

UNIVERSIDADE VEIGA DE ALMEIDA

JÔNATAS ELEOTÉRIO SILVA

**TRABALHO PARA O DESENVOLVIMENTO DE APLICAÇÃO COM
INTERFACE GRÁFICA E ACESSO A BANCO DE DADOS PARA
CONTROLE DE CARROS DE UM ESTACIONAMENTO**

Trabalho da Disciplina [AVA 2]

São João de Meriti - RJ
2021

JÔNATAS ELEOTÉRIO SILVA

**TRABALHO PARA O DESENVOLVIMENTO DE APLICAÇÃO COM
INTERFACE GRÁFICA E ACESSO A BANCO DE DADOS PARA
CONTROLE DE CARROS DE UM ESTACIONAMENTO**

Trabalho da Disciplina [AVA 2]

ALUNO: JÔNATAS ELEOTÉRIO SILVA

MATRÍCULA: 20191300220

MATÉRIA: PROGRAMAÇÃO ORIENTADA A OBJETOS II

PROFESSOR: MARCIO AURELIO NOVAES ESTEVES

TURMA: IL10329

ENTREGA DA AVALIAÇÃO - TRABALHO DA DISCIPLINA [AVA 1]

Trabalho para o desenvolvimento de Aplicação om interface gráfica e acesso a banco de dados para controle de carros de um estacionamento.

Desenvolvimento de projeto de aplicação em Java com uso da API JDBC para a manipulação de dados de uma tabela.

Uma empresa precisa controlar o estacionamento de seus funcionários. Sabe-se que o estacionamento só permite a entrada de automóveis.

Crie um banco de dados Access para armazenamento dos carros do estacionamento de acordo com os atributos da classe Carro. Crie a classe Carro e a classe CarroDAO para a manipulação da tabela, e uma classe para realizar a conexão com o banco de dados. Crie uma aplicação com interfaces gráficas para o controle do estacionamento e para as operações de manutenção da tabela, com inclusão, alteração e exclusão, além da geração de um relatório de carros cadastrados. Características da classe:

Introdução

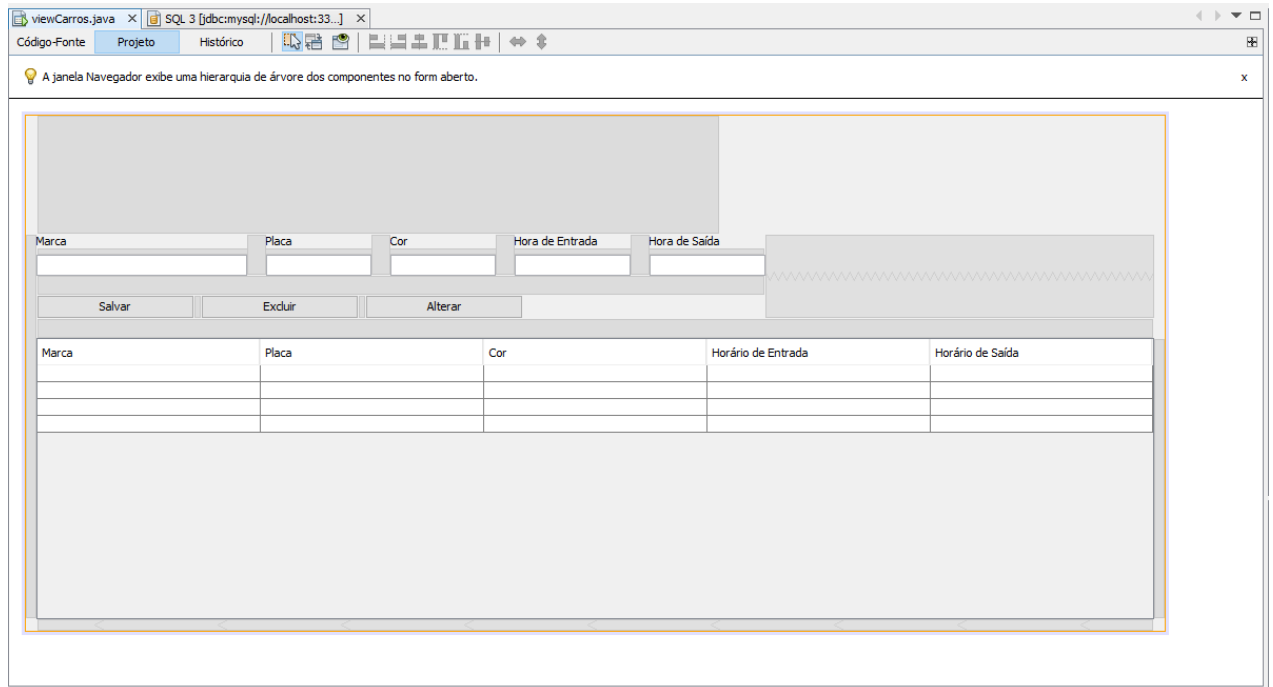
A aplicação foi desenvolvida e separada em quatro pacotes distintos a fim de manter o código escalável e de fácil compreensão. Os pacotes são os connection onde foi criada a classe ConnectionFactory de conexão com o banco de dados, model onde foi criada a classe Carro para definir construção do modelo de carro, model.dao foi criada a classe CarroDAO para a manipulação da tabela com o banco de dados e por fim o pacote view onde foi criado o JFrame da aplicação com a classe de nome viewCarros.

Foi criado o banco de dados bdcarros para que os dados fossem armazenados.

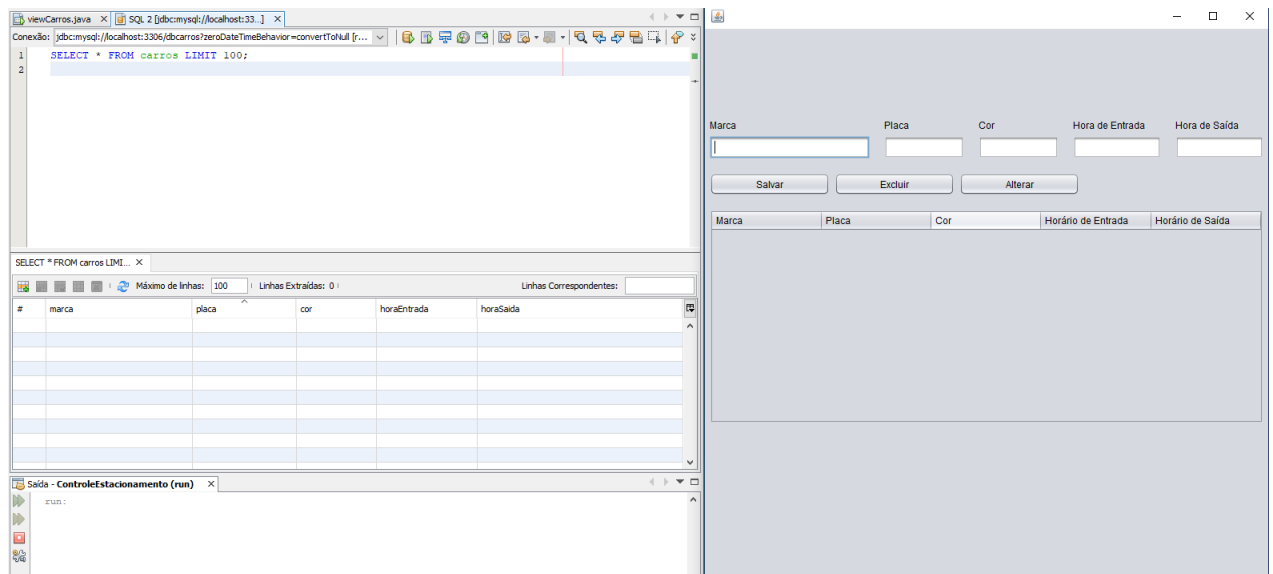
Conteúdo

Abaixo prints da aplicação construída interligada com o banco de dados

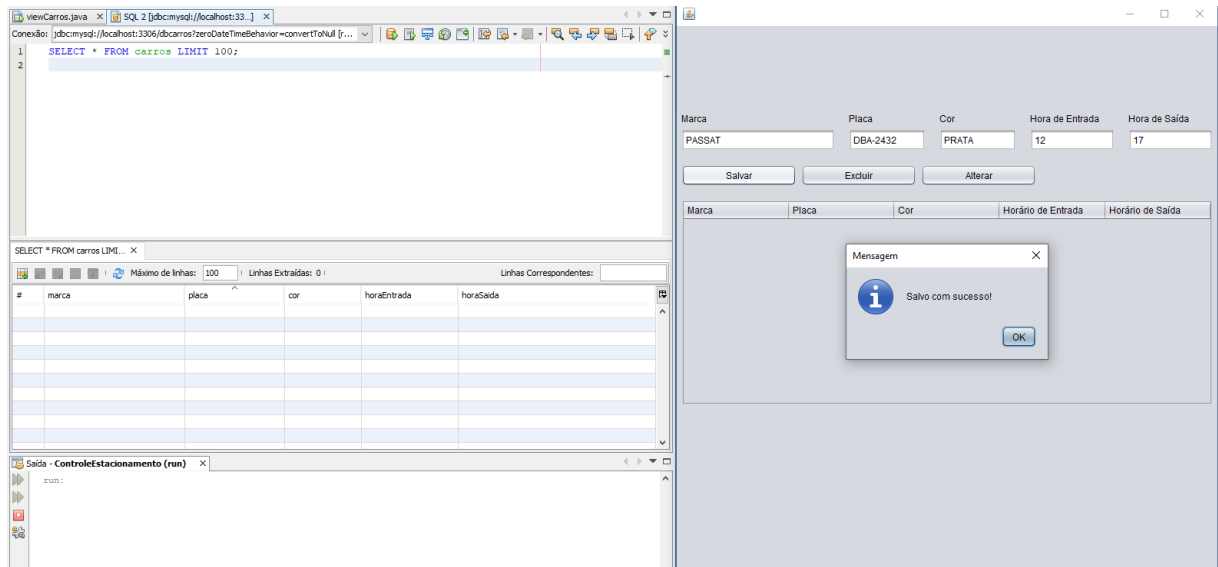
- JFrame construído.



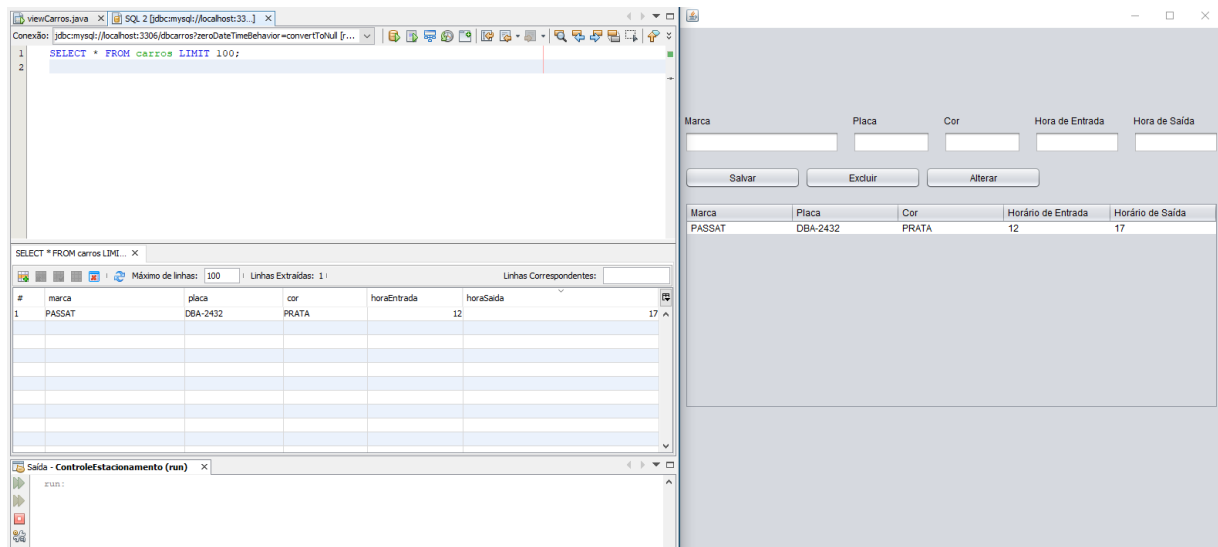
- Aplicação rodando com o banco de dados dbcarros vazio.



- Adicionando carro.



- Carro adicionado.



- Print com 6 carros já adicionados na aplicação para serem feitos as outras opções do crud.

The screenshot shows a Java IDE with a SQL query editor and a car management application window.

SQL Query Editor:

```
SELECT * FROM carros LIMIT 100;
```

Car Management Application:

Fields: Marca, Placa, Cor, Hora de Entrada, Hora de Saída

Buttons: Salvar, Excluir, Alterar

Marca	Placa	Cor	Horário de Entrada	Horário de Saída
PASSAT	DBA-2432	PRATA	12	17
GOL	VWM-3345	BRANCO	9	15
COROLLA	REE-8993	VERDE	10	14
HONDA CIVIC	FSD-0212	PRATA	9	19
ONIX	DAS-3928	AZUL	7	13
C4	FSD-3345	VERMELHO	13	20

- Excluindo campo.

The screenshot shows the same Java IDE and application window, but with a message dialog box indicating a successful exclusion.

SQL Query Editor:

```
SELECT * FROM carros LIMIT 100;
```

Car Management Application:

Fields: Marca, Placa, Cor, Hora de Entrada, Hora de Saída

Buttons: Salvar, Excluir, Alterar

Marca	Placa	Cor	Horário de Entrada	Horário de Saída
PASSAT	DBA-2432	PRATA	12	17
GOL	VWM-3345	BRANCO	9	15
COROLLA	REE-8993	VERDE	10	14
HONDA CIVIC	FSD-0212	PRATA	9	19
ONIX	DAS-3928	AZUL	7	13
C4	FSD-3345	VERMELHO	13	20

Mensagem

Excluído com sucesso!

OK

- Campo excluído

viewCarros.java

SQL 3 [jdbc:mysql://localhost:33...]

Conexão: jdbc:mysql://localhost:3306/dbcarros?zeroDateTimeBehavior=convertToNull [ro...]

1
2
SELECT * FROM carros LIMIT 100;

SELECT * FROM carros LIMIT 100

Máximo de linhas: 100 Linhas Extraídas: 5 Linhas Correspondentes:

#	marca	placa	cor	horaEntrada	horaSaida
1	PASSAT	DBA-2432	PRATA	12	17
2	GOL	VWM-3345	BRANCO	9	15
3	COROLLA	REE-8993	VERDE	10	14
4	HONDA CIVIC	FSD-0212	PRATA	9	19
5	ONIX	DAS-3928	AZUL	7	13

Saída - ControleEstacionamento (run)

27/03/2020

Marca

Placa

Cor

Hora de Entrada

Hora de Saída

Salvar

Excluir

Alterar

Marca	Placa	Cor	Horário de Entrada	Horário de Saída
PASSAT	DBA-2432	PRATA	12	17
GOL	VWM-3345	BRANCO	9	15
COROLLA	REE-8993	VERDE	10	14
HONDA CIVIC	FSD-0212	PRATA	9	19
ONIX	DAS-3928	AZUL	7	13

- Atualizando nome de HONDA CIVIC para HONDA CIVIC.

viewCarros.java

SQL 3 [jdbc:mysql://localhost:33...]

Conexão: jdbc:mysql://localhost:3306/dbcarros?zeroDateTimeBehavior=convertToNull [ro...]

1
2
SELECT * FROM carros LIMIT 100;

SELECT * FROM carros LIMIT 100

Máximo de linhas: 100 Linhas Extraídas: 5 Linhas Correspondentes:

#	marca	placa	cor	horaEntrada	horaSaida
1	PASSAT	DBA-2432	PRATA	12	17
2	GOL	VWM-3345	BRANCO	9	15
3	COROLLA	REE-8993	VERDE	10	14
4	HONDA CIVIC	FSD-0212	PRATA	9	19
5	ONIX	DAS-3928	AZUL	7	13

Saída - ControleEstacionamento (run)

27/03/2020

Marca

Placa

Cor

Hora de Entrada

Hora de Saída

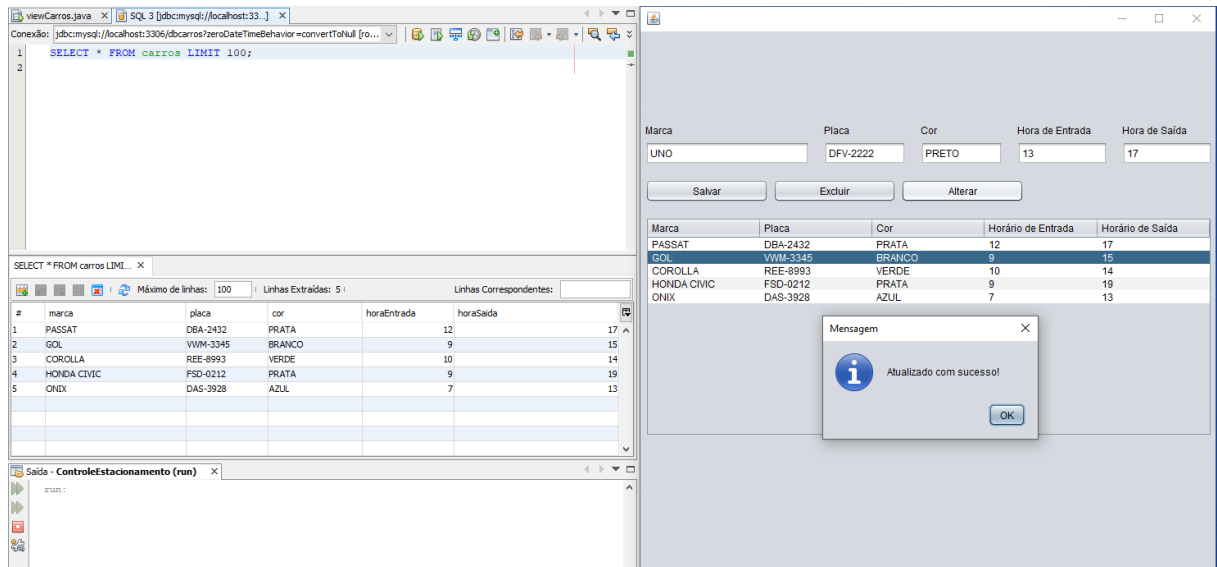
Salvar

Excluir

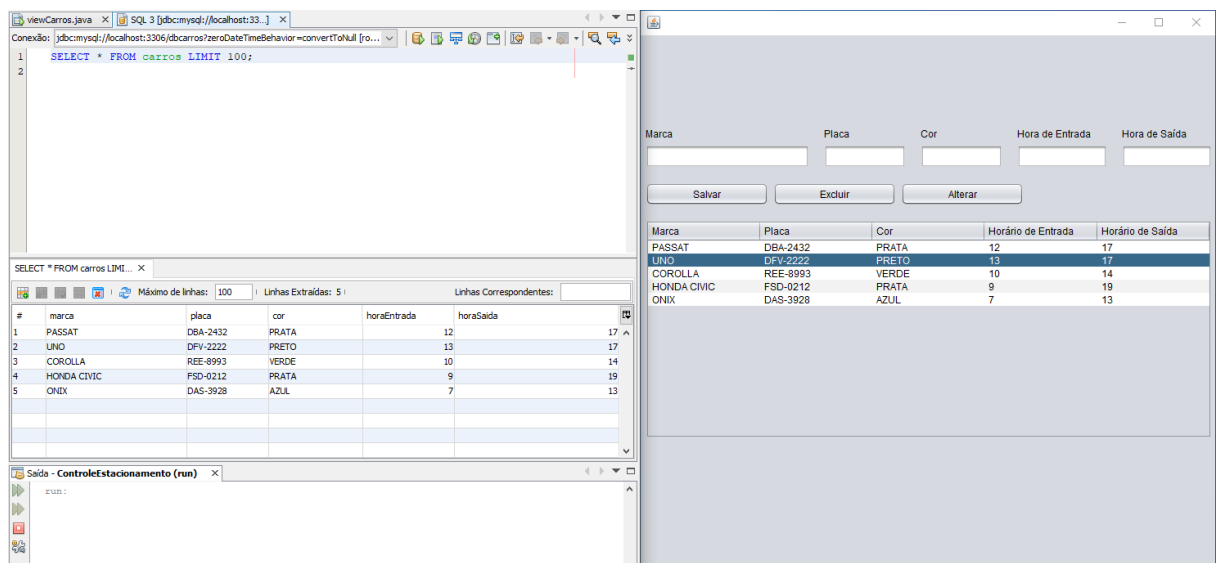
Alterar

Marca	Placa	Cor	Horário de Entrada	Horário de Saída
PASSAT	DBA-2432	PRATA	12	17
GOL	VWM-3345	BRANCO	9	15
COROLLA	REE-8993	VERDE	10	14
HONDA CIVIC	FSD-0212	PRATA	9	19
ONIX	DAS-3928	AZUL	7	13

- Atualizando todos os campos.



- Campos alterados.



Códigos de Implementação

- **ViewCarros.java**

package view;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableRowSorter;

import model.Carro;


```

import model.dao.CarroDAO;

/**
 *
 * @author Jônatas Eleotério Silva
 */
public class viewCarros extends javax.swing.JFrame {

    public viewCarros() {
        initComponents();
        DefaultTableModel modelo = (DefaultTableModel) tbCarros.getModel();
        tbCarros.setRowSorter(new TableRowSorter(modelo));
        readJTable();
    }

    public void readJTable () {
        DefaultTableModel modelo = (DefaultTableModel) tbCarros.getModel();

        modelo.setNumRows(0);

        CarroDAO cdao = new CarroDAO();

        cdao.read().forEach((c) -> {
            modelo.addRow(new Object[]{
                c.getMarca(),
                c.getPlaca(),
                c.getCor(),
                c.getHoraEntrada(),
                c.getHoraSaida()
            });
        });
    }
}

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    tbCarros = new javax.swing.JTable();
    txtMarca = new javax.swing.JTextField();
    txtPlaca = new javax.swing.JTextField();
    txtCor = new javax.swing.JTextField();
    txtHoraEntrada = new javax.swing.JTextField();
    txtHoraSaida = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    btnSalvar = new javax.swing.JButton();
    btnExcluir = new javax.swing.JButton();
    btnAlterar = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    tbCarros.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null},
            {null, null, null, null, null}
        },
        new String [] {
            "Marca", "Placa", "Cor", "Horário de Entrada", "Horário de Saída"

```

```

    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

tbCarros.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tbCarrosMouseClicked(evt);
    }
});

jScrollPane1.setViewportViewView(tbCarros);

txtHoraEntrada.setToolTipText("");
txtHoraEntrada.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtHoraEntradaActionPerformed(evt);
    }
});

jLabel1.setText("Marca");

jLabel2.setText("Placa");

jLabel3.setText("Cor");

jLabel4.setText("Hora de Entrada");

```

```
jLabel5.setText("Hora de Saída");
```

```
btnSalvar.setText("Salvar");
```

```
btnSalvar.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        btnSalvarActionPerformed(evt);  
    }  
});
```

```
btnExcluir.setText("Excluir");
```

```
btnExcluir.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        btnExcluirActionPerformed(evt);  
    }  
});
```

```
btnAlterar.setText("Alterar");
```

```
btnAlterar.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        btnAlterarActionPerformed(evt);  
    }  
});
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(layout.createSequentialGroup()  
            .addGap(10, 10, 10)  
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                .add(ContainerUtils.createFormComponent(jLabel5))  
                .add(ContainerUtils.createFormComponent(btnExcluir))  
                .add(ContainerUtils.createFormComponent(btnAlterar))  
            )  
            .addContainerGap(10, Short.MAX_VALUE)  
        )  
);
```

```

        .addComponent(jScrollPane1)

        .addGroup(layout.createSequentialGroup()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(layout.createSequentialGroup()

                    .addComponent(txtMarca, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(jLabel1))

                .addGroup(layout.createSequentialGroup()

                    .addGap(18, 18, 18)

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(layout.createSequentialGroup()

                            .addComponent(txtPlaca, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)

                            .addComponent(jLabel2))

                        .addGroup(layout.createSequentialGroup()

                            .addGap(18, 18, 18)

                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                                .addGroup(layout.createSequentialGroup()

                                    .addComponent(txtCor, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)

                                    .addComponent(jLabel3))

                                .addGroup(layout.createSequentialGroup()

                                    .addGap(18, 18, 18)

                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                                        .addGroup(layout.createSequentialGroup()

                                            .addComponent(txtHoraEntrada, javax.swing.GroupLayout.PREFERRED_SIZE,
110, javax.swing.GroupLayout.PREFERRED_SIZE)

                                            .addComponent(jLabel4))

                                        .addGroup(layout.createSequentialGroup()

                                            .addGap(18, 18, 18)

                                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                                                .addComponent(jLabel5)

                                                .addGroup(layout.createSequentialGroup()

                                                    .addComponent(txtHoraSaida, javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)))

                                                .addGroup(layout.createSequentialGroup()

                                                    .addGroup(layout.createSequentialGroup()

```

```

        .addComponent(btnSalvar, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(btnExcluir, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(btnAlterar, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGap(0, 369, Short.MAX_VALUE)))

    .addContainerGap()

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup()

        .addGap(112, 112, 112)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel1)

            .addComponent(jLabel2)

            .addComponent(jLabel3)

            .addComponent(jLabel4)

            .addComponent(jLabel5))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(txtMarca, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(txtPlaca, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(txtCor, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(txtHoraEntrada, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(txtHoraSaida, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

```

```

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(btnSalvar)
            .addComponent(btnExcluir)
            .addComponent(btnAlterar))
        .addGap(18, 18, 18)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 267,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

```

```

    pack();
} // </editor-fold>

```

```

private void txtHoraEntradaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void btnSalvarActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    Carro c = new Carro();
    CarroDAO dao = new CarroDAO();
    c.setMarca(txtMarca.getText());
    c.setPlaca(txtPlaca.getText());
    c.setCor(txtCor.getText());
    c.setHoraEntrada(Integer.parseInt(txtHoraEntrada.getText()));
    c.setHoraSaida(Integer.parseInt(txtHoraSaida.getText()));
    dao.create(c);
    txtMarca.setText("");
    txtPlaca.setText("");
    txtCor.setText("");
    txtHoraEntrada.setText("");

```

```
txtHoraSaida.setText("");  
readJTable();  
  
}
```

```
private void btnExcluirActionPerformed(java.awt.event.ActionEvent evt) {  
  
    if ( tbCarros.getSelectedRow() != -1 ) {  
        Carro c = new Carro();  
        CarroDAO dao = new CarroDAO();  
        c.setPlaca((String) tbCarros.getValueAt(tbCarros.getSelectedRow(), 1));  
        dao.delete(c);  
        txtMarca.setText("");  
        txtPlaca.setText("");  
        txtCor.setText("");  
        txtHoraEntrada.setText("");  
        txtHoraSaida.setText("");  
        readJTable();  
    }  
}
```

```
private void btnAlterarActionPerformed(java.awt.event.ActionEvent evt) {  
  
    if ( tbCarros.getSelectedRow() != -1 ) {  
        Carro c = new Carro();  
        CarroDAO dao = new CarroDAO();  
        c.setMarca(txtMarca.getText());  
        c.setPlaca(txtPlaca.getText());  
        c.setCor(txtCor.getText());  
        c.setHoraEntrada(Integer.parseInt(txtHoraEntrada.getText()));  
        c.setHoraSaida(Integer.parseInt(txtHoraSaida.getText()));  
    }  
}
```



```

        dao.update(c, tbCarros.getValueAt(tbCarros.getSelectedRow(), 1).toString());

        txtMarca.setText("");
        txtPlaca.setText("");
        txtCor.setText("");
        txtHoraEntrada.setText("");
        txtHoraSaida.setText("");
        readJTable();
    }
}

```

```

private void tbCarrosMouseClicked(java.awt.event.MouseEvent evt) {

```

```

    if ( tbCarros.getSelectedRow() != -1 ) {
        txtMarca.setText(tbCarros.getValueAt(tbCarros.getSelectedRow(), 0).toString());
        txtPlaca.setText(tbCarros.getValueAt(tbCarros.getSelectedRow(), 1).toString());
        txtCor.setText(tbCarros.getValueAt(tbCarros.getSelectedRow(), 2).toString());
        txtHoraEntrada.setText(tbCarros.getValueAt(tbCarros.getSelectedRow(), 3).toString());
        txtHoraSaida.setText(tbCarros.getValueAt(tbCarros.getSelectedRow(), 4).toString());
    }
}

```

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    <!--<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

```

```

        javax.swing.UIManager.setLookAndFeel(info.getClassName());
        break;
    }
}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(viewCarros.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(viewCarros.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(viewCarros.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(viewCarros.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    }
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new viewCarros().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton btnAlterar;
private javax.swing.JButton btnExcluir;

```

```

private javax.swing.JButton btnSalvar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tbCarros;
private javax.swing.JTextField txtCor;
private javax.swing.JTextField txtHoraEntrada;
private javax.swing.JTextField txtHoraSaida;
private javax.swing.JTextField txtMarca;
private javax.swing.JTextField txtPlaca;
// End of variables declaration
}

```

- **ConnectionFactory.java**

```

package connection;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Jônatas Eleotério Silva
 */
public class ConnectionFactory {

```

```

private static final String DRIVER = "com.mysql.jdbc.Driver";
private static final String URL = "jdbc:mysql://localhost:3306/dbcarros";
private static final String USER = "root";
private static final String PASS = "1234";
public static Connection getConnection() {
    try {
        Class.forName(DRIVER);
        return DriverManager.getConnection(URL, USER, PASS);
    } catch (ClassNotFoundException | SQLException ex) {
        throw new RuntimeException(" Erro de conexão: ", ex);
    }
}

public static void closeConnection(Connection con) {
    try {
        if(con != null){
            con.close();
        }
    } catch (SQLException ex) {
        Logger.getLogger(ConnectionFactory.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public static void closeConnection(Connection con, PreparedStatement stmt) {
    closeConnection(con);
    try {
        if(stmt != null){
            stmt.close();
        }
    } catch (SQLException ex) {
        Logger.getLogger(ConnectionFactory.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

public static void closeConnection(Connection con, PreparedStatement stmt, ResultSet rs) {
    closeConnection(con, stmt);
    try {
        if(rs != null){
            rs.close();
        }
    } catch (SQLException ex) {
        Logger.getLogger(ConnectionFactory.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

- **Carro.java**

```

package model;

/**
 *
 * @author Jônatas Eleotério Silva
 */
public class Carro {
    private String marca, placa, cor;
    private int horaEntrada, horaSaida;
    public String getMarca() {
        return marca;
    }
    public void setMarca(String marca) {
        this.marca = marca;
    }
    public String getPlaca() {
        return placa;
    }
}

```

```
public void setPlaca(String placa) {  
    this.placa = placa;  
}  
public String getCor() {  
    return cor;  
}  
public void setCor(String cor) {  
    this.cor = cor;  
}  
public int getHoraEntrada() {  
    return horaEntrada;  
}  
public void setHoraEntrada(int horaEntrada) {  
    this.horaEntrada = horaEntrada;  
}  
public int getHoraSaida() {  
    return horaSaida;  
}  
public void setHoraSaida(int horaSaida) {  
    this.horaSaida = horaSaida;  
}  
public Carro() {  
}  
public Carro(String marca) {  
    this.marca = marca;  
}  
public Carro(String marca, String placa) {  
    this.marca = marca;  
    this.placa = placa;  
}  
public Carro(String marca, String placa, String cor) {
```

```

        this.marca = marca;

        this.placa = placa;

        this.cor = cor;
    }

    public Carro(String marca, String placa, String cor, int horaEntrada, int horaSaida) {

        this.marca = marca;

        this.placa = placa;

        this.cor = cor;

        this.horaEntrada = horaEntrada;

        this.horaSaida = horaSaida;
    }
}

```

- CarroDAO.java

```

package model.dao;

import connection.ConnectionFactory;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import model.Carro;

/**
 *
 * @author Jônatas Eleotério Silva
 */
public class CarroDAO {

```

```

public void create(Carro c) {
    Connection con = ConnectionFactory.getConnection();
    PreparedStatement stmt = null;
    try {
        stmt = con.prepareStatement("INSERT INTO carros
(marca,placa,cor,horaEntrada,horaSaida)VALUES(?,?,?,?,?)");

        stmt.setString(1, c.getMarca());
        stmt.setString(2, c.getPlaca());
        stmt.setString(3, c.getCor());
        stmt.setInt(4, c.getHoraEntrada());
        stmt.setInt(5, c.getHoraSaida());
        stmt.executeUpdate();
        JOptionPane.showMessageDialog(null, "Salvo com sucesso!");
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Erro ao salvar: "+ ex);
    } finally {
        ConnectionFactory.closeConnection(con, stmt);
    }
}

public List<Carro> read() {
    Connection con = ConnectionFactory.getConnection();
    PreparedStatement stmt = null;
    ResultSet rs = null;
    List<Carro> carros = new ArrayList<>();
    try {
        stmt = con.prepareStatement("SELECT * FROM carros");
        rs = stmt.executeQuery();
        while(rs.next()) {
            Carro carro = new Carro();
            carro.setMarca(rs.getString("marca"));
            carro.setPlaca(rs.getString("placa"));

```



```

        carro.setCor(rs.getString("cor"));

        carro.setHoraEntrada(rs.getInt("horaEntrada"));

        carro.setHoraSaida(rs.getInt("horaSaida"));

        carros.add(carro);
    }
} catch (SQLException ex) {

    Logger.getLogger(CarroDAO.class.getName()).log(Level.SEVERE, null, ex);

} finally {

    ConnectionFactory.closeConnection(con, stmt, rs);

}

return carros;

}

public void update(Carro c, String p) {

    Connection con = ConnectionFactory.getConnection();

    PreparedStatement stmt = null;

    try {

        stmt = con.prepareStatement("UPDATE carros SET marca = ?,placa = ?,cor = ?,horaEntrada = ?,horaSaida = ? WHERE placa = ?");

        stmt.setString(1, c.getMarca());

        stmt.setString(2, c.getPlaca());

        stmt.setString(3, c.getCor());

        stmt.setInt(4, c.getHoraEntrada());

        stmt.setInt(5, c.getHoraSaida());

        stmt.setString(6, p);

        stmt.executeUpdate();

        JOptionPane.showMessageDialog(null, "Atualizado com sucesso!");

    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(null, "Erro ao atualizar: "+ ex);

    } finally {

        ConnectionFactory.closeConnection(con, stmt);

    }
}

```

```
}  
  
public void delete(Carro c) {  
    Connection con = ConnectionFactory.getConnection();  
    PreparedStatement stmt = null;  
    try {  
        stmt = con.prepareStatement("DELETE FROM carros WHERE placa = ?");  
        stmt.setString(1, c.getPlaca());  
        stmt.executeUpdate();  
        JOptionPane.showMessageDialog(null, "Excluido com sucesso!");  
    } catch (SQLException ex) {  
        JOptionPane.showMessageDialog(null, "Erro ao excluir: "+ ex);  
    } finally {  
        ConnectionFactory.closeConnection(con, stmt);  
    }  
}  
}
```

Referências

<https://www.devmedia.com.br/manipulando-dados-com-jdbc-em-java/27287>