

## Prática de Programação J2ME (5)

Especialização em Desenvolvimento Web com Interfaces Ricas

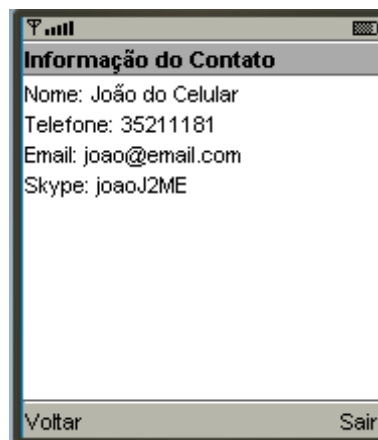
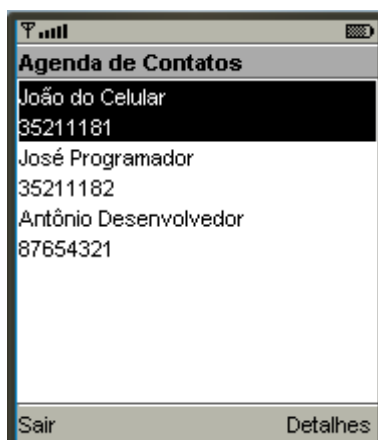
[www.especializacao.info](http://www.especializacao.info)

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

1. Crie a classe *Pessoa*, que armazena alguns atributos correspondentes: Nome completo, Telefone, Email e Skype-nickname ;
2. Crie um MIDlet *PeopleListMidlet*, que em sua tela inicial mostre uma lista de três contatos;
3. A lista de contatos a ser disponibilizada deverá ser um *List* do tipo *IMPLICIT*, que armazena um vetor de *Pessoas*, em que cada elemento é representado por nome e telefone de cada uma, linha à linha, em fonte de tamanho médio.
4. Na lista, ao clicar em “Sair”, o programa é finalizado. Ao clicar em “Detalhes”, inicia-se uma nova Thread, que abre um Formulário contendo todos os detalhes da pessoa;
5. No formulário *ContatoForm*, deve-se colocar os detalhes da pessoa e dois botões, de saída da aplicação e de “Recuo para tela anterior”.

## Resposta da Prática de Programação J2ME (5)

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

---

### PeopleListMidlet.java

---

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

public class PeopleListMidlet extends MIDlet {

    private Display display;

    public void startApp() {
        display = Display.getDisplay(this);
        Pessoa[] pessoas = new Pessoa[3];
        pessoas[0] = new Pessoa("João do Celular", "35211181",
                                "joao@email.com", "joaoJ2ME");
        pessoas[1] = new Pessoa("José Programador", "35211182",
                                "jose@email.com", "ZeJava");
        pessoas[2] = new Pessoa("Antônio Desenvolvedor",
                                "87654321", "antonio@email.com",
                                "TonhoDev");
        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);
    }
}
```

---

### PessoaList.java

---

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

public class PessoaList extends List implements CommandListener,
                                                Runnable{

    private PeopleListMidlet controlador;
    private Pessoa[] pessoas;
    private Command info, exit;
```

## Prática de Programação J2ME

```
public PessoaList(PeopleListMidlet controlador, Pessoa[]
                    pessoas) {
    super("Agenda de Contatos", List.IMPLICIT);
    this.controlador = controlador;
    this.pessoas = pessoas;
    setFitPolicy(List.TEXT_WRAP_ON);
    for(int i=0; i < pessoas.length;i++){
        append(pessoas[i].getNome() + "\n" +
                pessoas[i].getTelefone(), null);
    }
    Font fonte = Font.getFont(Font.FACE_SYSTEM,
                               Font.STYLE_PLAIN,
                               Font.SIZE_MEDIUM);
    for(int i = 0; i < size(); i++)
        setFont(i, fonte);
    exit = new Command("Sair", Command.EXIT, 0);
    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() {
    Pessoa selecionado = pessoas[selectedIndex()];
    Form f = new ContatoForm(selecionado, this);
    controlador.mudarTelaPara(f);
}

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

---

### ContatoForm.java

---

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

public class ContatoForm extends Form implements CommandListener{

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

    public ContatoForm(Pessoa selecionado, PessoaList uiLista){
        super("Informação do Contato");
        this.telaAnterior = uiLista;
        this.controlador = uiLista.getControlador();
        sair = new Command("Sair", Command.EXIT, 0);
        voltar = new Command("Voltar", Command.BACK, 1);
        addCommand(sair);
        addCommand(voltar);
        append("Nome: " + selecionado.getNome() + "\n" +
            "Telefone: " + selecionado.getTelefone() + "\n" +
            "Email: " + selecionado.getEmail() + "\n" +
            "Skype: " + selecionado.getNomeSkype());
        setCommandListener(this);
    }

    public void commandAction(Command arg0, Displayable arg1) {
        if(arg0 == sair)
            controlador.destroyApp(true);
        else
            if(arg0 == voltar)
                controlador.mudarTelaPara(telaAnterior);
    }
}
```