

Universidade Federal de Santa Catarina
Campus Universitário de Florianópolis
Curso de Sistemas de Informação
Desenvolvimento de sistemas orientados a objetos
Rosvelter João Coelho da Costa

PROJETO AGENDA TELEFÔNICA JAVA

Elisa Rosá
Géssica Lehmkul
Rafael Silva

Florianópolis
Julho de 2016

1 INTRODUÇÃO

O projeto final da disciplina de desenvolvimento de sistemas orientados a objetos será uma agenda telefônica, totalmente feita em linguagem *JAVA*, na qual terá as seguintes funcionalidades:

- Cadastro de dados: O usuário poderá criar um usuário e adicionar dados, como telefone e e-mail;
- Remoção de dados: O usuário poderá remover os dados de usuários que já foram anteriormente cadastrados;
- Edição de dados: Poderão ser atribuídos pequenos novos detalhes a um cadastro existente;
- Exportação para formato CSV: Será possível salvar os contatos que foram já cadastrados em um arquivo de extensão *.csv* (comma separated value).

1.1 Objetivos

- Implementar um sistema a partir dos conhecimentos adquiridos em aula;
- Torná-lo funcional e adicionar melhorias por meio do que já foi criado;
- Fazer com que o usuário, por mais inexperiente que seja, consiga fazer uso de tal maneira que ele se sinta a vontade.

2 METODOLOGIA

2.1 Requisitos do Sistema

Foi implementado um sistema com os principais requisitos:

- O usuário poderá adicionar apenas um telefone por contato
- Enquanto não mover o arquivo contatos.csv do local os contatos serão todos armazenados no mesmo

2.2 Implementação

Criou-se uma interface mostrando todas as funcionalidades básicas (Figura 2) obtida por meio do código (Figura 1) a seguir:

```
private static final String exportarCSV = "exportarCSV";

public static void main(String[] args) {

    Agenda agenda = new Agenda();

    JFrame janela = new JFrame("Agenda");
    JPanel meuPainel = new JPanel();
    janela.setContentPane(meuPainel);

    ActionListener listener = new Gestor(agenda);
    mk_Tela(meuPainel, listener);

    janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    janela.setSize(600, 400);
    janela.setVisible(true);

}

static void mk_Tela(JPanel q, ActionListener listener) {

    JButton x;

    x = new JButton("Deletar contato");
    q.add(x);
    x.addActionListener(listener);
    x.setActionCommand("DeletarContato");

    x = new JButton("Novo contato");
    q.add(x);
    x.addActionListener(listener);
    x.setActionCommand("AdicionarContato");

    x = new JButton("Atualizar contato");
    q.add(x);
    x.addActionListener(listener);
    x.setActionCommand("acharContato");

    x = new JButton("Exportar CSV");
    q.add(x);
    x.addActionListener(listener);
    x.setActionCommand("exportarCSV");
}
```

FIGURA 1 - Código Interface



FIGURA 2 - Interface Agenda telefônica.

Ao clicar em **Novo Contato** abrirá uma nova janela na qual poderão ser adicionados os dados do seu novo contato (Figura 3).



FIGURA 3 - Interface Tela de Contato

```

public class Contato {

    private String name;
    private String fone;

    public Contato(String name, String fone) {
        this.name = name;
        this.fone = fone;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getFone() {
        return fone;
    }

    public void setFone(String fone) {
        this.fone = fone;
    }
}

```

FIGURA 4 - Entidade Contato

O trecho de código (Figura 4) vai pegar os valores nos campos nome e fone e colocá-los em memória. Eles ficarão salvos enquanto o programa estiver em execução, ao fechar o programa todos os dados inseridos serão perdidos.

Para que isto não se torne um problema e o programa se torne funcional, foi inserido o botão **Exportar CSV**, no qual o usuário terá a opção de exportar seus contatos em um arquivo para possivelmente manipulá-lo em um terceiro programa, como por exemplo o Excel.

```

class Gestor implements ActionListener {

    private Agenda agenda;
    final private String path = "contatos.csv";

    public Gestor(Agenda agenda) {
        this.agenda = agenda;
    }

    public void actionPerformed(ActionEvent c) {
        String cmd = c.getActionCommand();
        switch (cmd) {
            case "AdicionarContato":
                AdicionarContato();
                break;
            case "acharContato":
                acharContato();
                break;
            case "DeletarContato":
                DeletarContato();
                break;
            case "AtualizarNumero":
                AtualizarNome();
                break;
            case "AtualizarNome":
                AtualizarNome();
                break;
            case "exportarCSV":
                {
                    try{
                        exportarCSV();
                    } catch (IOException ex) {
                        System.out.println("Erro ao exportar!");
                    }
                }
                break;

            default:
                break;
        }
    }
}

```

FIGURA 5 - Exportar CSV

A parte implementada para exportação (Figura 5) foi criado um case *exportarCSV* que ao clicar no botão **Exportar CSV** ele irá gerar um arquivo na pasta do próprio projeto chamado *contatos.csv*. O interessante desta funcionalidade é que enquanto o arquivo não for movido da pasta do projeto poderão ser adicionados mais contatos nele, mesmo com o programa já sendo reaberto várias vezes. Conseguido por causa do *buffWrite* que sempre vai escrevendo no arquivo (Figura 6).

```
private void exportarCSV() throws IOException {  
    String formatoCSV = "";  
    for(Contacto it : agenda.getList()){  
        formatoCSV+=it.getName()+", "+it.getFone()+"\n";  
    }  
    BufferedWriter buffWrite = new BufferedWriter(new FileWriter(path));  
    buffWrite.append(formatoCSV);  
    buffWrite.close();  
}
```

FIGURA 6 - Buffwrite

3 CONCLUSÃO

Tiveram resultados esperados no código, o tornando funcional e não apenas sendo um projeto. Apesar de simples a interface se mostrou bem funcional, fazendo com que até o mais inexperiente usuário consiga fazer uso. Foi implementado uma funcionalidade para exportar os contatos, fazendo com que o programa se torne seguro e utilizável