

# Java GUI

## AWT e Swing

## AWT e Swing

### Introdução :

1. Construção de Componentes simples com JOptionPane. Ex:

AWT -> Abstract Window Toolkit

Object < Component (awt) < Container (awt) < JComponent (swing)

O AWT utiliza o ambiente gráfico do Sistema Operacional.

O Swing tem um ambiente próprio do Java. Utiliza sempre a mesma aparência em qualquer Sistema Operacional. Look and Feel.

javax.swing package.

javax.awt package.

## AWT e Swing

### Mais sobre Swing...

Introduzido na versão 1.2

A maioria dos componentes Swing são puros e manipulados e exibidos pela JVM. Fazem parte da JFC Java Foundation Classes.

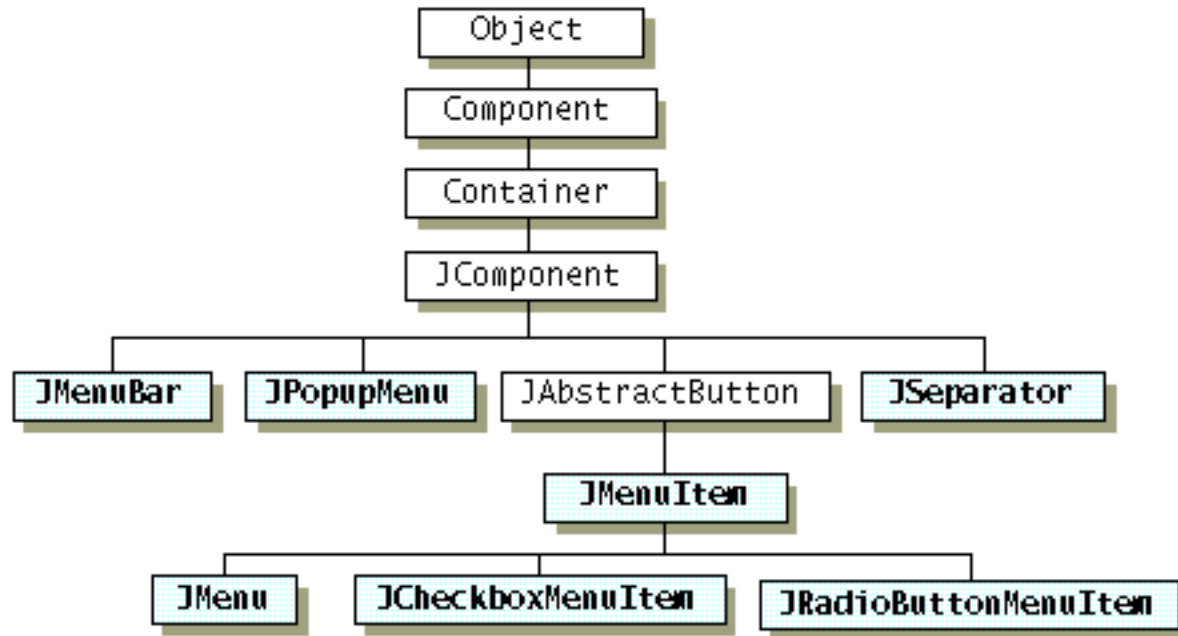
Ex: JLabel, JTextField, JButton, JCheckBox, JComboBox, JList, JPanel, JFrame...

Diferenças:

AWT: Componentes são anexados a Containers como Janelas. Todos os componentes do AWT são subclasses de `java.awt.Component`

Swing: É um substituto do AWT no que compete a substituir componentes heavyweight (pesados) por componentes lightweight. (leves) deste modo criando uma implementação com mesma aparência em diferentes sistemas operacionais. O Swing reaproveita alguns componentes raízes do AWT.

## Hierarquia de Classes AWT e Swing



**Component (awt)** -> Declara muitos comportamentos e atributos comuns dos componentes GUI.

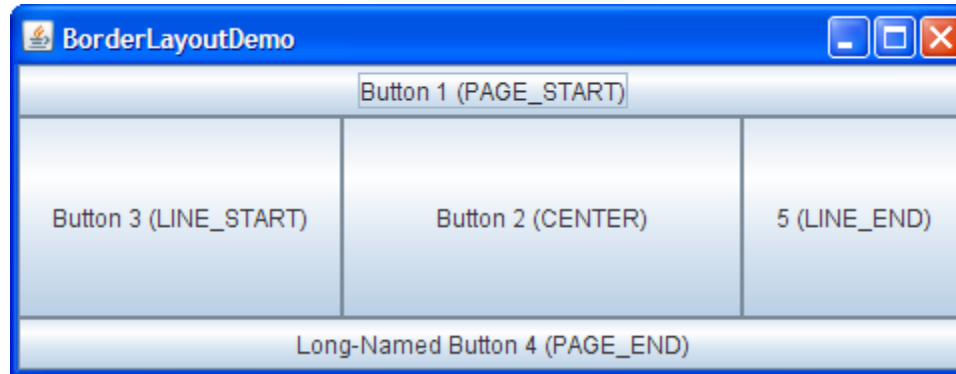
**Container (awt)** -> Componentes são anexados a Containers como Janelas. Os componentes são organizados em Layouts para então serem anexados aos containers.

**JComponent (swing)** -> subclasse de Container, É uma superclasse de todos componentes leves Swing

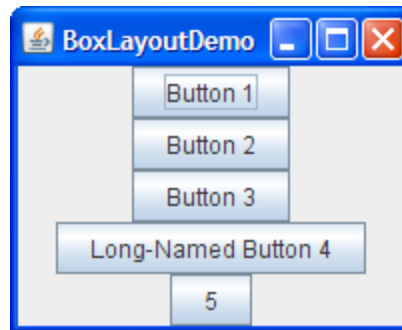
### Tipos de Layout

1. BorderLayout
2. BoxLayout
3. CardLayout
4. FlowLayout
5. GridBagLayout
6. GridLayout
7. GroupLayout
8. SpringLayout

## AWT e Swing

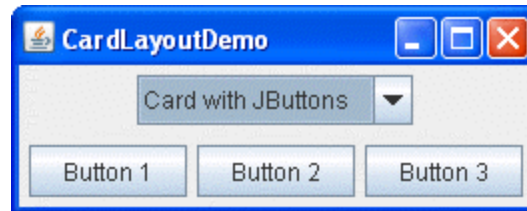


O BorderLayout é dividido em até cinco áreas: superior, inferior, esquerda, direita e centro. Todo o espaço extra é colocado na área central. Normalmente a lateral direita é usada para barra de rolagem

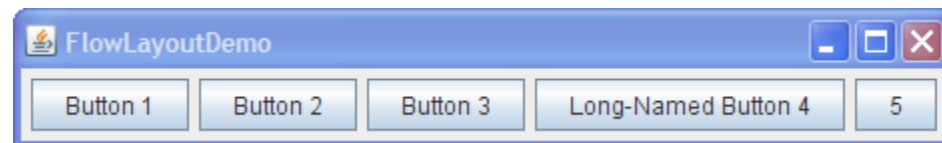


A classe BoxLayout coloca os componentes em uma única linha ou coluna. Respeita os tamanhos dos componentes máxima solicitada e também permite alinhar componentes.

## AWT e Swing

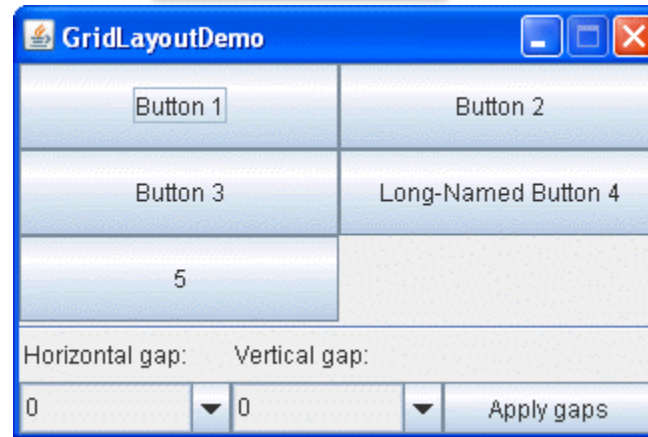


A classe `CardLayout` permite implementar uma área que contém componentes diferentes em momentos diferentes. É freqüentemente controlado por uma combobox, o estado desta combo pode determinar quais painéis (grupo de componentes) serão apresentados.

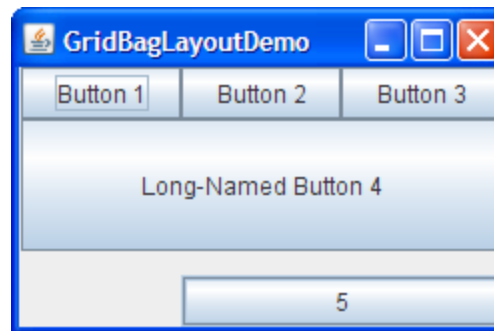


Estabelece os componentes em uma única linha, iniciando uma nova linha se o espaço não for suficiente.

## AWT e Swing



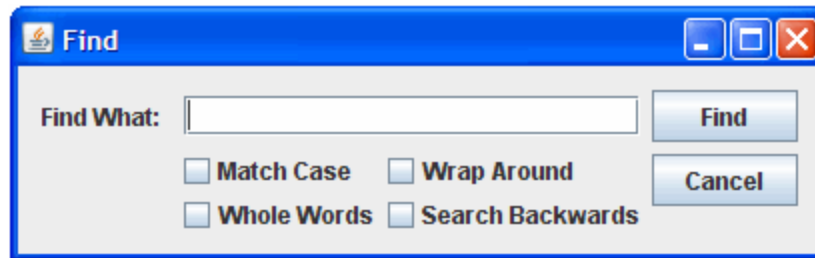
GridLayout faz um monte de componentes iguais em tamanho e os exibe no número solicitado de linhas e colunas.



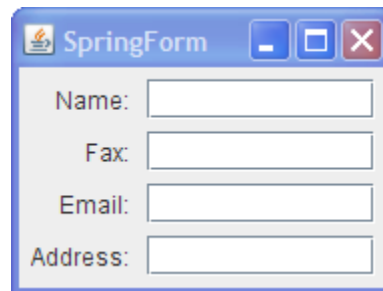
GridBagLayout é um sofisticado gerenciador de layout. Alinha os componentes, colocando-os dentro de uma grid (grade), permitindo que os componentes se estendam por mais de uma célula. As linhas da grade podem ter alturas diferentes, e as colunas diferentes larguras.



## AWT e Swing



GridLayout - trabalha com layout vertical e horizontal . Este layout é definido por dimensões independentes



SpringLayout é um layout flexível modelado para ser usado pelas GUI Builders. Permite relações precisas entre os componentes.

## Exemplos:

```
public class Painel extends JPanel {  
    public Painel() {  
        TitledBorder title =  
        BorderFactory.createTitledBorder("teste");  
        this.setBorder(title);  
    }  
}
```

```
public class FramePrincipal extends JFrame {  
    public FramePrincipal() {  
        getContentPane().setLayout(new BorderLayout());  
        getContentPane().add(new Painel(),  
        BorderLayout.NORTH);  
    }  
    public static void main(String[] args) {  
        new FramePrincipal().setVisible(true);  
    }  
}
```

## JPanel Vs JFrame

- JPanel
- JPanel estende Component, Container and JComponent.
- É útil para trabalhar com LayoutManagers como GridLayout, adicionando componentes em diferentes JPanels para posteriormente serem adicionados a um JFrame para criar uma GUI.
- É mais gerenciável em termos de Layout e reusabilidade
- É útil para pintar/desenhar no swing, sobrescrevendo paintComponent.
- Uma GUI Swing não pode existir sem um container de nível superior como Window, Frame, JFrame ou Applet, enquanto os outros possam existir sem um JPanel.

## JPanel Vs Frame

- **JFrame**
- Não estender o JFrame desnecessariamente. Somente se você precisar adicionar extras funcionalidades. (herança)
- JFrame estende Component e Container
- É um container de nível superior usado para representar os requisitos mínimos de uma janela, incluindo bordas, redimensionamento, barra de título, controles maximizar, minimizar e restaurar. Eventos WindowClose e windowOpened etc.

## JPanel Vs Frame

### **Conclusão:**

“JPanel é simples pode ser usado para quase tudo. A forma mais simples de trabalhar é agrupar os componentes em um JPanel e então hospedado dentro de um JFrame”

## JavaFX

A Sun Microsystems anunciou o JavaFX pela primeira vez na conferência JavaOne WorldWide Java Developer em Maio de 2007.

JavaFX é uma plataforma de software multimídia desenvolvida pela Oracle baseada em java para a criação e disponibilização de Aplicação Rica para Internet que pode ser executada em vários dispositivos diferentes.

A versão atual (JavaFX 2.1.0) permite a criação para desktop, browser e dispositivos móveis. TVs, video-games, Blu-rays players e há planos de adicionar novas plataformas no futuro. O suporte nos desktops e browsers é através da JRE e nos dispositivos móveis através do JavaME.

Para desenvolver em JavaFX existe uma IDE da Oracle chamada Scene Builder

Comparando...

## Swing vs JavaFX - Linha do Tempo



## JavaFX

JavaFX para dispositivos móveis.

- \* 02/2009: JavaFX Script 1.1 released com componentes Mobile.
- \* 08/2011: Suporte oficial para móveis foi cancelado.
- \* Java one 2012: somente expectativas
- \* Java one 2013: somente expectativas
- \* Devote 2013: removido do JavaFXPorts.



## JavaFX

### JavaFX para dispositivos móveis.

O Maven usa um conjunto de identificadores para identificar exclusivamente um projeto e especificar como o artefato do projeto deve ser empacotado:

**groupId:** um nome base exclusivo da empresa ou grupo que criou o projeto.

**artifactId:** um nome exclusivo do projeto.

groupId

- br.edu.cassio.farmacia

artifactId

- farmacia

version

- 0.0.1

## JavaFX

JavaFX para dispositivos móveis.

projeto GLUON HQ:

<http://gluonhq.com/products/mobile/vm/>

<http://gluonhq.com/products/mobile/>

openjdk:

<http://openjdk.java.net/projects/mobile/>

A ideia de ambos projetos eh ter uma JVM no ios e android. Com isso o JavaFX voltou a vida sendo uma boa alternativa para o desenvolvimento mobile.

JavaFx plugin eclipse ou netbeans

<https://openjfx.io/> se for java superior ao 8

<https://gluonhq.com/products/scene-builder/>

Novo projeto maven openjfx

# JavaFX pode ser rodado em navegadores web?

According to the [Java 9 release notes](#):

Java Applet and WebStart functionality, including the Applet API, The Java plug-in, the Java Applet Viewer, JNLP and Java Web Start including the javaws tool are all deprecated in JDK 9 and will be removed in a future release.

So, for Java 9+, deployment of JavaFX in a browser using a Java plug-in will only be possible using deprecated technology. Oracle notes in another part of the [Java 9 release notes](#):

Deprecates the Java Plug-in and associated applet technologies in Oracle's JDK 9 builds. While still available in JDK 9, these technologies will be considered for removal from the Oracle JDK and JRE in a future release. Applets and JavaFX applications embedded in a web page require the Java Plug-in to run. Consider rewriting these types of applications as Java Web Start or self-contained applications.

Note: Java Web Start isn't really an alternative as that is also deprecated (anyway, it's different from an application embedded and rendered within a browser page).

As an alternative which offers similar functionality, consider something like [jpro](#), which deploys JavaFX applications in a browser without a Java plugin (jpro is currently only in closed beta, so it is not a viable solution as of October 2017, but maybe someday in the future...).