

## Prática de Programação J2ME (6)

Especialização em Desenvolvimento Web com Interfaces Ricas

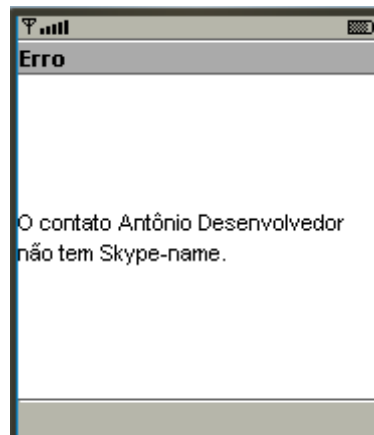
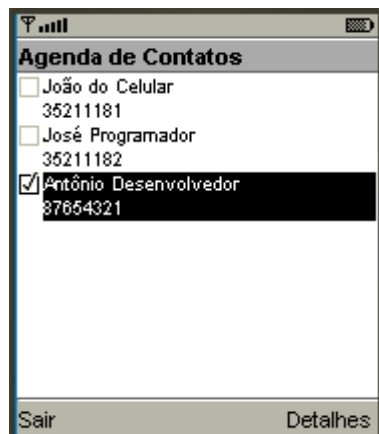
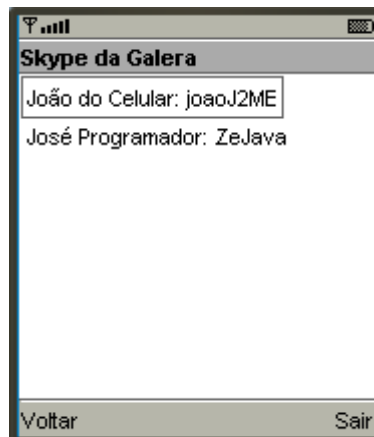
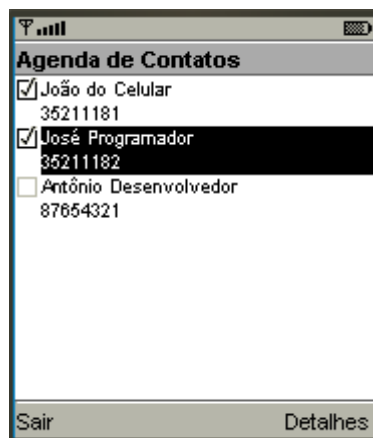
[www.especializacao.info](http://www.especializacao.info)

Crie a aplicação correspondente aos itens seguintes. Execute o programa e observe os resultados.

1. Crie uma nova agenda de contatos, à partir da Prática de Programação 5, porém com uma lista de 3 pessoas, do tipo MULTIPLE, que permite a escolha de mais de um item;
2. Ao clicar em “Detalhes”, abre-se um novo Formulário, que ao invés de conter todos os detalhes, mostra somente o Skype-name de cada pessoa selecionada, como no modelo de telas abaixo;
3. Ao se clicar em “Voltar” na lista de Skype-names, mostra-se novamente a lista de contatos. Ao se clicar em “Sair”, fecha a aplicação;
4. Caso algum usuário não tenha Skype-name, deve-se mostrar um alerta de ERRO, que deve durar 4 segundos;
5. Para ilustrar o erro, mude o Skype-name do último contato para *null*, selecione-o, clique em “Detalhes” e observe o resultado.

## Resposta da Prática de Programação J2ME (6)

TELA(S)



## CÓDIGO EM JAVA

---

Pessoa.java

---

```
public class Pessoa {

    String nome, telefone, email, nomeSkype;

    public Pessoa(String nome, String telefone, String email,
                  String nomeSkype) {
        setNome(nome);
        setTelefone(telefone);
        setEmail(email);
        setNomeSkype(nomeSkype);
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getTelefone() {
        return telefone;
    }

    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getNomeSkype() {
        return nomeSkype;
    }

    public void setNomeSkype(String nomeSkype) {
        this.nomeSkype = nomeSkype;
    }
}
```

## Prática de Programação J2ME

---

### SkypesMidlet.java

---

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class SkypesMidlet extends MIDlet {

    private Display display;

    public void startApp() {
        display = Display.getDisplay(this);
        Vector pessoas = new Vector();
        pessoas.addElement( new Pessoa("João do Celular",
            "35211181", "joao@email.com", "joaoJ2ME"));
        pessoas.addElement( new Pessoa("José Programador",
            "35211182", "jose@email.com", "ZeJava"));
        pessoas.addElement( new Pessoa("Antônio Desenvolvedor",
            "87654321", "antonio@email.com", null));
        PessoaList lista = new PessoaList(this, pessoas);
        display.setCurrent(lista);
    }

    public void pauseApp() { }

    public void destroyApp(boolean unconditional) {
        notifyDestroyed();
    }

    public void mudarTelaPara(Displayable d){
        display.setCurrent(d);
    }
}
```

---

### SkypeForm.java

---

```
import java.util.Vector;
import javax.microedition.lcdui.*;

public class SkypeForm extends Form implements CommandListener{

    private Command sair, voltar;
    private SkypesMidlet controlador;
    private PessoaList telaAnterior;

    public SkypeForm(Vector selecionados, PessoaList uiLista){
        super("Skype da Galera");
        this.telaAnterior = uiLista;
    }
}
```

## Prática de Programação J2ME

```
this.controlador = uiLista.getControlador();
sair = new Command("Sair", Command.EXIT, 0);
voltar = new Command("Voltar", Command.BACK, 1);
addCommand(sair);
addCommand(voltar);
for(int i = 0; i < selecionados.size(); i++){
    Pessoa pessoa = (Pessoa) selecionados.elementAt(i);
    append(pessoa.getNome() + ": " +
           pessoa.getNomeSkype());
}
setCommandListener(this);
}
public void commandAction(Command arg0, Displayable arg1) {
    if(arg0 == sair)
        controlador.destroyApp(true);
    else
        if(arg0 == voltar)
            controlador.mudarTelaPara(telaAnterior);
}
}
```

---

### PessoaList.java

---

```
import java.util.Vector;
import javax.microedition.lcdui.*;

public class PessoaList extends List implements CommandListener,
Runnable{

    private SkypesMidlet controlador;
    private Vector pessoas;
    private Command info, exit;

    public PessoaList(SkypesMidlet controlador, Vector pessoas) {
        super("Agenda de Contatos", List.MULTIPLE);
        this.controlador = controlador;
        this.pessoas = pessoas;
        setFitPolicy(List.TEXT_WRAP_ON);
        for(int i=0; i < pessoas.size();i++){
            Pessoa pessoa = (Pessoa) pessoas.elementAt(i);
            append(pessoa.getNome() + "\n" + pessoa.getTelefone(),
                  null);
        }
        Font fonte = Font.getFont(Font.FACE_SYSTEM,
                                   Font.STYLE_PLAIN,
                                   Font.SIZE_SMALL);
        for(int i = 0; i < size(); i++)
            setFont(i, fonte);
        exit = new Command("Sair", Command.EXIT, 0);
    }
}
```

## Prática de Programação J2ME

```
        addCommand(exit);
        info = new Command("Detalhes", Command.OK, 0);
        addCommand(info);
        setCommandListener(this);
    }

    public void commandAction(Command arg0, Displayable arg1) {
        if(arg0 == exit)
            controlador.destroyApp(true);
        else{
            if(arg0 == info)
                new Thread(this).start();
        }
    }

    public void run() {
        boolean[] selecionados = new boolean[3];
        getSelectedFlags(selecionados);
        Vector escolhidos = new Vector();
        for(int i = 0; i < selecionados.length; i++){
            if(selecionados.elementAt(i)){
                if(pessoas.elementAt(i).getNomeSkype() == null){
                    Alert alerta = new Alert("Erro", "O contato "
                                                + pessoas.elementAt(i).getNome()
                                                + " não tem Skype-name.",
                                                null, AlertType.ERROR);
                    controlador.mudarTelaPara(alerta);
                    return;
                }
                escolhidos.addElement(pessoas.elementAt(i));
            }
        }
        SkypeForm f = new SkypeForm(escolhidos, this);
        controlador.mudarTelaPara(f);
    }

    public SkypesMidlet getControlador() {
        return controlador;
    }
}
```