

## Prática de Programação J2ME (11)

Especialização em Desenvolvimento Web com Interfaces Ricas

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

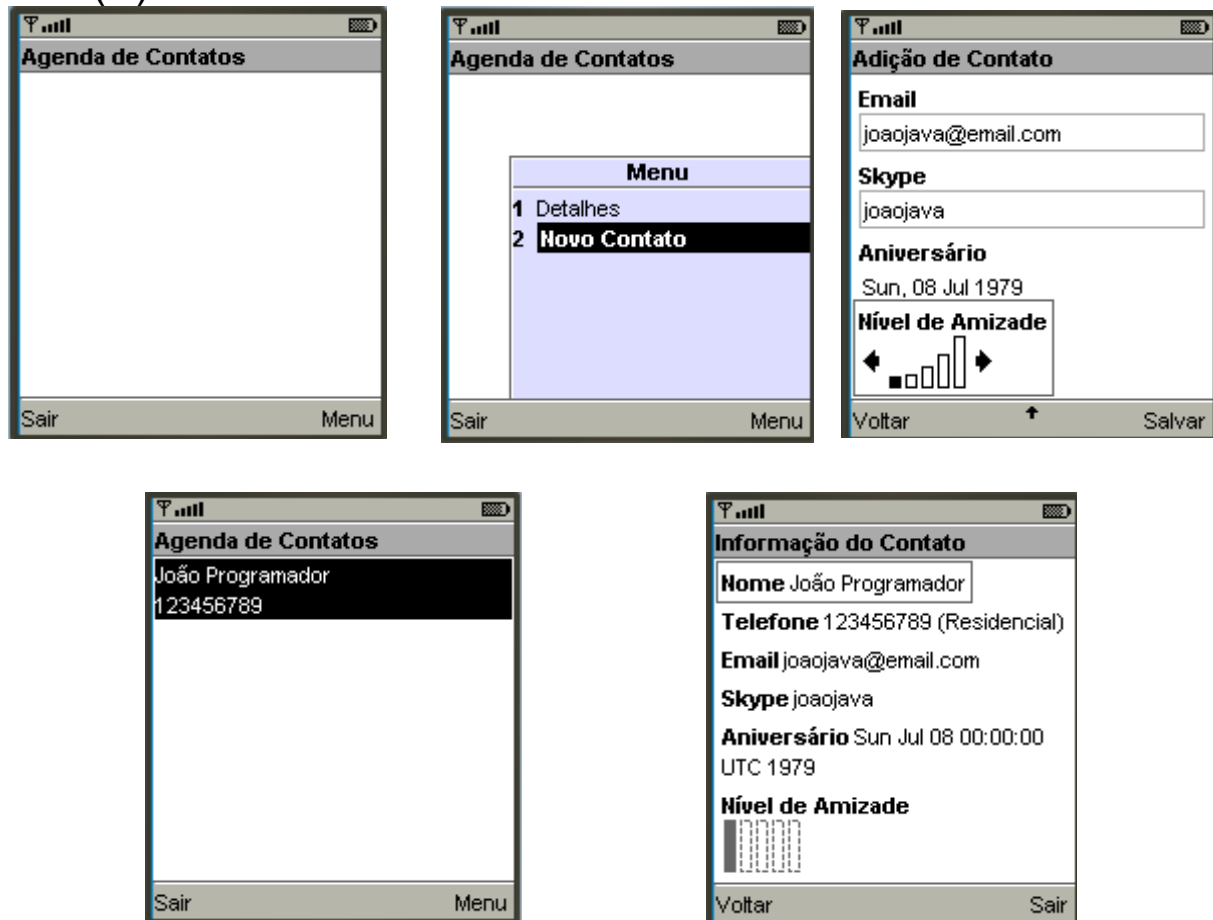
Crie a aplicação correspondente aos itens seguintes. Execute o programa e observe os resultados.

---

1. Crie um MIDlet simples, chamado *PeopleListMidlet.java*, como as práticas anteriores, que ao ser executado, mostra uma lista de contatos pessoais;
2. Os atributos de um contato devem seguir o padrão da Práticas de Programação 10, porém com a adição do atributo (int) *nivelAmizade*, que guarda o quão amigo a pessoa é;
3. A lista de contatos pessoais *PessoaList.java* a ser mostrada pelo MIDlet deve ser inicialmente vazia (sem dados), do tipo IMPLICIT, fonte média e plana, com título “Agenda de Contatos”, que permite um item se configurar em mais de uma linha e que possui três comandos: Sair, Detalhes e Novo Contato;
4. A lista de pessoas cadastradas deve ser guardada em um vetor (*Vector - java.util*) em *PessoaList.java*;
5. O comando “Sair” deve estar na posição BACK e fechar a aplicação ao ser clicado;
6. O comando “Detalhes” deve estar na posição “OK”, prioridade 0 e, ao selecionado, mostra-se os detalhes do contato em um novo formulário (como na Prática de Programação 10). Deve ser adicionado um *Gauge* não editável para o nível de amizade da pessoa;
7. O comando “Novo contato” deve estar na posição “OK”, prioridade 1 e, a selecionado, mostra um formulário para entrada de dados de uma nova pessoa a ser cadastrada;
8. O formulário *NovoContatoForm.java* deve conter quatro *TextFields*, sendo eles:
  - Nome – aceita até 40 caracteres quaisquer;
  - Telefone – aceita até 10 caracteres numéricos;
  - Email – aceita até 30 caracteres do tipo EMAILADDR;
  - Skype – aceita até 40 caracteres quaisquer.
9. O formulário *NovoContatoForm.java* deve conter um *ChoiceGroup* do tipo POPUP que ofereça duas opções para escolha: Residencial e Comercial.
10. O formulário *NovoContatoForm.java* deve conter a opção de escolha de data de nascimento da pessoa, que abre uma nova janela para a escolha do dia;
11. O formulário *NovoContatoForm.java* deve conter a opção de escolha de nível de amizade da pessoa, através de um *Gauge*, de mínimo 0 e máximo 5 unidades;
12. Tal formulário de cadastro de pessoas deve conter dois comandos: Voltar e Salvar.
13. Ao se selecionar o comando “Voltar”, a tela do aparelho deve retornar à lista de contatos cadastrados;
14. Ao se selecionar “Salvar”, cria-se uma nova Pessoa, que é armazenada no vetor de contatos de *PessoaList.java*;
15. Após adição da pessoa no vetor de cadastrados, a tela da lista é atualizada e mostrada ao usuário. Novas pessoas podem ser cadastradas.

## Resposta da Prática de Programação J2ME (11)

### TELA(S)



### CÓDIGO EM JAVA

#### PeopleListMidlet.java

```
package PP11;

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

public class PeopleListMidlet extends MIDlet {

    Display display;

    public void startApp() {
        display = Display.getDisplay(this);
        PessoaList lista = new PessoaList(this);
        display.setCurrent(lista);
    }
}
```

## Prática de Programação J2ME

```
public void pauseApp() {}

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

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

---

### Pessoa.java

---

```
package PP11;

import java.util.Date;

public class Pessoa {

    String nome, telefone, email, nomeSkype, tipoTel;
    Date diaAniversario;
    int nivelAmizade;

    public Pessoa(String nome, String telefone, String email, String
        nomeSkype,String tipoTel, Date diaAniversario,
        int nivelAmizade) {
        setNome(nome);
        setTelefone(telefone);
        setEmail(email);
        setNomeSkype(nomeSkype);
        setTipoTel(tipoTel);
        setDiaAniversario(diaAniversario);
        setNivelAmizade(nivelAmizade);
    }

    public int getNivelAmizade() {
        return nivelAmizade;
    }

    public void setNivelAmizade(int nivelAmizade) {
        this.nivelAmizade = nivelAmizade;
    }

    public Date getDiaAniversario() {
        return diaAniversario;
    }

    public void setDiaAniversario(Date diaAniversario) {
        this.diaAniversario = diaAniversario;
    }
}
```

## Prática de Programação J2ME

```
public String getTipoTel() {
    return tipoTel;
}

public void setTipoTel(String tipoTel) {
    this.tipoTel = tipoTel;
}

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;
    }
}
```

---

### PessoaList.java

---

```
package PP11;

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

public class PessoaList extends List implements CommandListener, Runnable
{
    private PeopleListMidlet controlador;
    private Vector pessoas;
```

## Prática de Programação J2ME

```
private Command info, exit, add;

public PessoaList(PeopleListMidlet controlador) {
    super("Agenda de Contatos", List.IMPLICIT);
    this.controlador = controlador;
    this.pessoas = new Vector();
    setFitPolicy(List.TEXT_WRAP_ON);
    carregaLista();
    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);
    info = new Command("Detalhes", Command.OK, 0);
    add = new Command("Novo Contato", Command.OK, 1);

    addCommand(exit);
    addCommand(add);
    addCommand(info);
    setCommandListener(this);
}

public void commandAction(Command arg0, Displayable arg1) {
    if (arg0 == exit) {
        controlador.destroyApp(true);
    } else {
        if (arg0 == info) {
            if (size() > 0) {
                new Thread(this).start();
            }
        } else {
            if (arg0 == add) {
                NovoContatoForm novo = new NovoContatoForm(this);
                controlador.mudarTelaPara(novo);
            }
        }
    }
}

public void run() {
    try{
        Pessoa selecionado = (Pessoa)
            pessoas.elementAt(getSelectedIndex());
        Form f = new ExibeContatoForm(selecionado, this);
        controlador.mudarTelaPara(f);
    }catch (Exception e ){
        e.printStackTrace();
    }
}

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

## Prática de Programação J2ME

```
public void carregaLista() {
    deleteAll();
    for (int i = 0; i < pessoas.size(); i++) {
        Pessoa pessoa = (Pessoa) pessoas.elementAt(i);
        append(pessoa.getNome() + "\n" + pessoa.getTelefone(), null);
    }
}

public Vector getPessoas() {
    return pessoas;
}
}
```

---

### NovoContatoForm.java

---

```
package PP11;

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

public class NovoContatoForm extends Form implements CommandListener,
Runnable{

    TextField nome, email, telefone, skype;
    DateField diaNiver;
    ChoiceGroup tipoTel;
    Gauge contAmizade;
    Command salvar, voltar;
    PeopleListMidlet controlador;
    PessoaList telaAnterior;

    public NovoContatoForm(PessoaList anterior) {
        super("Adição de Contato");
        this.telaAnterior = anterior;
        controlador = anterior.getControlador();

        nome = new TextField("Nome", "", 40, TextField.ANY);
        email = new TextField("Email", "", 30, TextField.EMAILADDR);
        telefone = new TextField("Telefone", "", 10, TextField.DECIMAL);
        skype = new TextField("Skype", "", 40, TextField.ANY);
        String tipos[] = {"Residencial", "Comercial"};
        tipoTel = new ChoiceGroup("Tipo de Telefone", ChoiceGroup.POPUP,
            tipos, null);
        diaNiver = new DateField("Aniversário", DateField.DATE);
        contAmizade = new Gauge("Nível de Amizade", true, 5, 0);
        salvar = new Command("Salvar", Command.OK, 0);
        voltar = new Command("Voltar", Command.BACK, 0);
        addCommand(salvar);
        addCommand(voltar);
        append(nome);
        append(tipoTel);
        append(telefone);
    }
}
```

## Prática de Programação J2ME

```
        append(email);
        append(skype);
        append(diaNiver);
        append(contAmizade);
        this.setCommandListener(this);
    }

    public void commandAction(Command arg0, Displayable arg1) {
        if(arg0 == voltar){
            controlador.mudarTelaPara(telaAnterior);
        }else{
            new Thread(this).start();
        }
    }

    public void run() {
        String n = nome.getString();
        String t = telefone.getString();
        String e = email.getString();
        String s = skype.getString();
        String tt = tipoTel.getString(tipoTel.getSelectedIndex());
        Date dA = diaNiver.getDate();
        int a = contAmizade.getValue();
        Pessoa pessoa = new Pessoa(n, t, e, s, tt, dA, a);
        Vector pessoas = telaAnterior.getPessoas();
        pessoas.addElement(pessoa);
        telaAnterior.carregaLista();
        controlador.mudarTelaPara(telaAnterior);
    }
}
```

---

### ExibeContatoForm.java

---

```
package PP11;

import javax.microedition.lcdui.*;

public class ExibeContatoForm extends Form implements CommandListener{

    private Command sair, voltar;
    private PeopleListMidlet controlador;
    private PessoaList telaAnterior;
    private StringItem nome, tel, email, skype, diaNiver;
    private Gauge nivelAmizade;

    public ExibeContatoForm(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);
        nome = new StringItem("Nome", selecionado.getNome());
    }
}
```

## Prática de Programação J2ME

```
tel = new StringItem("Telefone", selecionado.getTelefone() + " (" +
                    + selecionado.getTipoTel() + ")");
email = new StringItem("Email", selecionado.getEmail());
skype = new StringItem("Skype", selecionado.getNomeSkype());
diaNiver = new StringItem("Aniversário",
                          selecionado.getDiaAniversario() + "");
nivelAmizade = new Gauge("Nível de Amizade", false, 5,
                        selecionado.getNivelAmizade());
append(nome);
append(tel);
append(email);
append(skype);
append(diaNiver);
append(nivelAmizade);
addCommand(sair);
addCommand(voltar);
setCommandListener(this);
}
public void commandAction(Command arg0, Displayable arg1) {
    if(arg0 == sair)
        controlador.destroyApp(true);
    else
        if(arg0 == voltar)
            controlador.mudarTelaPara(telaAnterior);
}
}
```