



PROJETO CARTEIRA DE VACINA

PROJETO POR:

WANDERLANE DA SILVA PEREIRA

EDSON SOTERO DOS SANTOS

EMANOEL MORAES

DENYS MARQUES DE OLIVEIRA

WILLAMES DA SILVA

EDUARDO FLORÊNCIO ORICO BARROS

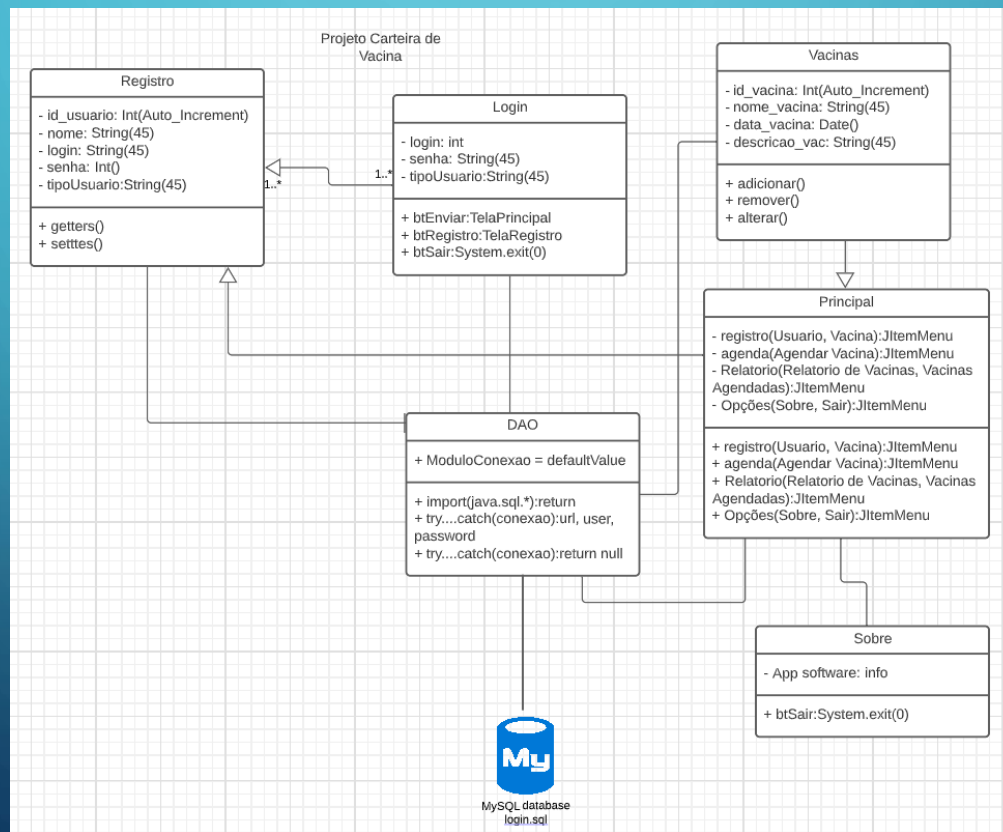
O PROJETO

Com o intuito de uma solução para ajudar no agendamento das vacinas nos postos de saúde, hospitais e também como solução para usuários. A carteira de Vacinas busca aproximar tanto o usuário comum e o profissional de saúde do agendamento e com relatórios de vacinas para cada usuário e também informa quais vacinas foram agendadas e faltam agendar.

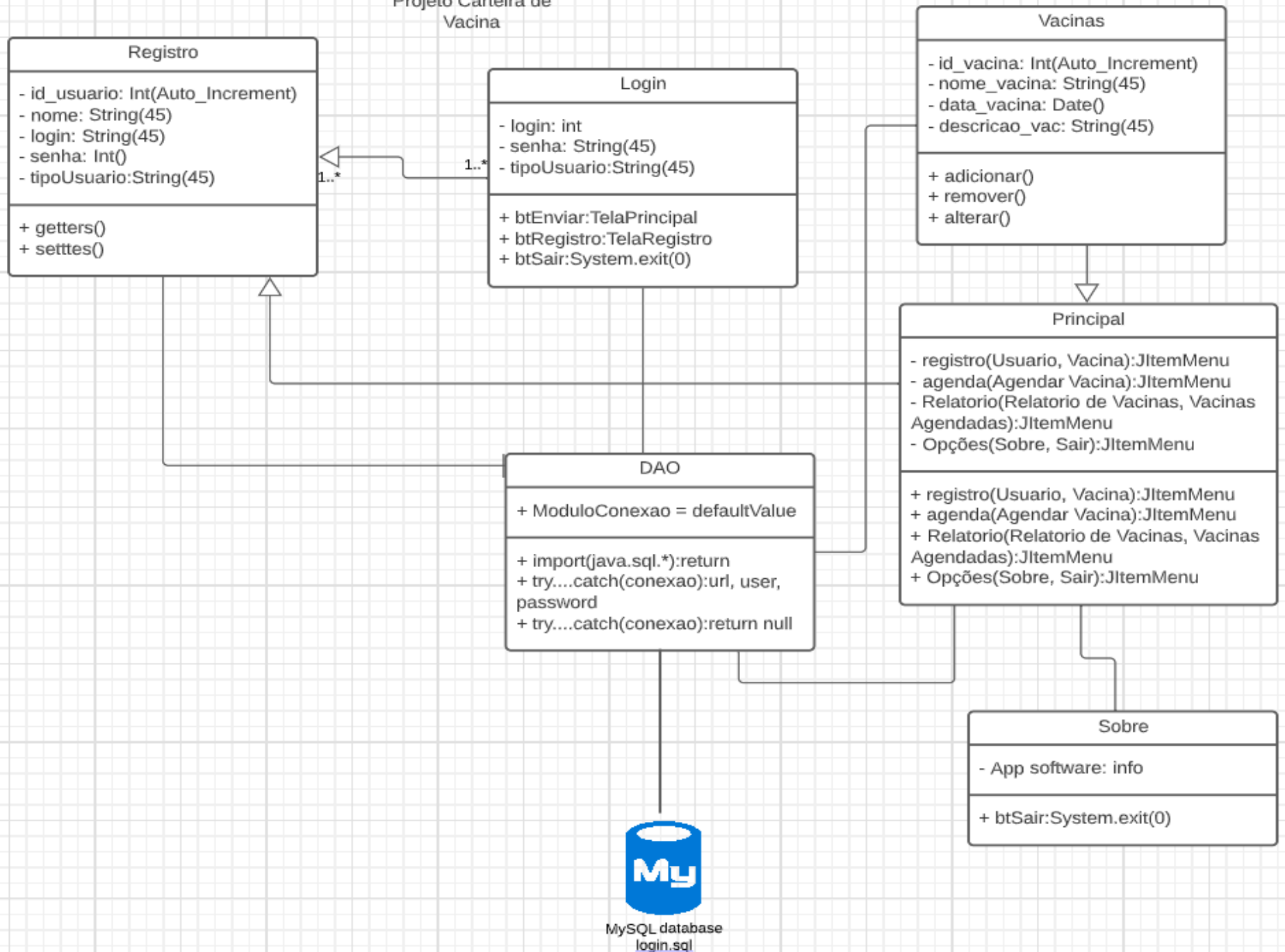
Assim essa App buscar ajudar a todos que necessitam e trabalham com saúde, especificamente com as vacinas. Podendo expandir para outras áreas da saúde.

A CARA DO APP

Iniciamos mostrando a UML.



Projeto Carteira de Vacina

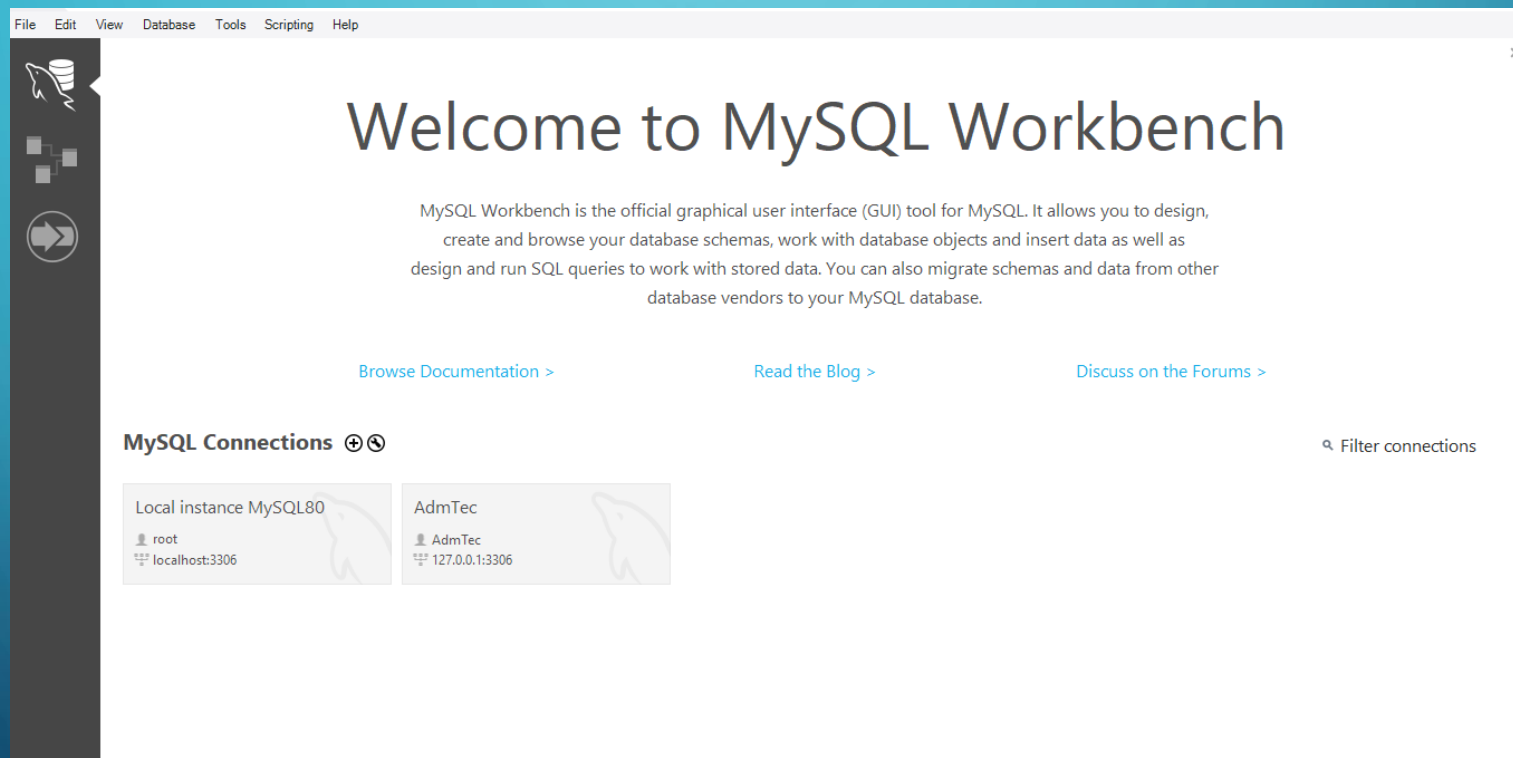


O DAO...

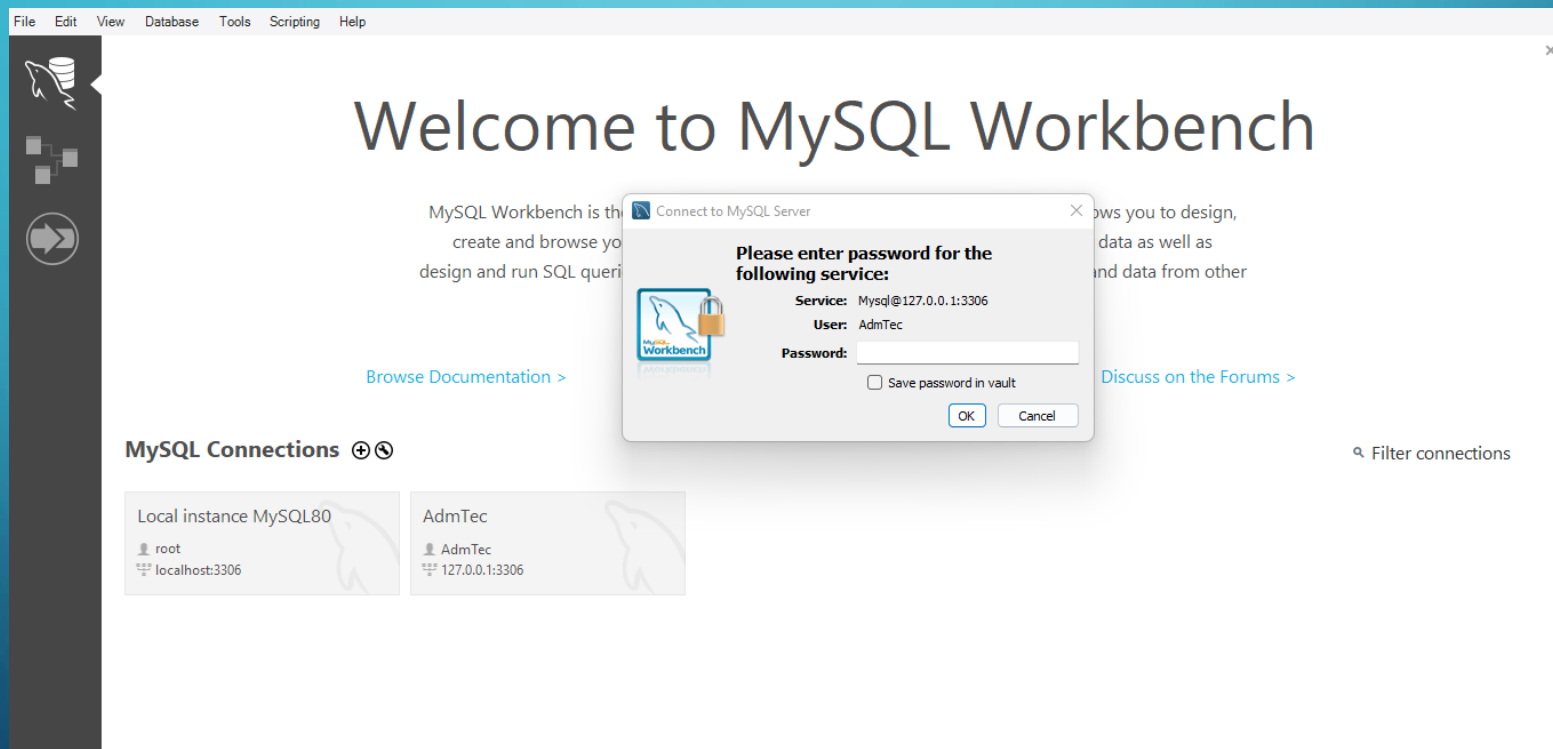
```
ModuloConexao.java X
1 package Dao;
2
3 import java.sql.*;
4
5 public class ModuloConexao {
6     // Metodo responsavel por estabelecer a conexão com o Banco
7     public static Connection conector() {
8         java.sql.Connection conexao = null;
9         //A linha abaixo "chama" o driver para biblioteca
10        String driver = "com.mysql.cj.jdbc.Driver";
11        //Criar três variaveis para armazenar informações ao banco
12        String url = "jdbc:mysql://localhost:3306/login";
13        String user = "AdmTec";
14        String password = "Pro-2022";
15        //Estabelecendo a conexão com o bando de dados
16        try {
17            Class.forName(driver);
18            conexao = DriverManager.getConnection(url, user, password);
19            return conexao;
20        }
21        catch (Exception e) {
22            //System.out.println(e);
23
24            return null;
25        }
26    }
27 }
```

```
1 package Dao;
2
3 import java.sql.*;
4
5 public class ModuloConexao {
6     // Metodo responsavel por estabelecer a conexão com o Banco
7     public static Connection conector() {
8         java.sql.Connection conexao = null;
9         //A linha abaixo "chama" o driver para biblioteca
10        String driver = "com.mysql.cj.jdbc.Driver";
11        //Criar três variaveis para armazenar informações ao banco
12        String url = "jdbc:mysql://localhost:3306/login";
13        String user = "AdmTec";
14        String password = "Pro-2022";
15        //Estabelecendo a conexão com o bando de dados
16        try {
17            Class.forName(driver);
18            conexao = DriverManager.getConnection(url, user, password);
19            return conexao;
20        }
21        catch (Exception e) {
22            //System.out.println(e);
23
24            return null;
25        }
26    }
27 }
```

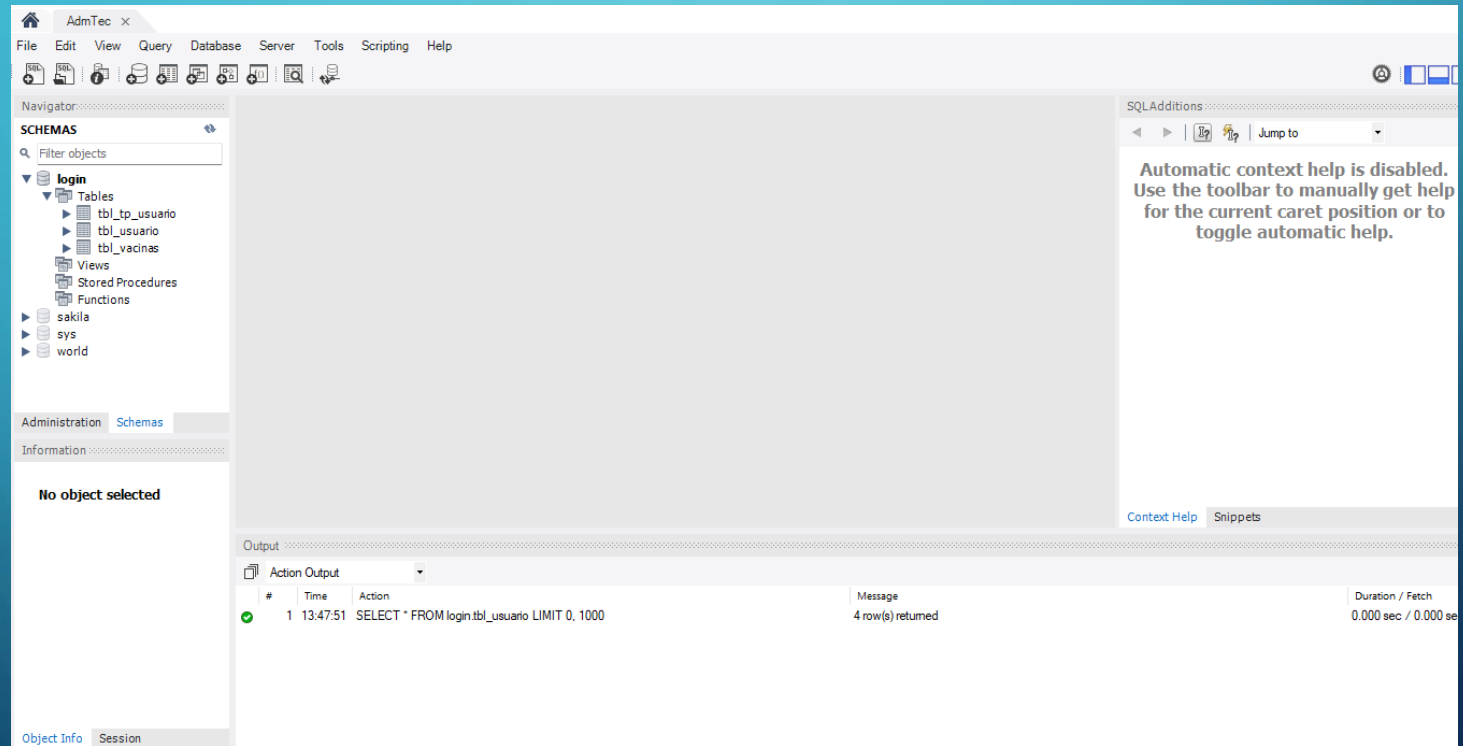
DAO - BANCO DE DADOS - MYSQL



DAO - BANCO DE DADOS - MYSQL



BANCO DE DADOS - MYSQL



BANCO DE DADOS - MYSQL

The image displays three overlapping screenshots of MySQL software interfaces, likely from a version like MySQL Workbench or a similar IDE.

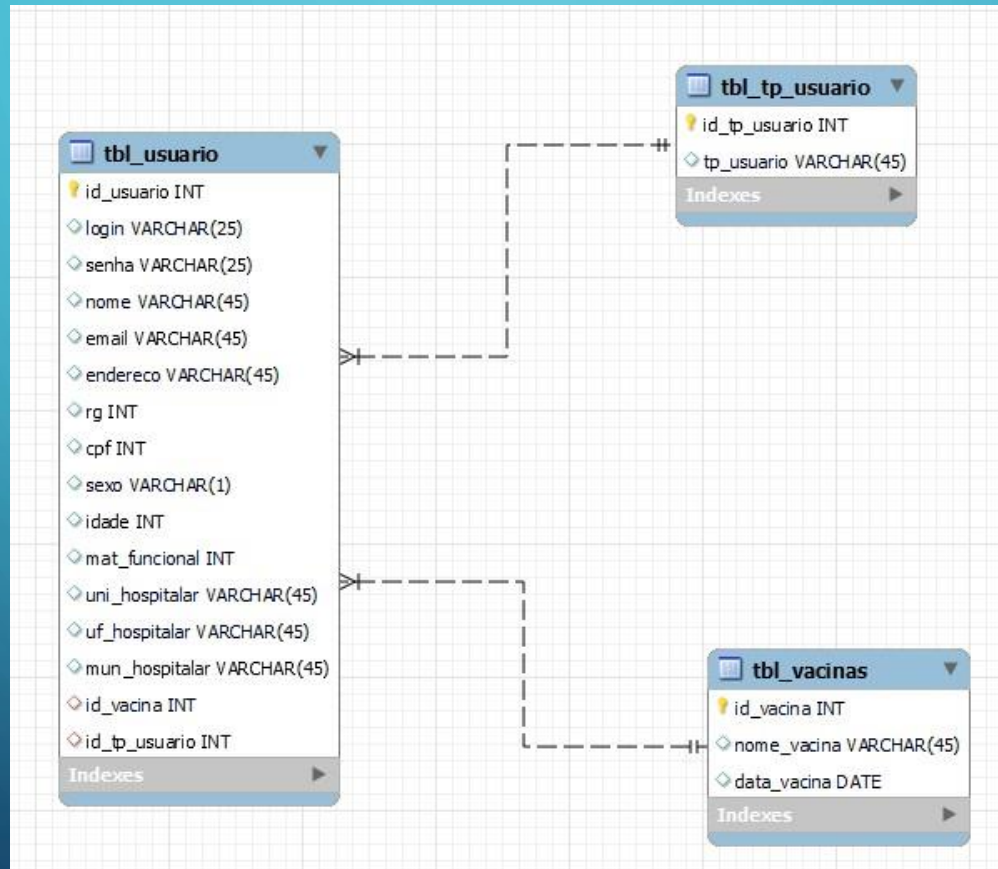
Left Screenshot (Navigator): Shows the 'Navigator' pane with the 'SCHEMAS' tab selected. A search bar 'Filter objects' is at the top. The tree view shows a hierarchy: 'login' (expanded) containing 'Tables' (tbl_tp_usuario, tbl_usuario, tbl_vacinas), 'Views', 'Stored Procedures', and 'Functions'. Below 'login' are 'sakila', 'sys', and 'world'. The 'Administration' tab is active, and the 'Schemas' sub-tab is selected. The 'Information' pane shows 'No object selected'. The 'Output' pane shows 'Action Output' with a table:

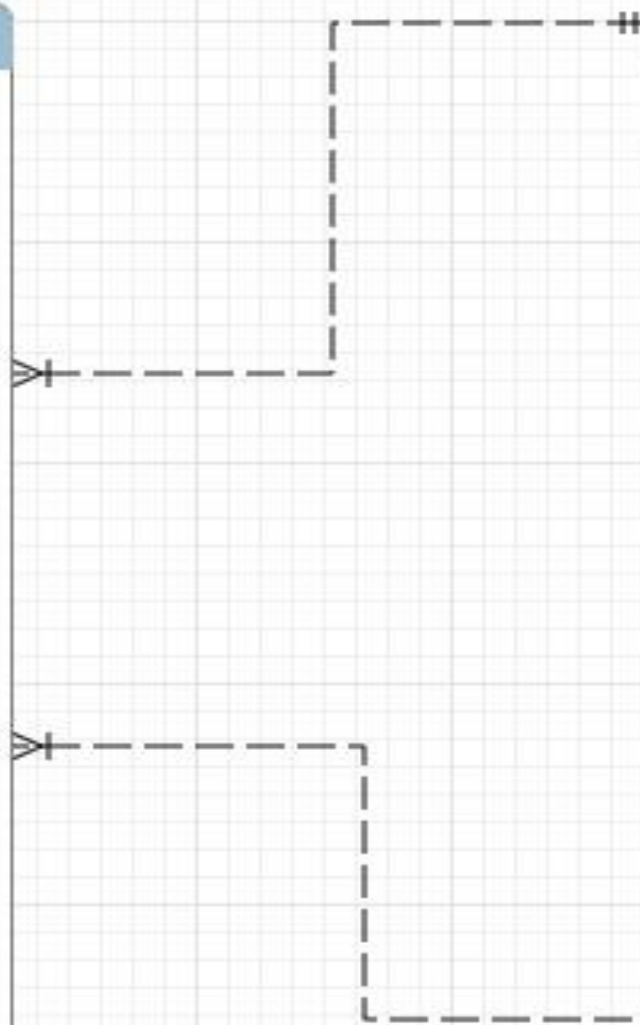
#	Time	Action
1	13:47:51	SELE

Middle Screenshot (SCHEMAS): A larger view of the 'SCHEMAS' pane. It features a search bar 'Filter objects'. The tree view shows: 'login' (expanded) containing 'Tables' (tbl_tp_usuario, tbl_usuario, tbl_vacinas), 'Views', 'Stored Procedures', and 'Functions'. Below 'login' are 'sakila', 'sys', and 'world'.

Right Screenshot (SQLAdditions): Shows the 'SQLAdditions' pane. It has a search bar 'Filter objects' and a 'Jump to' dropdown. A message states: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.' The 'Context Help' and 'Snippets' tabs are visible at the bottom. The 'Duration / Fetch' pane shows '0.000 sec / 0.000 se'.

BANCO DE DADOS - MYSQL





O APP... TELA DE LOGIN

Carteira de Vacina

Login

Senha

Status

Tipo de Usuario : Seleccione

Enviar Registro Sair

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor

	id_usuario	login	senha	nome	email	tipo_usuario	endereco	rg	cpf	sexo	idade	mat_funci
▶	1	Emanoel	12345	Emanoel Moraes	NULL	Profissional	NULL	NULL	NULL	NULL	NULL	NULL
	2	Denis	54321	Denis Oliveira	NULL	Usuario	NULL	NULL	NULL	NULL	NULL	NULL
	3	Wilames	98765	Wilames Santos	NULL	Profissional	NULL	NULL	NULL	NULL	NULL	NULL
	4	Mario	368800	Mario Santos	NULL	Usuario	NULL	NULL	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

tbl_usuario 1 x Apply Revert

○ APP – TELA DE LOGIN

```
1 package View;
2
3 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6 import javax.swing.JPanel;
7 import javax.swing.border.EmptyBorder;
8 import javax.swing.JLabel;
9 import javax.swing.JOptionPane;
10 import javax.swing.JTextField;
11 import javax.swing.JPasswordField;
12 import java.awt.Font;
13 import javax.swing.JButton;
14 import java.awt.Color;
15 import java.awt.event.ActionListener;
16 import java.awt.event.ActionEvent;
17
18 import java.sql.*;
19 import Dao.ModuloConexao;
20 import View.TelaRegistro;
21 import javax.swing.JComboBox;
22 import javax.swing.ComboBoxEditor;
23 import javax.swing.ComboBoxModel;
24 import javax.swing.DefaultComboBoxModel;
25
26 import DTO.TipoUsuarioDTO;
27 import View.TelaPrincipal;
```



```
1 package View;
2
3 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6 import javax.swing.JPanel;
7 import javax.swing.border.EmptyBorder;
8 import javax.swing.JLabel;
9 import javax.swing.JOptionPane;
10 import javax.swing.JTextField;
11 import javax.swing.JPasswordField;
12 import java.awt.Font;
13 import javax.swing.JButton;
14 import java.awt.Color;
15 import java.awt.event.ActionListener;
16 import java.awt.event.ActionEvent;
17
18 import java.sql.*;
19 import Dao.ModuloConexao;
20 import View.TelaRegistro;
21 import javax.swing.JComboBox;
22 import javax.swing.ComboBoxEditor;
23 import javax.swing.ComboBoxModel;
24 import javax.swing.DefaultComboBoxModel;
25
26 import DTO.TipoUsuarioDTO;
27 import View.TelaPrincipal;
28
```

```
28 public class TelaLogin extends JFrame {
29
30     Connection conexao = null;
31     PreparedStatement pst = null;
32     ResultSet rs = null;
33
34     private JComboBox cbxUsuario;
35
36     public void logar() {
37         String sql = "select * from tbl_usuario where login=? and senha=?";
38         try {
39             pst=conexao.prepareStatement(sql);
40             pst.setString(1,txtlogin.getText());
41             pst.setString(2,txtsenha.getText());
42
43             rs=pst.executeQuery();
44
45             if(rs.next()) {
46
47                 TelaPrincipal TLPrincipal = new TelaPrincipal();
48                 TLPrincipal.setVisible(true);
49                 this.dispose(); //fecha a tela login
50                 conexao.close(); //fecha a conexão
51             }else {
52                 JOptionPane.showMessageDialog(null, "Usuario ou Senha Invalido");
53             }
54         }
55         catch(Exception e) {
56             JOptionPane.showMessageDialog(null,e);
57         }
58     }
59 }
```



```
81  /**
82   * Create the frame.
83   */
84  public TelaLogin() {
85      setTitle("Carteira de Vacina");
86      setResizable(false);
87
88      conexao = ModuloConexao.conector();
89
90      setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
91      setBounds(100, 100, 380, 277);
92      contentPane = new JPanel();
93      contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
94
95      setContentPane(contentPane);
96      contentPane.setLayout(null);
97
98      JLabel lblNewLabel = new JLabel("Login");
99      lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 16));
100     lblNewLabel.setBounds(30, 23, 46, 14);
101     contentPane.add(lblNewLabel);
102
103     txtlogin = new JTextField();
104     txtlogin.setBounds(30, 47, 160, 30);
105     contentPane.add(txtlogin);
106     txtlogin.setColumns(10);
107
108     JLabel lblNewLabel_1 = new JLabel("Senha");
109     lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
116 JButton btnNewButton = new JButton("Enviar");
117 btnNewButton.addActionListener(new ActionListener() {
118     public void actionPerformed(ActionEvent e) {
119         logar();
120     }
121 });
122 btnNewButton.setForeground(Color.BLUE);
123 btnNewButton.setBounds(30, 205, 89, 23);
124 contentPane.add(btnNewButton);
```

```
127
128 if (conexao!=null) {
129     lblStatus.setText("Conectado");
130 }else {
131     lblStatus.setText("Não Conectado");
132 }
133
134
```

```
135
136 contentPane.add(btnNewButton_1);
```

```
137
138 JButton btnNewButton_2 = new JButton("Sair");
139 btnNewButton_2.addActionListener(new ActionListener() {
140     public void actionPerformed(ActionEvent e) {
141         System.exit(0);
142     }
143 });
```

○ APP... TELA REGISTRO

Tela de Registro

ID

Nome Completo

Login

Senha

Tipo de Usua... Selecione ▼

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id_usuario	login	senha	nome	email	tipo_usuario	endereco	rg	cpf	sexo	idade	mat_funcion
▶	1	Emanoel	12345	Emanoel Moraes	NULL	Profissional	NULL	NULL	NULL	NULL	NULL	NULL
	2	Denis	54321	Denis Oliveira	NULL	Usuario	NULL	NULL	NULL	NULL	NULL	NULL
	3	Wilames	98765	Wilames Santos	NULL	Profissional	NULL	NULL	NULL	NULL	NULL	NULL
	4	Mario	368800	Mario Santos	NULL	Usuario	NULL	NULL	NULL		NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

tbl_usuario 1 x

Apply Revert

Result Grid
Form Editor

```
114 JButton btnAdicionar = new JButton("Adicionar");
115 btnAdicionar.addActionListener(new ActionListener() {
116     public void actionPerformed(ActionEvent e) {
117         adicionar();
118     }
119 });
120 btnAdicionar.setBounds(10, 317, 89, 23);
121 contentPane.add(btnAdicionar);
122
123 JButton btnConsultar = new JButton("Consultar");
124 btnConsultar.addActionListener(new ActionListener() {
125     public void actionPerformed(ActionEvent e) {
126         consultar();
127     }
128 });
129 btnConsultar.setBounds(135, 317, 89, 23);
130 contentPane.add(btnConsultar);
131
132 JButton btnAlterar = new JButton("Alterar");
133 btnAlterar.addActionListener(new ActionListener() {
134     public void actionPerformed(ActionEvent e) {
135         alterar();
136     }
137 });
138 btnAlterar.setBounds(259, 317, 89, 23);
139 contentPane.add(btnAlterar);
140
141 JButton btnRemover = new JButton("Remover");
142 btnRemover.addActionListener(new ActionListener() {
```



```

165 private void consultar() {
166     String sql = "select * from tbl_usuario where id_usuario=?";
167     try {
168         pst=conexao.prepareStatement(sql);
169         pst.setString(1, txtRegId.getText());
170         rs=pst.executeQuery();
171         if (rs.next()) {
172             txtRegNome.setText(rs.getString(4));
173             txtRegLogin.setText(rs.getString(2));
174             txtRegSenha.setText(rs.getString(3));
175             //comboBox.setSelectedItem(rs.getString(6));
176         } else {
177             JOptionPane.showMessageDialog(null, "Usuario não Cadastrado");
178             txtRegNome.setText(null);
179             txtRegLogin.setText(null);
180             txtRegSenha.setText(null);

```

	id_usuario	login	senha	nome	email	tipo_usuario	endereco	rg	cpf	sexo	idade	mat_fundi
▶	1	Emanuel	12345	Emanuel Moraes	NULL	Profissional	NULL	NULL	NULL	NULL	NULL	NULL
	2	Denis	54321	Denis Oliveira	NULL	Usuario	NULL	NULL	NULL	NULL	NULL	NULL
	3	Wilames	98765	Wilames Santos	NULL	Profissional	NULL	NULL	NULL	NULL	NULL	NULL
	4	Mario	368800	Mario Santos	NULL	Usuario	NULL	NULL	NULL		NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

tbl_usuario 1 x

Apply Revert

```

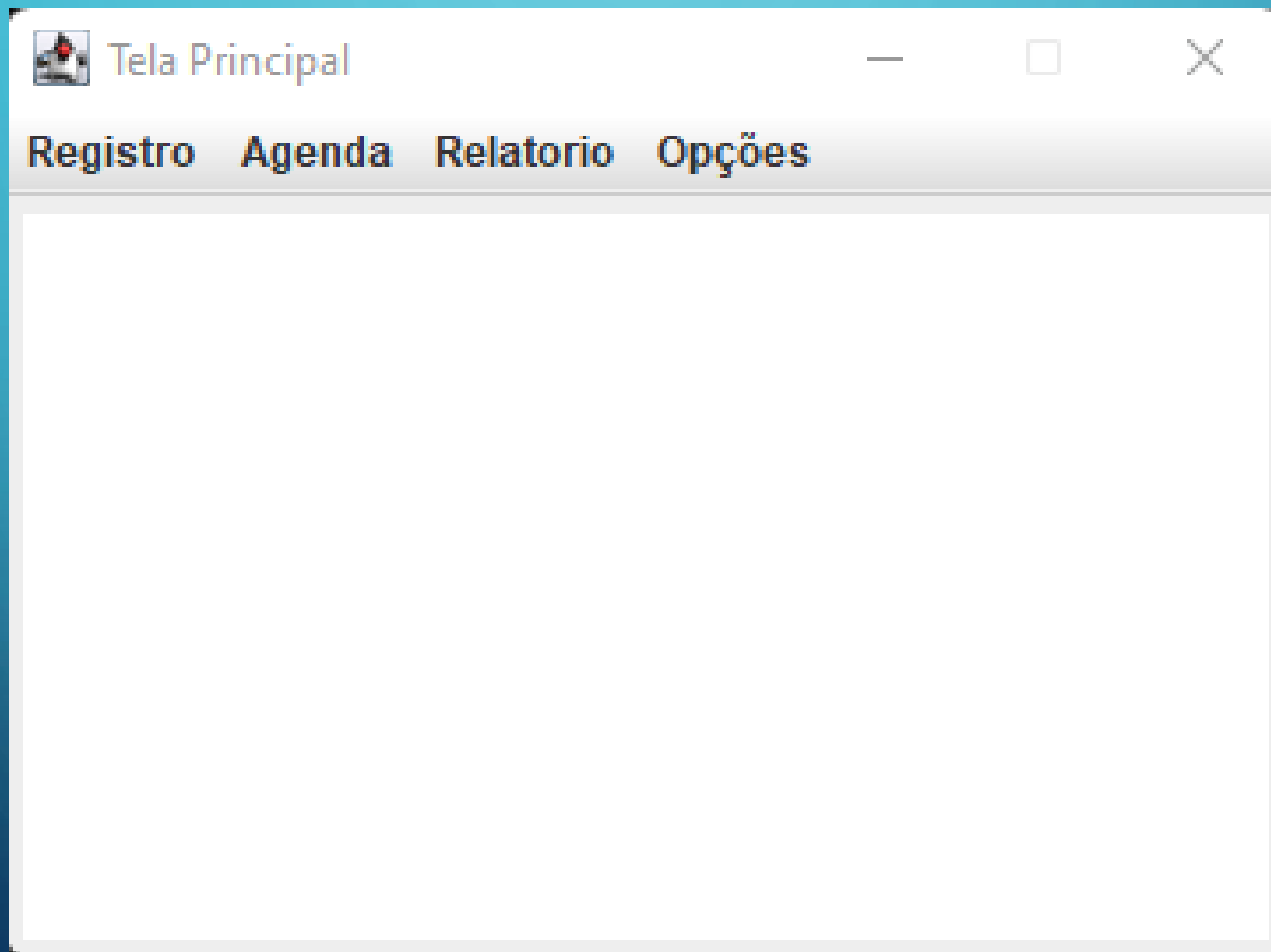
187 private void adicionar() {
188     String sql = "insert into tbl_usuario(id_usuario, nome, login, senha) values(?,?,?,?)";
189     try {
190         pst=conexao.prepareStatement(sql);
191         pst.setString(1, txtRegId.getText());
192         pst.setString(2, txtRegNome.getText());
193         pst.setString(3, txtRegLogin.getText());
194         pst.setString(4, txtRegSenha.getText());
195         //pst.setString(6, comboBox.getSelectedItem().toString());
196         int adicionado=pst.executeUpdate();
197         //System.out.println(adicionado);
198         if (adicionado > 0) {
199             JOptionPane.showMessageDialog(null, "Usuario Registrado com Sucesso");
200             txtRegId.setText(null);
201             txtRegLogin.setText(null);
202             txtRegNome.setText(null);
203             txtRegSenha.setText(null);
204         }
205     } catch (Exception e) {
206         JOptionPane.showMessageDialog(null, e);
207     }
208 }
209 private void alterar() {
210     String sql = "update tbl_usuario set login=?, nome=? , senha=? where id_usuario=?";
211     try {
212         pst=conexao.prepareStatement(sql);
213         pst.setString(1, txtRegLogin.getText());
214         pst.setString(2, txtRegNome.getText());
215         pst.setString(3, txtRegSenha.getText());
216         //pst.setString(6, comboBox. sql);
217         pst.setString(4, txtRegId.getText());
218         int alterado=pst.executeUpdate();
219         if (alterado > 0) {
220             JOptionPane.showMessageDialog(null, "Usuario Alterado com Sucesso");
221             txtRegId.setText(null);
222             txtRegLogin.setText(null);
223             txtRegNome.setText(null);
224             txtRegSenha.setText(null);
225         }
226     } catch (Exception e) {
227         JOptionPane.showMessageDialog(null, e);
228     }
229 }

```

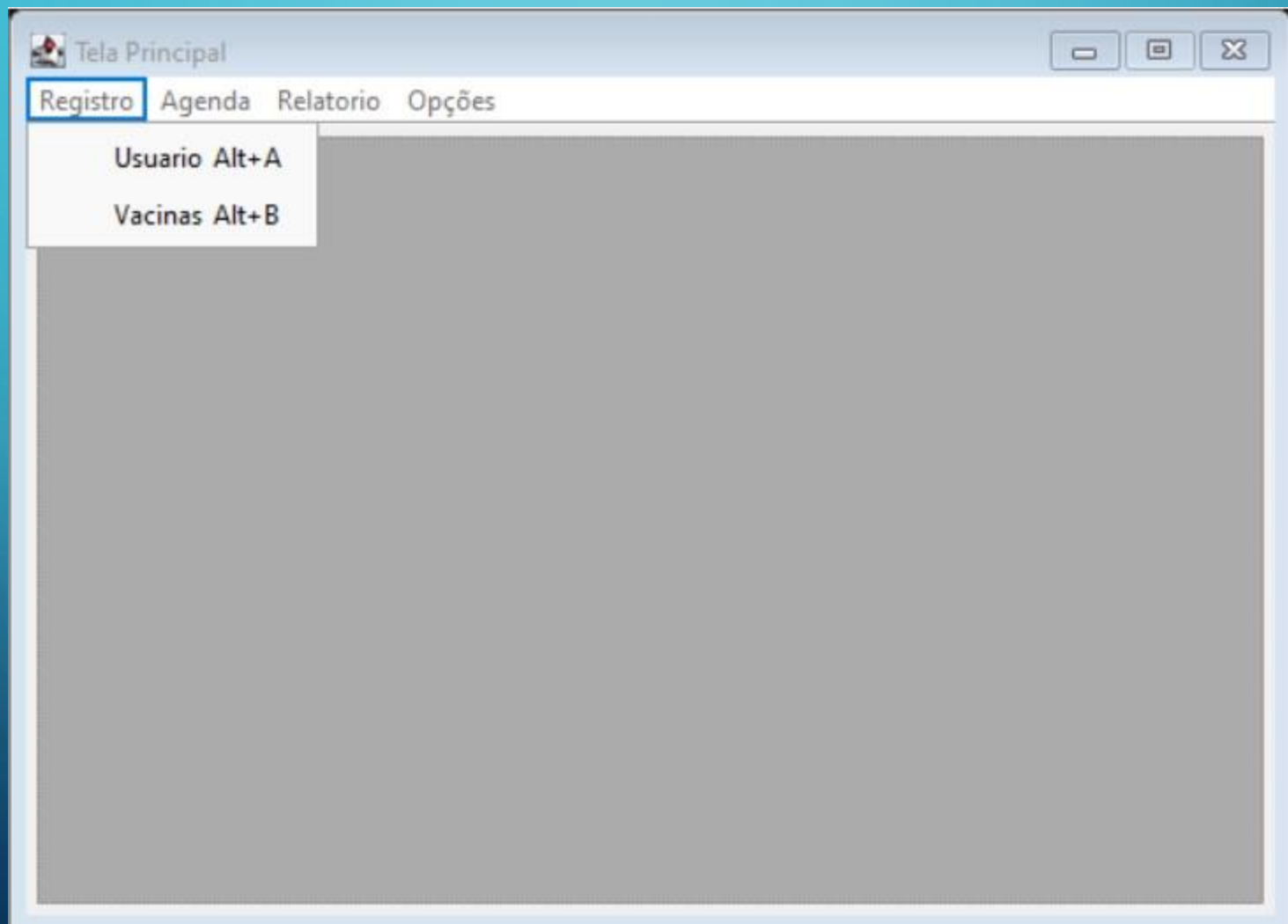
FINALIZAÇÃO DO CRUD

```
230 private void remover() {  
231     int confirma = JOptionPane.showConfirmDialog(null, "Tem Certeza que deseja Remover o usuario", "Atenção", JOptionPane.YES_NO_OPTION);  
232     if (confirma == JOptionPane.YES_OPTION) {  
233         String sql = "delete from tbl_usuario where id_usuario=?";  
234         try {  
235             pst = conexao.prepareStatement(sql);  
236             pst.setString(1, txtRegId.getText());  
237             int apagado = pst.executeUpdate();  
238             if (apagado > 0) {  
239                 JOptionPane.showMessageDialog(null, "Usuario Removido com Sucesso");  
240                 txtRegId.setText(null);  
241                 txtRegLogin.setText(null);  
242                 txtRegNome.setText(null);  
243                 txtRegSenha.setText(null);  
244             }  
245         } catch (Exception e) {  
246             JOptionPane.showMessageDialog(null, e);  
247         }  
248     }  
249 }
```

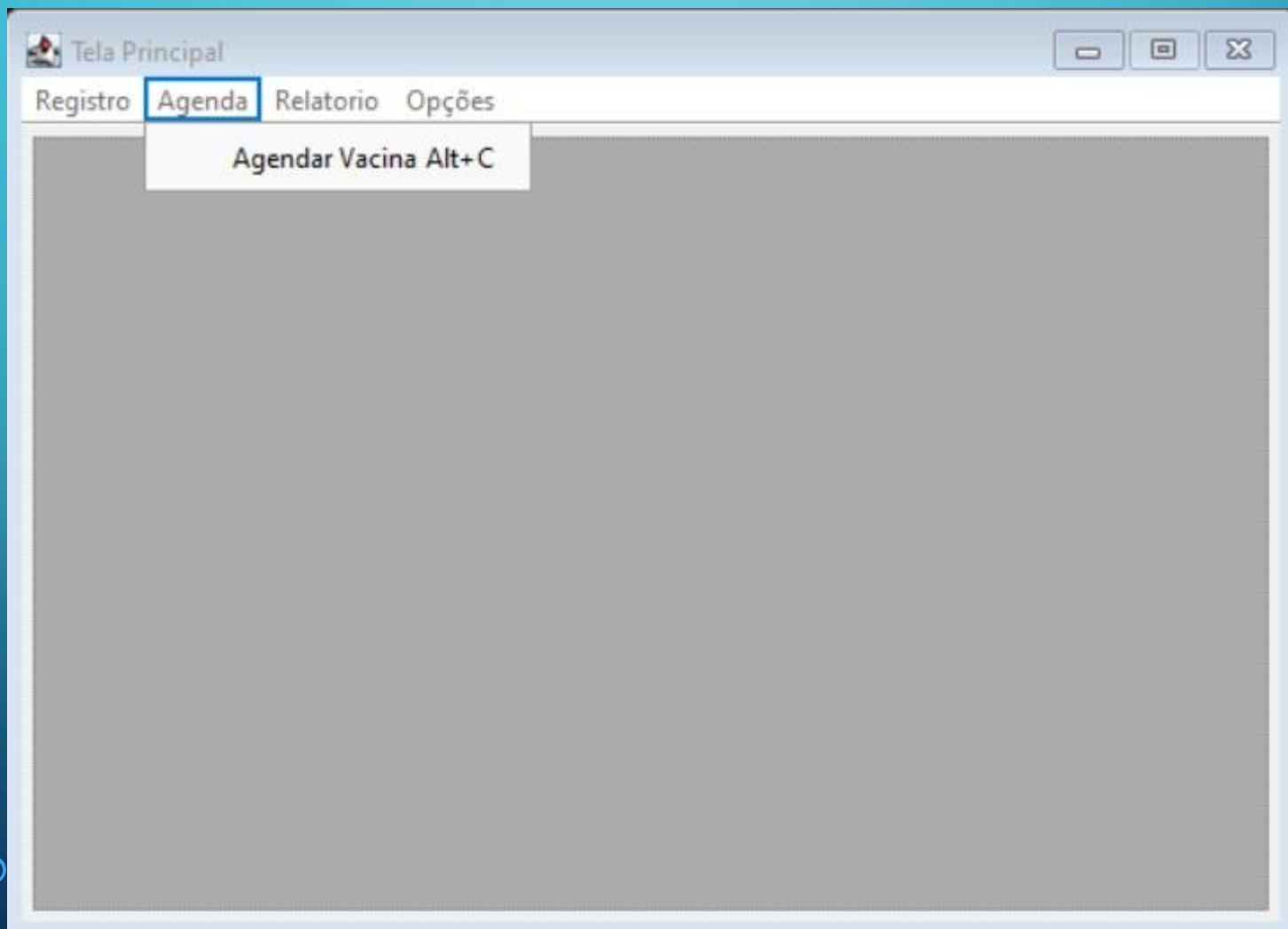
○ APP... TELA PRINCIPAL



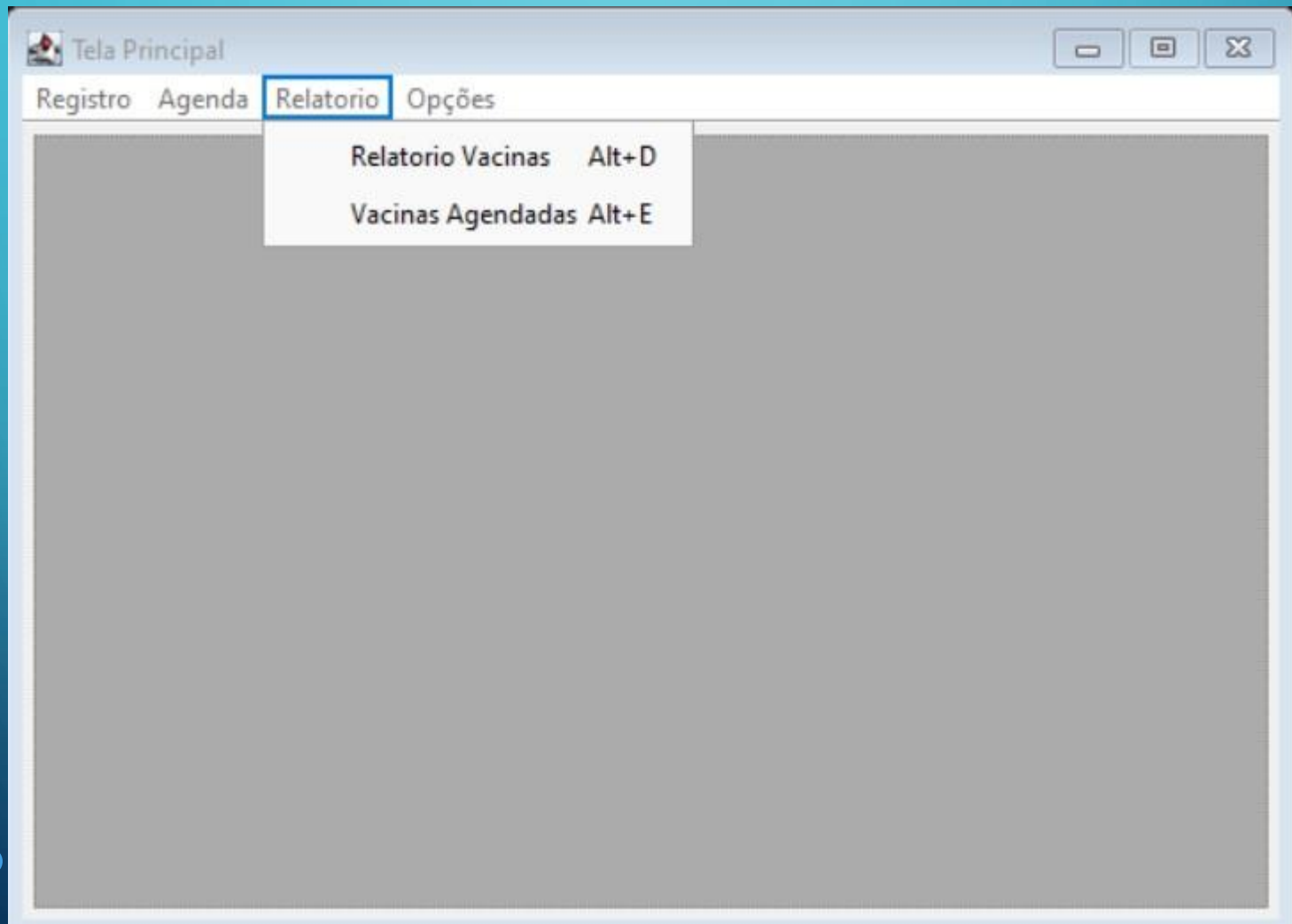
○ APP... TELA PRINCIPAL



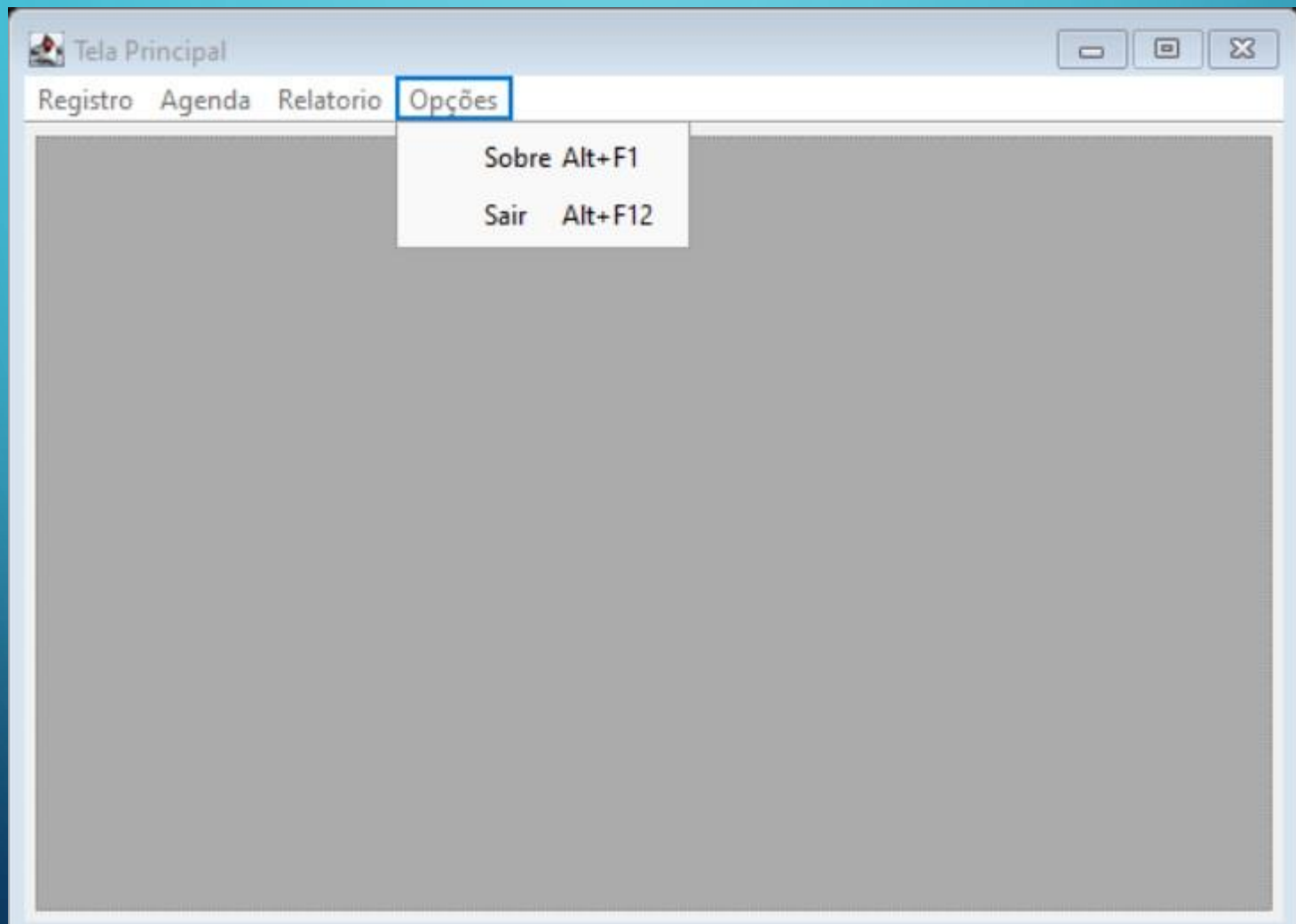
○ APP... TELA PRINCIPAL



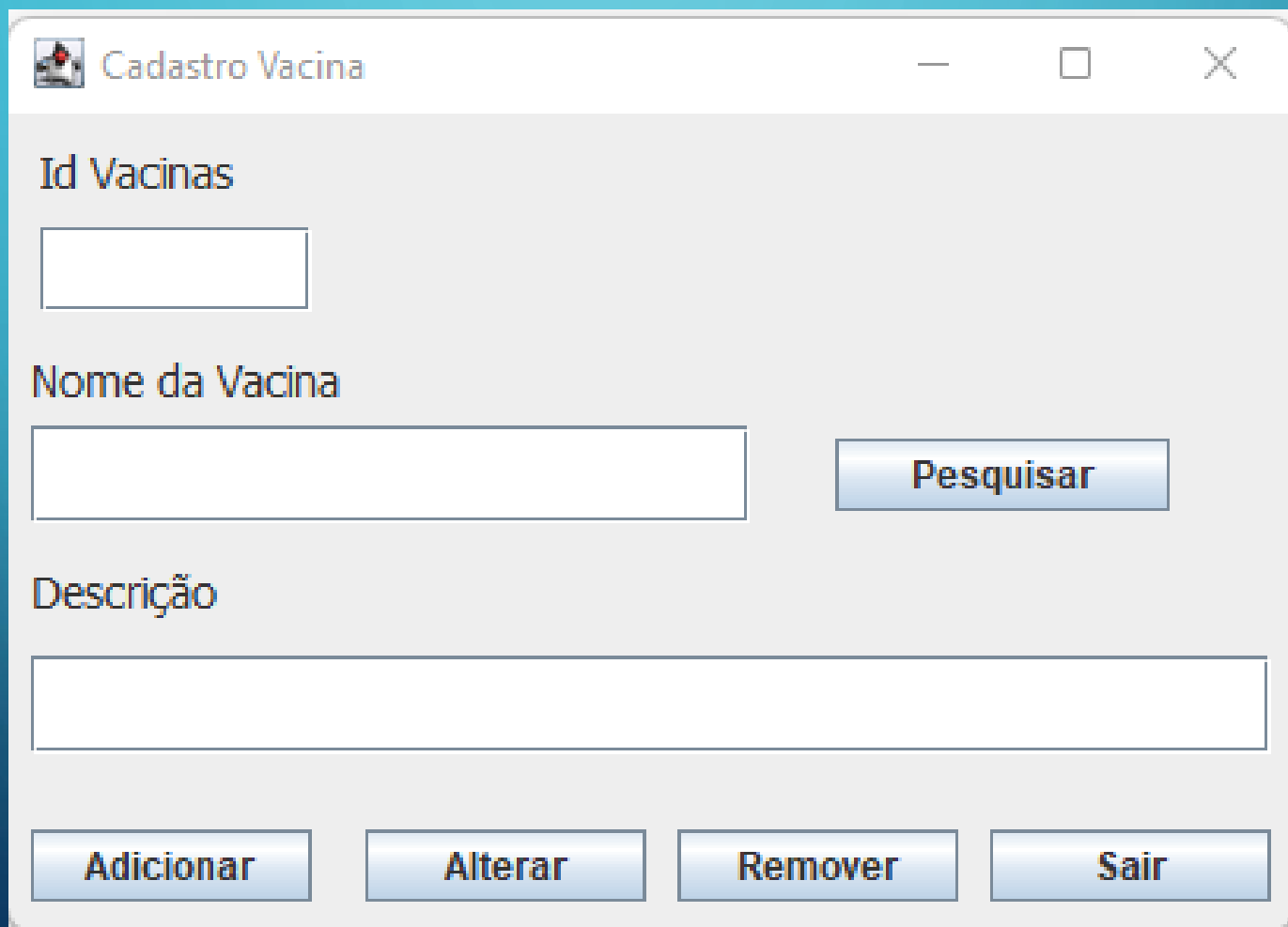
○ APP... TELA PRINCIPAL



○ APP... TELA PRINCIPAL



○ APP... TELA DE AGENDAMENTO



The image shows a screenshot of a software application window titled "Cadastro Vacina". The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area is light gray and contains three input fields and four buttons. The first input field is labeled "Id Vacinas" and is empty. The second input field is labeled "Nome da Vacina" and is also empty. To the right of the "Nome da Vacina" field is a button labeled "Pesquisar". Below these fields is a larger text area labeled "Descrição", which is empty. At the bottom of the window, there are four buttons: "Adicionar", "Alterar", "Remover", and "Sair".

Cadastro Vacina

Id Vacinas

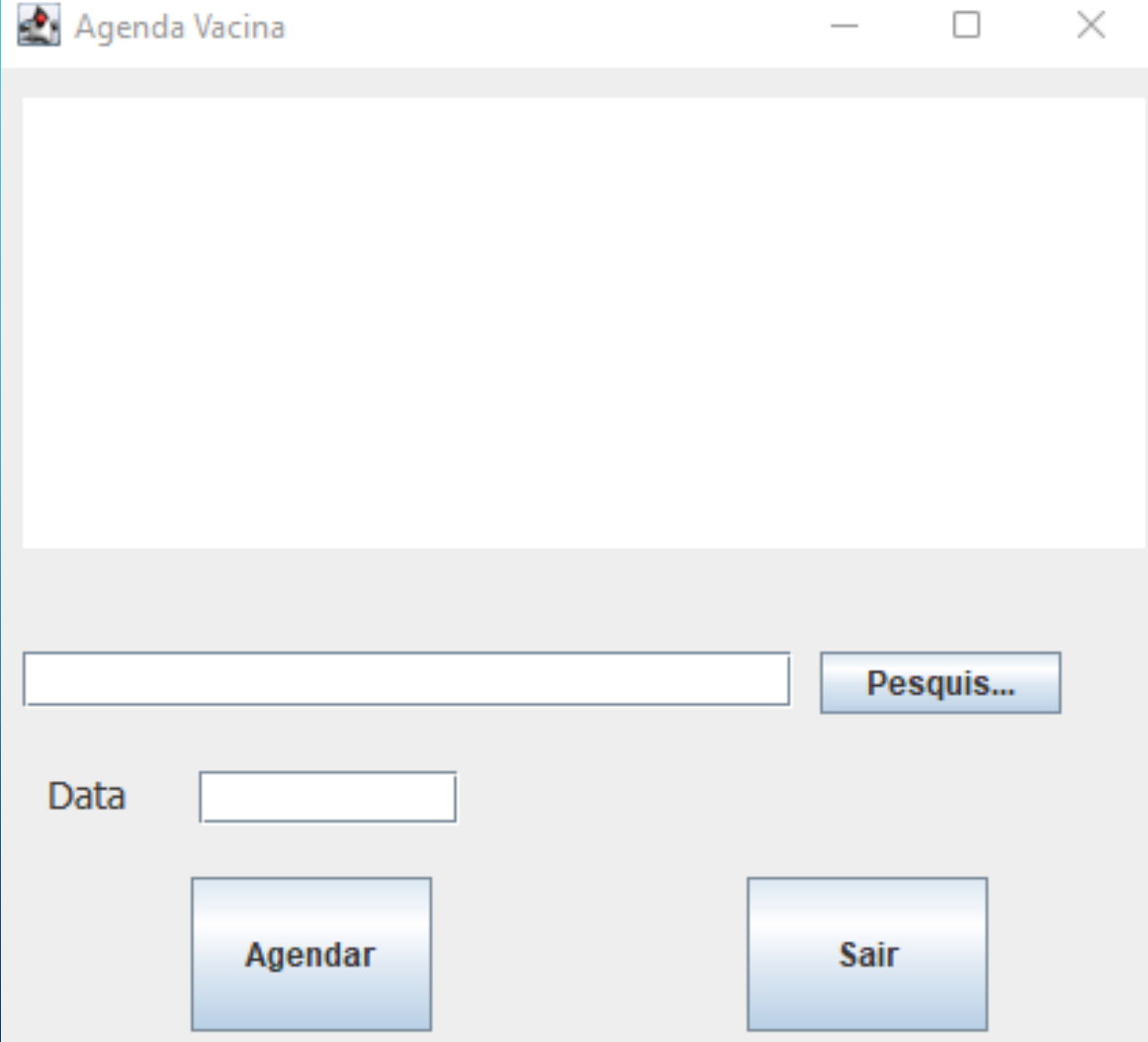
Nome da Vacina

Pesquisar

Descrição

Adicionar Alterar Remover Sair

O APP... TELA DE AGENDAMENTO



The image shows a screenshot of a software application window titled "Agenda Vacina". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main area of the window is a large, empty white rectangle, likely intended for a calendar or list of appointments. Below this area, there is a search bar with a "Pesquis..." button to its right. Further down, there is a label "Data" followed by a date input field. At the bottom of the window, there are two buttons: "Agendar" (Schedule) and "Sair" (Exit).

Agenda Vacina

Pesquis...

Data

Agendar

Sair

O APP... TELA SOBRE



The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing nodes.

OBRIGADO A TODOS

Agora vamos para Pratica...