

# Desenvolvimento Desktop

## Aula 08 - Componentes JPasswordField, JFormattedTextField e JTable

# Apresentação

---

Nesta aula, conheceremos mais três componentes do Java: o **Campo de senha (JPasswordField)**, utilizado para exibir o conteúdo digitado, geralmente, como asteriscos (\*); o **Campo formatado (JFormattedTextText)**, utilizado para a criação de campos já predefinidos, como por exemplo, número de CPF, data de nascimento, etc., de forma que o usuário não preencha esses campos com informações inválidas; e a **Tabela (JTable)**, utilizada para organizar dados alinhados em linhas e colunas. Vamos conhecer algumas particularidades desses componentes e como utilizá-los nas suas aplicações.



**Vídeo 01** - Apresentação

## Objetivos

Ao final desta aula, você será capaz de:

- Definir Campo de senha, Campo formatado e Tabela.
- Aplicar esses componentes, personalizando suas propriedades de forma a adaptar-se às suas necessidades.

# O Componente Campo de Senha (JPasswordField)

---

Esse componente é muito comum em qualquer interface que solicite uma senha de acesso. O Campo de senha é uma caixa de texto, cujo conteúdo digitado não é exibido como em uma caixa de texto convencional (**JTextField**), já visto anteriormente.

Nesse componente, o texto digitado geralmente aparece como asteriscos (por exemplo, "\*\*\*\*\*", e também pode aparecer como bolinhas "•••••"), independentemente do que seja digitado, a fim de manter o sigilo da informação.

Com certeza você já viu várias aplicações usando esse componente, tanto na internet, ao acessar seu e-mail, áreas restritas (como contas bancárias) ou até ao baixar um arquivo de uma área privada. O próprio Windows permite configurar uma senha no perfil do usuário, o Excel (uma planilha eletrônica da Microsoft) possibilita proteger o conteúdo de sua planilha através de uma senha, além de diversos aplicativos que disponibilizam a opção de cadastrar uma senha para liberar o acesso a todo, ou parte do conteúdo do programa, restringindo o acesso de certas informações a determinados usuários.

## Como Utilizar o Campo de Senha em uma Aplicação

Vejamos como utilizar esse componente em uma aplicação prática:

1. Execute o NetBeans e crie um novo projeto (**Arquivo > Novo projeto**).
2. Escolha um nome e salve seu projeto.
3. Crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **Campo de senha** para o **JFrame**.
5. Mesmo na **Paleta**, perceba que o desenho que representa o componente é um retângulo com algumas bolinhas dentro. É assim que o componente se apresenta, conforme ilustra a **Figura 1**.

**Figura 01** - Componente Campo de senha



6. Como visto em aulas anteriores, para alterar as suas dimensões (largura e altura), utilize as alças de dimensionamento, ou diretamente na paleta de **Propriedades**.
7. Executando a aplicação (tecla **F6**), note que aparecerá um retângulo branco com o cursor piscando no final de uma sequência de bolinhas, como na **Figura 2**.

**Figura 02** - Componente Campo de senha após execução da aplicação



8. Nessa situação, perceba que ao digitar qualquer texto o conteúdo apresentado sempre será exibido da mesma forma, como na figura anterior (**Figura 2**).

**Obs.:** Geralmente, para se cadastrar uma senha, são solicitados de 4 a 8 caracteres, incluindo caracteres numéricos e alfanuméricos. Nesse caso, esse componente não precisa ter dimensões muito grandes, visto que seu conteúdo será curto.

A seguir, veremos algumas propriedades essenciais na utilização desse componente.

---

## As Propriedades do Componente Campo de Senha

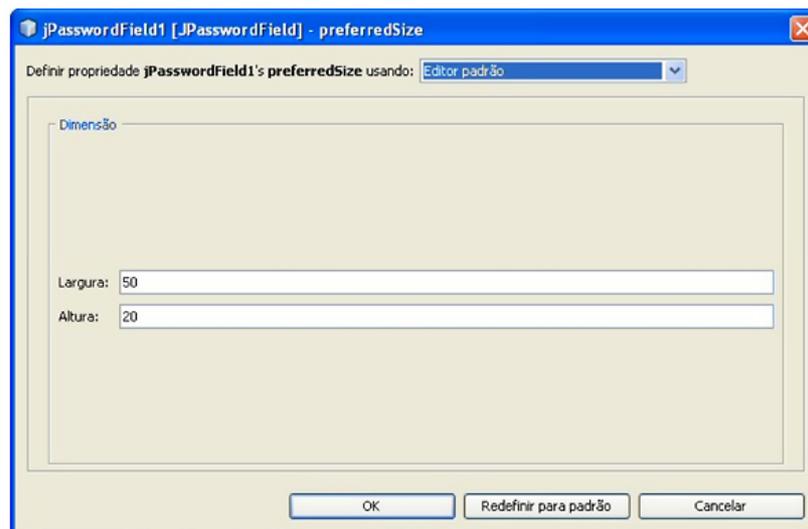
Dentre as propriedades que já vimos, experimente alterar algumas sem consultar as aulas anteriores. Lembrando que algumas propriedades são muito importantes para determinados componentes.

Em particular, assim como nos outros componentes de texto (**JTextField** e **JTextArea**, por exemplo) que geralmente não apresentem nenhum texto predefinido, a propriedade **text** desse componente deve ficar vazia, principalmente no **JPasswordField**, no qual qualquer texto digitado não será exibido.

Na seção **Outras propriedades**, você poderá configurar as seguintes propriedades:

- **caretColor** – funciona da mesma forma que outras propriedades de definição de cores, só que essa determina a cor do cursor dentro do componente.
- **preferredSize** – permite definir os valores de largura e altura do componente em número de pixels. A **Figura 3** apresenta a janela correspondente a essa propriedade.

**Figura 03** - Propriedade preferredSize do componente Campo de senha



- **caretPosition** – permite definir em que posição o cursor estará quando a aplicação for executada. Lembre-se de que deve haver algum texto predefinido para que você visualize o resultado dessa

propriedade, caso contrário, o cursor aparecerá no canto esquerdo do componente.

- **cursor** – permite alterar o formato do cursor quando o componente estiver com o foco.
- **echoChar** – essa propriedade permite alterar o caractere que deverá substituir o texto que for digitado no campo. Para isso, altere apenas o valor numérico.



**Vídeo 02** - JPasswordField

## Atividade 01

---

1. Cite duas situações em que você utilizaria o componente **Campo de senha**, descrevendo o objetivo do uso desse componente em cada uma das situações.
2. Que propriedade devemos utilizar para alterar a cor do cursor de um componente **Campo de senha**?
3. Especifique quatro propriedades que você poderia alterar para personalizar o componente **Campo de senha**.

## O Componente Campo Formatado (JFormattedTextField)

---

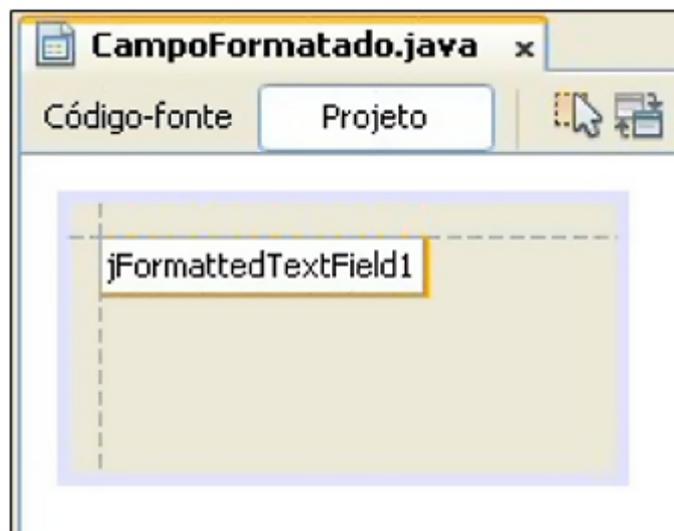
Campos formatados servem para especificar a formatação da informação que deverá ser digitada em um campo de texto (**JFormattedTextField**). A formatação pode servir, por exemplo, para definir formatos de datas, número de telefones, número do CEP, CPF etc. Ela não só facilita a entrada de dados, como evita que o usuário digite a informação errada.

# Como Utilizar o Campo Formatado em uma Aplicação

Vejamos passo a passo como utilizar esse componente:

1. Execute o NetBeans e crie um novo projeto (**Arquivo > Novo projeto**).
2. Defina um nome e salve seu projeto.
3. Crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **Campo formatado** para o **JFrame**.
5. Aparentemente, é idêntico ao **Campo de texto**, como mostra a **Figura 4**, entretanto, esse componente permite uma configuração que é bastante útil em determinadas situações.

**Figura 04** - Componente Campo formatado



6. A aparência de um campo de texto formatado ao ser executado é igual a um campo de texto convencional, não dá para distinguir um do outro (dependendo do tipo de formatação aplicado).
7. Execute a aplicação (tecla **F6**). Veja que aparentemente você não teria como perceber a diferença entre um **Campo de texto**, um **Campo de senha** e o **Campo formatado**. Confira com a **Figura 5**.

**Figura 05** - Semelhança entre três componentes de texto



**Obs.:** O componente **Campo formatado** é utilizado quando se precisa definir o formato da informação que deve ser digitada. Geralmente, tem dimensões curtas.

A seguir, você conhecerá algumas propriedades determinantes na utilização desse componente.

## As Propriedades do Componente Campo Formatado

A principal funcionalidade desse componente é poder definir um formato para a informação digitada. Na seção **Propriedades**, identifique a propriedade a seguir.

- **formatterFactory** – essa é a propriedade mais importante desse componente. Ela permite a configuração do formato a ser estipulado para o componente e dispõe das seguintes categorias:
- **Número** – essa categoria dispõe de diversos formatos de números. Confira com a **Figura 6**.

**Figura 06** - Padrões de formato do tipo número

Categoria	Formatar
número	padrão -1.234,56
data	padrão inteiro -1.235
hora	0 -1235
porcentagem	0.00 -1234,56
moeda	#,##0 -1.235
máscara	#,##0.00 -1.234,56
	#,###.00 -1.234,56
	personalizado

- **Data** – nessa categoria, você poderá definir entre vários formatos de data disponíveis, conforme mostra a **Figura 7**.

**Figura 07** - Padrões de formato do tipo data

Categoria	Formatar
número	padrão 27/08/2010
data	breve 27/08/10
hora	médio 27/08/2010
porcentagem	longo 27 de Agosto de 2010
moeda	completo Sexta-feira, 27 de Agosto de 2010
máscara	personalizado

- **Hora** – essa categoria traz os principais formatos de hora utilizados, do mais reduzido ao mais completo. Veja a **Figura 8**.

**Figura 08** - Padrões de formato do tipo hora

Categoria	Formatar
número	padrão 22:55:12
data	breve 22:55
hora	médio 22:55:12
porcentagem	longo 22h55min12s GMT-03:00
moeda	completo 22h55min12s GMT-03:00
máscara	personalizado

- **Porcentagem** – basicamente, os dois formatos mais utilizados são: sem casas decimais e com duas casas decimais (**Figura 9**).

**Figura 09** - Padrões de formato do tipo porcentagem

Categoria	Formatar
número	padrão -123.456%
data	0% -123456%
hora	0.00% -123456,00%
porcentagem	personalizado
moeda	
máscara	

- **Moeda** – formatos de moeda com e sem os centavos. Veja a **Figura 10**.

**Figura 10** - Padrões de formato do tipo moeda

Categoria	Formatar
número	padrão -R\$ 1.234,56
data	x#,##0 -R\$1.235
hora	x#,##0.00 -R\$1.234,56
porcentagem	x#,##0.-- -R\$1.235,--
moeda	xx#,##0 -BRL1.235
máscara	xx#,##0.00 -BRL1.234,56
	xx#,##0.-- -BRL1.235,--
	personalizado

- **Máscara** – nessa categoria, você define uma máscara qualquer. Podemos relembrar os exemplos do número de CEP e CPF, já citados, que são bastante utilizados.

**Figura 11** - Padrões de formato do tipo máscara

Categoria	Formatar
número	###-###
data	personalizado
hora	
porcentagem	
moeda	
máscara	

Você deve ter observado que em todas as categorias existe a possibilidade de personalizar o formato.



**Vídeo 03** - JFormattedTextField

## Atividade 02

1. Qual a principal razão de se utilizar um componente do tipo **Campo formatado**? Explique.

2. Cite pelo menos quatro categorias para se formatar um componente **Campo formatado**.
3. Cite pelo menos dois exemplos de formatação que poderíamos definir na categoria **máscara**, além de CEP e CPF.

## O Componente Tabela (JTable)

---

Entende-se por tabela uma espécie de planilha cuja utilidade principal é organizar informações provenientes de alguma fonte de dados, como bancos de dados ou de um arquivo texto, por exemplo. As informações também podem ser digitadas diretamente na planilha, mas não é aconselhável quando se tratar de muita informação, principalmente quando for necessário alterá-las.

Essas informações são mostradas em uma espécie de grade utilizando linhas e colunas. As colunas representam os nomes dos campos (nome, endereço, telefone, e-mail etc.) e as linhas representam os dados propriamente ditos. À área referente à interseção de uma coluna e uma linha dá-se o nome de célula (**Figura 12**). As tabelas são muito utilizadas quando se quer exibir informações de forma tabular. Como exemplo, podemos citar: um histórico escolar, uma lista de produtos, um cardápio, entre outros.

Para se trabalhar com tabelas, alguns passos precisam ser seguidos, como criar a tabela, determinar o tamanho das linhas de colunas, criar um modelo para a tabela etc. Esses passos serão detalhados a seguir.

---

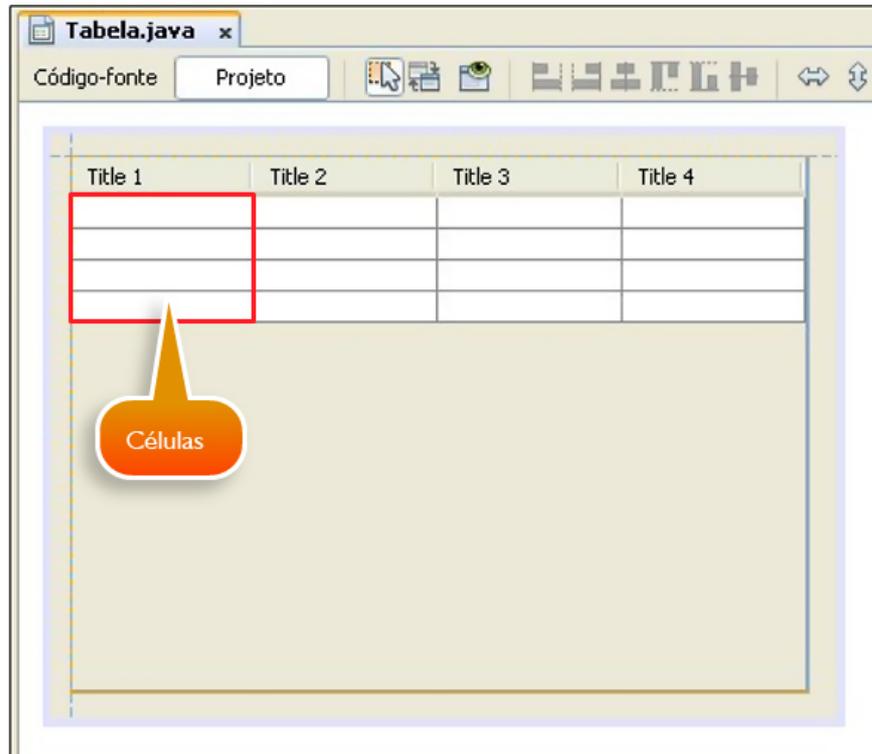
## Como Usar uma Tabela em uma Aplicação

Vejamos como implementar esse componente na prática:

1. Execute o NetBeans e crie um novo projeto (**Arquivo > Novo projeto**).
2. Defina um nome e salve seu projeto.
3. Crie um novo **Formulário JFrame**.
4. Clique e arraste um componente **Tabela** para a área do projeto.

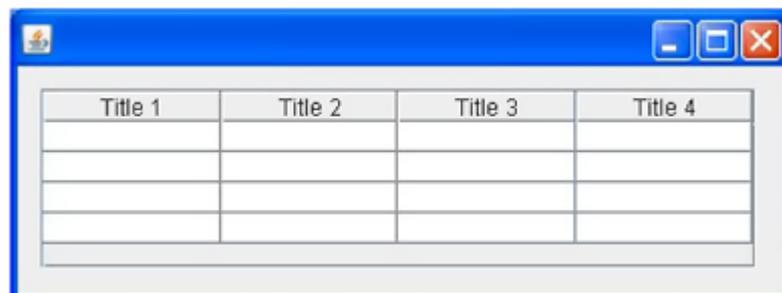
5. Visualmente, você já vai ver uma tabela pronta, de cinco linhas e quatro colunas, conforme mostra a **Figura 12**.

**Figura 12** - Componente Tabela com a estrutura padrão



6. Altere as dimensões (largura e altura), tanto do componente como do **JFrame**, utilizando as alças de dimensionamento.
7. Executando a aplicação (tecla **F6**), você verá o componente como apresentado na **Figura 13**, com sua estrutura padrão.

**Figura 13** - Componente Tabela após a execução da aplicação



8. Veja que agora você pode inserir informações nas células, alterá-las, apagá-las e até alterar a ordem das colunas, mas não pode alterar a identificação das colunas. Veja a **Figura 14** a seguir.

**Figura 14** - Componente Tabela com as possíveis alterações permitidas no aplicativo



A seguir, veremos como alterar algumas propriedades desse componente definindo-o de acordo com a sua necessidade.

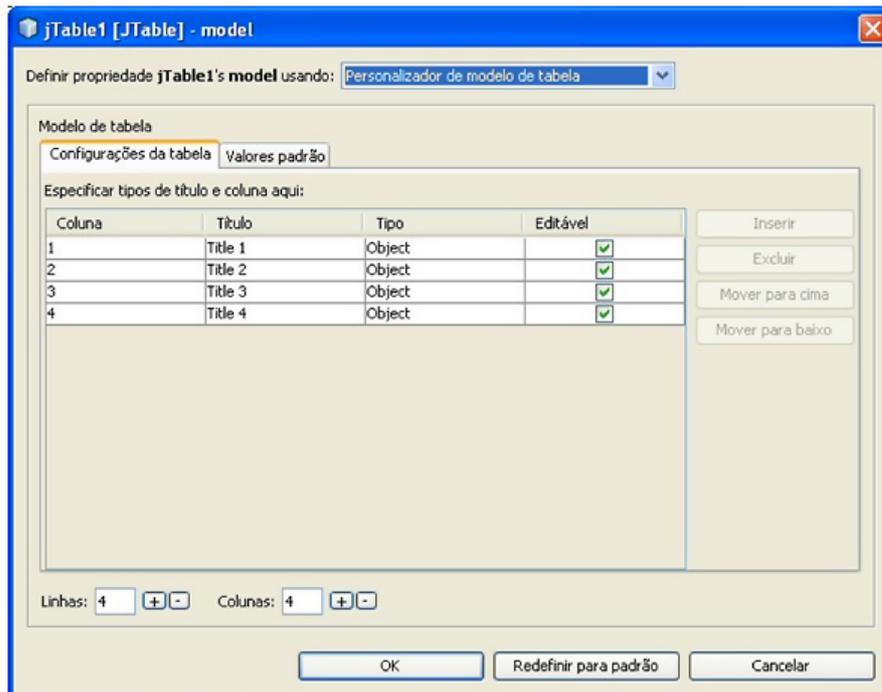
---

## As Propriedades do Componente Tabela

Você verá agora algumas propriedades desse componente que poderão ser alteradas para uso em uma aplicação. Na seção **Propriedades**, você encontrará várias propriedades já vistas em aulas anteriores, mas para esse componente uma em particular é essencial:

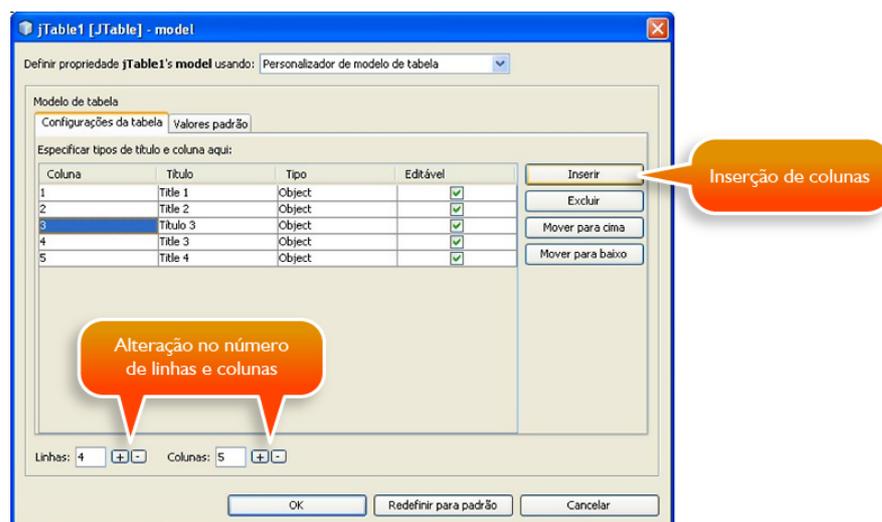
- **model** – através dessa propriedade você poderá definir várias características importantes desse componente. Passo a passo, explorando as telas, veremos quais são essas características e como alterá-las.
1. Clicando no pequeno quadrado com reticências ao lado dessa propriedade, você abrirá a janela mostrada na **Figura 15**.

**Figura 15** - Explorando a propriedade model



2. Perceba, inicialmente, que na parte inferior da janela existem dois campos de texto: o primeiro para a definição da quantidade de linhas e o segundo para a quantidade de colunas. Você poderá alterar diretamente, modificando os valores, ou utilizando os botões de adição e subtração à direita (**Figura 16**).

**Figura 16** - Alterando o número de linhas e colunas



3. Na Guia **Configurações da tabela**, observe as seguintes opções:

- **Coluna** – o valor em cada linha define a ordem de exibição das colunas quando o aplicativo for executado.

- **Título** – nesse campo, você poderá alterar o conteúdo, definindo o texto que aparecerá no topo de cada coluna. Veja a **Figura 14**.
- **Tipo** – clicando nessa opção, você verá as diversas possibilidades para esse campo. Algumas opções são: **Objeto, String, Boleano, Inteiro**, entre outras.
- **Editável** – esse **checkbox** define se o campo estará disponível para edição ou não. Caso não esteja marcado, o usuário não poderá alterar o conteúdo da coluna na aplicação.

4. Na Guia **Valores padrão**, você poderá definir diretamente o valor padrão para cada célula (se for o caso).

Na seção **Outras propriedades**, você poderá também configurar as seguintes propriedades:

- **autoCreateRowSorter** – essa propriedade permite ordenar o conteúdo de cada coluna, de **A** para **Z** e vice-versa. Para isso, basta clicar na coluna desejada.
- **columnCount** – essa propriedade mostra a quantidade de colunas da tabela, definida na propriedade anterior (**model**).
- **gridColor** – através dessa propriedade, você define a cor das linhas de grade da tabela, caso estejam em exibição de acordo com as propriedades **showHorizontalLines** e **showVerticalLines**.
- **rowCount** – essa propriedade exibe a quantidade de linhas da tabela.
- **rowHeight** – o valor dessa propriedade define a altura de cada linha da tabela.
- **showHorizontalLines** – se o **checkbox** estiver marcado, indica que as linhas horizontais da tabela serão exibidas. Caso contrário, estarão ausentes.
- **showVerticalLines** – funciona de forma similar à propriedade anterior. Se o **checkbox** estiver marcado, as linhas verticais serão exibidas. Caso contrário, estarão ausentes.

- **tableHeader** – essa propriedade dispõe de dois checkboxes. O primeiro permite, ou não, que o usuário dimensione a largura das colunas. O segundo permite, ou não, que o usuário permute as colunas.

Vejamos agora como utilizar esse componente em uma aplicação prática:

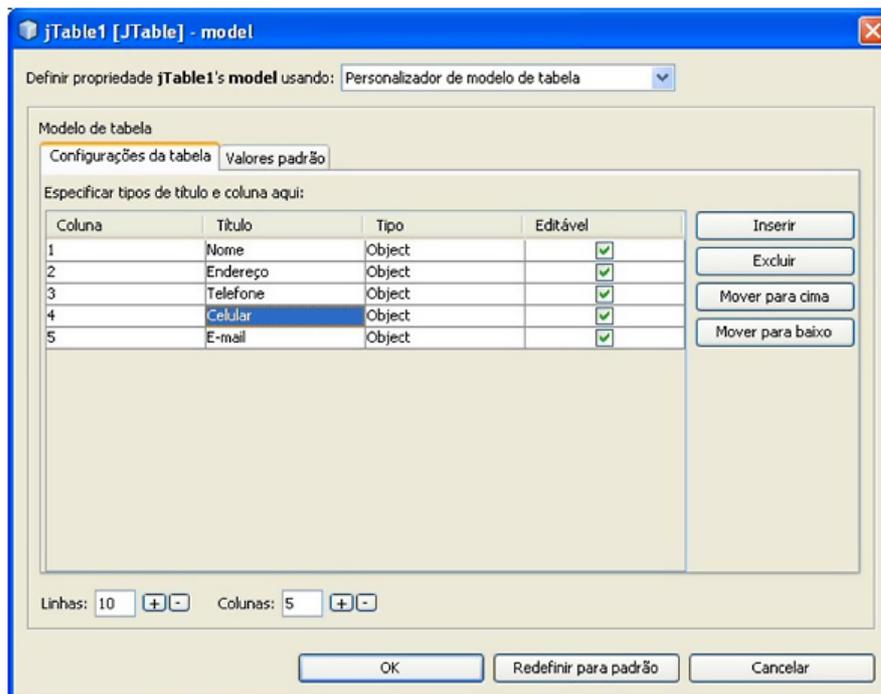
1. Crie um novo projeto (**Arquivo > Novo projeto**).
2. Salve seu projeto com o nome de: **ComponenteTabela**.
3. Crie um novo **Formulário JFrame** com o nome: **Tabela**.
4. Clique e arraste um componente **Tabela** para a área do projeto.
5. Com o componente selecionado, clique no pequeno botão à direita da propriedade **model** (**Figura 17**).

**Figura 17** - Editando a propriedade model



6. Altere os itens dessa lista pelos mostrados na **Figura 18**.

**Figura 18** - Janela como os novos itens da Tabela



7. Clique em **OK** para concluir.
8. De volta ao projeto, utilize as alças do componente para redimensionar a largura e a altura.
9. Executando (**F6**) sua aplicação agora, sua planilha deverá ficar semelhante à da **Figura 19**.

**Figura 19** - Componente Tabela em execução com alterações



10. Observe que foi criada uma barra de rolagem vertical, já que o número de linhas da tabela ultrapassa a altura anteriormente definida. Entretanto, repare que ao rolar a barra para baixo, os títulos das colunas mantêm-se visíveis.
11. Se quiser alterar a ordem das colunas, basta clicar no título da coluna desejada e arrastá-la para a esquerda ou direita.

12. Para preencher um campo da tabela, basta clicar na célula correspondente e digitar a informação necessária.

13. Utilize as teclas TAB e as SETAS para navegar na tabela.



**Vídeo 04** - jTable

## Atividade 03

---

1. Qual a propriedade fundamental que devemos utilizar para configurar um componente do tipo **Tabela**?
2. Que propriedade devemos utilizar para alterar a cor da grade de uma **Tabela** caso a mesma esteja ativada?
3. Que propriedade devemos utilizar para evitar que o usuário altere as informações contidas em uma tabela?
4. Qual a finalidade das propriedades **showHorizontalLines** e **showVerticalLines**?

## Conclusão

---

Chegamos ao fim de mais uma aula referente a componentes do Java. Já vimos os componentes mais comuns em diversas aplicações, mas ainda não terminamos. Ainda teremos mais uma aula, na qual exploraremos mais alguns componentes. Além de utilizar os componentes vistos nesta aula, procure ir adicionando outros vistos desde a primeira aula de componentes. Outra forma de exercício é procurar identificar os componentes em um aplicativo qualquer que você utilize para outras atividades como, por exemplo, em editores de texto, imagens ou navegadores. Assim, ao visualizar o componente, você pode verificar as diversas situações em que ele pode ser utilizado.

# Resumo

---

Nesta aula, você aprendeu sobre os componentes: **Campo de senha**, muito utilizado para restringir o acesso a determinados usuários; o componente **Texto formatado**, muito útil para definir o formato da informação que o usuário deve inserir, de acordo com o seu interesse; e o componente **Tabela**, que apresenta ao usuário as informações organizadas sob a estrutura de linhas e colunas, além de algumas propriedades para configurá-los de acordo com a necessidade da aplicação que você irá desenvolver.

## Autoavaliação

---

1. Crie uma aplicação utilizando um componente **Campo de senha**, de forma que tenha: cor de fundo amarelo, cor da fonte azul e o cursor seja verde.
2. Crie uma aplicação que contenha um **Campo formatado**, de forma que possa ser digitado um número de telefone no formato: (84) 9999-9999.
3. Na propriedade **model** de uma tabela, na guia **Configurações da tabela**, podemos de dado **Integer** (inteiro, por exemplo) e, durante a execução da aplicação, for digitada uma informação do tipo String (um nome, por exemplo)?
4. Crie uma aplicação utilizando um componente **Tabela** no seguinte formato:

<b>Produto</b>	<b>Vlr.Unitário</b>	<b>Unidade</b>	<b>Quant.</b>	<b>Total</b>
Papel	100	Resma	10	1000
Lápis	5	Dúzia	5	25
Borracha	7	Caixa	2	14

A **Tabela** deverá obedecer às seguintes condições:

- a. As colunas **Produto** e **Unidade** devem ser preenchidas durante o projeto e não poderão ser editadas pelo usuário quando a aplicação for executada.
- b. Deverá ser permitido ao usuário classificar o conteúdo da coluna **Produtos** durante a execução da aplicação.
- c. O tipo de dado das colunas **Vlr.Unitário**, **Quant.** e **Total** deverá ser **Float**. As outras colunas poderão ser **String**.

## Referências

---

HUBBARD, John R. **Programação com Java (Coleção Schaum)**. 2. ed. Porto Alegre: Artmed, [ 2004 ]. 328 p.

RODRIGUES Filho, Renato; Érica (ed). **Desenvolva Aplicativos com Java 6**. 1. ed. 384 p. ISBN 9788536502120.

THE JAVA TUTORIALS. **How to Use Password Fields**. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/passwordfield.html>>. Acesso em: 5 jul. 2012.

\_\_\_\_\_. **How to Use Formatted Text Fields**. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/formattedtextfield.html>>. Acesso em: 5 jul. 2012.

\_\_\_\_\_. **How to Use Tables**. Disponível em: <<http://download.oracle.com/javase/tutorial/uiswing/components/table.html>>. Acesso em: 5 jul. 2012.