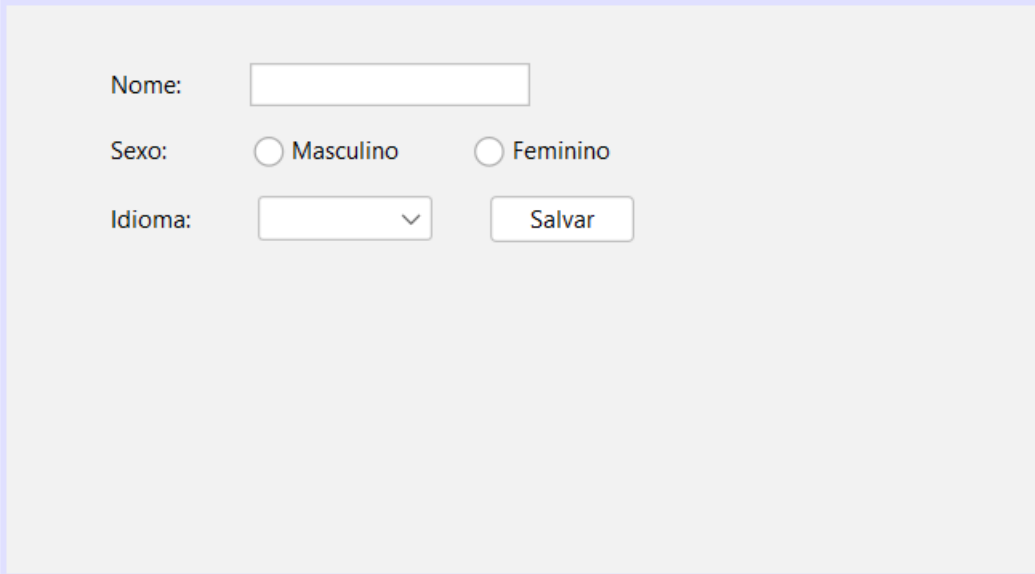


Laboratório de desenvolvimento de software

Java Swing + Banco de dados

Projeto com banco MySQL

- Vamos fazer a interface?
- Atividade 1: Faça a interface e faça com que seja cadastrado pessoas a partir dela no banco de dados

A user registration form with a light gray background and a thin blue border. It contains three input fields: a text box for 'Nome:', two radio buttons for 'Sexo:' (labeled 'Masculino' and 'Feminino'), and a dropdown menu for 'Idioma:'. A 'Salvar' button is positioned to the right of the 'Idioma:' dropdown.

Nome:

Sexo: ☐ Masculino ☐ Feminino

Idioma:

Projeto com banco MySQL

- Vamos fazer a interface?
- Atividade 1: Faça a interface e faça com que seja cadastrado pessoas a partir dela no banco de dados

```
private void btn_SalvarActionPerformed(java.awt.event.ActionEvent evt) {  
    if(rdo_Masculino.isSelected()){  
        sexo = "M";  
        /* JOptionPane.showMessageDialog(null,  
        .....  
        "Sexo selecionado: "+sexo, "Alerta!" , JOptionPane.OK_OPTION);*/  
    }  
    else if(rdo_Feminino.isSelected()){  
        sexo = "F";  
        /*JOptionPane.showMessageDialog(null,  
        .....  
        "Sexo selecionado: "+sexo, "Alerta!" , JOptionPane.OK_OPTION);*/  
    }  
    Pessoa p = new Pessoa();  
    p.setNome(nome: txt_Nome.getText());  
    p.setSexo(sexo);  
    p.setIdioma(idioma: cmb_Idioma.getSelectedItem().toString());  
  
    PessoaDAO pDAO = new PessoaDAO();  
    pDAO.inserir(pessoa:p);  
    limparFormulario();  
}
```

Projeto com banco MySQL

```
private void btn_SalvarActionPerformed(java.awt.event.ActionEvent evt) {  
    if(rdo_Masculino.isSelected()){  
        sexo = "M";  
        /* JOptionPane.showMessageDialog(null,  
            "Sexo selecionado: "+sexo, "Alerta!" , JOptionPane.OK_OPTION);*/  
    }  
    else if(rdo_Feminino.isSelected()){  
        sexo = "F";  
        /*JOptionPane.showMessageDialog(null,  
            "Sexo selecionado: "+sexo, "Alerta!" , JOptionPane.OK_OPTION);*/  
    }  
    Pessoa p = new Pessoa();  
    p.setNome(nome: txt_Nome.getText());  
    p.setSexo(sexo);  
    p.setIdioma(idioma: cmb_Idioma.getSelectedItem().toString());  
  
    PessoaDAO pDAO = new PessoaDAO();  
    pDAO.inserir(pessoa:p);  
    limparFormulario();  
}
```

Projeto com banco MySQL

```
private void limparFormulario() {  
    txt_Nome.setText(t: "");  
    btnGrp_Sexo.clearSelection();  
    cmb_Idioma.setSelectedIndex(anIndex: 0);  
}
```

Projeto com banco MySQL

- Vamos criar uma consulta de pessoa pelo ID.
- Para isso, precisaremos trabalhar na classe PessoaDAO inserindo o método getPessoa

```
public Pessoa getPessoa(int id){
    String sql = "SELECT * FROM pessoa WHERE id = ?";
    try {
        PreparedStatement stmt = conn.prepareStatement(string:sql,1: ResultSet.TYPE_SCROLL_INSENSITIVE,
            11: ResultSet.CONCUR_UPDATABLE);
        //1º parâmetro é o SQL
        //2º parâmetro é o tipo do ResultSet -
        //ResultSet scroll, ou seja, o cursor se move para frente ou para trás.
        //Este tipo de ResultSet é sensível às alterações feitas no banco de dados, ou seja, as
        //modificações feitas no banco de dados são refletidas no ResultSet.
        //3º parâmetro é sobre os parâmetros de concorrência - pode ser "read only" ou atualizável
        stmt.setInt(1: 1, 11: id);
        ResultSet rs = stmt.executeQuery(); //obtenho o retorno da consulta e armazeno no ResultSet
        Pessoa p = new Pessoa(); //Preparo um objeto que vou armazenar a consulta
        //Primeiramente, vamos posicionar o retorno da consulta (ResultSet) na primeira posição da consulta
        //Em alguns casos, a consulta terá mais de um resultado de retorno
        rs.first();
        p.setId(id);
        p.setNome(nome: rs.getString(string:"nome"));
        p.setSexo(sexo: rs.getString(string:"sexo"));
        p.setIdioma(idioma:rs.getString(string:"idioma"));
        return p;
    } catch (SQLException ex) {
        System.out.println("Erro ao consultar pessoa: "+ex.getMessage());
        return null;
    }
}
```

```

public Pessoa getPessoa(int id){
    String sql = "SELECT * FROM pessoa WHERE id = ?";
    try {
        PreparedStatement stmt = conn.prepareStatement(string:sql,i: ResultSet.TYPE_SCROLL_INSENSITIVE,
            il: ResultSet.CONCUR_UPDATABLE);
        //1º parâmetro é o SQL
        //2º parâmetro é o tipo do ResultSet -
        //ResultSet scroll, ou seja, o cursor se move para frente ou para trás.
        //Este tipo de ResultSet é sensível às alterações feitas no banco de dados, ou seja, as
        //modificações feitas no banco de dados são refletidas no ResultSet.
        //3º parâmetro é sobre os parâmetros de concorrência - pode ser "read only" ou atualizável
        stmt.setInt(i: 1, il: id);
        ResultSet rs = stmt.executeQuery(); //obtenho o retorno da consulta e armazeno no ResultSet
        Pessoa p = new Pessoa(); //Preparo um objeto que vou armazenar a consulta
        //Primeiramente, vamos posicionar o retorno da consulta (ResultSet) na primeira posição da consulta
        //Em alguns casos, a consulta terá mais de um resultado de retorno
        rs.first();
        p.setId(id);
        p.setNome(nome: rs.getString(string:"nome"));
        p.setSexo(sexo: rs.getString(string:"sexo"));
        p.setIdioma(idioma:rs.getString(string:"idioma"));
        return p;
    } catch (SQLException ex) {
        System.out.println("Erro ao consultar pessoa: "+ex.getMessage());
        return null;
    }
}

```

Projeto com banco MySQL

- E para consultar, precisaremos ter uma interface.
- Ela servirá para consultar (e daqui a pouco, editar e excluir o registro)
- Nesta interface, inicialmente, teremos estes campos:

Editar e Excluir Pessoa:

ID:

Nome:

Sexo:

Idioma:

Projeto com banco MySQL

- Agora, vamos criar um outro formulário (Jframe Form)

Editar e Excluir Pessoa:

ID:

Nome:

Sexo:

Idioma:

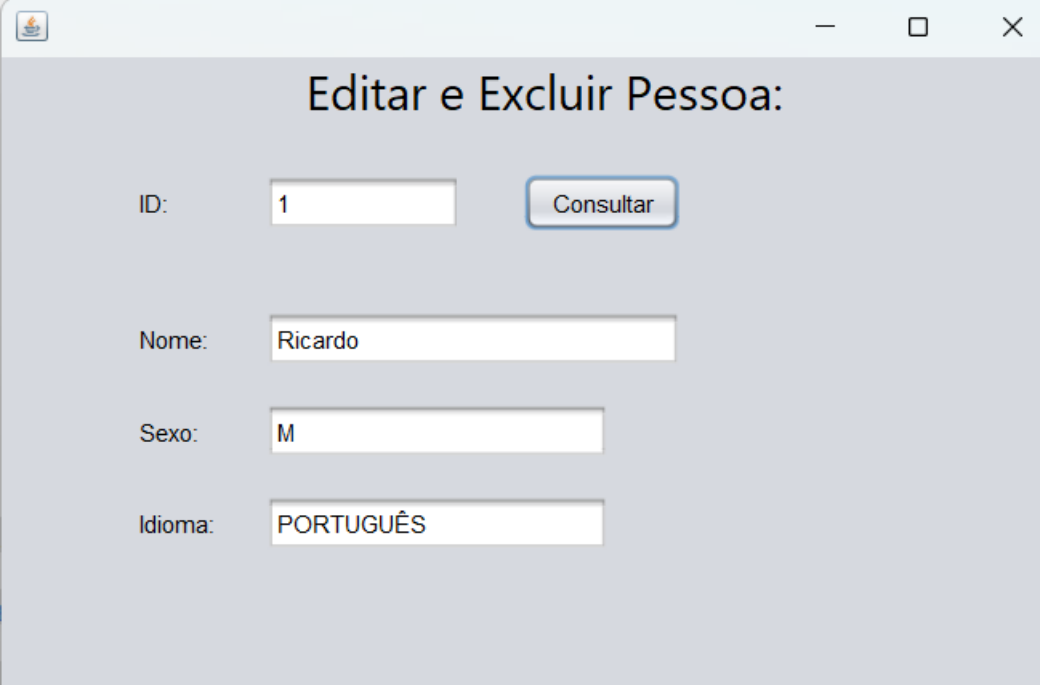
Projeto com banco MySQL

- E no clique do botão consultar, vamos chamar o método getPessoa da classe PessoaDAO:

```
private void btn_ConsultarActionPerformed(java.awt.event.ActionEvent evt) {  
    int idPessoa = Integer.parseInt(s: txt_IDPessoa.getText());  
    PessoaDAO pDAO = new PessoaDAO();  
  
    Pessoa p = pDAO.getPessoa(id: idPessoa);  
    if(p == null){  
        txt_Nome.setText(t: "");  
        txt_Sexo.setText(t: "");  
        txt_Idioma.setText(t: "");  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Pessoa não encontrada!");  
    }  
    else{  
        txt_Nome.setText(t: p.getNome());  
        txt_Sexo.setText(t: p.getSexo());  
        txt_Idioma.setText(t: p.getIdioma());  
    }  
}
```

Projeto com banco MySQL

- E deve funcionar desta forma:



A screenshot of a web application window titled "Editar e Excluir Pessoa:". The window has a light blue header bar with standard window controls (minimize, maximize, close). The main content area is light gray and contains a form with four labeled input fields and a button. The fields are: "ID:" with the value "1", "Nome:" with the value "Ricardo", "Sexo:" with the value "M", and "Idioma:" with the value "PORTUGUÊS". A blue button labeled "Consultar" is positioned to the right of the ID field.

Label	Value
ID:	1
Nome:	Ricardo
Sexo:	M
Idioma:	PORTUGUÊS

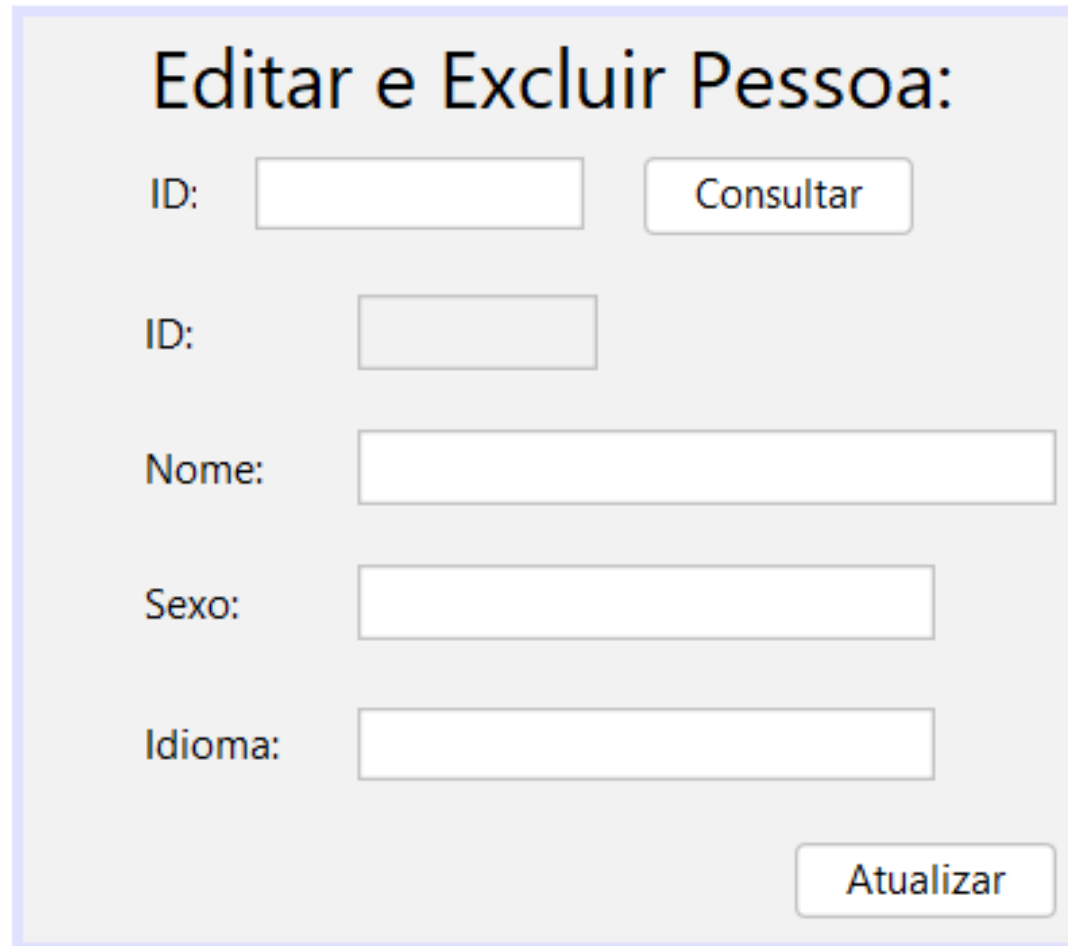
Projeto com banco MySQL

- E agora, podemos implementar o editar (update) e para isso, vamos trabalhar novamente na classe PessoaDAO:

```
public void editar(Pessoa pessoa) {  
    try {  
        String sql = "UPDATE pessoa set nome=?, sexo=?, idioma=? WHERE id=?";  
  
        PreparedStatement stmt = conn.prepareStatement(string:sql);  
        stmt.setString(i: 1, string:pessoa.getNome());  
        stmt.setString(i: 2, string:pessoa.getSexo());  
        stmt.setString(i: 3, string:pessoa.getIdioma());  
        stmt.setInt(i: 4, i: pessoa.getId());  
        stmt.execute();  
    } catch (SQLException ex) {  
        System.out.println("Erro ao atualizar pessoa: "+ex.getMessage());  
    }  
}
```

Projeto com banco MySQL

- E vamos atualizar a nossa interface gráfica com os seguintes componentes:



Editar e Excluir Pessoa:

ID:

ID:

Nome:

Sexo:

Idioma:

The image shows a graphical user interface for a database application. It has a light gray background with a blue border. At the top, the title 'Editar e Excluir Pessoa:' is displayed in a large, bold, black font. Below the title, there are five input fields and two buttons. The first row contains an 'ID:' label, a text input field, and a 'Consultar' button. The second row contains an 'ID:' label and a text input field. The third row contains a 'Nome:' label and a text input field. The fourth row contains a 'Sexo:' label and a text input field. The fifth row contains an 'Idioma:' label and a text input field. At the bottom right, there is an 'Atualizar' button.

Projeto com banco MySQL

- E no botão atualizar, vamos colocar o seguinte código:

```
private void btn_AtualizarActionPerformed(java.awt.event.ActionEvent evt) {  
    Pessoa p = new Pessoa();  
    p.setId(id: Integer.parseInt(s: txt_ID.getText()));  
    p.setNome(nome: txt_Nome.getText());  
    p.setSexo(sexo: txt_Sexo.getText());  
    p.setIdioma(idioma: txt_Idioma.getText());  
  
    PessoaDAO pDAO = new PessoaDAO();  
    pDAO.editar(pessoa:p);  
    limparFormulario();  
}
```

```
private void limparFormulario() {  
    txt_ID.setText(t: "");  
    txt_Nome.setText(t: "");  
    txt_Sexo.setText(t: "");  
    txt_Idioma.setText(t: "");  
}
```

Projeto com banco MySQL

- E agora, vamos criar um método de exclusão dentro da classe PessoaDAO:

```
public void excluir(int id) {  
    try {  
        String sql = "delete from pessoa WHERE id=?";  
  
        PreparedStatement stmt = conn.prepareStatement(string:sql);  
        stmt.setInt(i: 1, il: id);  
        stmt.execute();  
    } catch (SQLException ex) {  
        System.out.println("Erro ao atualizar pessoa: "+ex.getMessage());  
    }  
}
```

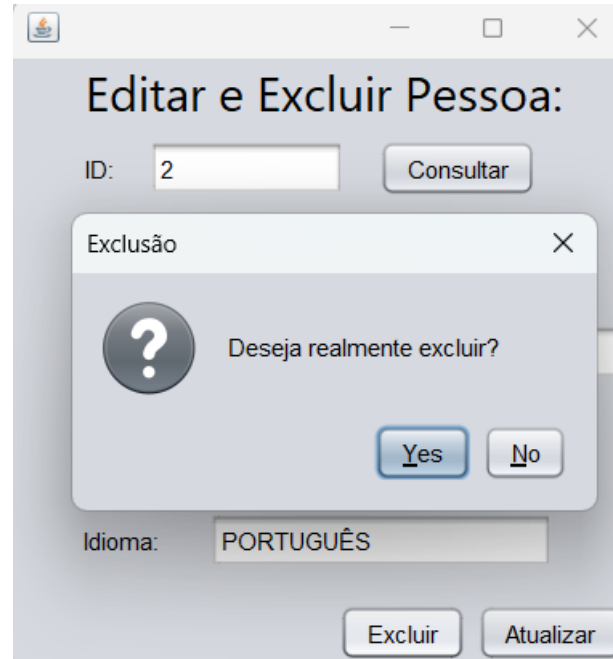
Projeto com banco MySQL

- Após, vamos adicionar um botão na nossa interface que irá chamar a exclusão:

```
private void btn ExcluirActionPerformed(java.awt.event.ActionEvent evt) {  
    int resposta = JOptionPane.showConfirmDialog(parentComponent: null, message: "Deseja realmente excluir?",  
        title: "Exclusão", optionType: JOptionPane.YES_NO_OPTION);  
    if (resposta == JOptionPane.YES_OPTION) {  
        PessoaDAO pDAO = new PessoaDAO();  
        pDAO.excluir(id: Integer.parseInt(s: txt_ID.getText()));  
        limparFormulario();  
    }  
}
```


Projeto com banco MySQL

- E o resultado deve ser este:



Projeto com banco MySQL

- E agora, vamos melhorar o nosso editar e excluir, exclua os TextField dos campos sexo e idioma e adiciona radio button para o sexo e combo box para o idioma:

Editar e Excluir Pessoa:

ID:

ID:

Nome:

Sexo: ☐ Masculino ☐ Feminino

Idioma:

Projeto com banco MySQL

- Depois, vamos alterar no consultar:

```
private void btn_ConsultarActionPerformed(java.awt.event.ActionEvent evt) {  
    int idPessoa = Integer.parseInt(s: txt_IDPessoa.getText());  
    PessoaDAO pDAO = new PessoaDAO();  
  
    Pessoa p = pDAO.getPessoa(id: idPessoa);  
    if(p == null){  
        limparFormulario();  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Pessoa não encontrada!");  
    }  
    else{  
        txt_ID.setText(t: String.valueOf(i: p.getId()));  
        txt_Nome.setText(t: p.getNome());  
        //txt_Sexo.setText(p.getSexo());  
        if(p.getSexo().equals(anObject: "M")){  
            rdo_Masculino.setSelected(b: true);  
        }  
        else{  
            rdo_Feminino.setSelected(b: true);  
        }  
        //txt_Idioma.setText(p.getIdioma());  
        cmb_Idioma.setSelectedItem(anObject: p.getIdioma());  
    }  
}
```

```
private void btn_ConsultarActionPerformed(java.awt.event.ActionEvent evt) {  
    int idPessoa = Integer.parseInt(s: txt_IDPessoa.getText());  
    PessoaDAO pDAO = new PessoaDAO();  
  
    Pessoa p = pDAO.getPessoa(id: idPessoa);  
    if(p == null){  
        limparFormulario();  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Pessoa não encontrada!");  
    }  
    else{  
        txt_ID.setText(t: String.valueOf(i: p.getId()));  
        txt_Nome.setText(t: p.getNome());  
        //txt_Sexo.setText(p.getSexo());  
        if(p.getSexo().equals(anObject: "M")){  
            rdo_Masculino.setSelected(b: true);  
        }  
        else{  
            rdo_Feminino.setSelected(b: true);  
        }  
        //txt_Idioma.setText(p.getIdioma());  
        cmb_Idioma.setSelectedItem(anObject: p.getIdioma());  
    }  
}
```

Projeto com banco MySQL

- Depois, vamos alterar no atualizar:

```
private void btn_AtualizarActionPerformed(java.awt.event.ActionEvent evt) {  
    String sexo = null;  
    if(rdo_Masculino.isSelected()){  
        sexo = "M";  
    }  
    else if(rdo_Feminino.isSelected()){  
        sexo = "F";  
    }  
    Pessoa p = new Pessoa();  
    p.setId(id: Integer.parseInt(s: txt_ID.getText()));  
    p.setNome(nome: txt_Nome.getText());  
    p.setSexo(sexo);  
    p.setIdioma(idioma: cmb_Idioma.getSelectedItem().toString());  
  
    PessoaDAO pDAO = new PessoaDAO();  
    pDAO.editar(pessoa:p);  
    limparFormulario();  
}
```

```
private void btn_AtualizarActionPerformed(java.awt.event.ActionEvent evt) {  
    String sexo = null;  
    if(rdo_Masculino.isSelected()){  
        sexo = "M";  
    }  
    else if(rdo_Feminino.isSelected()){  
        sexo = "F";  
    }  
    Pessoa p = new Pessoa();  
    p.setId(id: Integer.parseInt(s: txt_ID.getText()));  
    p.setNome(nome: txt_Nome.getText());  
    p.setSexo(sexo);  
    p.setIdioma(idioma: cmb_Idioma.getSelectedItem().toString());  
  
    PessoaDAO pDAO = new PessoaDAO();  
    pDAO.editar(pessoa:p);  
    limparFormulario();  
}
```

Projeto com banco MySQL

- Depois, conseqüentemente alterar o método limparFormulario:

```
private void limparFormulario() {  
    txt_IDPessoa.setText(t: "");  
    txt_ID.setText(t: "");  
    txt_Nome.setText(t: "");  
    btnGrp_Sexo.clearSelection();  
    cmb_Idioma.setSelectedIndex(anIndex: 0);  
}
```

Atividade

- Utilizando o exemplo e os exercícios da aula passada, faça o consultar e editar as a tabela da escola (com as tabelas Alunos e de Professores)