

JAVA AWT

Carlos Arruda Baltazar
UNIP – Cidade Universitária

```
Microsoft (R) MS-DOS 8.0 (R) Startup Menu

1. Normal
2. Logged (\BOOTLOG.TXT)
3. Safe mode
4. Safe mode with network support
5. Step-by-step confirmation
6. Command prompt only
7. Safe mode command prompt only

Enter a choice: 1

F5=Safe mode  Shift+F5=Command prompt  Shift+F8=Step-by-step confirmation [N]
```

```
-rwxr-xr-x 1 sys      52850 Jun  8  1979 hptmunix
drwxrwxr-x 2 bin        320 Sep 22 05:33 lib
drwxrwxr-x 2 root       96 Sep 22 05:46 mdec
-rwxr-xr-x 1 root    50990 Jun  8  1979 rkunix
-rwxr-xr-x 1 root    51982 Jun  8  1979 rl2unix
-rwxr-xr-x 1 sys     51790 Jun  8  1979 rphtunix
-rwxr-xr-x 1 sys     51274 Jun  8  1979 rptmunix
drwxrwxrwx 2 root       48 Sep 22 05:50 tmp
drwxrwxr-x12 root      192 Sep 22 05:48 usr
# ls -l /usr
total 11
drwxrwxr-x 3 bin      128 Sep 22 05:45 dict
drwxrwxrwx 2 dmr       32 Sep 22 05:48 dmr
drwxrwxr-x 5 bin     416 Sep 22 05:46 games
drwxrwxr-x 3 sys     496 Sep 22 05:42 include
drwxrwxr-x10 bin     528 Sep 22 05:43 lib
drwxrwxr-x11 bin     176 Sep 22 05:45 man
drwxrwxr-x 3 bin     208 Sep 22 05:46 mdec
drwxrwxr-x 2 bin       80 Sep 22 05:46 pub
drwxrwxr-x 6 root       96 Sep 22 05:45 spool
drwxrwxr-x13 root     208 Sep 22 05:42 src
# ls -l /usr/dmr
total 0
#
```

[illegible]

```

__ SIAFI2010-CONTABIL-DEMONSTRA-CONRAZAO (CONSULTA RAZAO POR C. CONTABIL)____
28/09/10 09:52                                USUARIO : GLORIA
                                                PAGINA : 1

UG EMITENTE      : 250022 - NUCLEO ESTADUAL DO MS/MA
GESTAO EMITENTE  : 00001 - TESOURO NACIONAL
POSICAO          : SETEMBRO - ABERTO
CONTA CONTABIL   : 199720200 - CONTRATOS DE SERVICOS

CONTA CORRENTE                                SALDO EM R$
_ 01414525000136 - G M GARCEZ SILVA              7.031,24 D
    00105 - J. F. ROCHA SANTOS                    2.093,00 D
    00157 - RR - COMERCIO SERVICOS E REPRESENT     17.576,73 D
    00177 - MDAT SERVICOS E REPRESENTACOES LTD      8.559,48 D
    00184 - OSTENSIVA SEGURANCA PRIVADA LTDA        76.181,48 D
    00184 - COMPANHIA ENERGETICA DO MARANHAO        46.253,46 D
    00150 - COMPANHIA DE AGUAS E ESGOTOS DO MA     21.324,60 D
    00197 - S O S SERVICOS E COMERCIO LTDA - M      8.077,13 D
    00131 - AGUIAR LOCACAO E TURISMO LTDA          474.312,88 D
    00186 - ATLANTA TURISMO LTDA                   163.739,36 D
    00170 - D. G. CASTRO CUTRIM - ME.              1.448,00 D

CONTINUA ...

F2=RAZAO PF3=SAI PF7=RECUA PF8=AVANCA PF12=RETORNA

```

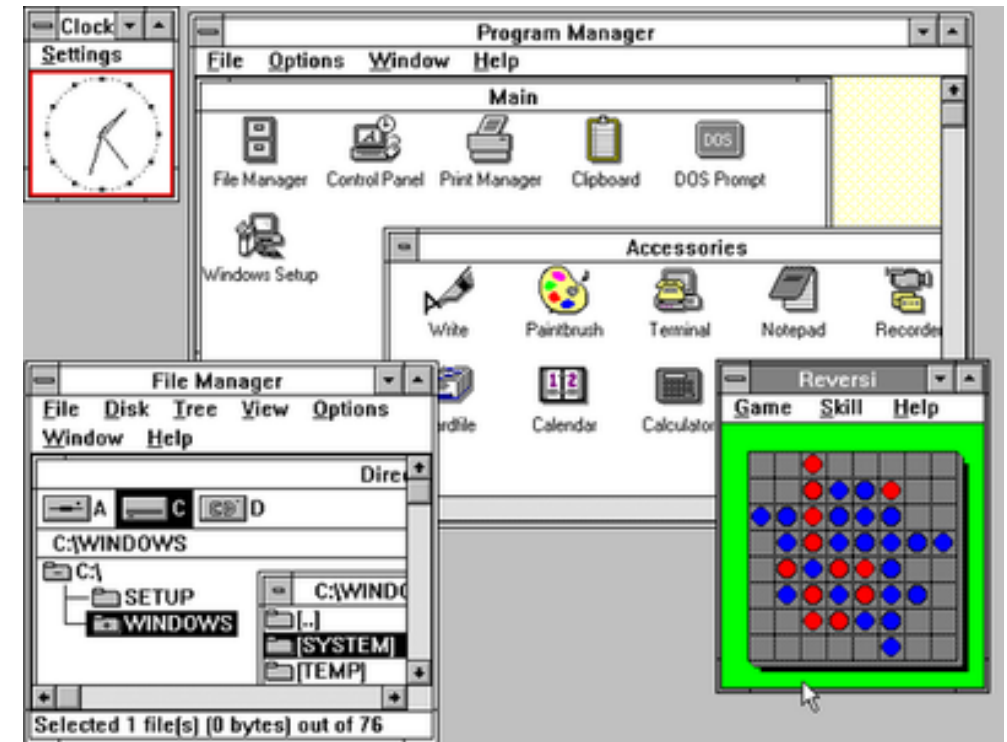
```

SENHA :
SISTEMA : _____
                                     BASE ...: BSB02
                                     DATA ...: 02/03/2016
                                     HORA ...: 10:43:40
                                     TERMINAL: AWMN1907

SEN0501 - INFORME A SUA SENHA.
PF1=AJUDA PF3=SAIDA PF6=TELA PF11=INFORMACOES GERAIS
                                     T01
MA + a                                     16/012

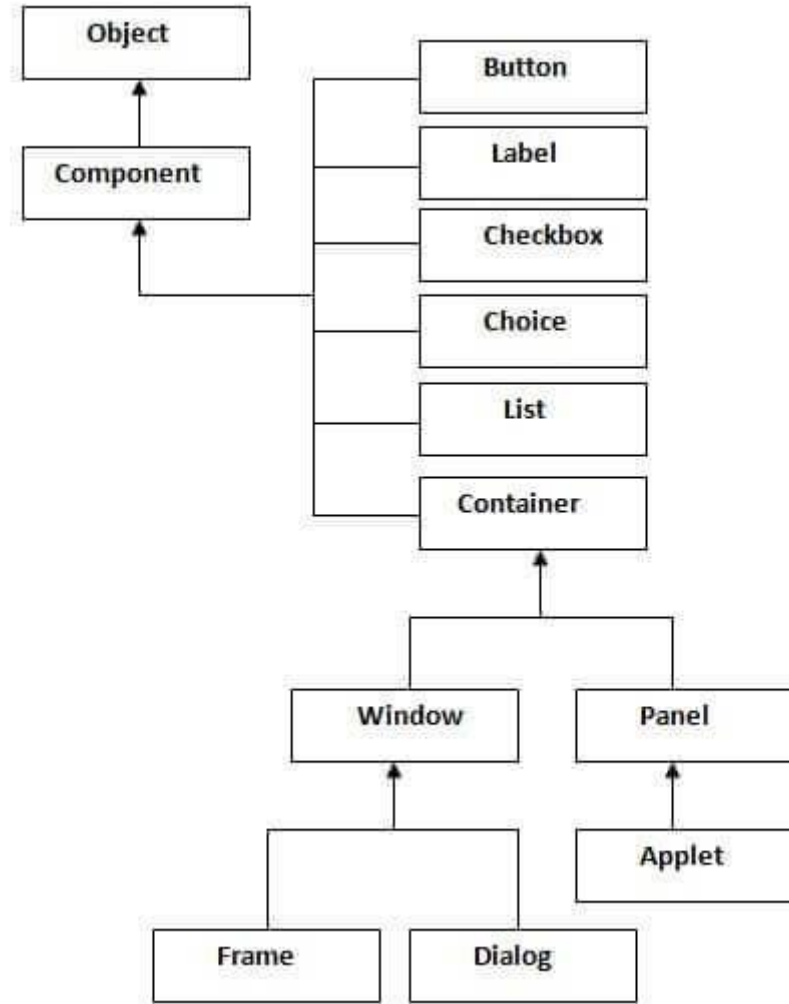
```

Criar uma aparência que “facilita” o uso e, deixa a aplicação com uma cara mais amigável



- Uma biblioteca GUI (Graphical User Interface), que é responsável por montar interface para o usuário, utilizando a plataforma nativa.
- Tendo em mente que as telas criadas pela biblioteca tem o intuito de ser aplicada em qualquer sistema cuja possui uma JVM.

AWT (Abstract Window Toolkit) é o *toolkit* gráfico original da linguagem de programação Java. Atualmente, *AWT* é parte da *JFC (Java Foundation Classes)*, a API padrão para uma interface gráfica do usuário (GUI) em um programa em Java. Um conjunto mais recente de interfaces da GUI, *Swing*, estende o *AWT* para que o programador possa criar objetos GUI generalizados independentes do sistema de janelas de um sistema operacional específico.



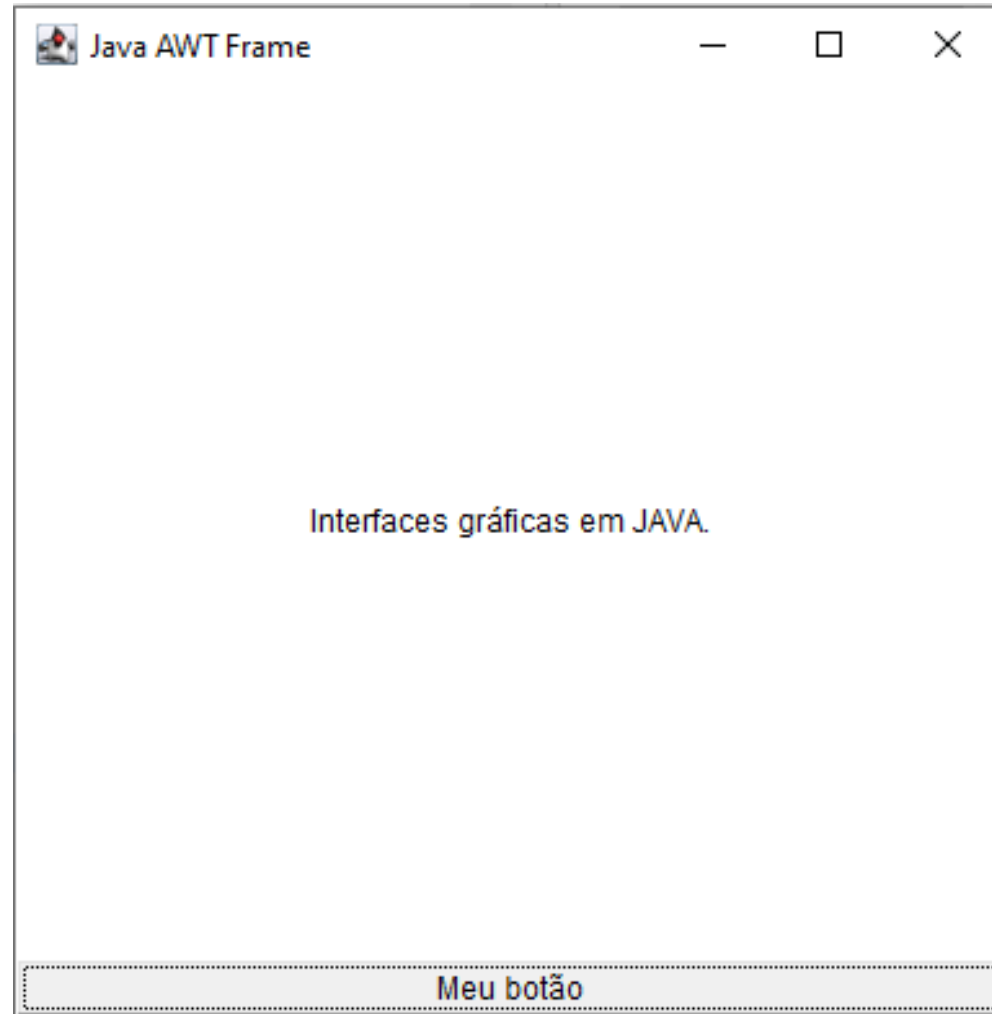
Container é um componente da classe *AWT* que pode conter outros componentes como: botões; campos de texto; etiquetas e etc. Qualquer classe que herde a classe *container* torna-se um *container*.

Um componente do tipo *Window* é um container que não tem borda nem barras de menu. Um componente *window* somente pode ser usado dentro de um frame, diálogo ou dentro de outro componente *window*.

Um componente *Panel* é um container que não tem uma barra de título e não tem barra de menu. Ele pode portar outros componentes como: botões; campos de texto e etc.

Um componente do tipo Frame é um container que possui barra de título e pode ter barras de menu. Este componente também pode conter outros componentes como: botões; campos de texto e etc.

```
1 package Pck_MeuPrimeiroFrame;
2
3 import java.awt.*;
4
5 public class MeuPrimeiroFrame
6 {
7     public static void main(String[] args)
8     {
9         Frame frm = new Frame("Java AWT Frame");
10        Label lbl = new Label("Interfaces gráficas em JAVA.",Label.CENTER);
11        Button b = new Button("Meu botão");
12        b.setBounds(30,100,80,30);
13        frm.add(lbl, BorderLayout.CENTER);
14        frm.add(b, BorderLayout.SOUTH);
15        frm.setSize(400,400);
16        frm.setVisible(true);
17    }
18 }
19
```



As aplicações em geral precisam de alguns eventos para o funcionamento completo do programa. Normalmente as GUIs do Java são baseadas em eventos, que são tarefas realizadas quando um usuário faz a interação com algum componente GUI, convertendo as ações do usuário em eventos, como por exemplo:

- Quando ocorre uma ação (*ActionEvent*);
- Clique do mouse (*MouseEvent*);
- Apertar em uma tecla (*KeyEvent*);
- Fechar uma janela (*WindowEvent*);

As tarefas de respostas realizadas em um evento são conhecidas como *handler* de evento e o processo total de responder a eventos é conhecido como tratamento de evento. Para cada tipo de evento precisa ser implementada uma interface de escuta.

Existem três elementos para o funcionamento e o tratamento de um evento, são eles:

- Origem do Evento: é o componente GUI com qual o usuário interage.

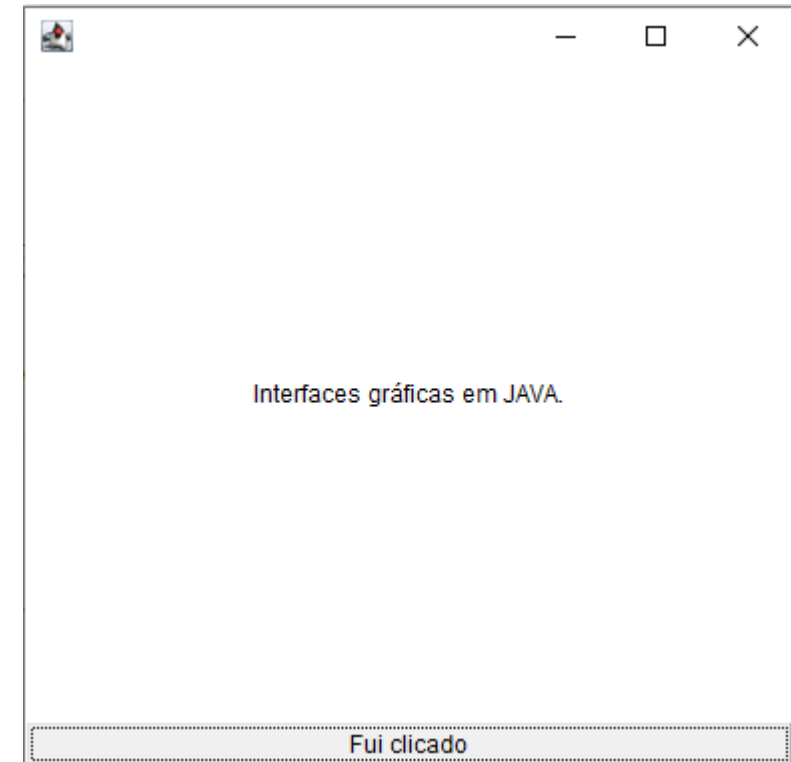
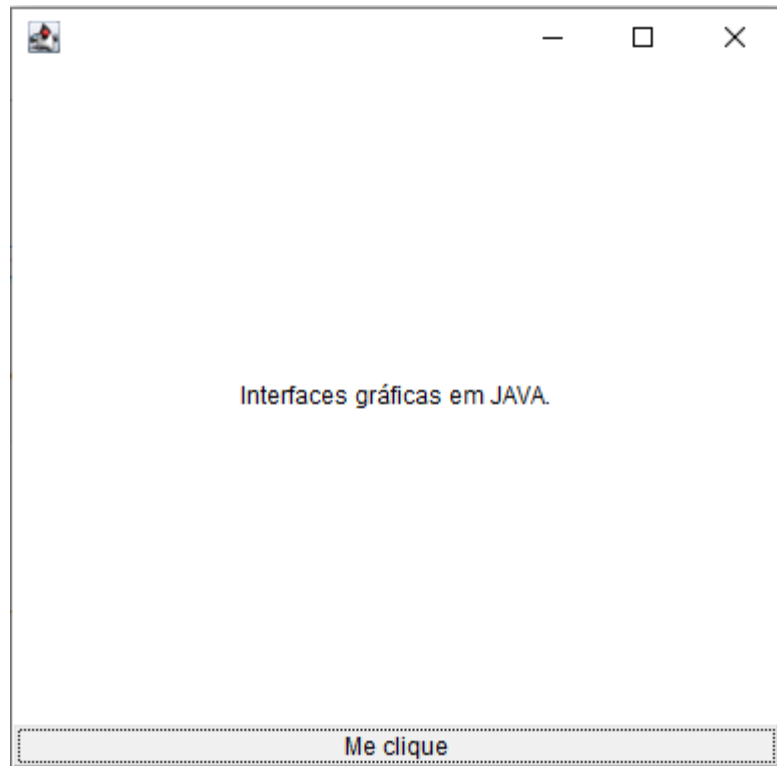
- Objeto do Evento: contém os dados do evento invocado, sendo como uma referência à origem do evento e quaisquer informações específicas do evento que podem ser exigidas pelo ouvinte para testar o evento.
- Ouvinte do Evento: é um objeto que é notificado pela origem de evento quando um evento ocorre.

Event Classes	Listener Interfaces
ActionEvent	ActionListener
MouseEvent	MouseListener and MouseMotionListener
MouseWheelEvent	MouseWheelListener
KeyEvent	KeyListener
ItemEvent	ItemListener
TextEvent	TextListener
AdjustmentEvent	AdjustmentListener
WindowEvent	WindowListener
ComponentEvent	ComponentListener
ContainerEvent	ContainerListener
FocusEvent	FocusListener

- Para registrar o componente no Listener, diversas classes proporcionam métodos de registro.
Por exemplo:
- Button
 - `public void addActionListener(ActionListener a){}`
- MenuItem
 - `public void addActionListener(ActionListener a){}`
- TextField
 - `public void addActionListener(ActionListener a){}`
 - `public void addTextListener(TextListener a){}`

- TextArea
 - `public void addTextListener(TextListener a){}`
- Checkbox
 - `public void addItemListener(ItemListener a){}`
- Choice
 - `public void addItemListener(ItemListener a){}`
- List
 - `public void addActionListener(ActionListener a){}`
 - `public void addItemListener(ItemListener a){}`

```
1 package Pck_AWT_Evento;
2
3 import java.awt.*;
4 import java.awt.event.*;
5
6 public class MeuPrimeiroFrameEvento extends Frame implements ActionListener
7 {
8     Label lbl;
9     Button bt;
10
11 public MeuPrimeiroFrameEvento()
12 {
13     lbl = new Label("Interfaces gráficas em JAVA.", Label.CENTER);
14     bt = new Button("Me clique");
15     bt.setBounds(30, 100, 80, 30);
16     bt.addActionListener(this);
17
18     this.add(lbl, BorderLayout.CENTER);
19     this.add(bt, BorderLayout.SOUTH);
20     this.setSize(400, 400);
21     this.setVisible(true);
22     this.addWindowListener(new WindowAdapter()
23     {
24         public void windowClosing(WindowEvent e)
25         {
26             System.exit(0);
27         }
28     });
29 }
30 public void actionPerformed(ActionEvent e)
31 {
32     bt.setLabel("Fui clicado");
33 }
34
35
36 public static void main (String args[])
37 {
38     MeuPrimeiroFrameEvento frame = new MeuPrimeiroFrameEvento();
39 }
40 }
```



OBRIGADO