**Como alterar o componente JTextField para aceitar apenas números**

**Veja neste artigo como alterar o componente JTextField para que ele passe a aceitar a digitação apenas de números, impedindo a entrada letras e caracteres especiais.**

 2

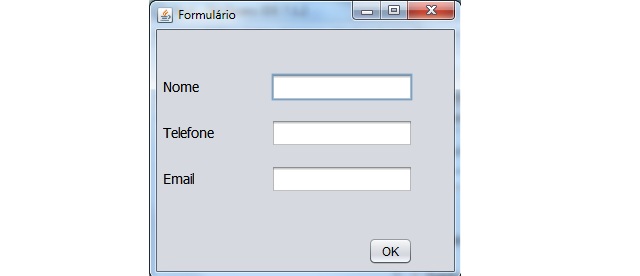
http://www.devmedia.com.br/imagens/portal2010/handUp.png

[Gostei (2)](http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152)

[http://www.devmedia.com.br/articles/img/handDown.png](http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152)

[(0)](http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152)

Neste artigo vamos alterar o componente JTextField para aceitar apenas números, funcionalidade bastante útil e necessária, por exemplo, quando um campo for utilizado para entrada de valores numéricos.



**Figura 1:** Exemplos de JTextFields

Para começar, é necessário adicionar um componente JTextField no formulário alterar seu nome, para que quando for necessário acionar as suas propriedades tenhamos uma maior facilidade.

Existe um componente que é o JFormattedTextField que é um campo de texto que é utilizado para definirmos a máscara do campo e assim determinarmos como as informações devem ser digitadas.

Não é o nosso objetivo trabalhar nesse post sobre o campo JFormattedTextField e sim operações que podemos aplicar sobre o JTextField para manipulá-lo de modo a exibir as informações com as quais estamos trabalhando, também de forma personalizada.

Vamos ao código para aceitar apenas números. De forma mais geral podemos apenas criar o evento Key Typed para o componente JTextField e acrescentar o seguinte código.

O evento Key Typed: pelo observado durante os testes em que foi desenvolvido esse código, é gerado cada vez que um novo caracter tiver de ser adicionado a string, diferente do key pressed e do key relesead, sendo o pressed acionado quando a tecla é pressionada e o key relesead quando a tecla é solta, ambos não gerando eventos enquanto a tecla permanece pressionada. No nosso caso causando erro durante a execução do algoritmo, por este motivo estamos trabalhando com o Key Typed.

**Listagem 1:** Evento para o JTextField aceitar apenas números

String caracteres="0987654321";

if(!caracteres.contains(evt.getKeyChar()+"")){

evt.consume();

}

Neste código na variável String estamos determinando quais caracteres devem ser aceitos. Neste caso estamos apenas aceitando números. E no if verificamos se o caracter que gerou o evento não está na variável caracteres, caso esteja acionamos a propriedade consume do evento.

Acrescentamos o ! que é a negação do teste, ou seja se o caractere que gerou o evento **NÃO** estiver na lista, esse caractere deve ser removido, no caso acionando **Consume**.

O método consume pode ser interpretado como: consome este evento de modo que não será processado na maneira padrão pela fonte que a originou. Traduzindo, este evento não deve ser executado.

Para aprimorarmos e conseguirmos reutilizar o código, podemos criar uma classe que herde as características de um JTextField normal, e definir essas propriedades. Além disso, podemos também definir o tamanho máximo que o campo deve suportar, dessa maneira também trabalhando a nivel de validação do tamanho da string que deve ser aceita em um JTextField, da mesma maneira que trabalhamos com campo para apenas texto.

**Listagem 2:** Classe JtextFieldSomenteNumeros

public final class JtextFieldSomenteNumeros extends JTextField {

private int maximoCaracteres=-1;// definição de -1

// como valor normal de um textfield sem limite de caracters

public JtextFieldSomenteNumeros() {

super();

addKeyListener(new java.awt.event.KeyAdapter() {

@Override

public void keyTyped(java.awt.event.KeyEvent evt) {

jTextFieldKeyTyped(evt);}});

}

public JtextFieldSomenteLetras(int maximo) {

super();

setMaximoCaracteres(maximo);// define o tamanho máximo

//que deve ser aceito no jtextfield que foi recebido no construtor

addKeyListener(new java.awt.event.KeyAdapter() {

@Override

public void keyTyped(java.awt.event.KeyEvent evt) {

jTextFieldKeyTyped(evt);}});

}

private void jTextFieldKeyTyped(KeyEvent evt) {

String caracteres="0987654321";// lista de caracters que não devem ser aceitos

if(!caracteres.contains(evt.getKeyChar()+"")){// se o caracter que gerou o evento estiver não estiver na lista

evt.consume();//aciona esse propriedade para eliminar a ação do evento

}

if((getText().length()>=getMaximoCaracteres())&&(getMaximoCaracteres()!=-1)){

//if para saber se precisa verificar também o tamanho da string do campo

// maior ou igual ao tamanho máximo, cancela e nao deixa inserir mais

evt.consume();

setText(getText().substring(0,getMaximoCaracteres()));

// esta linha acima é para remover os caracters inválidos caso o usuário tenha copiado o //conteúdo de algum lugar, ou seja, com tamanho maior que o definido

}//fim do if do tamanho da string do campo

}

public int getMaximoCaracteres() {

return maximoCaracteres;

}

public void setMaximoCaracteres(int maximoCaracteres) {

this.maximoCaracteres = maximoCaracteres;

}

}

Ao criar essa classe, optamos por ja definirmos no construtor a definição de duas opções a primeira sem parâmetro, que apenas aceita os números dentro do componente e a segunda que recebe como parâmetro o tamanho máximo da string do jtextField e verifica e deixa esse caracteres serem inseridos somente enquanto o tamanho for menor ao informado.

Para utilizar esta classe: após criar a classe, e colocar os imports necessários, temos duas opções que vamos explicar abaixo.

No NetBeans, basta inserir um JTextField normal, clicar com o botão direito sobre o componente e ir em personalizar código. Vai existir no lado esquerdo da janela, um combobox com duas opções, “código padrão” e “código personalizado”. Vamos trocar para código personalizado.

Após a definição de new vamos mudar a classe que está sendo instanciada:

**Listagem 3:** Código original do campo

jTextFieldRG = new javax.swing.JTextField();

Vamos Deixar assim:

**Listagem 4:** Código modificado do campo

jTextFieldRG = new JtextFieldSomenteNumeros();

Neste caso apenas acionamos a opção que vai aceitar apenas os números do teclado.

Ou podemos deixar assim:

**Listagem 5:** Código modificado do campo para militar o tamanho do texto

jTextFieldRG = new JtextFieldSomenteNumeros(50);

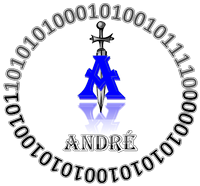
Neste caso acionamos a opção que vai aceitar apenas os números do teclado e também definimos o tamanho máximo do campo em 50 caracteres.

Também podemos instanciar normal caso não esteja utilizando o NetBeans. Basta ir no local que está instanciando a trocar o nome da classe para o JtextFieldSomenteNumeros.

Dessa maneira, basta repetir esse processo para todos os campos que pretende aceitar apenas a digitação de números e definir também o tamanho máximo do Campo. Facilitando assim o processo de validação dos campos do formulário.

No artigo ["JTextField aceitando apenas letras"](http://www.devmedia.com.br/jtextfield-aceitando-apenas-letras/26019) utilizamos o mesmo padrão de código, mas para permitir apenas letras no JTextField.

Até a próxima.

[](http://www.devmedia.com.br/autor/andre-camargo/272384)  
[André Camargo](http://www.devmedia.com.br/autor/andre-camargo/272384)

Analista de Tecnologia da Informaçao:  Bacharel em Ciência da Computação pela URI - Campus Santiago.

O que você achou deste post?

[http://www.devmedia.com.br/imagens/portal2010/handUp.png](http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152)

[Gostei (2)](http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152)

[http://www.devmedia.com.br/articles/img/handDown.png](http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152)

[(0)](http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152)

[Comentário | Tire sua dúvida](http://www.devmedia.com.br/rte3/addcomment.asp?idcomp=26152)

http://www.devmedia.com.br/articles/img/usuariosemfoto.png

Israel Faustino

Eu fiz e deu certo...   
Só que encontrei um problema:   
Como é que eu faço para apagar os dados do campo usando o backspace quando o JTextfield já estiver no tamanho máximo?

[há +1 ano] - [Responder](http://www.devmedia.com.br/rte3/addcomment.asp?idcomp=26152&idco_pai=34256)



Diogo Souza

Olá Israel,   
  
Faz o seguinte, tenta encaixar a ideia do artigo dele com a deste: http://www.dreamincode.net/forums/topic/191947-how-can-recognize-backspace-in-jtextfield/

[há +1 ano] - [Responder](http://www.devmedia.com.br/rte3/addcomment.asp?idcomp=26152&idco_pai=34308)

http://www.devmedia.com.br/articles/img/usuariosemfoto.png

Derci Santos

não seria mais interessante implementar um document para setar no JtextField?

[há +1 ano] - [Responder](http://www.devmedia.com.br/rte3/addcomment.asp?idcomp=26152&idco_pai=34592)



**[autor]** André Camargo

Pode ser sim, posta um artigo como fazer.

[há +1 ano] - [Responder](http://www.devmedia.com.br/rte3/addcomment.asp?idcomp=26152&idco_pai=34612)



Felipe Mendes Silva

Tbm tive o mesmo problema e o post do Diogo não me ajudou... alguem descobriu alguma coisa, favor me ajudem!

[há +1 ano] - [Responder](http://www.devmedia.com.br/rte3/addcomment.asp?idcomp=26152&idco_pai=41418)



Wesley Yamazack

Opa Felipe, obrigado pelo comentário.   
  
Enviamos a solicitação ao Diogo, para fazermos um artigo explicativo, tudo bem? Ele será postado em breve na home.   
  
Um abraço e até mais.

[há +1 ano] - [Responder](http://www.devmedia.com.br/rte3/addcomment.asp?idcomp=26152&idco_pai=41792)

Conhece a assinatura MVP?

Serviços

[Inclua um comentário](http://www.devmedia.com.br/rte3/addcomment.asp?idcomp=26152)

[Adicionar aos Favoritos](http://www.devmedia.com.br/favorite/addfavorite.asp?comp=26152)

[Marcar como lido/assistido](http://www.devmedia.com.br/favorite/add_finished.asp?comp=26152)

[Incluir anotação pessoal](http://www.devmedia.com.br/anotacoes/add/?idcomp=26152)

[Versão para impressão](http://www.devmedia.com.br/articles/viewcomp_forprint.asp?comp=26152)

[+Java](http://www.devmedia.com.br/java)

Mais posts

* Artigo

[Serializando objetos com byte em Java e PostgreSQL](http://www.devmedia.com.br/serializando-objetos-com-byte-em-java-e-postgresql/32089)

* Artigo

[T-SQL Subqueries: Serializando objetos em Java e PostgreSQL](http://www.devmedia.com.br/t-sql-subqueries-serializando-objetos-em-java-e-postgresql/32072)

* Pocket Video

[Integrando o DropBox com o Java](http://www.devmedia.com.br/integrando-o-dropbox-com-o-java/31961)

* Revista

[Revista easy Java Magazine 47](http://www.devmedia.com.br/revista-easy-java-magazine-47/32020)

* Artigo

[Programação paralela em Java](http://www.devmedia.com.br/programacao-paralela-em-java/32010)

* Artigo

[Criando aplicações Desktop em Java](http://www.devmedia.com.br/criando-aplicacoes-desktop-em-java/32009)

* Artigo

[Anti-padrões de Projeto: o que são, como identificar e evitar](http://www.devmedia.com.br/anti-padroes-de-projeto-o-que-sao-como-identificar-e-evitar/32008)

* Video aula

[Implementando o filtro - Curso de Java Web: Servlet, JSP, JSTL e Tags - Aula 100](http://www.devmedia.com.br/implementando-o-filtro-curso-de-java-web-servlet-jsp-jstl-e-tags-aula-100/31933)

* Video aula

[SQL de filtragem - Curso de Java Web: Servlet, JSP, JSTL e Tags - Aula 99](http://www.devmedia.com.br/sql-de-filtragem-curso-de-java-web-servlet-jsp-jstl-e-tags-aula-99/31932)

[Listar mais conteúdo](http://www.devmedia.com.br/java)

Leia mais em: [Como alterar o componente JTextField para aceitar apenas números](http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152#ixzz3Qh6jgrND) <http://www.devmedia.com.br/como-alterar-o-componente-jtextfield-para-aceitar-apenas-numeros/26152#ixzz3Qh6jgrND>