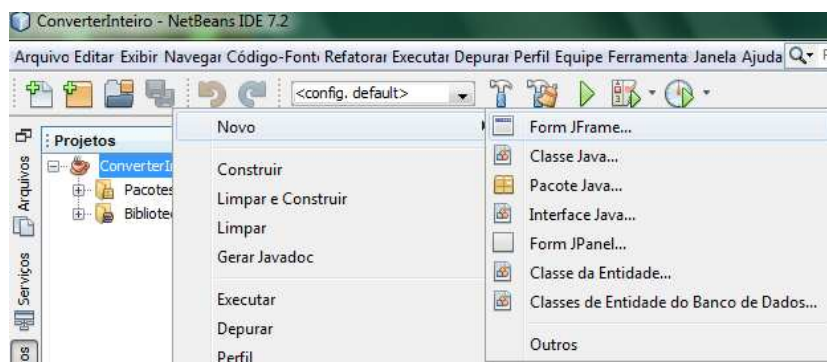


Figura 4 - Criação de Janelas

Agora vamos definir um nome para Janela Figura 5 - Definindo o nome do JFrame no campo pacote, que não pode ficar vazio, temos que colocar dentro de um pacote, no caso vamos escrever janela que será nossa pasta que irá possuir todos os JFrames/Janelas do sistema.

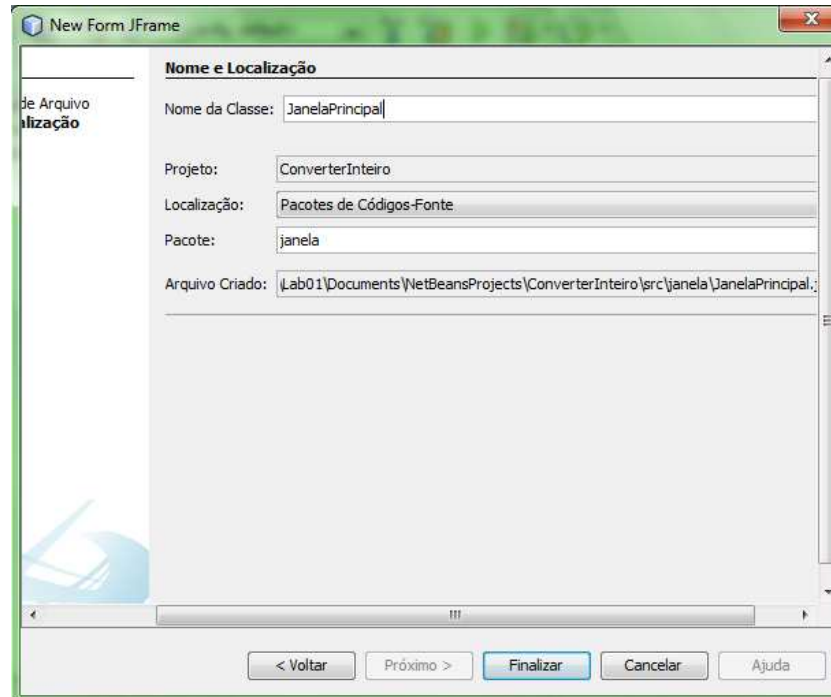


Figura 5 - Definindo o nome do JFrame

O programa criará uma nova janela Figura 6 - Nova Janela/JFrame para que possamos adicionar outros componentes visuais.

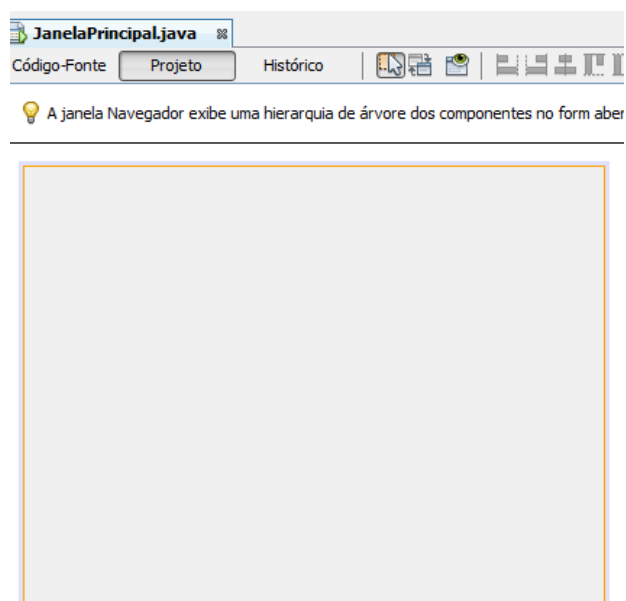


Figura 6 - Nova Janela/JFrame

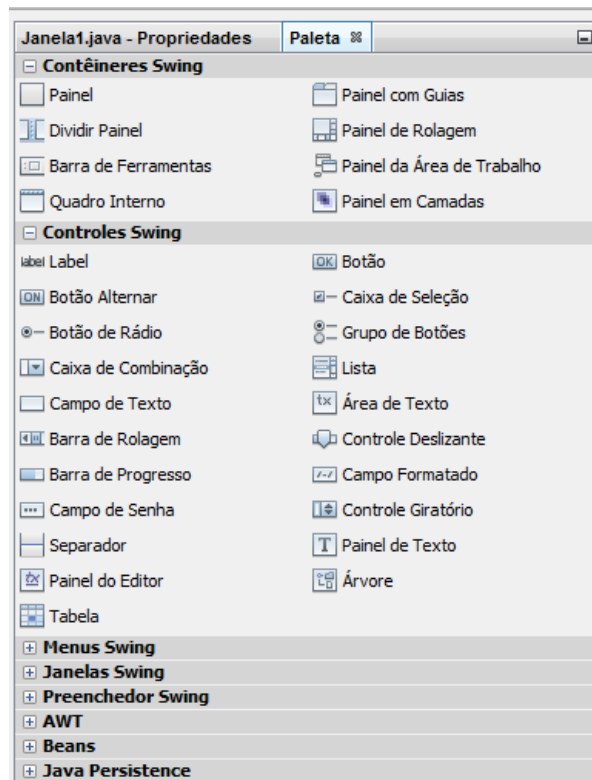


Figura 7 - Janela Paleta do Netbeans

Com a criação da nova janela Figura 6 - Nova Janela/JFrame o programa abre numa aba do lado direito alguns componentes visuais da Figura 7 - Janela Paleta do Netbeans. Os componentes básicos da paleta são:

- **JFrame** é a classe do Swing do Java que é responsável por desenhar uma tela de um programa. (Java Free)
- **JButton** é um componente para acionar ações dentro do programa, ou ações externas ao programa como salvar em um banco de dados ou salvar em um arquivo entre outros; (DEVMEDIA)
- **JLabel** é um componente para a criação de formulários onde é necessária a inserção de textos fixos ou imagens. (DEVMEDIA)
- **JTextField** é um componente que recebe a inserção dos dados pelo teclado. (DEVMEDIA)
 - **JPanel**: O painel é um componente utilizado para fazer subdivisões na tela, ou para separar de forma organizada componentes dispostos na tela. (DEVMEDIA)

Com o componente selecionado dá para alterar aparência, texto e tamanho dele, abaixo segue uns dos principais atributos que podemos modificar:

- Background -> cor de fundo;
- Font-> Tipo de letra e tamanho;
- Foreground -> Cor da letra;
- horizontalAlignment -> Se o texto vai ficar no centro, esquerda ou à direita
- icon -> colocar figura
- text -> O texto que vai aparecer na tela

A tela/janela do aplicativo vai ficar parecido com a Figura 10 - Tela final do projeto. Para executar o Projeto pressione F6.

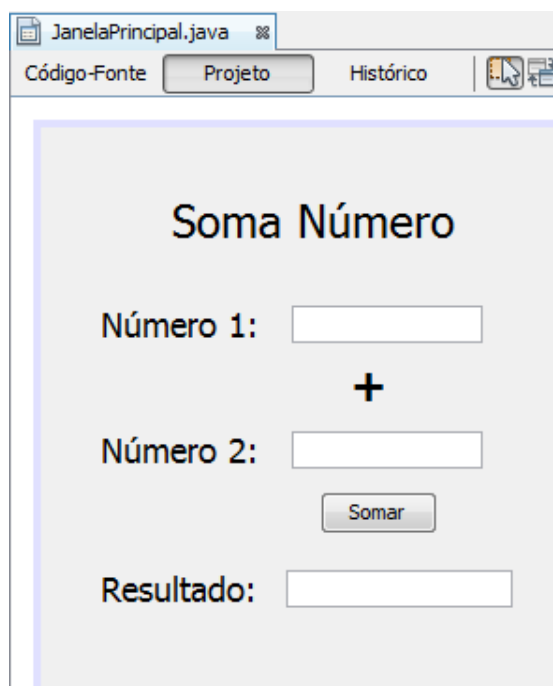


Figura 10 - Tela final do projeto

Agora vamos partir para criação do código do botão Somar, para acionar o evento é só dar dois cliques no botão somar, vai aparecer à tela de codificação do projeto. Percorrer a tela e verificar onde o cursor está piscando para adicionar o código abaixo Figura 11 - Código botão somar:

```

Generated Code

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    //Pegando número e convertendo para inteiro
    int numero1 = Integer.valueOf(jTextField1.getText());
    int numero2 = Integer.valueOf(jTextField2.getText());
    int resultado;
    //Somando números
    resultado = numero1 + numero2;
    //Agora é só mostrar o resultado
    jTextField3.setText(String.valueOf(resultado));
}

```

Figura 11 - Código botão somar

Obs.: Todo campo de texto gera ou recebe apenas tipos de dados String e para utilizar os tipos de dados diferentes, como int de Inteiro, nos temos que converter de String para Integer, ou outro tipo que veremos melhor no capítulo Conversão de tipos

Agora vamos adicionar um novo botão e adicionar alguns comandos para limpar os dados digitados Figura 12 - Código botão Limpar.

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setText("");
    jTextField2.setText("");
    //Altera o conteúdo para nada ou vazio
    jTextField3.setText("0");
}

```

Figura 12 - Código botão Limpar

Importante:

- nomeComponente.getText(); ->Esse comando pega um valor, ou retornar um valor, do componente;
- nomeComponente.setText(texto) -> Esse comando passa um valor para o componente, altera o valor existente por um novo;

Conversão de tipos de dados

Agora vamos criar um novo projeto Media (Sem acento, pois o nome do arquivo não pode ter acentuação), criar uma nova janela e desenhar a tela colocando os componentes para ficar como na Figura 13 - Tela projeto Média.

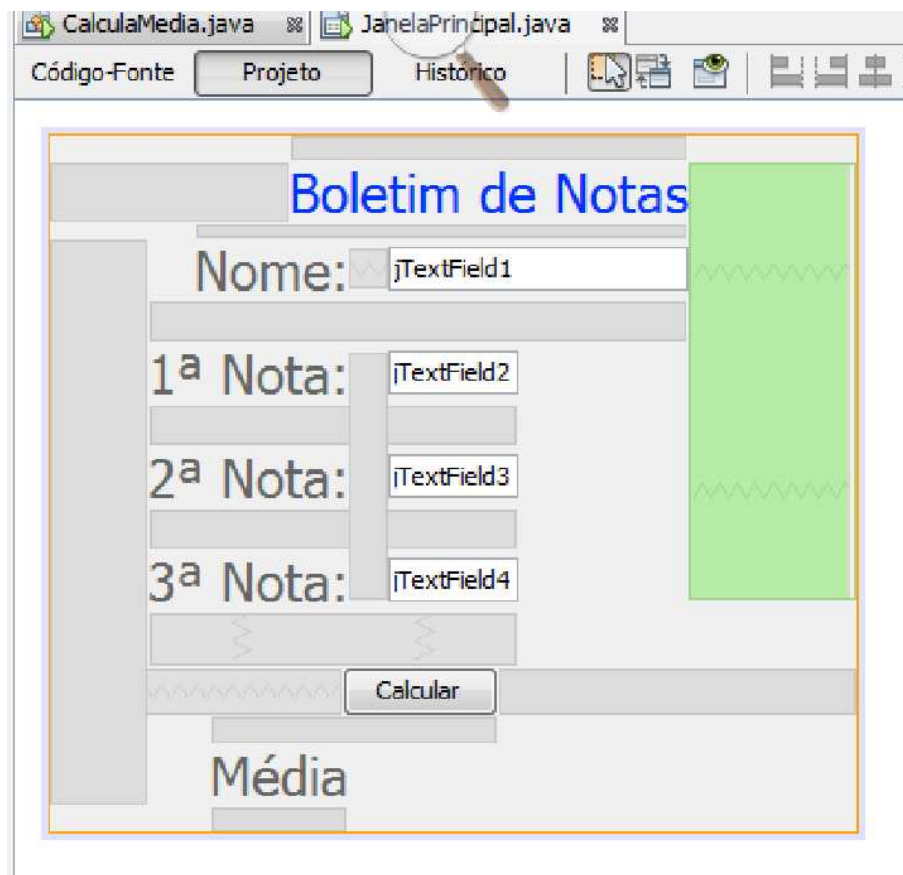


Figura 13 - Tela projeto Média

Importante

- Criando um valor como double ou float (estes valores permitem usar números decimais, ou seja, que possuem casas decimais, famoso número após a vírgula)
- Estes números possuem a notação americana, ou seja, os números decimais são separados por ponto (.) não por vírgula (,).

As figuras abaixo são do mesmo botão, porém em um método nos usamos double Figura 14 - Código botão calcular com double e no outro float Figura 15 - Código botão calcular com float. Basicamente a diferença dos dois é o tamanho de casas que cada um suporta, o double é muito superior ao float. Porém como a intenção aqui é mostra a funcionalidade, por isso, foi demonstrado usando os dois tipos.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    //peguei o nome da pessoa  
    String nome = jTextField1.getText();  
  
    //Agora pegar as notas  
    double nota1 = Double.valueOf(jTextField2.getText());  
    double nota2 = Double.valueOf(jTextField3.getText());  
    double nota3 = Double.valueOf(jTextField4.getText());  
    //Fazer a média  
    double resultado;  
    resultado = (nota1 + nota2 + nota3)/3;  
    //Falta Mostra o resultado  
    jLabel6.setText(nome + " sua média é " +resultado);  
}
```

Figura 14 - Código botão calcular com double

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    //peguei o nome da pessoa  
    String nome = jTextField1.getText();  
  
    //Agora pegar as notas  
    float nota1 = Float.valueOf(jTextField2.getText());  
    float nota2 = Float.valueOf(jTextField3.getText());  
    float nota3 = Float.valueOf(jTextField4.getText());  
    //Fazer a média  
    float resultado;  
    resultado = (nota1 + nota2 + nota3)/3;  
    //Falta Mostra o resultado  
    jLabel6.setText(nome + " sua média é " +resultado);  
}
```

Figura 15 - Código botão calcular com float

O resultado está sendo mostrado num jLabel6 criado que fica embaixo do botão, mas que na Figura 13 - Tela projeto Média não ficou explicita devido ao valor do text que é nulo no começo do programa.

Novamente vamos adicionar um botão para limpar a tela dos valores antigos a codificação ficará Figura 16 - Código botão limpar de Média.

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText("");  
    jTextField2.setText("");  
    jTextField3.setText("");  
    jTextField4.setText("");  
    jLabel6.setText("Digite novas notas.");  
}
```

Figura 16 - Código botão limpar de Média

Exercício

1. Crie um novo projeto e uma nova janela conforme Figura 17 - Exercício Reajuste Salarial o resultado final deverá mostrar o nome do funcionário e seu novo salário.

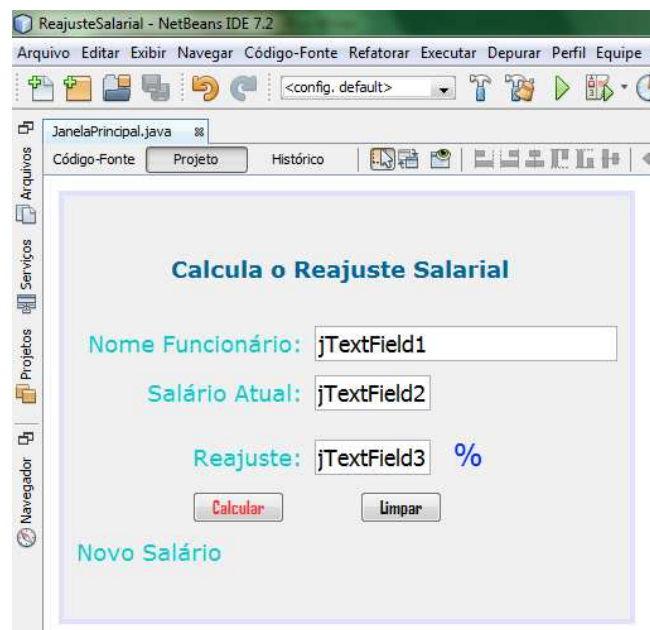


Figura 17 - Exercício Reajuste Salarial

Inserindo figuras/ícones

Criar um projeto com o nome **Projetolcone**, tudo junto sem espaço ou acentos e caracteres especiais, busque ícones em sites específicos nos exemplos desta apostila utilizamos o site: <http://www.iconfinder.com>. Baixar ícones com extensão png ou jpg, escolher o tamanho em pixel para os ícones na barra superior direita. **(Parece uma régua é o tamanho do ícone que você quer)**

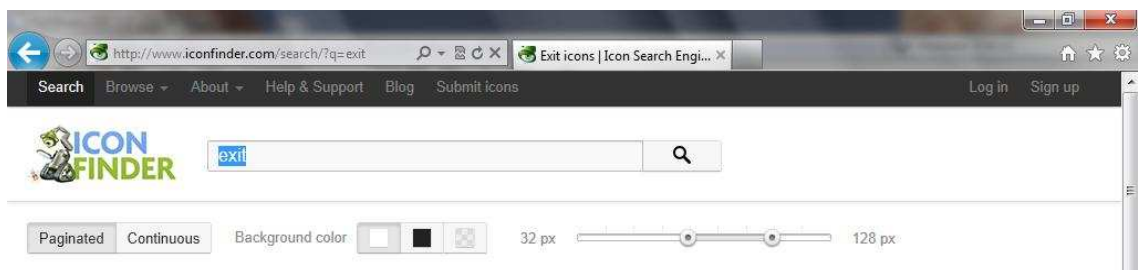


Figura 18 - Site iconfinder

Para buscar um ícone digite no campo de busca **Figura 18 - Site iconfinder** e mude o tamanho dos pixels na régua que aparece na tela **Figura 19 - Ícone exit**, depois é só clicar em png, que é a extensão do arquivo que você deseja salvar.

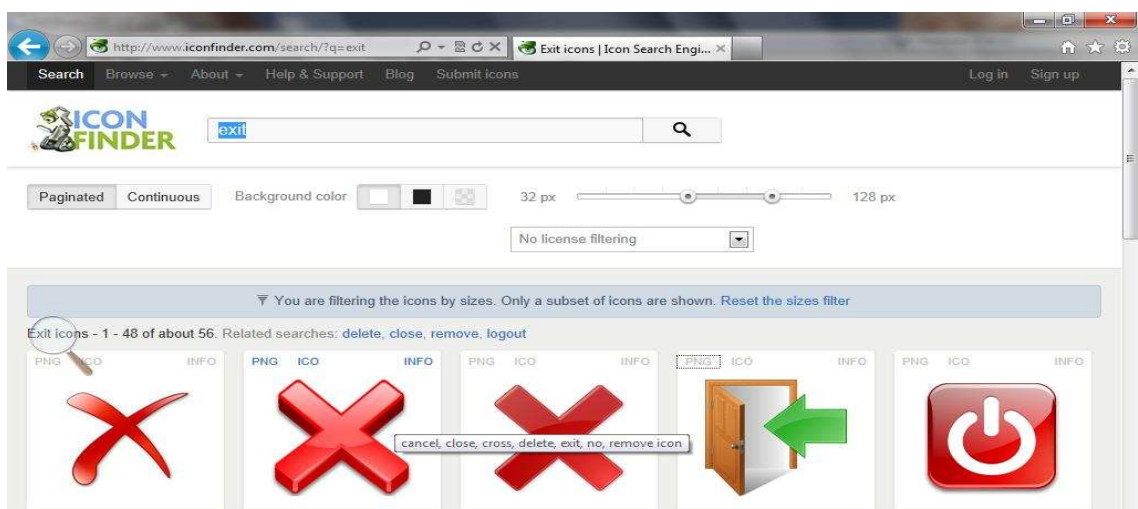


Figura 19 - Ícone exit

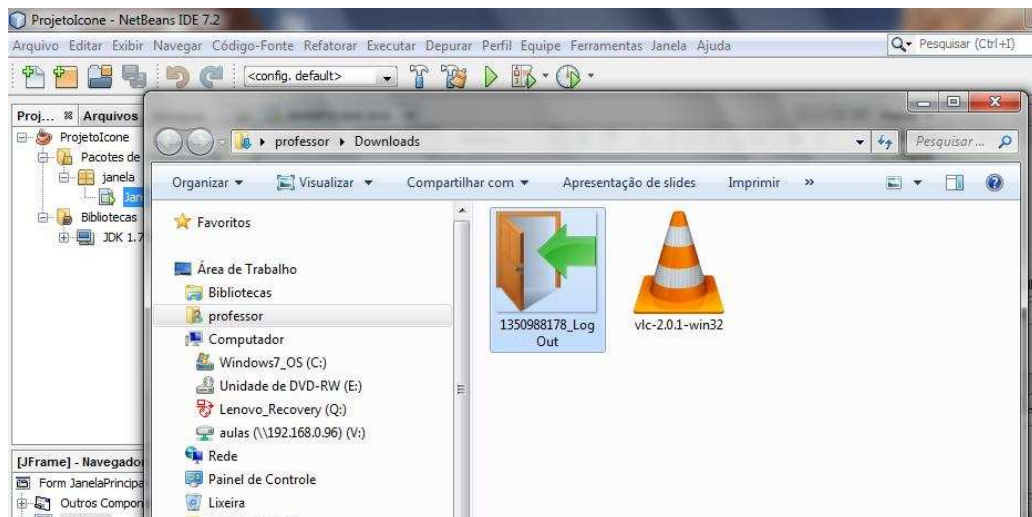


Figura 20 - Pasta do arquivo

Uma maneira de adicionar figuras em seu projeto é ir para pasta que baixamos o arquivo Figura 21 - Aba projetos com figura depois arrastar essa figura para um pacote do projeto Figura 21 - Aba projetos com figura.

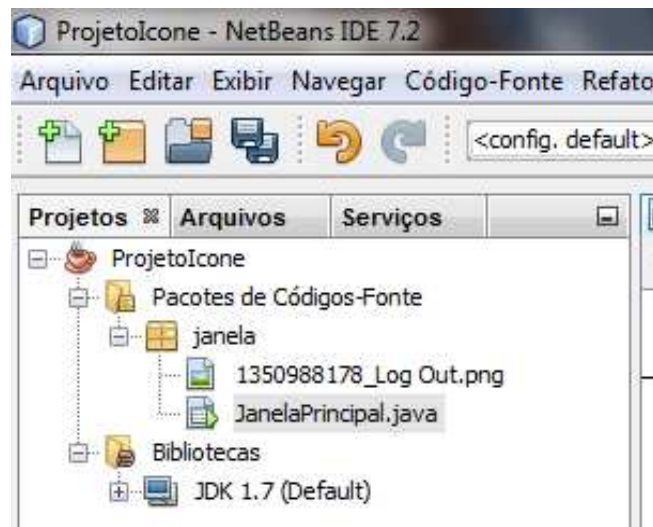


Figura 21 - Aba projetos com figura

Agora vamos adicionar um componente Botão/jButton na Janela, selecione o botão e vá até propriedades campo icon Figura 22 - Propriedade do botão, clique nos três pontinhos a direita do campo.

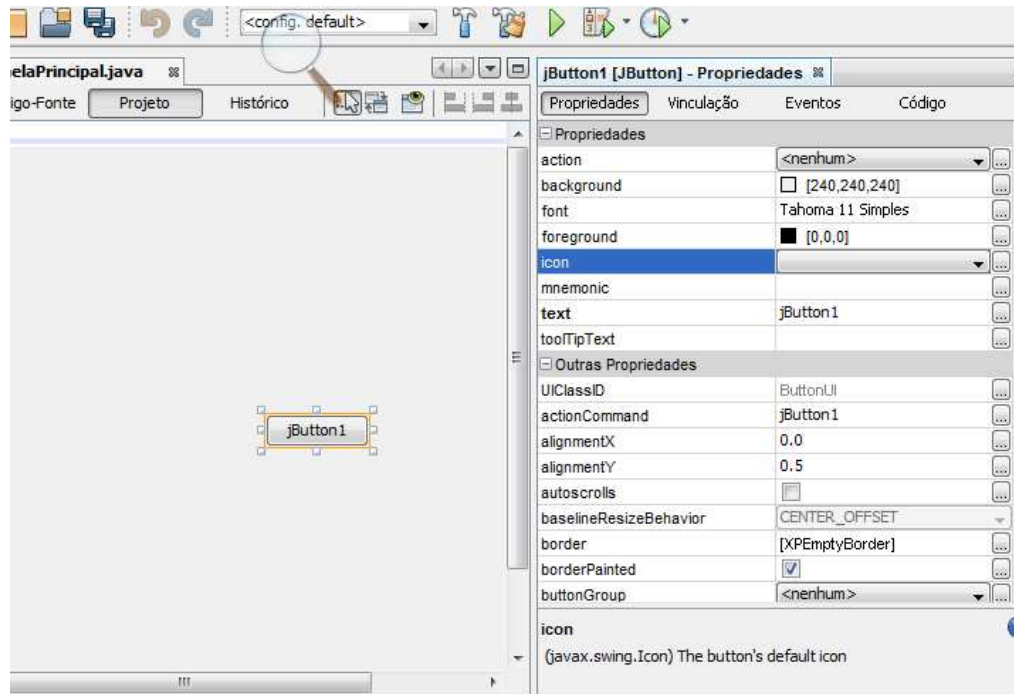


Figura 22 - Propriedade do botão

Agora é só selecionar a imagem, primeiro selecione o pacote que a imagem pertence, depois procure nome do arquivo Figura 23 - Selecionando a imagem.

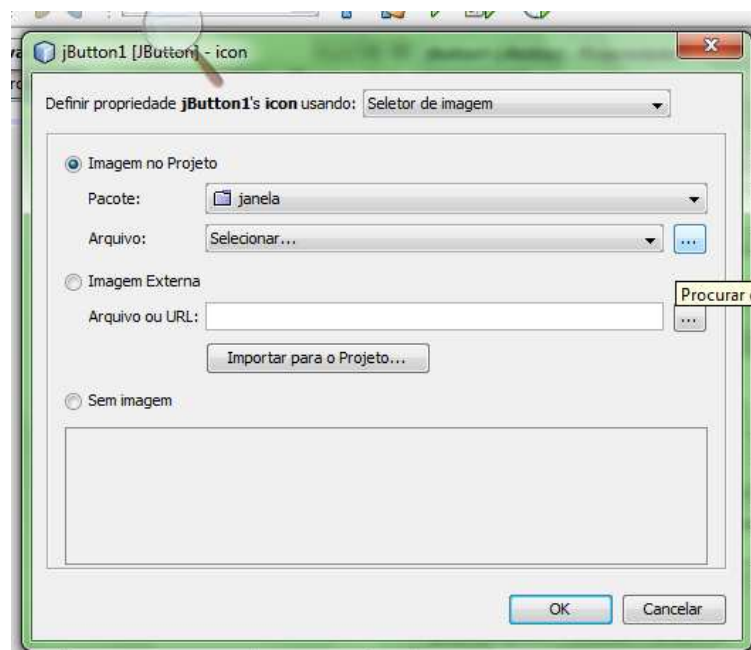


Figura 23 - Selecionando a imagem

Com a imagem selecionada a figura aparece para você verificar se é ela mesmo que você deseja adicionar no projeto Figura 24 - Imagem selecionada

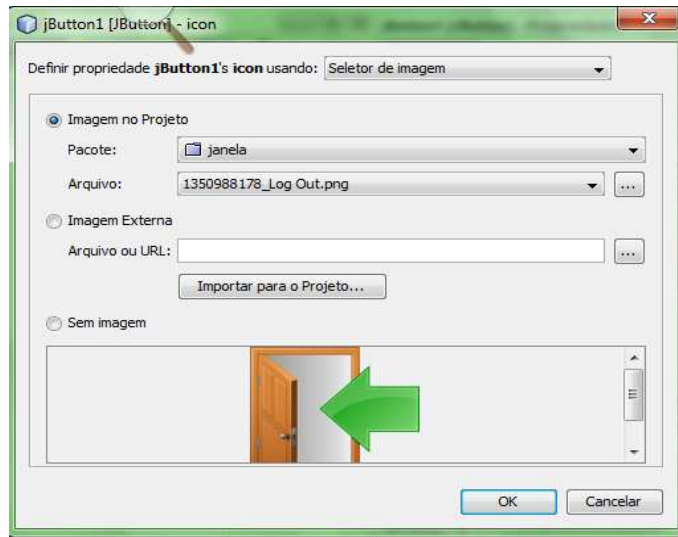


Figura 24 - Imagem selecionada

O componente botão irá mudar o visual onde irá acrescentar a imagem que você selecionou Figura 25 - Botão com figura.

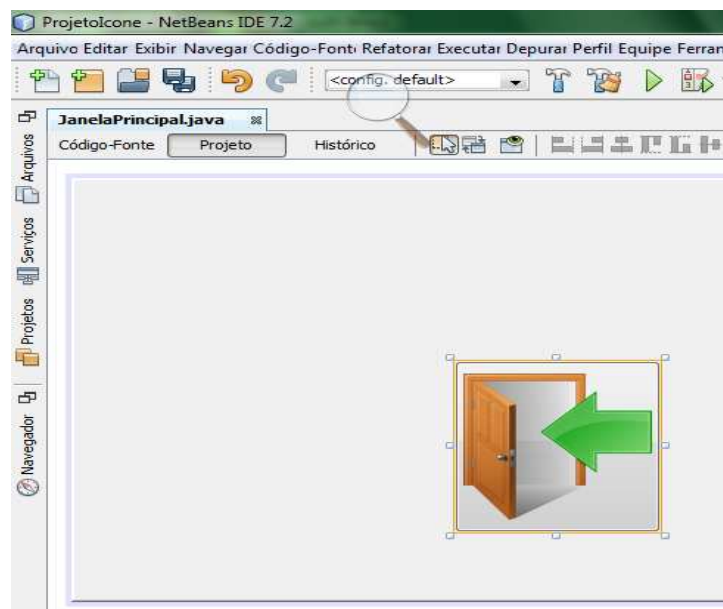


Figura 25 - Botão com figura

Exercício

2. Vamos calcular quanto gastamos com o consumo de combustível de um determinado veículo.

$$\text{Gasto} = \text{distancia} * \text{consumo} * \text{preço}$$

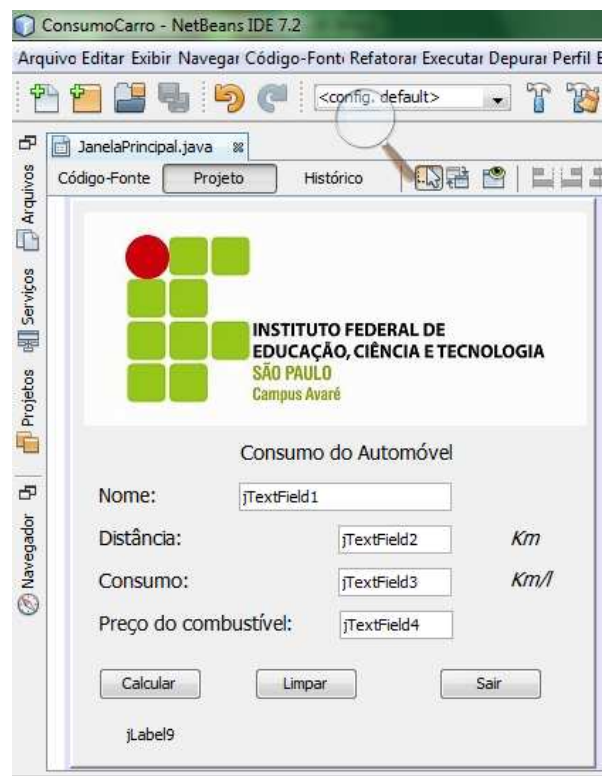


Figura 26 - Tela Exercício 2

Renomeando componentes e adicionando legendas

Renomear um componente é necessário para não se perder na notação padrão, ou seja, tem um monte de campo de texto `(jTextField)` e você precisa pegar o valor de um, mas qual é aquele que tem o dado que você quer? Então precisamos dar nomes mais significativos para os componentes.

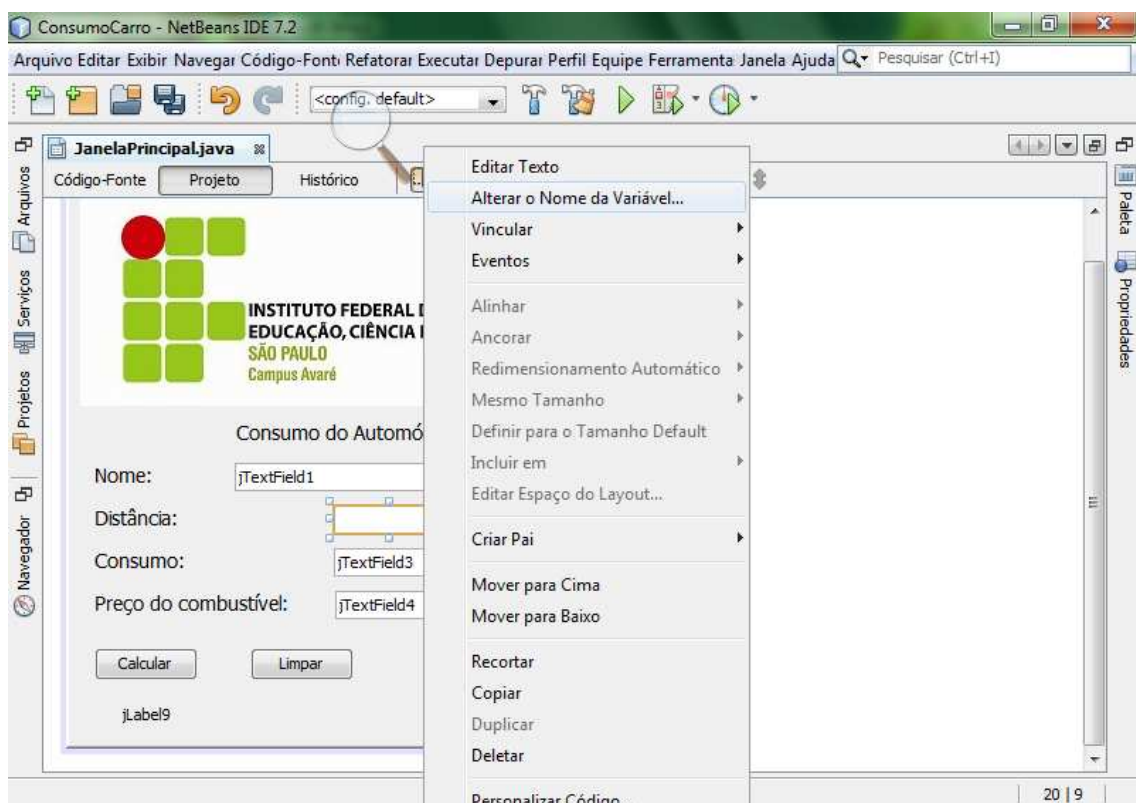


Figura 27 - Renomear componente

Para renomear o componente primeiramente precisamos clicar com o botão direito em cima dele aparecerá uma janela Figura 27 - Renomear componente depois clicar na opção para Alterar o Nome da Variável abrirá uma janela Figura 28 - Renomear componente 2, depois é só renomear o componente.

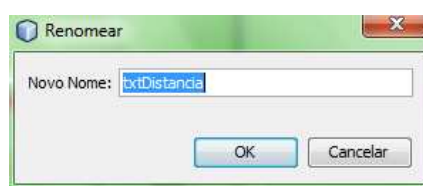


Figura 28 - Renomear componente 2

Legenda, também conhecida como Hint, é a propriedade que você digita um texto e quando o usuário descansar o ponteiro do mouse em cima do componente aparece uma legenda que pode descrever a função do componente ou dar uma dica. Para adicionar essa legenda basta selecionar um componente, tipo botão, selecionar a propriedade `toolTipText`, conforme Figura 29 - Legenda nos componentes, depois adicione a legenda ou a dica do componente.

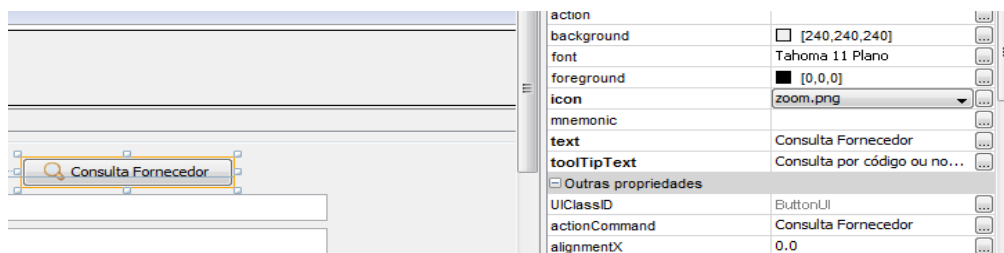


Figura 29 - Legenda nos componentes

Menu, itens de menu e navegação entre as janelas.

Neste capítulo vamos criar um projeto que utilize menus **Figura 30 - Criando Menus**, que vão servir para chamar outras telas do aplicativo ou uma função externa, como abrir uma página no navegador.

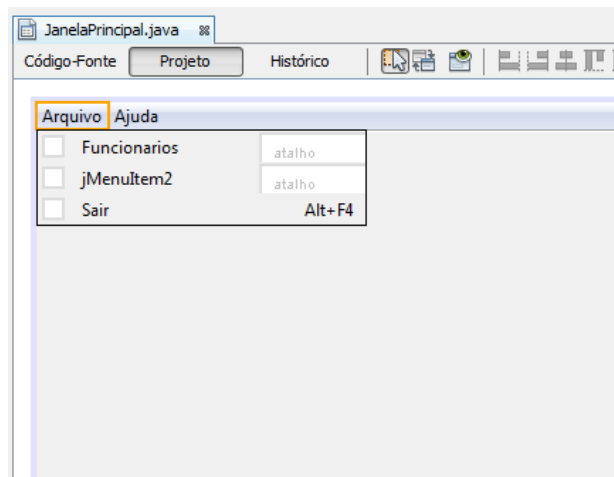


Figura 30 - Criando Menus

Para fazer este programa crie um novo projeto e desta vez selecione criar classe Principal. Para inserir o Menu contido na Paleta: arraste o componente Barra de Menu, que é a barra fixa, o Menu são os itens que possuem Item de Menu, ou seja, no exemplo, **Figura 30 - Criando Menus**, possui dois Menus que foram nomeados como *Arquivo* e *Ajuda* e dentro do menu *Arquivo* foi inserido três Itens de Menu: *Funcionários*, *Sair*, *JMenuItem2*.

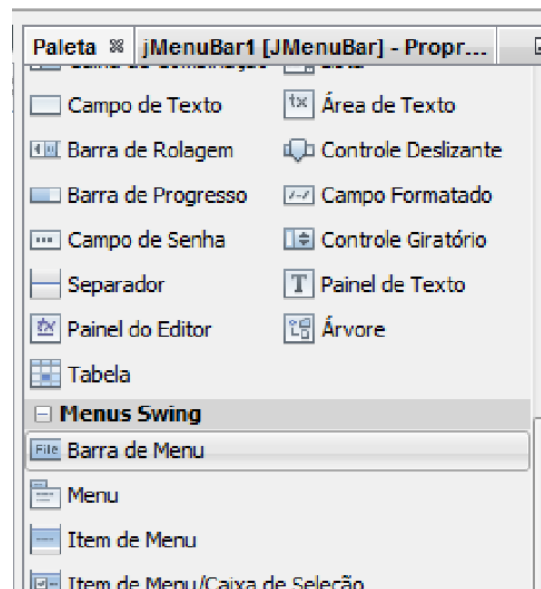


Figura 31 - Paleta Menus Swing

Nesta Figura 32 - Classe Principal fizemos um código para chamar/invocar uma janela, que nosso caso chamamos de *JanelaPrincipal* mas que pode ser alterado com base no nome que você atribuiu ao JFrame que criou.

```
public class Menu {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        JFrame janela = new JanelaPrincipal();  
        janela.setVisible(true);  
        janela.setExtendedState(JFrame.MAXIMIZED_BOTH);  
    }  
}
```

Figura 32 - Classe Principal

- **Códigos para criar uma Janela:**

```
JFrame janela = new Sobre();  
janela.setVisible(true);
```

Onde está sobre mudar
para o Nome que você deu

- **Deixar a nova janela no centro da tela:**

```
janela.setLocationRelativeTo(null);
```

- **Deixar a tela Maximizada (Tela Cheia):**

```
janela.setExtendedState(JFrame.MAXIMIZED_BOTH);
```

- **Fechar a janela atual:**

```
dispose();
```

Ao executar o projeto, com F6, o programa vai abrir a Figura 33 - Janela Principal, mas de uma maneira diferente, pois você solicitou que a janela fosse maximizada, ou seja, que a janela preencha todo espaço da tela do computador.

Agora vamos programar para Janela Principal abrir outra Janela a de Sobre. Para abirmos outras janelas podemos adicionar o código no item Sobre Figura 34 – Código para invocar a Janela Sobre ou em um botão.

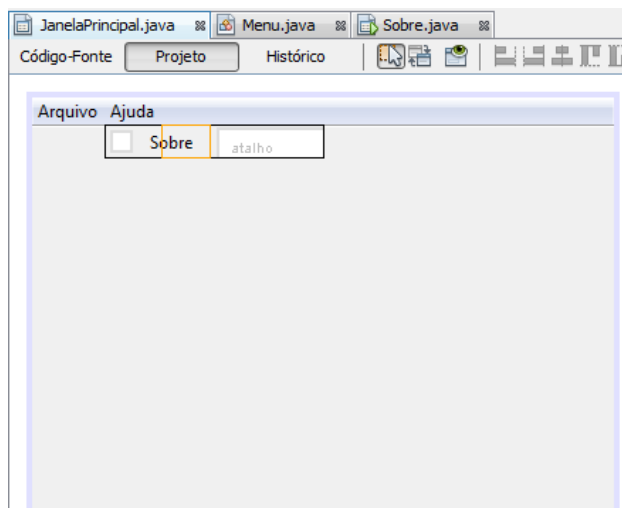


Figura 33 - Janela Principal

Dentro da ação do Item de Menu que chama a janela Sobre, adicione o código da Figura 34 – Código para invocar a Janela Sobre.

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent) {  
    JFrame janela = new Sobre();  
    janela.setVisible(true);  
    janela.setLocationRelativeTo(null);  
}
```

Figura 34 – Código para invocar a Janela Sobre

Depois disso uma nova janela vai surgir na tela. (Desde que haja um JFrame com este nome).

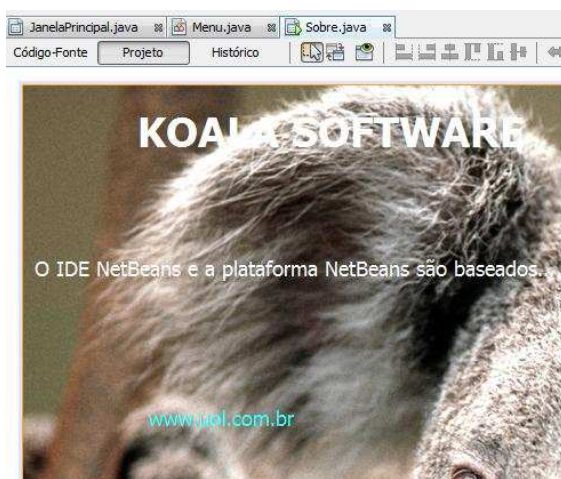


Figura 35 - Janela Sobre

Botão de Rádio e Grupo de Botões

As funções dos componentes `RadioButton` e `CheckBox` são semelhantes, conhecidos como botões de opções neles os usuário podem marcar ou desmarcar as opções. A diferença é que o usuário seleciona um botão de opção qualquer e o botão de rádio selecionado anteriormente no mesmo grupo torna-se desmarcado.

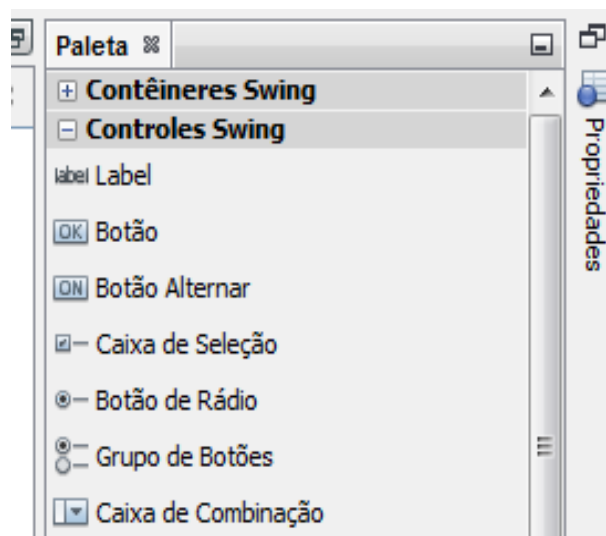


Figura 36 - Paleta Botão e Grupo de Rádio

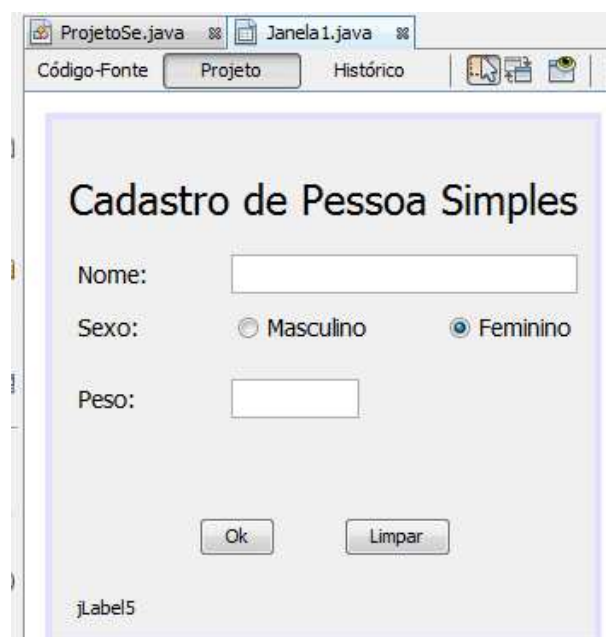


Figura 37 - Tela Cadastro de Sexo

Neste projeto estamos utilizando o componente Botão de Rádio, para mostrar as opções que o usuário pode selecionar. Para criar a tela Figura 37 - Tela Cadastro de Sexo precisamos adicionar dois novos componentes contidos na Figura 36 - Paleta Botão e Grupo de Rádio, além dos componentes já estudados aqui, o primeiro componente é o Botão de Rádio ou `RadioButton`, para cada opção que você deseja adicionar tem que adicionar um Botão de Rádio, neste caso nos vamos adicionar apenas dois, um para masculino e outro para feminino. Depois precisamos adicionar o Grupo de botões, ele é arrastado para o projeto e pode ficar apenas 1 (neste caso), que será para indicar para os componentes masculino e feminino que eles pertencem ao mesmo grupo e quando a pessoa selecionar masculino, não pode selecionar feminino junto, ou vice-versa.

Quando você arrasta um componente Grupo de botões o painel Navegação fica como na Figura 38 - Aba navegador programa com grupo de botões, este componente aparece fora do `JFrame`, isto é normal, pois ele está atrelado ao projeto, mas não é um elemento visual.

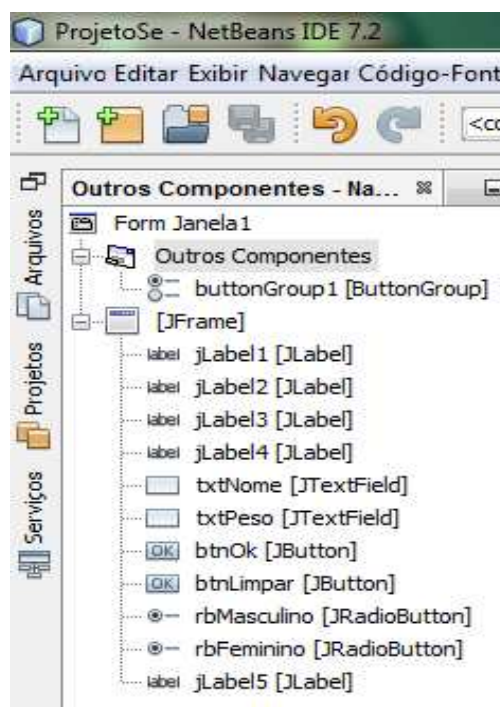


Figura 38 - Aba navegador programa com grupo de botões

Para vincular os componentes primeiro selecione o botão de opção feminino e altere o valor da propriedade buttonGroup Figura 39 - Propriedade do botão de opção, na propriedade selecione a opção buttonGroup1, que é o componente arrastado na tela e que não ficou visível.

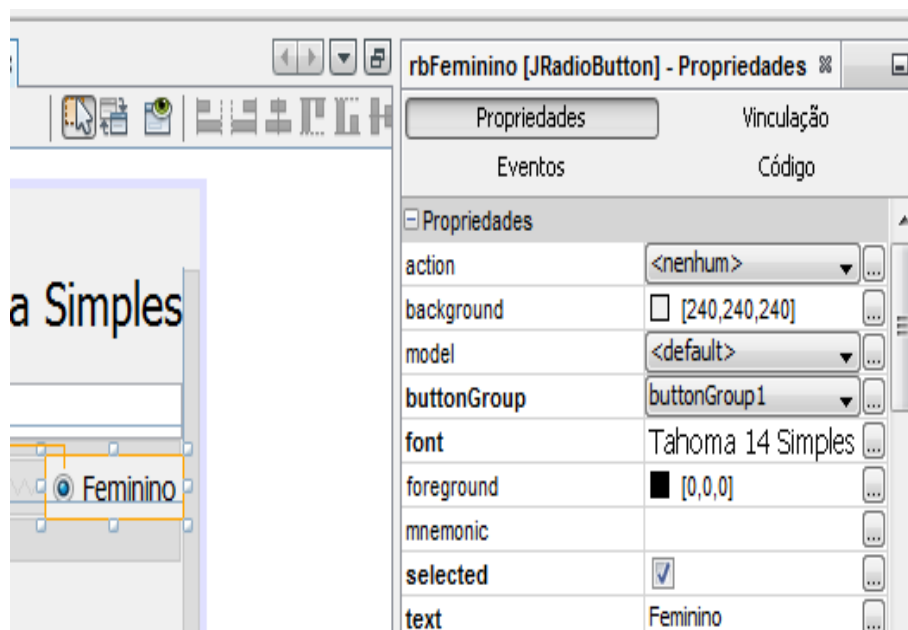


Figura 39 - Propriedade do botão de opção

No botão de opção de masculino fazer o mesmo passo e depois disso quando você executar o projeto o usuário do seu sistema não vai poder selecionar duas opções, ficando opção exclusiva, ou seja, ou ele cadastra a pessoa com o sexo masculino ou com sexo feminino, nunca as duas.

Agora precisamos codificar, conforme exemplo Figura 40 - Código do botão OK usando Opção, para que quando o usuário do sistema digitar um nome e escolher a opção do sexo o programa vai mostrar uma mensagem que vai utilizar o pronome de tratamento de senhor para sexo masculino e senhora ou senhorita, além de pegar o peso de ambos.

```

private void btnOkActionPerformed(java.awt.event.ActionEvent evt) {
    String nome = txtNome.getText();
    Double peso = Double.valueOf(txtPeso.getText());
    //Verificar o sexo da pessoa
    if (rbFeminino.isSelected()) {
        jLabel5.setText("Senhora " + nome + " você pesa "
            + peso + " kg, você emagreceu.");
    } else {
        jLabel5.setText(nome + " você está gordo cara "
            + peso + " kg.");
    }
}

```

Figura 40 - Código do botão OK usando Opção

Exercícios

3. Fazer um programa que pegue o nome e a idade de uma pessoa e mostre uma mensagem. Na mensagem mostrar o nome da pessoa e a qual faixa etária ela pertence de acordo com Tabela 2 - Índice de Massa Corporal:

Tabela 1 - Mensagem por faixa etária

Faixa etária (em anos)	Mensagem
De 0 a 2	Bebê
De 3 a 11	Criança
De 12 a 19	Adolescente
De 20 a 30	Jovem
De 31 a 60	Adulto
Maior de 60	Melhor idade

4. Calcule o IMC – Índice de Massa Corporal Figura 41- Tela IMC, conforme fórmula abaixo:

Onde: $imc = peso / altura^2$



Figura 41- Tela IMC

O resultado verificar o Índice que a pessoa pertence na Tabela 2 - Índice de Massa Corporal:

Tabela 2 - Índice de Massa Corporal

Índice IMC	Descrição
Menor que 18,5	Peso abaixo do Normal
Entre 18,5 e 24,4	Peso Ideal
Entre 24,5 e 29,9	Pré-Obesidade
Entre 30 e 34,9	Obesidade classe I
Entre 35 e 39,9	Obesidade classe II (severa)
Maior que 39,9	Obesidade classe III (mórbida)

Campo formatado e caixa de seleção

Agora vamos aprender a deixar o campo de entrada com máscaras, recurso bastante utilizado para entrada de dados que possui um padrão como por exemplo: CPF possui valor fixo, número de telefone, CNPJ, título de eleitor e outros. Para desenvolver os exemplos vamos criar um projeto e desenhar uma tela igual Figura 42 - Tela Cadastro de Pessoas, nos campos que estão com nome de `JFormattedTextField` você adiciona o componente Campo Formatado da Figura 43 - Paleta Campo Formatado.

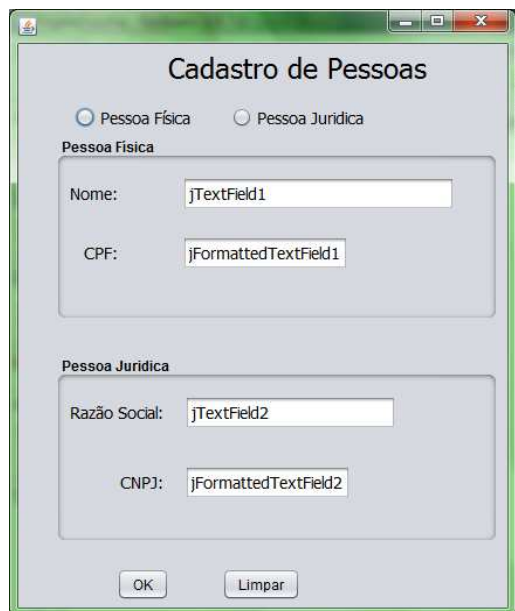


Figura 42 - Tela Cadastro de Pessoas

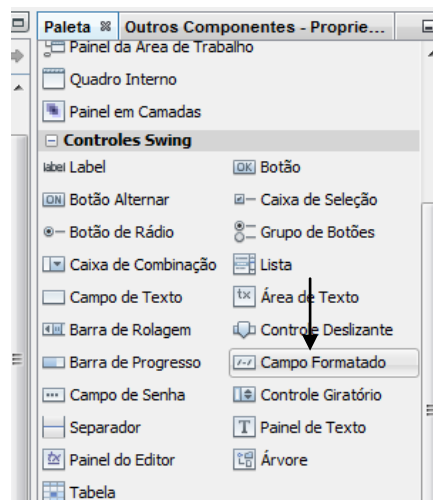


Figura 43 - Paleta Campo Formatado

Para alterar o valor de entrada primeiramente selecione o campo `JFormattedTextField1`, e na aba de propriedades e selecione os três pontinhos da propriedade `formatterFactory`, Figura 44 - Componente campo formatado - incluindo máscara 1.

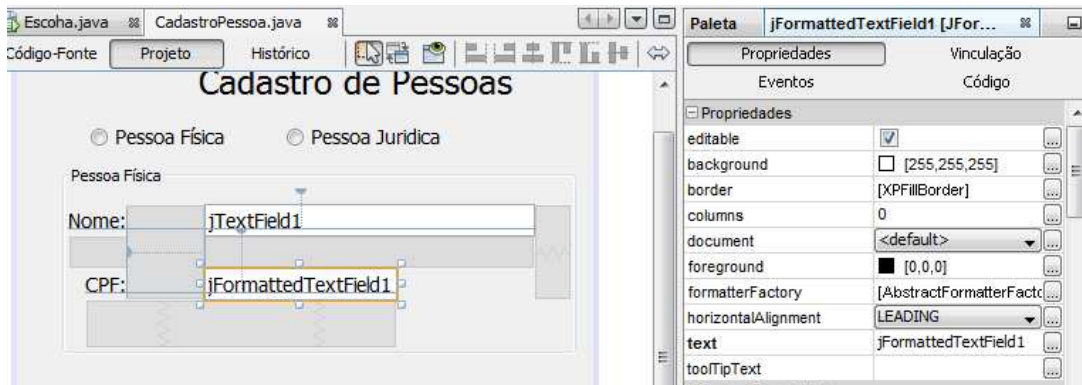


Figura 44 - Componente campo formatado - incluindo máscara 1

Logo após selecione os três pontinhos da propriedade `formatterFactory`, surgirá à tela Figura 45 - Componente campo formatado - incluindo máscara 2,

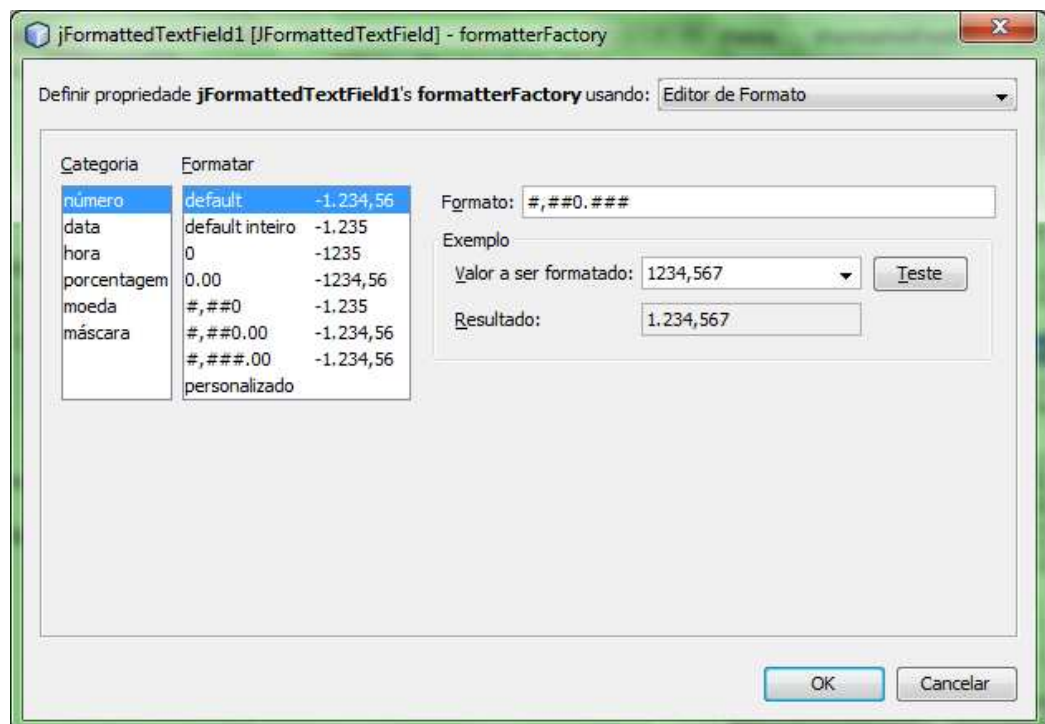


Figura 45 - Componente campo formatado - incluindo máscara 2

Nesta janela selecione categoria máscara, formatar escolha personalizado agora teremos que definir o padrão da máscara, pois o campo em questão é de CPF, então todo CPF tem uma máscara igual que é: ###.###.###-##, onde # significa um número qualquer, conforme Figura 46 - Componente campo formatado - incluindo máscara 3.

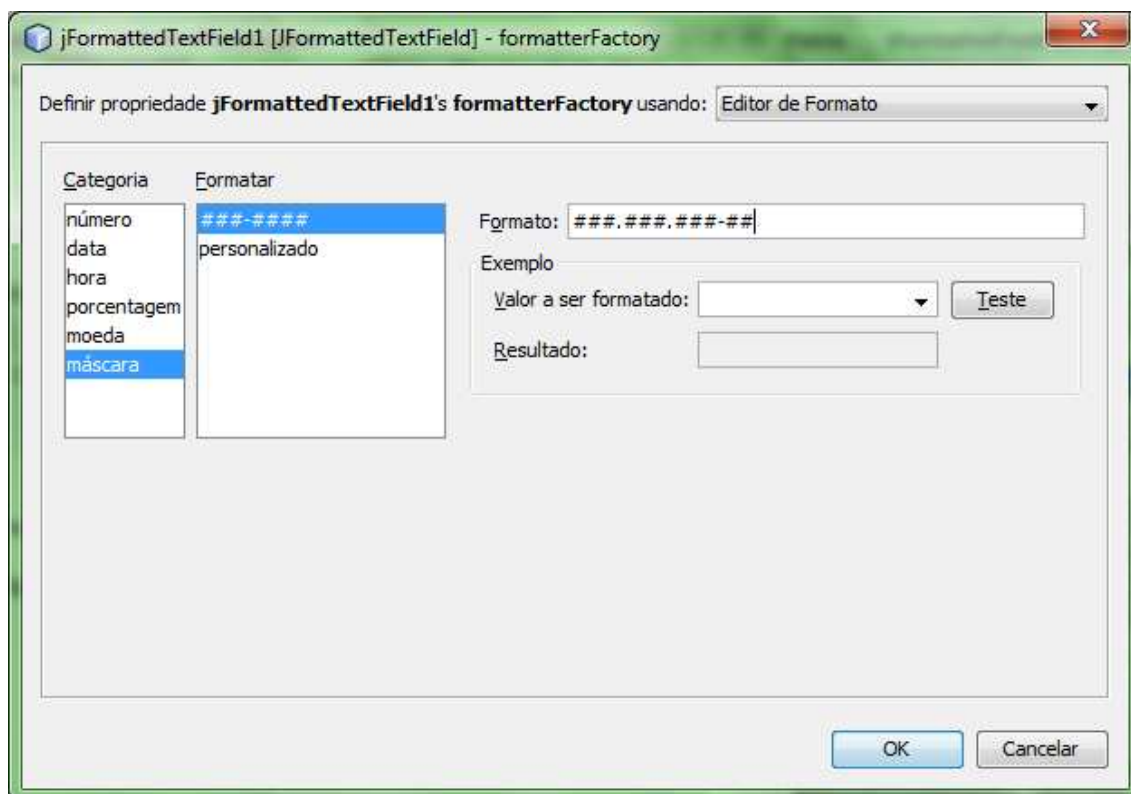


Figura 46 - Componente campo formatado - incluindo máscara 3

Depois de realizar este procedimento o campo CPF ficará travado e aceitará apenas no formato definido, que é o padrão para o CPF Figura 47 - Tela final do campo formatado:

Cadastro de Pessoas

☐ Pessoa Física ☐ Pessoa Juridica

Pessoa Física

Nome:

CPF:

Pessoa Juridica

Razão Social:

CNPJ:

OK Limpar

Figura 47 - Tela final do campo formatado

Depois faremos o mesmo processo para o campo formatado de CNPJ, tem formato um pouco diferente do CPF, então temos que verificar o formato e adicionar nas propriedades.

Como já foi dito o componente Caixa de Seleção ele é parecido com Botão de Rádio de selecionar as opções que são verdadeiras, mas com a diferença de poder selecionar várias opções ao mesmo tempo, por exemplo, vamos desenvolver um aplicativo que verifica qual é o gosto musical das pessoas, conforme Figura 48 - Tela Gosto Musical, repare que há várias opções e que o usuário pode selecionar todas, nenhuma ou qualquer uma.

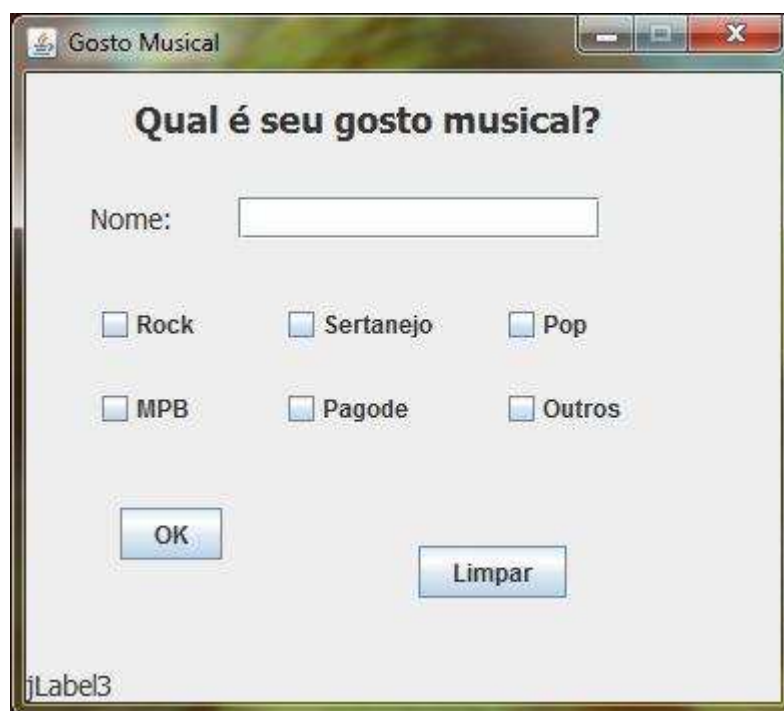


Figura 48 - Tela Gosto Musical

Para adicionar um componente Caixa de Seleção no projeto basta arrastar ele da Figura 49 - Paleta Caixa de Seleção para a Janela/JFrame do projeto.



Figura 49 - Paleta Caixa de Seleção

Na hora de programar temos que testar todas as Caixas de Seleção para ver se alguma caixa esta selecionada, Figura 50 - Código botão OK do gosto musical, neste código estamos verificando apenas duas caixas de seleção a de Rock e de Pagode, mas com esta demonstração dá para programar as outras caixas restantes

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String nome = txtNome.getText();  
    String mensagem = "Caro "+nome + " você gosta de: ";  
  
    if(cbRock.isSelected()){  
        //Ele pega a mensagem antiga mais o que vc adicionou  
        mensagem = mensagem + " rock ";  
    }  
    if(cbPagode.isSelected()){  
        mensagem = mensagem + " pagode ";  
    }  
    //Fazer o resto do códigos  
  
    jLabel13.setText(mensagem);  
  
}
```

Figura 50 - Código botão OK do gosto musical

Para codificar o botão limpar tem que informar que todas as caixas de seleção terão que assumir que não ficaram selecionadas, Figura 51 - Código do botão Limpar do gosto musical.

```
private void btnLimparActionPerformed(java.awt.event.ActionEvent evt) {  
    txtNome.setText("");  
    cbRock.setSelected(false);  
    cbMpb.setSelected(false);  
    cbOutros.setSelected(false);  
    cbPagode.setSelected(false);  
    cbPop.setSelected(false);  
    cbSertanejo.setSelected(false);  
}
```

Figura 51 - Código do botão Limpar do gosto musical

Exercício

5. Faça um programa que receba alguns dados da pessoa e mostre o resultado do que a pessoa selecionou ou digitou no programa. Figura 52 - Tela Ficha de Cadastro



The image shows a Java Swing window titled "Ficha de Cadastro". The window has a light gray background and a title bar with standard Windows controls. The main content area contains the following elements:

- Nome:** A text input field.
- Estado Civil:** A dropdown menu currently showing "Solteiro".
- Fuma:** Two radio buttons labeled "Sim" and "Não", with "Não" selected.
- Estudou/Estuda em Avaré:** A group box containing five checkboxes:
 - ☐ 1° ao 6° ano
 - ☐ 7° ao 9° ano
 - ☐ Ensino Médio
 - ☐ Ensino Técnico
 - ☐ Ensino Superior
- Buttons:** Two buttons at the bottom, "Cadastrar" and "Limpar".
- Label:** A label "jLabel6" is located at the bottom left of the window.

Figura 52 - Tela Ficha de Cadastro

Obras Citadas

DEVMEDIA. (s.d.). *DEVMEDIA*. Acesso em 01 de 04 de 2013, disponível em DEVMEDIA: <http://www.devmedia.com.br/>

Java Free. (s.d.). *UOL*. Acesso em 01 de 04 de 2013, disponível em Java Free: <http://javafree.uol.com.br/wiki/JFrame>