Programação para Celular com Java – Tipos de Tela

Objetivo:

Capacitar o aluno a desenvolver aplicações para celular utilizando janelas dos tipos

- Alert
- List
- Item : ChoiceGroup, DateField e ImageItem

Sumário

Prof. Vida
vida@utfr.edu.br

7.3 A classe javax.microedition.lcdui.Alert

Alert é um tipo de tela com tempo de apresentação definido. Deve-se utilizar esse tipo de tela para avisos de forma geral.

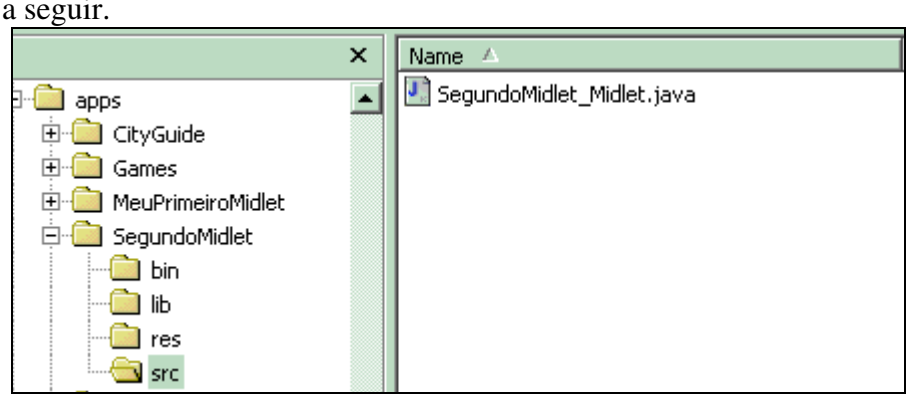
Tipos de Alert:	Uma tela do tipo Alert ainda pode ser definida como: • `Alarm` • `Confirmation` • `Error` • `Info` • `Warning`

Construtor:	O construtor é chamado através da passagem do parâmetro titulo da janela, por exemplo:
	`Alert(String titulo)`
	
	O construtor também pode ser chamado enviando as seguintes instruções:
	`Alert(String title`
	`, String alertText`
	`, Image alertImage`
	`, AlertType alertType)`

Tipos de Alerta:	A classe `javax.microedition.lcdui.AlertType` define os seguinte tipos de alerta: • `AlertType.`ALARM • `AlertType.`CONFIRMATION • `AlertType.`ERROR • `AlertType.`INFO • `AlertType.`WARNING O alerta ainda pode ser associado a um som através da instrução `playSound(Display display)`
Métodos Importantes:	• `void addCommand(Command c)` • `AlertType getType()` • `void removeCommand(Command c)` • `void setCommandListener(CommandListener l)` • `void setTimeout(int tempo)` • `void setType(AlertType t)` • `void setImage(Image i)`

Exemplo:

1. Inicie um novo projeto no WTK2.5.2. Como sugestão, recomenda-se o nome SegundoMidlet com a classe SegundoMidlet_Midlet. Essa recomendação é ilustrada na figura a seguir.



2. No arquivo Segundo_Midlet.java, forneça o código a seguir:

```
/////// arquivo SegundoMidlet_Midlet.java
import javax.microedition.midlet.MIDlet.*;
import javax.microedition.lcdui.*;

public class SegundoMidlet_Midlet
    extends javax.microedition.midlet.MIDlet
    implements CommandListener {

 private Display display;
 private Command exit = new Command("Exit",Command.EXIT,0);

 private TextBox inicio;
 private Alert[] alerta = new Alert[5];

 private void criaAlertas()
 {
      alerta[0] = new Alert("Alarme","Alerta do tipo Alarm", null,
AlertType.ALARM);
      alerta[1] = new Alert("Alarme","Alerta do tipo Confirmation", null,
AlertType.CONFIRMATION);
      alerta[2] = new Alert("Alarme","Alerta do tipo ERROR", null,
AlertType.ERROR);
      alerta[3] = new Alert("Alarme","Alerta do tipo INFO", null,
```

```
AlertType.INFO);
      alerta[4] = new Alert("Alarme","Alerta do tipo WARNING", null,
AlertType.WARNING);
      Command back = new Command("Back",Command.BACK,1);
      for (int i = 0; i < alerta.length; i++)
        alerta[i].addCommand(back);
 }

 public SegundoMidlet_Midlet() {  }

 public void startApp()
 {
   criaAlertas();
   inicio = new TextBox("Digite um numero entre 0 e 4:", "0", 2
                        , TextField.NUMERIC);
   inicio.addCommand(exit);
   inicio.addCommand(new Command("ShowMe",Command.OK,1));
   inicio.setCommandListener(this);

   display = Display.getDisplay(this);
   display.setCurrent(inicio);
  }
 public void pauseApp () {}

 public void destroyApp(boolean unconditional){}

 public void commandAction(Command c, Displayable s)
 { if (c.getCommandType() == Command.EXIT)
      notifyDestroyed();
   else
      if (c.getLabel() == "ShowMe")
       {
             int op = Integer.parseInt(inicio.getString().trim());
             if (op==0) display.setCurrent(alerta[0]);
             else if (op == 1) display.setCurrent(alerta[1]);
             else if (op == 2) display.setCurrent(alerta[2]);
             else if (op == 3) display.setCurrent(alerta[3]);
             else if (op == 4) display.setCurrent(alerta[4]);
       }
 }
}
```

7.4 A classe javax.microedition.lcdui.List

Alert é um tipo de tela que oferece uma lista de opções ao usuário.

Tipos:	Um objeto List pode ser de um dos seguintes tipos: • EXCLUSIVE : pode selecionar apenas um único elemento • MULTIPLE : pode selecionar vários elementos. • INPLICIT : associa um elemento a um CommandListener
Construtor:	O construtor pode ser chamado de duas formas: • `List(String titulo, int tipo)` • `List(String titulo, int tipo, String[] elementos, Image[] img)`
Métodos Importantes:	• `void append(String n, Image i)` • `void delete(int p)` • `int getSelectedIndex()` • `boolean isSelected(int p)` • `int size()`

Exemplo:

```
/////// arquivo SegundoMidlet_Midlet.java
import javax.microedition.midlet.MIDlet.*;
import javax.microedition.lcdui.*;

public class SegundoMidlet_Midlet
    extends javax.microedition.midlet.MIDlet
    implements CommandListener {

 private Display display;
 private Command exit = new Command("Exit",Command.EXIT,0);

 private TextBox inicio;
 private Alert[] alerta = new Alert[5];

 private List lista;
 private Command chamaLista = new
Command("Escolhas",Command.OK,1);

 private void criaLista()
 {
     String[] op = {"Laranja","Limão","Uva", "Maracujá", "Mamão"};
```

```
      lista = new List("Selecione a sua opção"
                       ,List.MULTIPLE,op,null);
      lista.addCommand(new Command("Back",Command.BACK,1));
      lista.setCommandListener(this);
 }

 private void criaAlertas()
 {
      alerta[0] = new Alert("Alarme","Alerta do tipo Alarm", null,
AlertType.ALARM);
      alerta[1] = new Alert("Alarme","Alerta do tipo Confirmation",
null, AlertType.CONFIRMATION);
      alerta[2] = new Alert("Alarme","Alerta do tipo ERROR", null,
AlertType.ERROR);
      alerta[3] = new Alert("Alarme","Alerta do tipo INFO", null,
AlertType.INFO);
      alerta[4] = new Alert("Alarme","Alerta do tipo WARNING",
null, AlertType.WARNING);
      Command back = new Command("Back",Command.BACK,1);
      for (int i = 0; i < alerta.length; i++)
        alerta[i].addCommand(back);
 }

 public SegundoMidlet_Midlet() { }

 public void startApp()
 {
   criaAlertas();
   criaLista();
   inicio = new TextBox("Digite um numero entre 0 e 4:", "0", 2
                       , TextField.NUMERIC);
   inicio.addCommand(exit);
   inicio.addCommand(new Command("ShowMe",Command.OK,1));
   inicio.addCommand(chamaLista);
   inicio.setCommandListener(this);



   display = Display.getDisplay(this);
   display.setCurrent(inicio);
```

```
  }
 public void pauseApp () {}

 public void destroyApp(boolean unconditional){}

 public void commandAction(Command c, Displayable s)
 { if (c.getCommandType() == Command.EXIT)
      notifyDestroyed();
   else
      if (c.getLabel() == "ShowMe")
       {
            int op = Integer.parseInt(inicio.getString().trim());
            if ((op>=0) && (op<5)) display.setCurrent(alerta[op]);
       }
       else if (c.getCommandType() == Command.BACK)
             {display.setCurrent(inicio);}
       else
       { display.setCurrent(lista);   }
 }
}
```

7.5. A classe javax.microedition.lcdui.Item

O que é?	A superclasse Item representa os elementos que podem ser adicionados em um formulário.
Classes Componentes de Item:	• ChoiceGroup • TextField • DateField • Gauge • ImageItem • Spacer • StringItem • CustomItem
Definição de LayOut	O alinhamento de componentes na tela pode seguir as seguintes instruções para a definição de LayOut: • `LAYOUT_DEFAULT` • `LAYOUT_LEFT` • `LAYOUT_RIGHT` • `LAYOUT_CENTER` • `LAYOUT_TOP` • `LAYOUT_BOTTOM` • `LAYOUT_VCENTER` • `LAYOUT_NEWLINE_BEFORE` • `LAYOUT_NEWLINE_AFTER` • `LAYOUT_SHRINK` • `LAYOUT_VSHRINK` • `LAYOUT_EXPAND` • `LAYOUT_VEXPAND` • `LAYOUT_2`

Exemplo:	Para a definição do Layout de um componente basta chamar o método setLayout. Por exemplo :
	`TextField t = new TextField(`
	`"Digite um numero entre 0 e 4:"`
	`, "0"`
	`, 2`
	`, TextField.NUMERIC);`
	`t.setLayout(Item.LAYOUT_RIGHT );`

7.5.1 ChoiceGroup	Define um grupo de botões. Exemplo:
	`private ChoiceGroup GrupoBotao(`
	`String[] a )`

```
{    ChoiceGroup saida;
     saida = new ChoiceGroup(
               "Escolha uma opção"
              ,ChoiceGroup.EXCLUSIVE
              , a
              , null);
     saida.setLayout(Item.LAYOUT_LEFT);
     return saida;
}
```

7.5.2 DateField

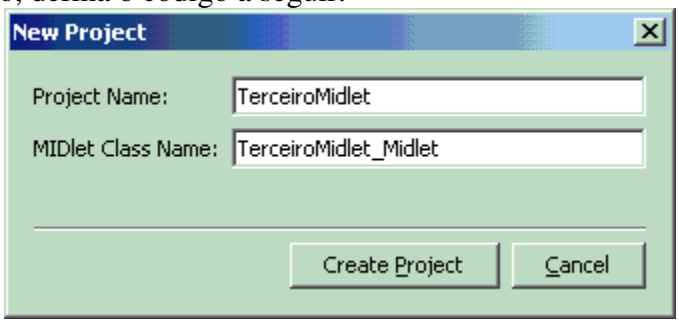
Componente utilizado para mostrar data e hora.

Exemplo:

```
java.util.Calendar folha =
     java.util.Calendar.getInstance();
folha.setTime(new java.util.Date());

DateField quando = new DateField(
                   "Calendário"
                   ,DateField.DATE_TIME);
quando.setDate(folha.getTime());
quando.setLayout(Item.LAYOUT_CENTER);
```

Exemplo Completo:

Em um novo projeto, defina o código a seguir.




```
/////// arquivo TerceiroMidlet_Midlet.java
import javax.microedition.midlet.MIDlet.*;
import javax.microedition.lcdui.*;

public class TerceiroMidlet_Midlet
    extends javax.microedition.midlet.MIDlet
```

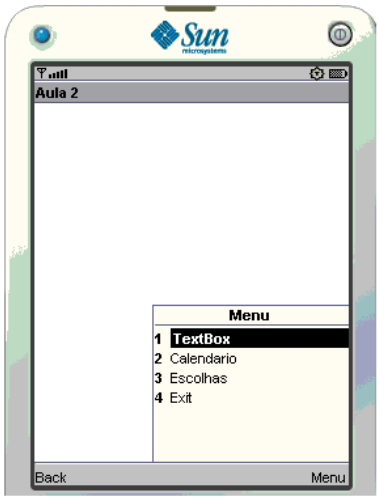
```
    implements CommandListener {

 private Display display;
 private Command exit = new Command("Exit",Command.EXIT,0);
 private Command back = new Command("Back",Command.BACK,1);
 private Command chamaLista = new
Command("Escolhas",Command.OK,1);

 private Alert[] alerta = new Alert[5];
 private Form janela = new Form("Aula 2");
 private TextBox inicio = CriaTextBoxForm();

 private List criaListaForm()
 {
     List lista;
     String[] op = {"Laranja","Limão","Uva", "Maracujá", "Mamão"};
     lista = new List("Selecione a sua opção",
List.MULTIPLE,op,null);
     lista.addCommand(back);
     lista.setCommandListener(this);
     return lista;
 }

 private void criaAlertas()
 {
     alerta[0] = new Alert("Alarme","Alerta do tipo Alarm", null
                            , AlertType.ALARM);
     alerta[1] = new Alert("Alarme","Alerta do tipo Confirmation"
                       , null, AlertType.CONFIRMATION);
     alerta[2] = new Alert("Alarme","Alerta do tipo ERROR", null
                            , AlertType.ERROR);
     alerta[3] = new Alert("Alarme","Alerta do tipo INFO", null
                          , AlertType.INFO);
     alerta[4] = new Alert("Alarme","Alerta do tipo WARNING"
                           , null, AlertType.WARNING);

     for (int i = 0; i < alerta.length; i++)
       alerta[i].addCommand(back);
 }
```

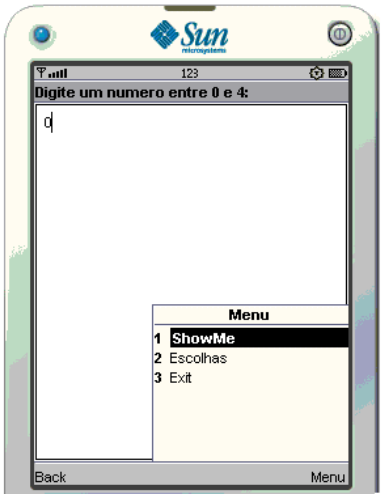
```
 public TerceiroMidlet_Midlet() { }

 private Form componenteItem()
 {
   Form saida = new Form("inicio 0");
   TextField t = new TextField("Digite um numero entre 0 e 4:",
"0", 2, TextField.NUMERIC);
    t.setLayout(Item.LAYOUT_RIGHT );

   saida.append(t);

   String ops[] = {"clarinete","saxofone","piano","guitarra"};
   ChoiceGroup aux = new ChoiceGroup("Escolha uma opção"
                          , ChoiceGroup.EXCLUSIVE, ops, null);
   aux.setLayout(Item.LAYOUT_LEFT);
   saida.append(aux);
   saida.addCommand(back);
   saida.setCommandListener(this);
   return saida;
 }

 private Form FormCalendario()
 {
     java.util.Calendar folha = java.util.Calendar.getInstance();
     folha.setTime(new java.util.Date());

     DateField quando = new DateField("Calendário"
                            ,DateField.DATE_TIME);
     quando.setDate(folha.getTime());
     quando.setLayout(Item.LAYOUT_CENTER);

     Form saida = new Form("Calendário");
     saida.append(quando);
     saida.addCommand(back);
     saida.setCommandListener(this);
     return saida;
 }
```

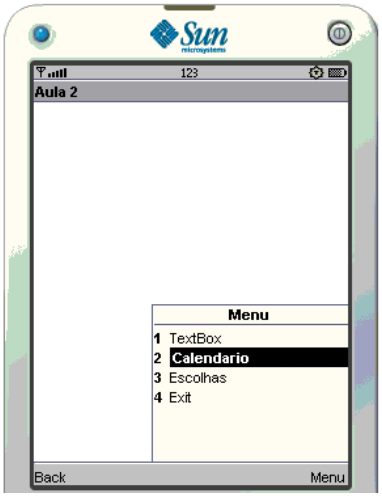
```
 private TextBox CriaTextBoxForm()
 {
   TextBox textbox;
   textbox = new TextBox("Digite um numero entre 0 e 4:", "0"
                         , 2, TextField.NUMERIC);
   textbox.addCommand(exit);
   textbox.addCommand(new Command("ShowMe",Command.OK,1));
   textbox.addCommand(chamaLista);
   textbox.addCommand(back);
   textbox.setCommandListener(this);
   return textbox;
 }

 public void startApp()
 {
   criaAlertas();

   display = Display.getDisplay(this);
   janela.addCommand(back);
   janela.addCommand(exit);
   janela.addCommand(new Command("TextBox",Command.OK,1));
   janela.addCommand(new Command("Calendario", Command.OK,2));
   janela.addCommand(new Command("Escolhas", Command.OK,3));

   janela.setCommandListener(this);
   display.setCurrent(janela);
  }
 public void pauseApp () {}

 public void destroyApp(boolean unconditional){}

 public void commandAction(Command c, Displayable s)
 { if (c.getCommandType() == Command.EXIT)
      notifyDestroyed();

   if (c.getCommandType() ==
Command.BACK){display.setCurrent(janela);}

   if (c.getLabel() == "TextBox") {display.setCurrent(inicio);}
```

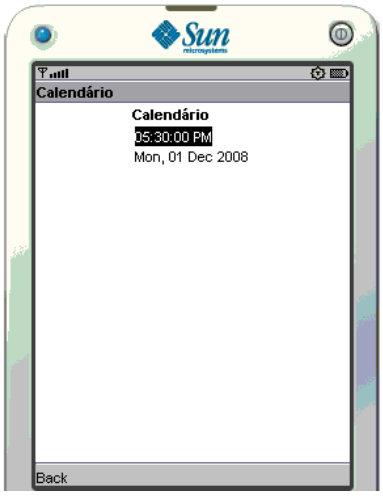
```
   if (c.getLabel() == "ShowMe")
        {
             int op = Integer.parseInt(inicio.getString().trim());
             if ((op>=0) && (op<5)) display.setCurrent(alerta[op]);
        }
   if (c.getLabel() == "Escolhas")
                  {display.setCurrent(componenteItem());}
   if (c.getLabel() == "Calendario")
                  {display.setCurrent(FormCalendario());}
 }
}
```

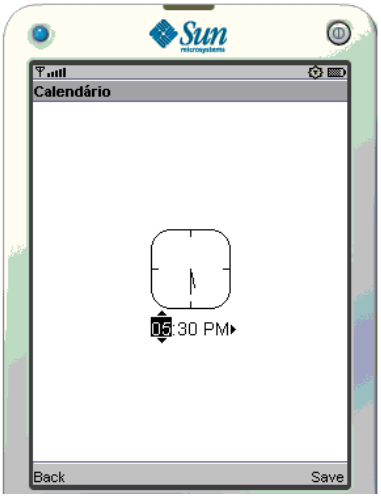
Ilustrações da Execução do programa:



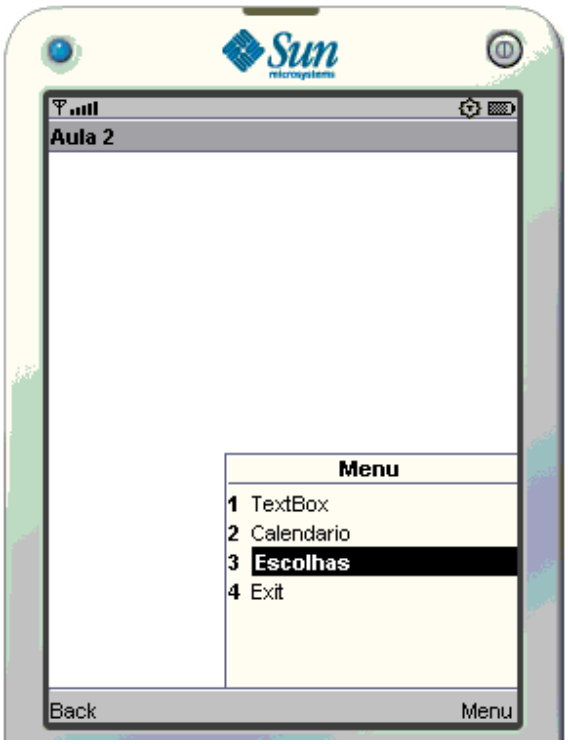


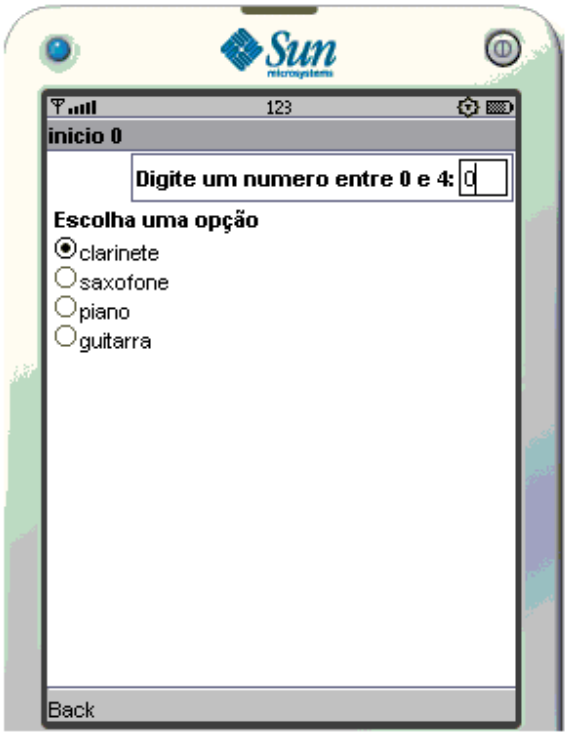






Prof. Vida
vida@utfr.edu.br



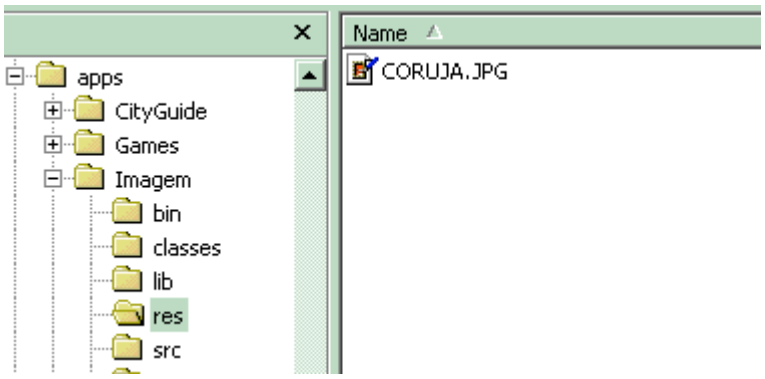


Prof. Vida
vida@utfr.edu.br

7.5.3 ImageItem

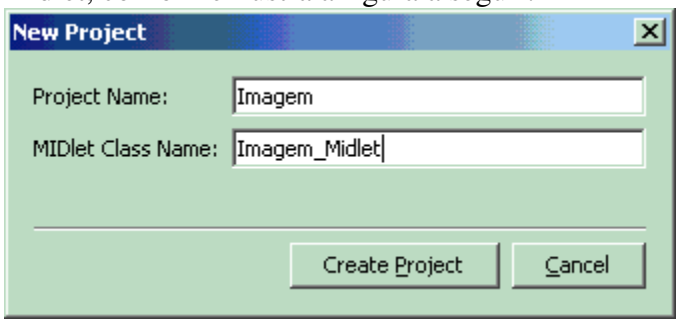
O que é?	O `ImageItem` contém uma referência a uma image ou objeto `Image`. Ele permite mostrar imagem na tela do aplicativo.

Chamada do Construtor:	O Construtor pode ser chamado conforme a seguir:
	`new ImageItem(String rotulo,`
	`Image figura,`
	`int layout,`
	`String textoAlternativo);`
	Exemplo:
	`new ImageItem("Texto"`
	`, Image.createImage("coruja.jpg")`
	`, ImageItem.LAYOUT_CENTER`
	`, null)`

Valores possíveis do layout:	• Item.LAYOUT_CENTER • Item.LAYOUT_LEFT • Item.LAYOUT_RIGHT • Item.LAYOUT_DEFAULT • Item.LAYOUT_NEWLINE_AFTER • Item.LAYOUT_NEWLINE_BEFORE
Aonde guardar a imagen?	As imagens devem ser armazenadas no diretório /res da aplicação, conforme ilustra a figura a seguir. 

Exemplo de aplicação que permite carregar imagens.

1. Inicie um novo Midlet, conforme ilustra a figura a seguir.



2. Desenvolva o código a seguir.

```
//////////////////////////////// arquivo Imagem_Midlet.java
import javax.microedition.midlet.MIDlet.*;
import javax.microedition.lcdui.*;

public class Imagem_Midlet
       extends javax.microedition.midlet.MIDlet
       implements CommandListener {

 private Display display;
 private Command exitCommand = new Command("Exit"
                             , Command.EXIT, 0);

 public Imagem_Midlet() { }

 public void carregaImagem(String i, Form f)
 {
     try
     {
          Image figura = Image.createImage(i);
          f.append(new ImageItem("Imagem Carregada:"
                                 , figura
                                 , ImageItem.LAYOUT_CENTER
                                 , null));
     } catch (java.io.IOException e)
         {f.append("Erro ao carregar a imagem:"+i+"\n"+e);}
 }
```

```
 public void startApp()
 {  display = Display.getDisplay(this);
    Form mainForm = new Form("Mostra Imagem");
    carregaImagem("/coruja.jpg",mainForm);
    mainForm.addCommand(exitCommand);
    mainForm.setCommandListener(this);
    display.setCurrent(mainForm);
  }
 public void pauseApp () {}

 public void destroyApp(boolean unconditional){}

 public void commandAction(Command c, Displayable s)
 { if (c.getCommandType() == Command.EXIT)
      notifyDestroyed();
 }
}
```

Prof. Vida
vida@utfr.edu.br

Exemplo de Execução:

Prof. Vida
vida@utfr.edu.br