

# Laboratório de desenvolvimento de software

---

Java Swing + Banco de dados

# Projeto com banco MySQL

---

- Primeiramente, precisamos criar uma tabela nova
  - Vamos criar uma tabela veículo que cada pessoa terá um veículo
  - Abra o MySQL Command Line Client

# Projeto com banco MySQL

---

```
mysql> desc pessoa;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
nome	varchar(50)	NO		NULL	
sexo	varchar(1)	NO		NULL	
idioma	varchar(1)	NO		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> CREATE TABLE veiculo(
  -> id int AUTO_INCREMENT PRIMARY KEY,
  -> modelo varchar(30),
  -> placa varchar(7),
  -> id_pessoa int NOT NULL,
  -> foreign key(id_pessoa) references pessoa(id)
  -> );
Query OK, 0 rows affected (0.01 sec)
```

# Projeto com banco MySQL

---

- Agora, com a tabela criada, vamos criar uma classe veiculo
  - Onde ela ficará?

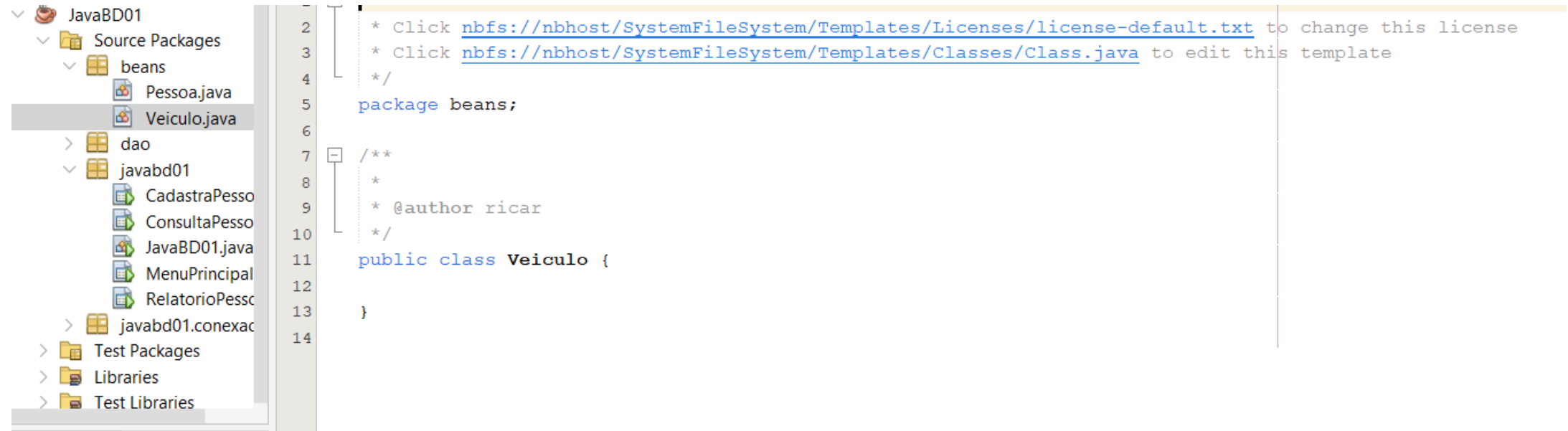
# Projeto com banco MySQL

---

- Agora, com a tabela criada, vamos criar uma classe veiculo
  - Onde ela ficará?
  - Qual delas?
- Precisaremos de uma classe Veiculo dentro do pacote beans que ficarão os atributos (colunas) da tabela
- E também de um VeiculoDAO dentro do pacote DAO que acessará o banco de dados

# Projeto com banco MySQL

---



# Projeto com banco MySQL

---

```
public class Veiculo {
    private int id;
    private String modelo;
    private String placa;
    private Pessoa pessoaid;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getModelo() {
        return modelo;
    }
    public void setModelo(String modelo) {
        this.modelo = modelo;
    }
    public String getPlaca() {
        return placa;
    }
    public void setPlaca(String placa) {
        this.placa = placa;
    }
    public Pessoa getPessoaid() {
        return pessoaid;
    }
    public void setPessoaid(Pessoa pessoaid) {
        this.pessoaid = pessoaid;
    }
}
```

```
public class Veiculo {
    private int id;
    private String modelo;
    private String placa;
    private Pessoa pessoaid;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getModelo() {
        return modelo;
    }
    public void setModelo(String modelo) {
        this.modelo = modelo;
    }
    public String getPlaca() {
        return placa;
    }
    public void setPlaca(String placa) {
        this.placa = placa;
    }
    public Pessoa getPessoaid() {
        return pessoaid;
    }
    public void setPessoaid(Pessoa pessoaid) {
        this.pessoaid = pessoaid;
    }
}
```



# Projeto com banco MySQL

---

- Agora, vamos criar o VeiculoDAO

```
public class VeiculoDAO {  
    private Conexao conexao;  
    private Connection conn;  
  
    public VeiculoDAO() {  
        this.conexao = new Conexao();  
        this.conn = this.conexao.getConexao();  
    }  
}
```

# Projeto com banco MySQL

---

```
public void inserir (Veiculo veiculo){
    String sql = "INSERT INTO veiculo (modelo, placa, id_pessoa) VALUES (?, ?, ?)";

    try {
        PreparedStatement stmt = this.conn.prepareStatement(string:sql);
        stmt.setString(i: 1, string:veiculo.getModelo());
        stmt.setString(i: 2, string:veiculo.getPlaca());
        stmt.setInt(i: 3, i1: veiculo.getPessoaid().getId());

        stmt.execute();
    } catch (SQLException ex) {
        System.out.println("Erro ao inserir Veiculo: "+ex.getMessage());
    }
}
```

# Projeto com banco MySQL

---

```
public void editar(Veiculo veiculo){
    try {
        String sql = "UPDATE veiculo set modelo=?, placa=?, id_pessoa=? WHERE id=?";

        PreparedStatement stmt = conn.prepareStatement(string:sql);
        stmt.setString(i: 1, string:veiculo.getModelo());
        stmt.setString(i: 2,string:veiculo.getPlaca());
        stmt.setInt(i: 3,il: veiculo.getPessoaid().getId());
        stmt.setInt(i: 4, il: veiculo.getId());
        stmt.execute();
    } catch (SQLException ex) {
        System.out.println("Erro ao atualizar Veiculo: "+ex.getMessage());
    }
}
```

# Projeto com banco MySQL

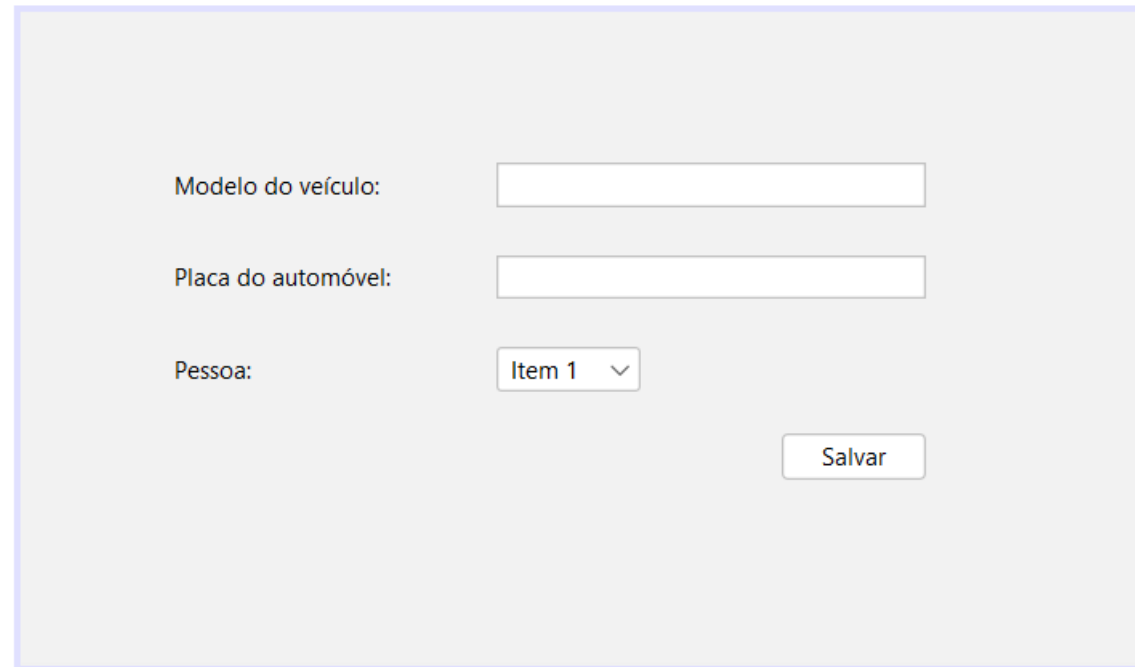
---

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

# Projeto com banco MySQL

---

- Agora, vamos criar um formulário para cadastrar o veículo:



Modelo do veículo:

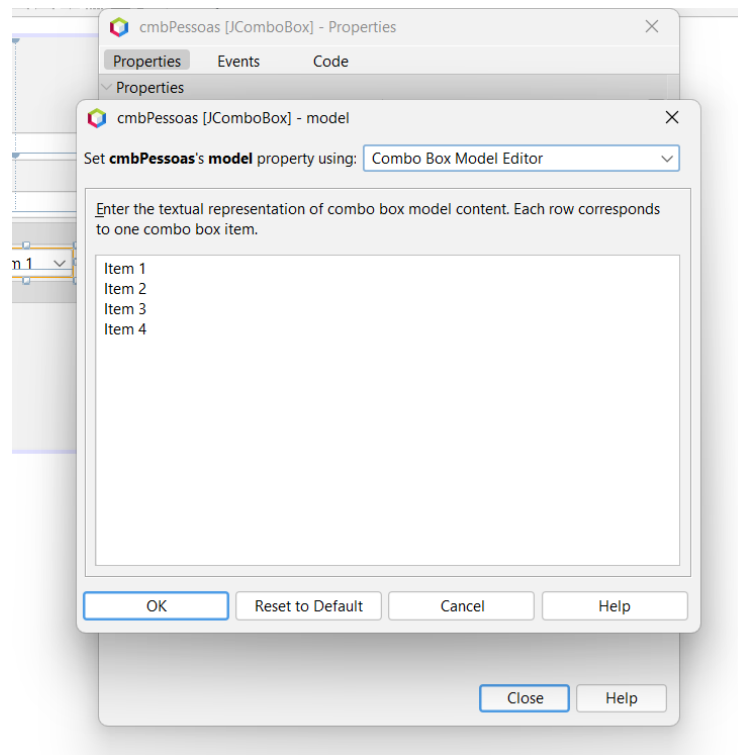
Placa do automóvel:

Pessoa:

# Projeto com banco MySQL

---

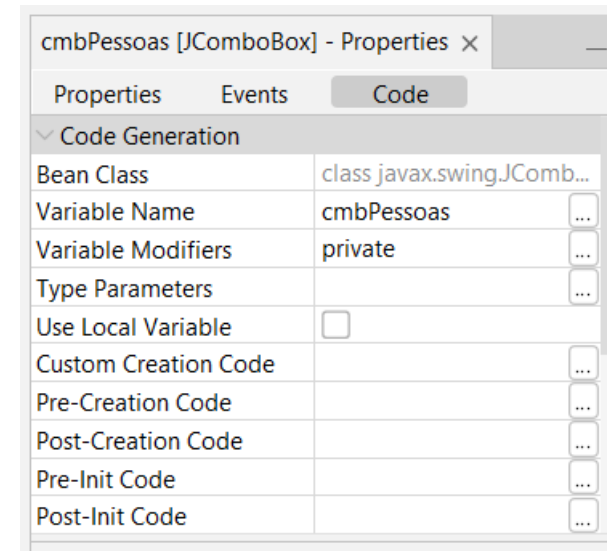
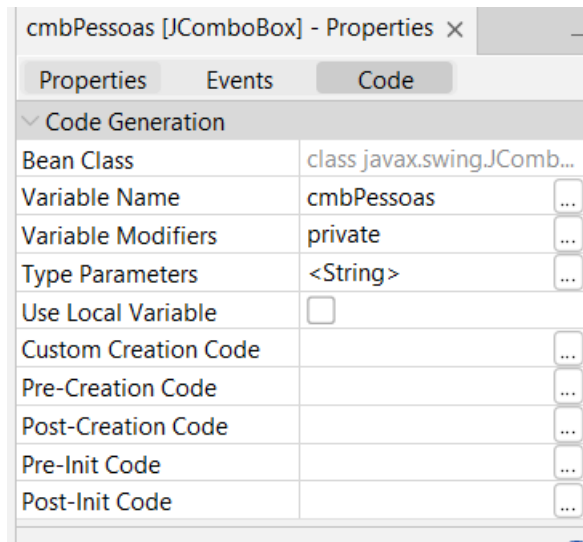
- Limpe todos os dados do combo box pois neste combo serão carregadas as pessoas do banco de dados:



# Projeto com banco MySQL

---

- Precisamos mudar nas propriedades para ele aceitar qualquer objeto dentro do combo box, no caso, um objeto do tipo Pessoa



# Projeto com banco MySQL

---

```
public class FormVeiculo extends javax.swing.JFrame {  
  
    /**  
     * Creates new form FormVeiculo  
     */  
    public FormVeiculo() {  
        initComponents();  
        preencherComboPessoas();  
    }  
  
    public void preencherComboPessoas() {  
        PessoaDAO pDAO = new PessoaDAO();  
        List<Pessoa> listaP = pDAO.getPessoas();  
        for(Pessoa p : listaP){  
            cmbPessoas.addItem(item: p);  
        }  
    }  
}
```



```
public List<Pessoa> getPessoas() {

    String sql = "SELECT * FROM pessoa";
    try {

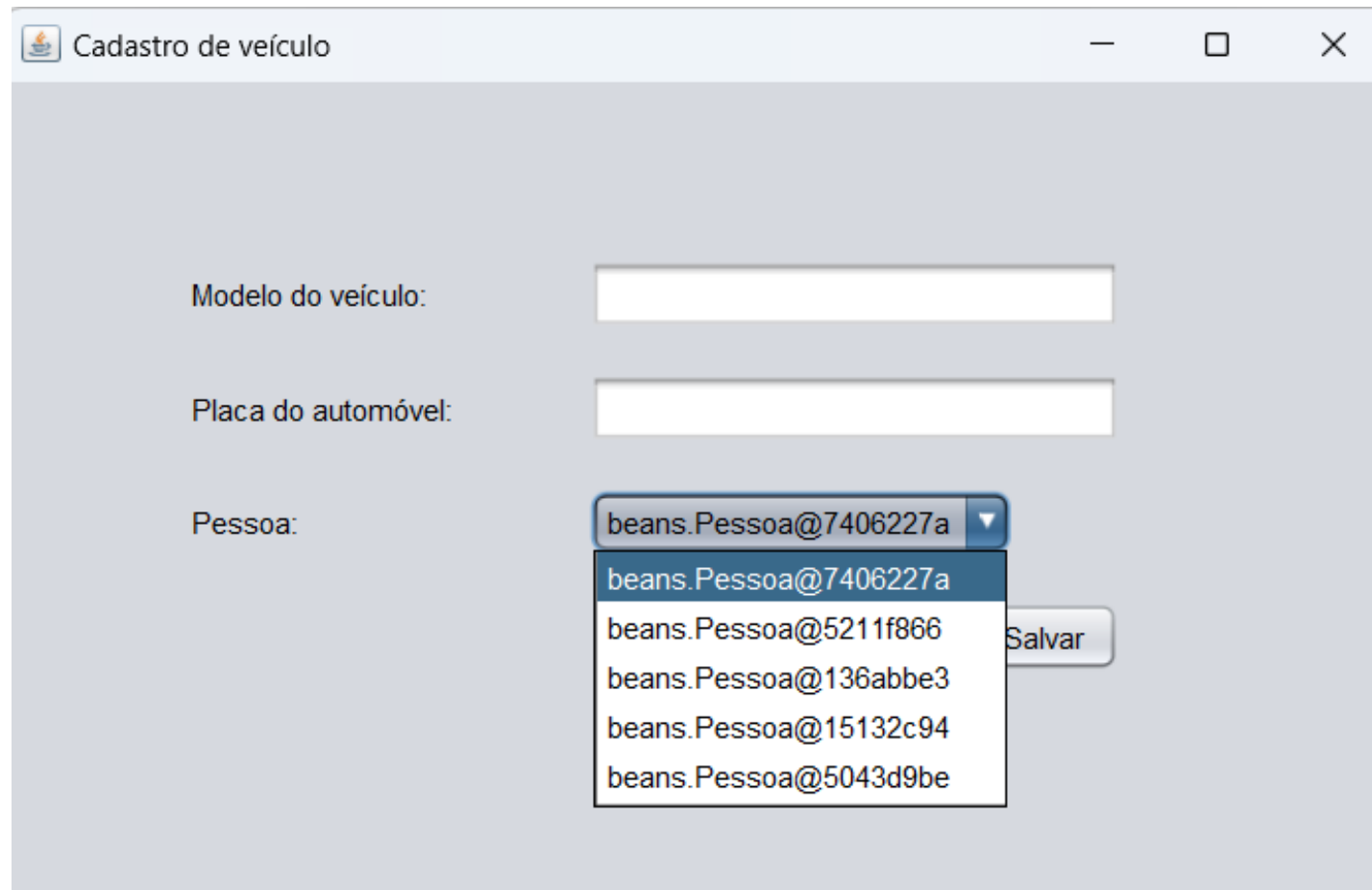
        PreparedStatement stmt = conn.prepareStatement(string:sql,i: ResultSet.TYPE_SCROLL_INSENSITIVE,
            il: ResultSet.CONCUR_UPDATABLE);

        ResultSet rs = stmt.executeQuery(); //obtenho o retorno da consulta e armazeno no ResultSet
        List<Pessoa> listaPessoas = new ArrayList(); //Preparo uma lista de objetos que vou armazenar a consulta
        //Percorre rs e salvar as informações dentro de um objeto Pessoa e depois adiciona na lista
        while (rs.next()){
            Pessoa p = new Pessoa();
            p.setId(id: rs.getInt(string:"id"));
            p.setNome(nome: rs.getString(string:"nome"));
            p.setSexo(sexo: rs.getString(string:"sexo"));
            p.setIdioma(idioma:rs.getString(string:"idioma"));
            listaPessoas.add(e: p);
        }
        return listaPessoas;
    } catch (SQLException ex) {
        System.out.println("Erro ao consultar todas as pessoas: "+ex.getMessage());
        return null;
    }
}
```

# Projeto com banco MySQL

---

- Funcionou? Ficou estranho né?



Cadastro de veículo

Modelo do veículo:

Placa do automóvel:

Pessoa: beans.Pessoa@7406227a ▼

- beans.Pessoa@7406227a
- beans.Pessoa@5211f866
- beans.Pessoa@136abbe3
- beans.Pessoa@15132c94
- beans.Pessoa@5043d9be

Salvar

# Projeto com banco MySQL

---

- Vamos melhorar, vá na classe Pessoa e implemente um método toString assim:

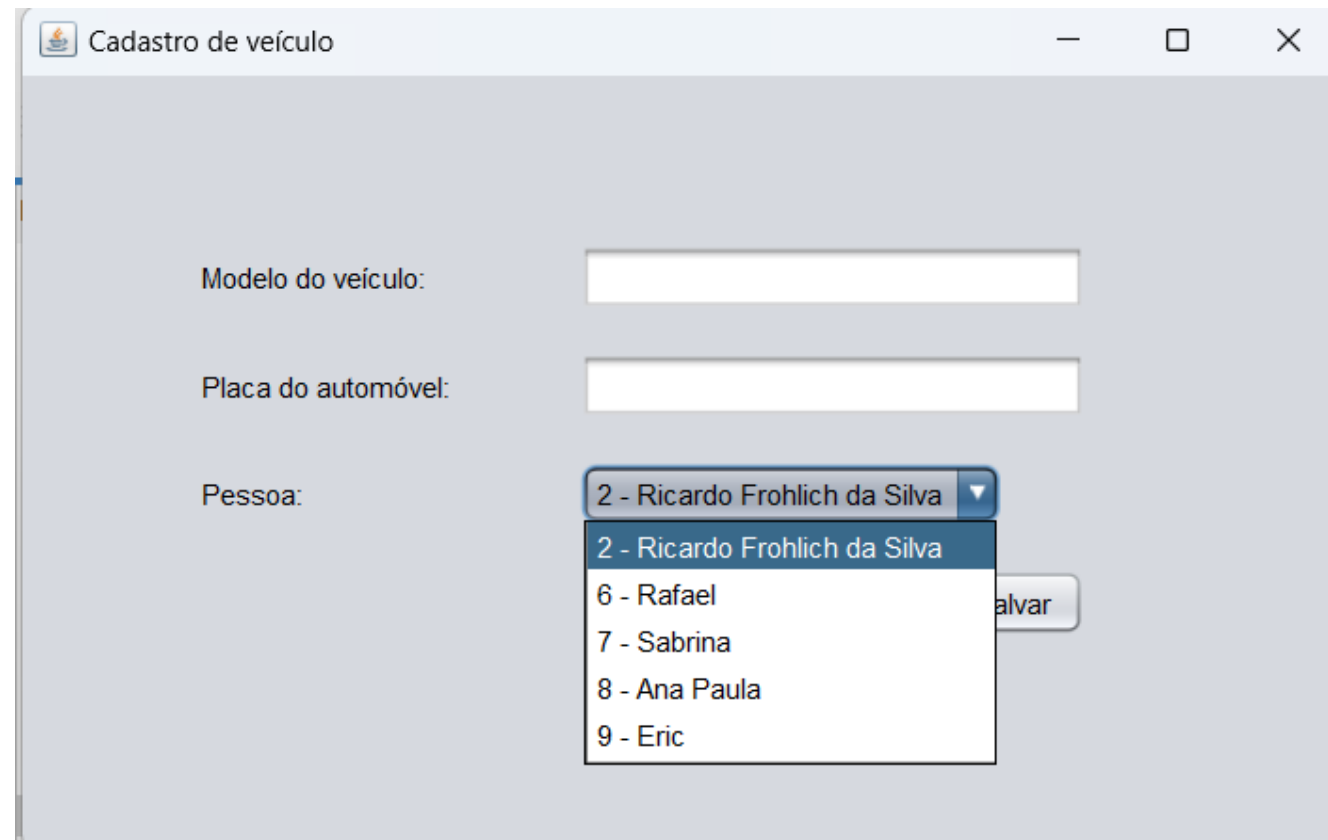
```
public String toString() {  
    return this.id+" - "+this.nome;  
}
```

- E agora?

# Projeto com banco MySQL

---

- E agora?



Cadastro de veículo

Modelo do veículo:

Placa do automóvel:

Pessoa: 2 - Ricardo Frohlich da Silva ▼

- 2 - Ricardo Frohlich da Silva
- 6 - Rafael
- 7 - Sabrina
- 8 - Ana Paula
- 9 - Eric

Salvar

# Projeto com banco MySQL

---

- Desafio:
  - Tente implementar para adicionar o veículo!

# Projeto com banco MySQL

---

- Dica 1:

```
private void btnSalvarActionPerformed(java.awt.event.ActionEvent evt) {  
    System.out.println(x: cmbPessoas.getSelectedItem());  
}
```

# Projeto com banco MySQL

---

- Dica 2:

```
private void btnSalvarActionPerformed(java.awt.event.ActionEvent evt) {  
    System.out.println(x: cmbPessoas.getSelectedItem());  
  
    Pessoa p = (Pessoa) cmbPessoas.getSelectedItem();  
  
    System.out.println(x: p.getNome());  
    System.out.println(x: p.getSexo());  
    System.out.println(x: p.getIdioma());  
}
```