

Curso Profissional: Programador/a de Informática

PSD – 11.º ano: UFCD 10791 – Desenvolvimento de aplicações web em JAVA

Ficha de Trabalho 4

Ano letivo 22/23

1. Cria no phpMyAdmin a BD com o nome **agenda**.

2. Cria a seguinte tabela na BD agenda:

NOTA: O campo CodContacto é AUTO_INCREMENT

3. No Eclipse cria um projeto de nome **AgendaBD**.

Nota: a imagem será adicionada no exercício 10

4. Cria uma Janela Application Window com o nome **AgendaMenu**.

(File→New→Other→ Application Window)

5. A partir da vista de **Design**, opta por um layout – Absolute layout e substitui o ícone pela imagem *agenda_icon.png* (deve criar, previamente, uma pasta para adicionares aí as imagens do projeto).

(Clicar com o botão direito no GetContentPane→ Set layout→Absolute layout)

6. A partir da vista de **Design**, adiciona à janela:

a. Um componente **JMenuBar**;

b. No componente anterior inclui:

i. Dois componentes **JMenu**:
CONTACTOS e PROCURAR;

Um componente **JMenu**: SAIR.


c. Ao componente **CONTACTOS** acrescenta:

i. Dois componentes **JMenuItem**: NOVO e LISTAR.

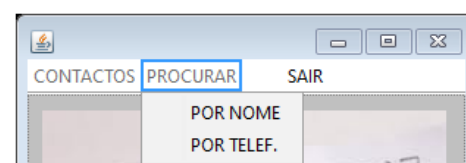
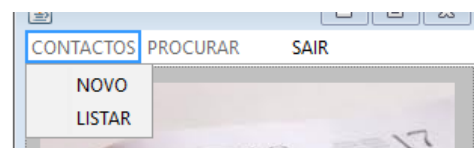
d. Ao componente **PROCURAR** acrescenta:

i. Dois componentes **JMenuItem**:

POR NOME e POR TELEF.



agenda contactos	
CodContacto	: int(11)
Nome	: char(50)
Telefone	: char(9)
Telemovel	: char(9)
Email	: char(20)



7. Associa ao menu **SAIR** um evento para fechar a aplicação (System.exit(0);).

É necessário o código seguinte:

Neste caso, é necessário usar um Listener para Jmenu:

```
userMenu.addMenuListener(new MenuListener(){
    @Override
    public void menuSelected(MenuEvent e)
        System.exit(0);
    }
    @Override
    public void menuCanceled(MenuEvent e) {
        // Invoked when the menu is canceled.
    }
    @Override
    public void menuDeselected(MenuEvent e) {
        // Invoked when the menu is deselected.
    }
});
```

8. Adiciona o seguinte título à janela: AGENDA.

9. Adiciona um fundo, de cor à tua escolha, à janela.

10. Adiciona uma imagem de fundo à janela, a aplicar a partir de um componente JLabel.

1.º passo: No workspace - copiar previamente a imagem para o package/pasta correspondente ao projeto Java;

2.º passo: No Eclipse criar um novo package com o nome **recursos**, no package **src** da aplicação (não esquecer de fazer Refresh para que a imagem seja assumida no projeto- botão direito no package recursos e clicar no item Refresh);

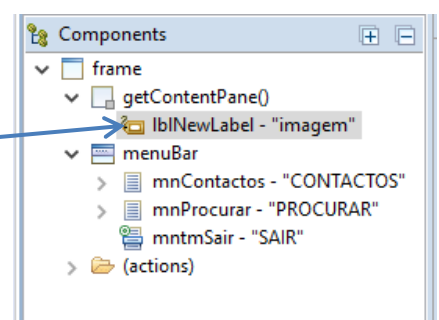
3.º passo: Criar um componente JLabel e atribuir-lhe o nome **imagem**.

//é criado o seguinte código no programa fonte (source)

```
JLabel lblNewLabel = new JLabel("imagem");
lblNewLabel.setBounds(10, 11, 279, 166);
// A
frame.getContentPane().add(lblNewLabel);
```

4.º passo: Duplo clique no JLabel e no código construído acrescentar as instruções **A**, na posição A no código gerado, importando as bibliotecas necessárias.

A



```
ImageIcon icon = new ImageIcon(AgendaMenu.class.getResource("/recursos/agenda.jpg"))
lblNewLabel.setIcon(icon);
```

11. Cria uma nova classe que te permita fazer a ligação à base de dados criada em 1 e também fechar essa ligação.

12. Cria agora uma janela do tipo JFrame, com o nome NovoContacto semelhante à figura à direita (o botão fechar será para fechar a janela e não toda a aplicação):

- As caixas de texto devem ter os nomes: textNome, textTelefone, textTelemovel e textEmail
- Os rótulos devem ter os nomes: lblNome, lblTelefone, lblTelemovel e lblEmail
- Os botões terão os nomes: guardar e cancelar

13. O botão guardar ao ser clicado irá permitir guardar os dados na tabela contactos da BD agenda.

Código associado ao evento ActionPerformed:

```
if (cxtNome.getText().equals("") || cxtTelefone.getText().equals("") ||
cxtTelemovel.getText().equals("") || cxtEmail.getText().equals(""))

JOptionPane.showMessageDialog(null,"Todos os campos são de preenchimento obrigatório!
Se não conhece um dos dados escreva: 'NC'.")
else {
    try {
        PreparedStatement pstmt;
        //O valor null utiliza-se para indicar que o primeiro campo da tabela é de numeração
        automática (auto increment)
        String sqlNovoContacto = "INSERT INTO contactos VALUES(null,?,?,?,?)";
        LigacaoBD ligacaoBD = new LigacaoBD();
        Connection con = ligacaoBD.obterLigacao();
        pstmt = con.prepareStatement(sqlNovoContacto);
        pstmt.setString(1, cxtNome.getText());

        //O método de classe parseInt da classe Integer permite converter o valor do tipo String
        num valor do tipo int
        pstmt.setInt(2, Integer.parseInt(cxtTelefone.getText()));
        pstmt.setInt(3, Integer.parseInt(cxtTelemovel.getText()));
        pstmt.setString(4, cxtEmail.getText());
        pstmt.executeUpdate();
        ligacaoBD.fecharLigacao(con);
        JOptionPane.showMessageDialog(null,"Os dados foram guardados com
sucesso!");
        limpaCampos();    // a adicionar como método na classe NovoContacto antes
de usar!!!
    }
    catch(SQLException sqle) {
        JOptionPane.showMessageDialog(null,"Não foi possível efetuar a operação sobre a
BD! "+sqle.getMessage());
    }
}
```

//limpar os campos

```
private void limpaCampos() {
    cxtNome.setText("");
    cxtTelefone.setText("");
    cxtTelemovel.setText("");
    cxtEmail.setText("");
}
```

14. O botão **cancelar** ao ser clicado irá permitir enviar uma mensagem e limpar as caixas de texto.

Código associado ao evento ActionPerformed:

```
JOptionPane.showMessageDialog(null, "Operação cancelada!");  
limpaCampos();
```

15. Efetua passos semelhantes aos efetuados na questão 7, mas agora para abrir a janela NovoContacto

Código associado ao evento ActionPerformed:

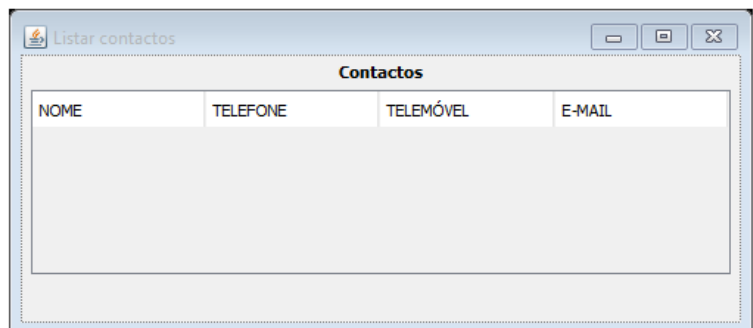
```
NovoContacto novocontacto=new NovoContacto(); //cria uma instância da classe  
novocontacto.setVisible(true); //”abre” a janela
```

16. Cria uma janela do tipo JFrame, com o nome ListarContactos que irá ser semelhante à figura abaixo (o botão fechar será para fechar a janela e não toda a aplicação):

Título do JFrame: Listar contactos

Layout: FlowLayout

Para isso...



17. Acrescenta uma JTable ao projeto e na propriedade Model, procede às seguintes alterações:

model

Table model items:

NOME	TELEFONE	TELEMÓVEL	E-MAIL
------	----------	-----------	--------

Columns

Count: 4

Insert

Delete

Move Left

Move Right

Rows

Count: 1

Insert

Delete

Move Up

Move Down

Note that table above is editable. Select a cell and start typing. Use Return to commit, Esc to cancel and arrow keys to move between cells.

Column properties

Here you can edit the properties of the column selected on the above table.

No.: 0 Title: NOME Pref.width: 120

Type: Object Values: Edit...

Min.width: 120

Max.width: 120

☒ editable ☒ resizable

Define a largura de cada coluna (bsta apenas o **Max.Width**) com as medidas 120, 100, 120 e 135 que irá corresponder à largura total do JScrollPane.

No método construtor será gerado o código seguinte:

```
table = new JTable();
table.setModel(new DefaultTableModel(
    new Object[][] {
    },
    new String[] {
        "NOME", "TELEFONE", "TELEMOVEL", "E-MAIL"
    }
));

table.getColumnModel().getColumn(0).setMaxWidth(120);
table.getColumnModel().getColumn(1).setMaxWidth(100);
table.getColumnModel().getColumn(2).setMaxWidth(120);
table.getColumnModel().getColumn(3).setMaxWidth(135);
contentPane.add(table);
```

No “Design mode” obterás uma frame semelhante à seguinte:



18. Adiciona à JFrame um JScrollPane (Quando o tamanho da tela é limitado, usamos um painel de rolagem para exibir um componente grande ou um componente cujo tamanho possa mudar dinamicamente). Altera as dimensões para Width = 475 e Height=125:

```
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 247, 475, 125);
contentPane.add(scrollPane);
```

19. No código gerado após a execução das ações em 17, deves proceder, manualmente às seguintes alterações:

```
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 247, 475, 125);
contentPane.add(scrollPane);
```

```
table = new JTable();
scrollPane.setViewportView(table);
table.setModel(new DefaultTableModel(
    new Object[][] {
        {null, null, null, null},
    },
    new String[] {
        "NOME", "TELEFONE", "TELEMOVEL", "E-MAIL"
    }
));
```

```
table.getColumnModel().getColumn(0).setMaxWidth(120);
table.getColumnModel().getColumn(1).setMaxWidth(100);
table.getColumnModel().getColumn(2).setMaxWidth(120);
table.getColumnModel().getColumn(3).setMaxWidth(135);
```

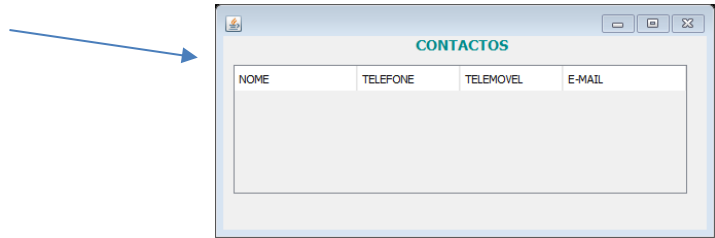
```
contentPane.add(table);

table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
DefaultTableModel tabela = (DefaultTableModel) table.getModel();
```

Retirar a instrução
(pois a tabela foi
associada ao
JScrollPane)

Permite selecionar
uma linha de cada
vez

20. Executa o projeto. Deverás obter uma janela semelhante à que se encontra exibida no exercício 16:



21. Ainda no construtor e após as instruções anteriores adiciona o código, para listar os registos da tabela no JTable do teu projeto:

//listar os contactos existentes na BD

```
try {
    Statement stmt;
    ResultSet rs;
    String sql = "SELECT * FROM contactos";
    int i = 0;
    String [] campos = new String[] {null, null, null, null};

    LigacaoBD ligacaoBD = new LigacaoBD();
    Connection con = ligacaoBD.obterLigacao();

    stmt = con.createStatement();
    rs = stmt.executeQuery(sql);

    while (rs.next()) {
        //Adiciona uma linha vazia à tabela
        tabela.addRow(campos);
        //Preenche as células da linha vazia. A numeração das colunas começa em 0
        tabela.setValueAt(rs.getString("nome"),i,0);
        tabela.setValueAt(rs.getInt("telefone"),i,1);
        tabela.setValueAt(rs.getInt("telemovel"),i,2);
        tabela.setValueAt(rs.getString("email"),i,3);
        i++;
    }
    ligacaoBD.fecharLigacao(con);
}
catch(SQLException sqle) {
    JOptionPane.showMessageDialog(null,"Não foi possível efetuar a operação
    sobre a BD! "+sqle.getMessage())
}
}
```

22. Efetua passos semelhantes aos da questão 16, mas agora para abrir a janela ListarContactos.

23. Cria uma janela do tipo JFrame, com o nome ProcurarPorNome que irá ser semelhante à figura à direita (o botão fechar será para fechar a janela e não toda a aplicação):

- a. Os nomes dos rótulos e caixas de texto podem ser iguais aos definidos na janela NovoContacto;

- b. Apenas a caixa de texto referente ao nome poderá estar EDITÁVEL;
- c. A inserção dos dados na caixa de texto do NOME é obrigatória;
- d. Ao clicar com o botão do rato na caixa de texto nome os dados deverão ser “limpos”;
Evento: mouseClicked
- e. Ao clicar no botão procurar deverá ser efetuada uma pesquisa na bd agenda e, se existir, exibido o registo referente ao nome, caso contrário será exibida uma caixa de mensagem com a seguinte mensagem:

Não foi encontrado nenhum contacto com esse nome!

Código associado ao evento ActionPerformed do botão Procurar (usa a verificação relativa ao preenchimento da caixa de texto nome dentro do try e associa-lhe uma exceção definida pelo utilizador):

```
boolean encontrouNome = false; // variável para testar se foi encontrado o registo na BD
if (cxtNome.getText().equals("")) // obriga à introdução de dados para o nome
{
    JOptionPane.showMessageDialog(null, "Preencha o campo nome!");
    limpaCampos();
}
else {
    try {
        PreparedStatement pstmt;
        ResultSet rs;
        String sqlProcuraNome = "SELECT * FROM contactos WHERE nome = ?";
        LigacaoBD ligacaoBD = new LigacaoBD();
        Connection con = ligacaoBD.obterLigacao();
        pstmt = con.prepareStatement(sqlProcuraNome);
        pstmt.setString(1, cxtNome.getText());
        rs = pstmt.executeQuery();
        if (rs.next()) {
            encontrouNome = true;
            cxtTelefone.setText(String.valueOf(rs.getInt("telefone")));
            cxtTelemovel.setText(String.valueOf(rs.getInt("telemovel")));
            cxtEmail.setText(rs.getString("email"));
        }
        if (encontrouNome == false) {
            JOptionPane.showMessageDialog(null, "Não foi encontrado nenhum contacto com esse nome!");
            limpaCampos();
        }
        ligacaoBD.fecharLigacao(con);
    }
    catch (SQLException sqle) {
        System.out.println("Não foi possível efetuar a operação sobre a BD!");
        sqle.printStackTrace();
    }
}
}
```

- 24. Efetua passos semelhantes aos da questão 16, mas agora abrir a janela ProcurarPorNome.
 - 25. Efetua passos semelhantes aos da questão 19, mas agora para abrir e programar a janela ProcurarPorTelefone (ficando apenas ativa a caixa de texto cxtTelefone).
 - 26. Efetua passos semelhantes aos da questão 16, mas agora abrir a janela ProcurarPorTelefone.
- Compacta a aplicação e envia-a para a Classroom**