



Atividade 07

Codificação do Aplicativo

GRUPO: PAUS

Líder

Lucas Alkimim - 239799

Integrantes

Enzo Fujimoto - 233930

Gustavo Alves - 236146

João Cabral - 237584

Henrique Arakaki - 236540

Marcos Gabriel Nocetti Santos - 240820

Mayara Alves - 260608

Vitor Roberto Kogawa de Moraes - 245582

Vinícius Ito - 225526

Vitoria Sue - 225743

Sumário

- **Descrição da atividade**
- **Descrição do que foi pesquisado e implementado**
- **Indicação de demissão**

Descrição da atividade

Nesta atividade, as equipes vão realizar a implementação do modelo Orientado a Objetos definido na aula anterior para o projeto CoronaApp.

➤ **Entregável:**

- O relatório com tudo que fora pesquisado e implementado (incluindo o diagrama de classes feito na aula anterior e eventualmente revisado pela equipe a partir dos meus comentários).

➤ **Níveis de Avaliação:**

- A avaliação se dará em vários níveis de conclusão do projeto dependendo do esforço dedicado pelas equipes.
- Mínimo: implementação das classes definidas no diagrama de classes da equipe para o projeto CoronaApp modelado na aula anterior. A implementação pode ser feita em C++ ou Java. Nesta versão mínima o sistema não precisa estar rodando, mas a definição das classes precisa estar de acordo com o modelo apresentado.
- Intermediário: As classes implementadas de acordo com o modelo e um programa principal funcionando para demonstrar que os requisitos são atendidos mesmo que parcialmente. Não há necessidade de implementar interface gráfica ou persistência de dados. Mas o programa precisa ser funcional, mesmo que parcialmente.
- Avançado: Os requisitos do projeto precisam estar implementados integralmente e pelo menos um item avançado precisa estar implementado, podendo ser a Interface Gráfica ou a Persistência de Dados. Nos dois casos as equipes devem pesquisar como esses itens avançados são implementados na linguagem escolhida e colocar o que foi pesquisado no relatório.

Descrição do que foi pesquisado e implementado

- **O que foi pesquisado?**

- Interface Gráfica

- ➔ Como implementar a interface gráfica em Java:

- <https://documentcloud.adobe.com/link/track?uri=urn%3Aaaid%3Asc%3AUS%3Aa284deb4-5f53-4a67-b60d-b244ac560b86>

Algumas dúvidas que surgiram durante a implementação:

- ➔ Como criar a tela de cadastro:

- <https://www.youtube.com/watch?v=eTEDBVcpf50>

- ➔ Como definir o JComboBox dinamicamente e obter valor do item:

- <https://www.youtube.com/watch?v=2JbrqY2EP24>

- ➔ Método de preencher JComboBox em Java:

- <https://www.youtube.com/watch?v=yxpgplPTIPY>

- ➔ Como carregar as cidades a partir do estado selecionado:

- <https://www.youtube.com/watch?v=A-KGJXax71Y&t=219s>

- <https://www.guj.com.br/t/carregar-cidades-apartir-do-estado-selecionado/43184/4>

- <https://pt.stackoverflow.com/questions/270290/ajuda-com-combox-selecionar-cidades-de-acordo-com-estado-escolhido>

- ➔ Como usar o debug no netbeans:

- <https://www.youtube.com/watch?v=qIMJytCpKJg>

- ➔ Evento MouseClicked para preencher JTextField em Java:

- <https://www.youtube.com/watch?v=3rtGGCJvRcl>

- ➔ Como mudar a cor da barra de progresso - Java NetBeans:

- <https://www.youtube.com/watch?v=VylkyBCu8uo>

- ➔ Como mudar a cor do JTextField e do Texto - Validando os campos:

- <https://www.youtube.com/watch?v=7sMkOdbHbe8>

- ➔ Como filtrar linhas em uma JTable:

- <https://java.docow.com/8792/como-possso-filtrar-linhas-em-uma-jtable.html>

- <https://www.guj.com.br/t/filtro-em-jtable-resolvido/301520>

- ➔ Tabela não mostra o título da coluna:

- <https://www.guj.com.br/t/resolvido-tabela-nao-mostra-o-titulo-da-coluna-so-mostra-as-linhas/83985>

- Como fazer uma JLabel mudar de cor quando passado o mouse em cima:
<https://www.guj.com.br/t/como-fazer-uma-jlabel-mudar-de-cor-quando-passa-o-mouse/235115/3>
 - Como aplicar o estilo sublinhado em um texto dentro de um JTextField:
<https://www.guj.com.br/t/texto-sublinhado-num-jtextfield/24597>
 - Como selecionar item do método setSelectedItem(x) do JComboBox:
<https://www.guj.com.br/t/jcombobox-setselecteditem-obj/223240>
 - Como encher um JComboBox a partir de um ResultSet:
<https://www.guj.com.br/t/defaultcomboboxmodel/53220>
 - Como adicionar itens na Jlist:
<https://www.guj.com.br/t/resolvido-adicionando-itens-na-jlist/234629>
<https://pt.stackoverflow.com/questions/263882/adicionar-elemento-em-jlist>
<https://pt.stackoverflow.com/questions/263882/adicionar-elemento-em-jlist>
<https://stackoverflow.com/questions/8176965/how-to-add-element-to-existing-jlist>
 - Usar o Set para um valor no JTextField:
<https://www.guj.com.br/t/setar-valor-no-jtextfield/382162>
 - Como passar um vetor como parâmetro para um método:
<https://www.guj.com.br/t/como-passar-um-vetor-como-parametro-para-um-metodo/65227>
 - Como adicionar um elemento dentro de um array:
<https://www.guj.com.br/t/adicionando-elemento-dentro-de-um-array/91360/5>
 - Diferenças entre execute(), executeUpdate() e executeQuery() em um PreparedStatement:
<https://www.guj.com.br/t/diferencas-entre-execute-executeupdate-e-executequery-em-um-preparedstatement/76048>
 - Como Popular uma JTable com um Array List de Objetos:
<https://respostas.guj.com.br/9696-como-popular-uma-jtable-com-um-array-list-de-objetos>
- Persistência de Dados
- O que é Persistência de Dados:
<https://pt.stackoverflow.com/questions/140411/o-que-%C3%A9-persist%C3%Aancia-de-dados>
 - Persistindo dados em JAVA:
<https://www.devmedia.com.br/persistindo-dados-em-java-com-jpa/29797>

Algumas dúvidas que surgiram durante a implementação:

- O que é um SGBD:
<https://www.oficinadanet.com.br/post/16631-o-que-e-um-sgbd>
- O que é SQL e para que ele serve:
<https://www.tecmundo.com.br/software/146482-sql-que-ele-serve.htm>
- Trazer data do banco no formato dd/MM/yyyy:
<https://www.guj.com.br/t/trazer-data-do-banco-no-formato-dd-mm-yyyy/70188>
- Retornar dados do BD usando List/ArrayList:
<https://pt.stackoverflow.com/questions/275274/retornar-dados-do-bd-usando-list-arraylist-java>
- converter data no formato dd/mm/yyyy para yyyy-mm-dd:
<https://pt.stackoverflow.com/questions/58850/converter-data-no-formato-dd-mm-yyyy-para-yyyy-mm-dd>
- Validação tela login com o banco de dados:
https://www.youtube.com/watch?v=uzc7M_gIS5A
- Banco de Dados Access:
<https://www.youtube.com/watch?v=l89Jlf8L-SU>
- Carregando JTable com registro de BD usando ArrayList:
https://www.youtube.com/watch?v=_XtR56iXRdM&list=PLJOO6wuBlgaUuaEbZvTjmwVAokMmeGqQr&index=15
- JAVA com BD Select:
<https://www.youtube.com/watch?v=FeIAIYgiiT8>
- Carregando JTable com registro de BD:
<https://www.youtube.com/watch?v=faPR87RX2c0>
- MySql para SQL Server - Mantendo padrão DBO:
https://www.youtube.com/watch?v=_phkptSCUNM

• Diagrama de classes antigo

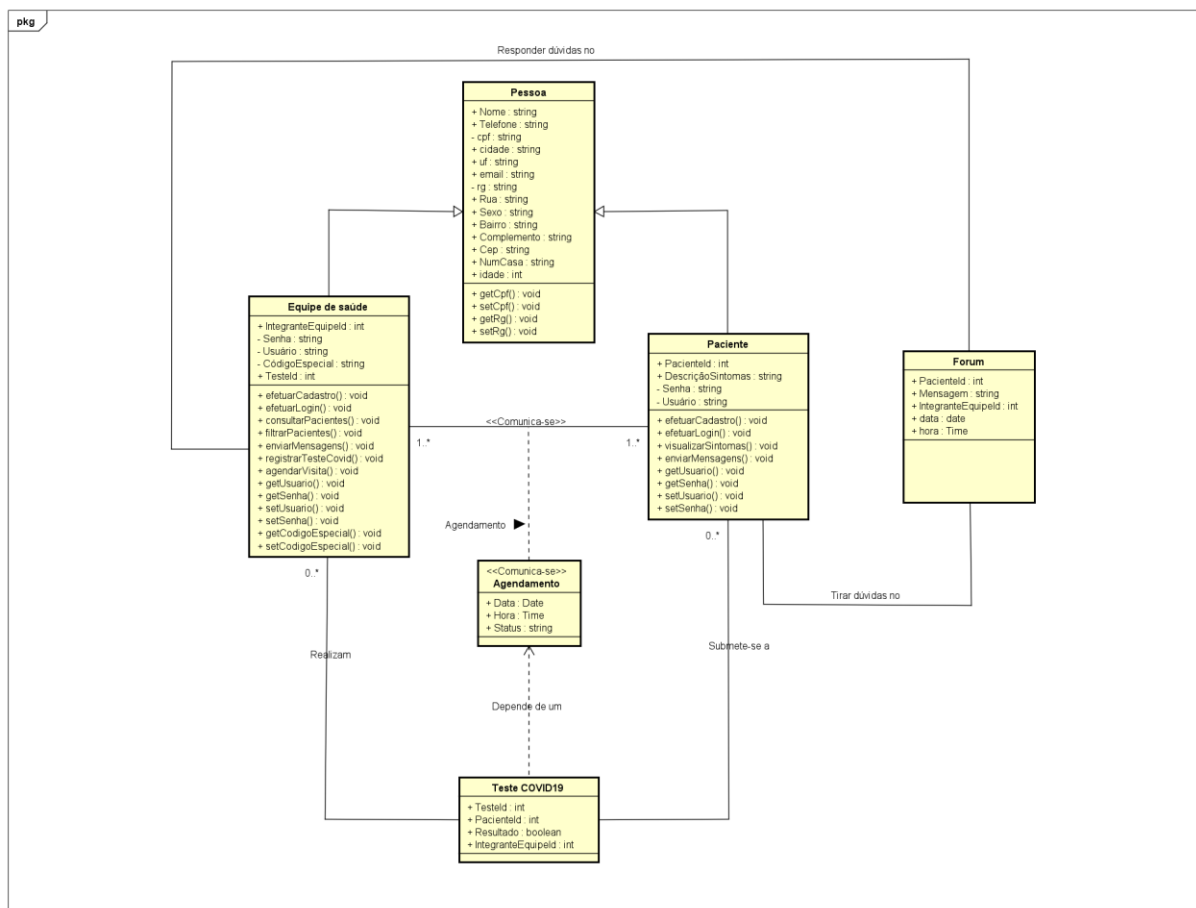


Figura 01 - Diagrama de classes

- Pessoa: Esta classe representa uma classe base para as classes derivadas que são “Equipe de saúde” e “Paciente”. O que ocorre é uma associação de generalização ou herança em que as classes “Equipe de Saúde” e “Paciente” contém todas as características e comportamentos que são compartilhados da classe Pessoa. Observe que a classe pessoa possui os atributos genéricos que todo indivíduo possui.

- Paciente: Classe contendo atributos e métodos, como por exemplo o código do paciente (int), descrição dos sintomas (string), usuário (string) e senha (string). Observe também que a classe herda todos os atributos compartilhados pela classe “Pessoa”.

Além disso, na classe temos métodos para efetuar cadastro, login, visualização de sintomas e envio de mensagens no aplicativo. No método efetuarCadastro() se preencherá todos os atributos herdados da classe “Pessoa” e os específicos da classe “Paciente”, como “descrição dos sintomas” (string), “usuario” (string) e “senha” (string). Este aspecto é demonstrado na associação entre ambas as classes.

Outrossim, o método efetuarLogin() é autoexplicativo e tem como finalidade o usuário se logar no aplicativo. No que diz respeito ao método visualizarSintomas(), temos também uma associação autoexplicativa cujo o objetivo é a visualização dos sintomas do paciente, sintomas estes que foram descritos no método efetuarCadastro(). Por fim, o método enviarMensagens() tem como objetivo o envio de mensagens do paciente a um fórum (aba da interface).

- Equipe de Saúde: Classe contendo atributos e métodos, como o código do integrante da equipe de saúde (int), usuario (string) e senha (string) que são para o login da equipe de saúde. Além de possuir métodos com o objetivo de consultar pacientes, filtrar pacientes, registrar os testes dos pacientes e registrar agendamentos de visitas no aplicativo. Observe também que a classe herda todos os atributos compartilhados pela classe “Pessoa”.

O método efetuarCadastro() tem como objetivo o cadastro dos membros da equipe de saúde, incluindo no mesmo os atributos “herdados” da classe Pessoa. Observe que este aspecto é demonstrado nas associações entre ambas as classes (“Pessoa” e “Equipe de saúde”). Já no que diz respeito ao método efetuarLogin(), perceberemos que ele é autoexplicativo e tem como finalidade o usuário integrante da equipe de saúde se logar no aplicativo. O próximo método que será analisado é o consultarPaciente() que tem como objetivo a consulta dos pacientes cadastrados na aplicação. Veja que, o método filtrarPacientes() oferece uma extensão do

consultar pacientes e tem como objetivo filtrar os pacientes por segmentos e parâmetros.

Como últimos métodos que analisaremos da classe, começaremos com o método `enviarMensagens()`, que tem como objetivo o envio de mensagens de um integrante da equipe de saúde a uma aba/ fórum. Outro método que temos é o `agendarVisita()`, que tem como objetivo o registro do agendamento da visita aos pacientes no aplicativo. Por fim, o último que analisaremos é o método `registrarTesteCovid()`, que possui um nome autoexplicativo e nele se registram todos os testes de COVID-19.

- Agendamento: Classe responsável por receber os dados do agendamento da consulta, como Data e Hora. Também possui o Status da consulta, que varia entre realizada, cancelada ou pendente. Observe que , a classe está associada com o método `agendarVisita()` da classe “Equipe de saúde”. Observe que esta é uma classe associativa , pois é gerada através de uma comunicação entre as classes “Equipe de Saúde “ e “Paciente”.
- Teste COVID-19: Classe responsável por receber os dados, como código do paciente cujo realizou o teste do COVID-19. Recebe também os dados do membro da equipe de saúde que registrou o teste. Além disso, a classe possui o atributo boolean “Resultado” no qual será guardado “true” caso dê positivo para a doença COVID-19 e “false” para negativo do mesmo. Observe que esta classe depende da classe “Agendamento”, pois sem um agendamento não se tem uma visita, e sem uma visita não se tem um teste, e que sem um teste não se tem um registro do mesmo no aplicativo.
- Fórum: Classe relacionada a comunicação entre pacientes e integrantes da equipe de saúde , através de mensagens que serão postadas em uma aba do aplicativo, como por exemplo, dúvidas/respostas sobre o COVID-19. Nesta classe, pega-se o código do paciente, o código do membro da equipe de saúde, e recebe-se a mensagem e seus respectivos dados, como a data e hora. Observe que a classe está associada às classes “Equipe de saúde” e “Paciente” e o respectivo método de ambas `enviarMensagens()`.
- Comentários gerais do professor:

O diagrama de classes parece estar ok, no entanto não entendi como os sintomas apresentados pelo paciente ficarão registrados. De qualquer forma a modelagem pode ter que ser atualizada à medida que você tentarem fazer a implementação. Essa experiência de modelar só se adquire com a prática. Vamos fazer isso na atividade seguinte...

• Diagrama de classes revisado

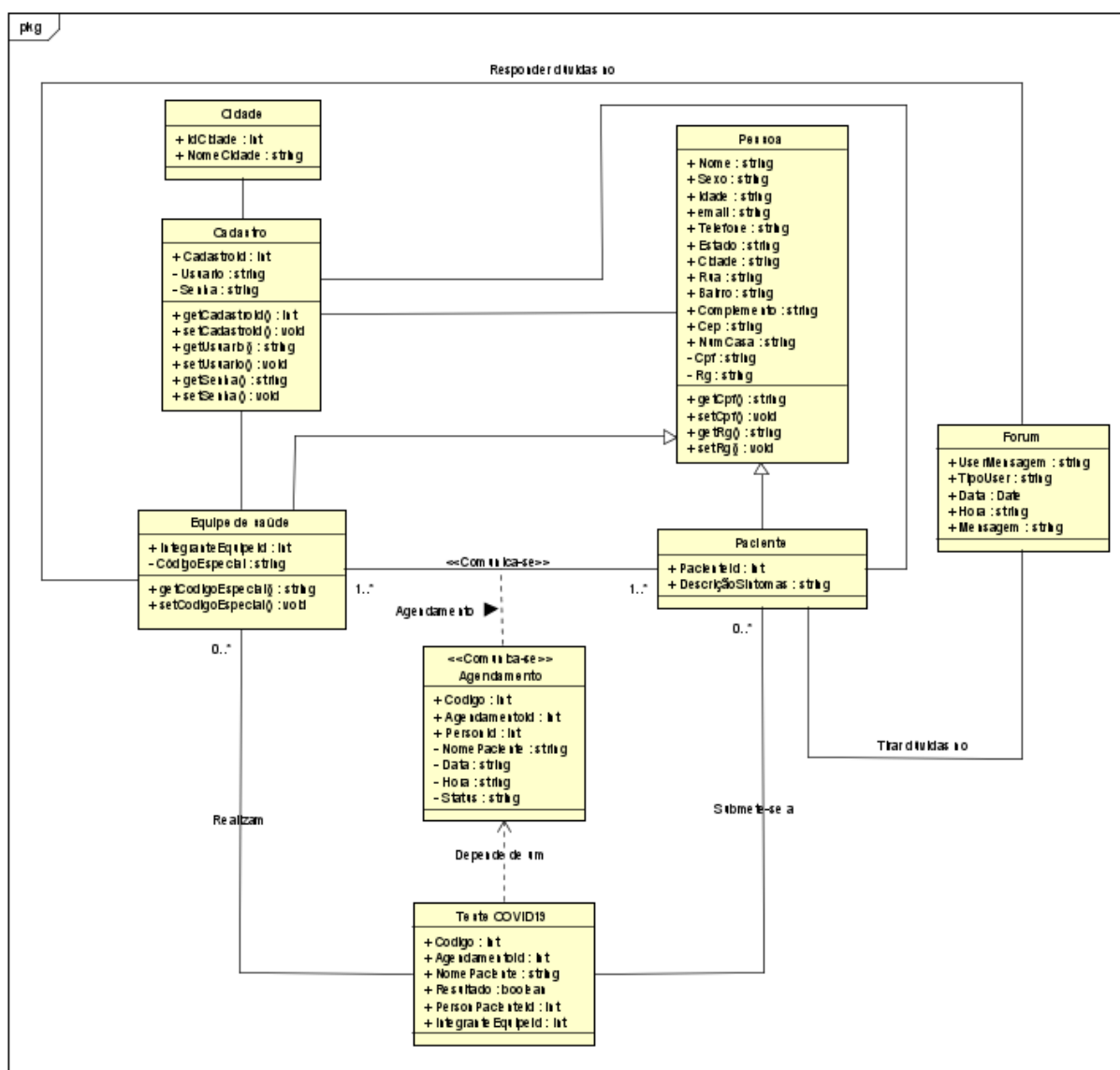


Figura 02 - Diagrama de classes (revisado)

Houveram alterações no Diagrama de classes, foram adicionadas novas classes e atributos. Além de uma reorganização dos atributos em suas

respectivas classes. As classes adicionadas foram “Cidade” e “Cadastro”, com seus respectivos atributos e métodos.

- Teste COVID: Classe responsável por receber os dados, como código do agendamento “Agendamentold” (int) que serve para identificar com mais facilidade o agendamento. Além disso recebe a string “nomePaciente”, que auxilia na identificação do paciente. Recebe “resultado”(boolean) no qual será guardado “true” caso dê positivo para a doença COVID-19 e “false” para negativo do mesmo. Além disso, são recebidos o Id do Paciente “personPacienteld” e o Id do integrante do Equipe da Saúde que realizou o teste “integranteEquipeld”.
- Agendamento: Classe responsável por receber os dados do agendamento da consulta, como Data e Hora (string). Também possui o Status da consulta (string), que varia entre realizada, cancelada ou pendente. Além disso, recebe o código do agendamento “Agendamentold” (int) que serve para identificar com mais facilidade o agendamento. Recebe o nome do usuário que está logado através da string “usuarioLogado”. Por fim, recebe a identificação da pessoa através da “personId” (int).
- Pessoa: Esta classe representa uma classe base para as classes derivadas que são “Equipe de saúde” e “Paciente”. O que ocorre é uma associação de generalização ou herança em que as classes “Equipe de Saúde” e “Paciente” contém todas as características e comportamentos que são compartilhados da classe Pessoa. Observe que a classe pessoa possui os atributos genéricos que todo indivíduo possui.
- Paciente: Classe contendo o código do paciente “pacienteld” (int) e a descrição dos sintomas “descricaoSintomas” (string). Observe também que a classe herda todos os atributos compartilhados pela classe “Pessoa”.
- Equipe de Saúde: Classe contendo atributos e métodos, como o código do integrante da equipe de saúde (int), para a identificação do membro da equipe da saúde. Além disso contém o código especial (string). que serve para que seja salvo no banco de dados.

- Fórum: Classe relacionada a comunicação entre pacientes e integrantes da equipe de saúde , através de mensagens que serão postadas em uma aba do aplicativo, como por exemplo, dúvidas/respostas sobre o COVID-19. Nesta classe, pega-se o código do paciente, o código do membro da equipe de saúde, e recebe-se a mensagem e seus respectivos dados, como a data e hora. Observe que a classe está associada às classes “Equipe de saúde” e “Paciente”.
- Cadastro: Classe que recebe alguns dados do cadastro. Recebe a identificação do cadastro “IdCadastro” (int), o usuário (string) e a senha (string) e os armazena para que posteriormente esses dados sejam registrados no banco de dados para posterior acesso.
- Cidade: Classe que recebe os dados da cidade do cadastrado. Recebe o código de identificação da cidade “IdCidade” (int) e o nome da cidade “NomeCidade” (string).

- **Implementação**

- ❑ **Interface gráfica**

Os estudantes da Faculdade de Tecnologia da Universidade Estadual de Campinas realizaram um projeto na disciplina de Engenharia de Software II, que tem como objetivo a conscientização e gerenciamento da população em meio a esta pandemia do Coronavírus. Foi gerado um aplicativo que possui as interfaces gráficas demonstradas abaixo:



Figura 03 - Interface

- **Breve descrição :** Ao acessar o aplicativo, o usuário observa a apresentação do ambiente inicial que possui o símbolo do Coronavírus, com a opção de entrada.

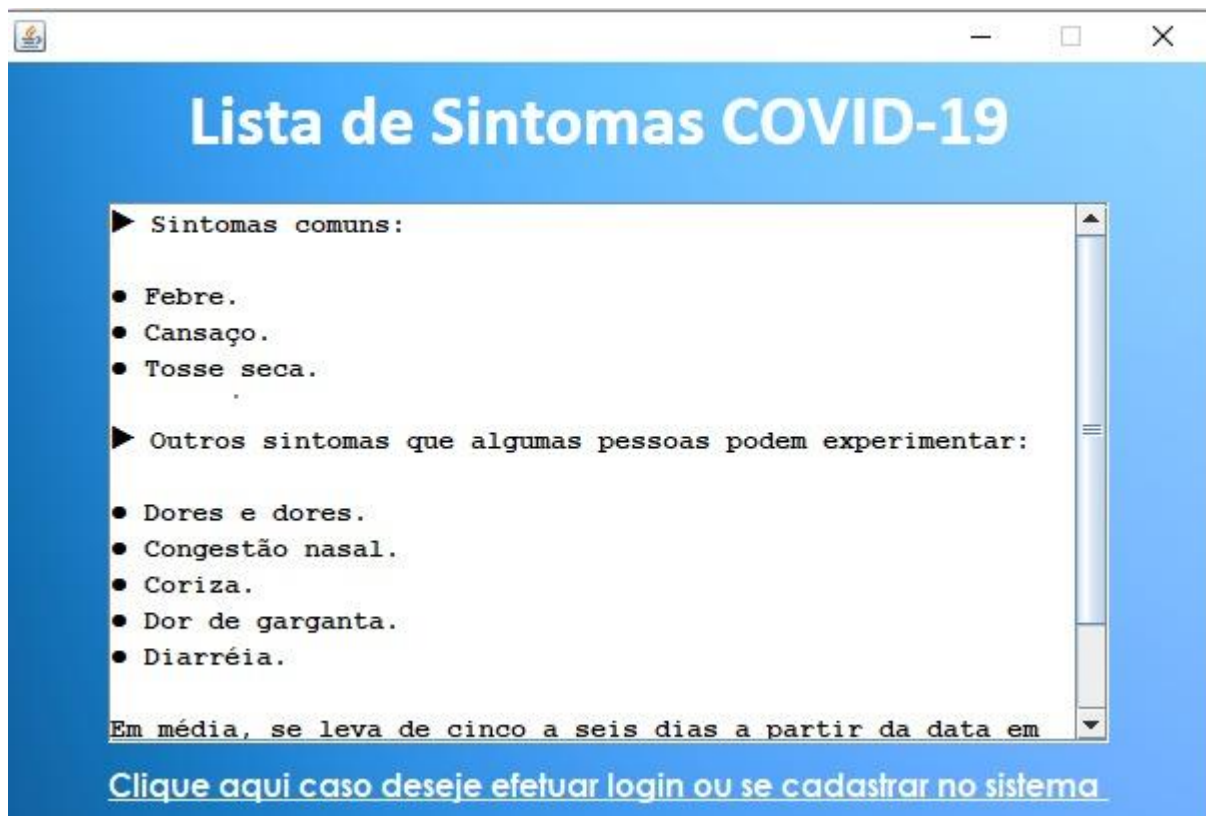
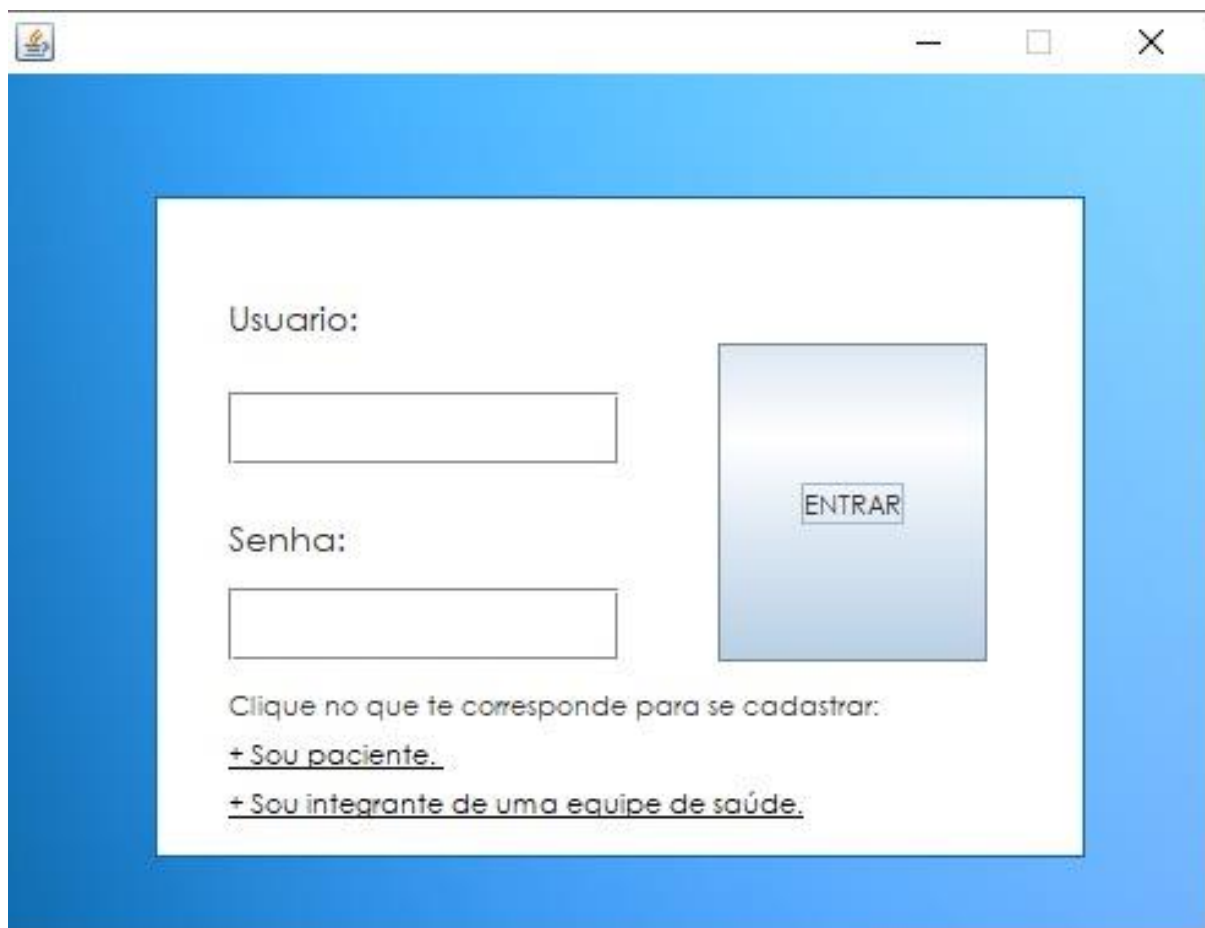


Figura 04 - Interface

- **Breve descrição :** Nesta tela, é possível visualizar os sintomas da COVID-19. Caso o usuário apresente alguns dos sintomas, ele prossegue clicando em "Clique aqui caso deseje efetuar login ou se cadastrar no sistema". Caso o usuário seja um membro da equipe de saúde, não há necessidade de verificar se há adequação com os sintomas, realizando o mesmo procedimento de clique anteriormente apresentado.



The image shows a login window with a blue border. Inside, there is a white rectangular area containing the following elements:

- A label "Usuario:" followed by a text input field.
- A label "Senha:" followed by a text input field.
- A large rectangular button with a blue gradient and the text "ENTRAR" in a small box in the center.
- Text: "Clique no que te corresponde para se cadastrar:"
- Two links: [+ Sou paciente.](#) and [+ Sou integrante de uma equipe de saúde.](#)

Figura 05 - Interface

- **Breve descrição :** Já nesta outra interface, o usuário se depara com a possibilidade de realizar o Login, caso já tenha sua identificação cadastrada. Pode também registrar-se, sendo algum novo Paciente ou algum integrante da Equipe Médica.

CADASTRO PACIENTE

Nome completo :

Sexo :

Idade :

Telefone :

Email :

Estado :

Cidade :

Rua :

Bairro :

Número da casa :

Complemento :

CEP :

CPF :

RG :

Descrição sintomas :

Agora escreva os dados que serão usados para seu acesso ao sistema:

Nome de Usuário:

Senha:

CADASTRAR

Figura 06 - Interface

- **Breve descrição :** Nesta tela, o usuário que se identificou como paciente deve preencher seus dados conforme os requisitos do sistema para que possa ser realizada a visita e para que esse possa acessar as funcionalidades do sistema.



The image shows a web application window titled "CADASTRO INTEGRANTE EQUIPE DE SAÚDE". The form is divided into two main sections. The left section contains personal and contact information fields: "Nome completo:", "Sexo:" (a dropdown menu), "Idade:", "Telefone:", "Email:", "Estado:" (a dropdown menu), "Cidade:" (a dropdown menu), "Rua:", "Bairro:", "Número da casa:", "Complemento:", "CEP:", "CPF:", and "RG:". The right section contains a "Código Especial:" field, followed by the instruction "Agora escreva os dados que serão usados para seu acesso ao sistema:", then "Nome de Usuário:" and "Senha:" fields. At the bottom center of the form is a button labeled "CADASTRAR".

Figura 07 - Interface

- **Breve descrição :** Caso o usuário opte por se cadastrar na Equipe de Saúde, aparecerá a interface supracitada, que lhe permite concretizar a inscrição. Dessa maneira, o sistema solicitará as identificações necessárias dele e, após isso, o conduzirá para a decisão do nome de usuário e sua respectiva senha. Só assim, será possível registrar-se. Os dados são salvos clicando no botão “Cadastrar”.

CADASTRAR AGENDAMENTO

Pesquisa :

Nome : Código Paciente :

Data agendada : Estado Visita: ▼

Horário agendado :

CADASTRAR

[Caso deseje alterar os dados de alguma visita clique aqui](#)

Figura 08 - Interface

- **Breve descrição :** Esta tela é restrita aos usuários identificados como sendo da Equipe de Saúde. Nela é possível consultar os pacientes já cadastrados no sistema e registrar o agendamento de visitas aos mesmos no sistema. É possível também ir para a tela de alteração dos dados da visita do paciente clicando em “Caso deseje alterar os dados de alguma visita clique aqui”. Os dados são salvos clicando no botão “Cadastrar”.

ALTERAR AGENDAMENTO

Pesquisa :

Nome paciente agendado :

Código Agendamento :

Data agendada :

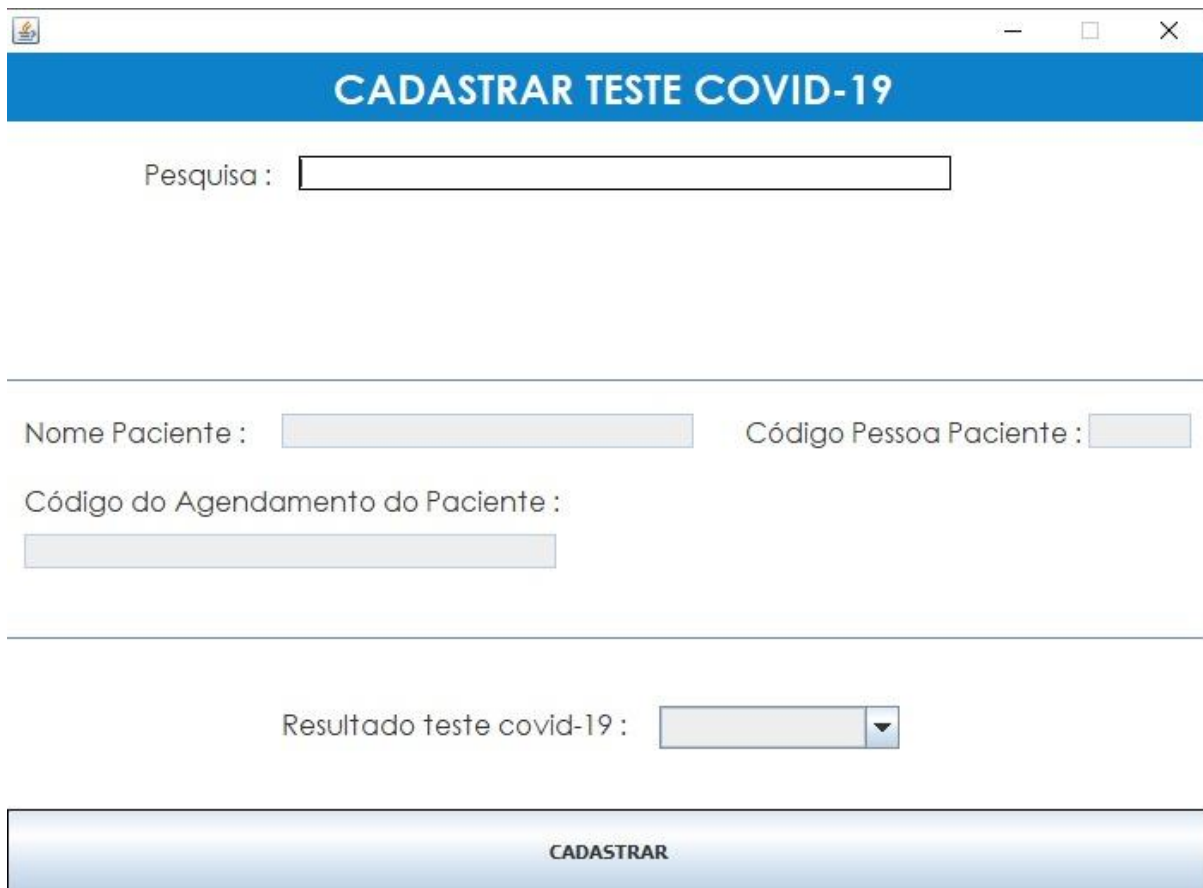
Horário agendado :

Estado da visita agendada:

ALTERAR

Figura 09 - Interface

- **Breve descrição :** Esta tela é acessada através da tela anterior. Nela é possível atualizar os dados da visita agendada. A barra de pesquisa é utilizada para encontrar mais facilmente o paciente por seu nome. Os dados são alterados clicando no botão “Alterar”.



The image shows a web browser window with the title "CADASTRAR TESTE COVID-19". The interface is divided into several sections. At the top, there is a search bar labeled "Pesquisa :". Below this, there are two input fields: "Nome Paciente :" and "Código Pessoa Paciente :". Underneath these, there is a label "Código do Agendamento do Paciente :" followed by an input field. Further down, there is a label "Resultado teste covid-19 :" followed by a dropdown menu. At the bottom of the form, there is a large blue button labeled "CADASTRAR".

Figura 10 - Interface

- **Breve descrição :** Nesta tela, após a visita ao paciente, a equipe de saúde cadastra o resultado do teste realizado, que pode ser positivo ou negativo. Além disso, os campos são preenchidos com os dados do paciente no qual foi realizado o teste. Os dados são salvos clicando no botão "Cadastrar".

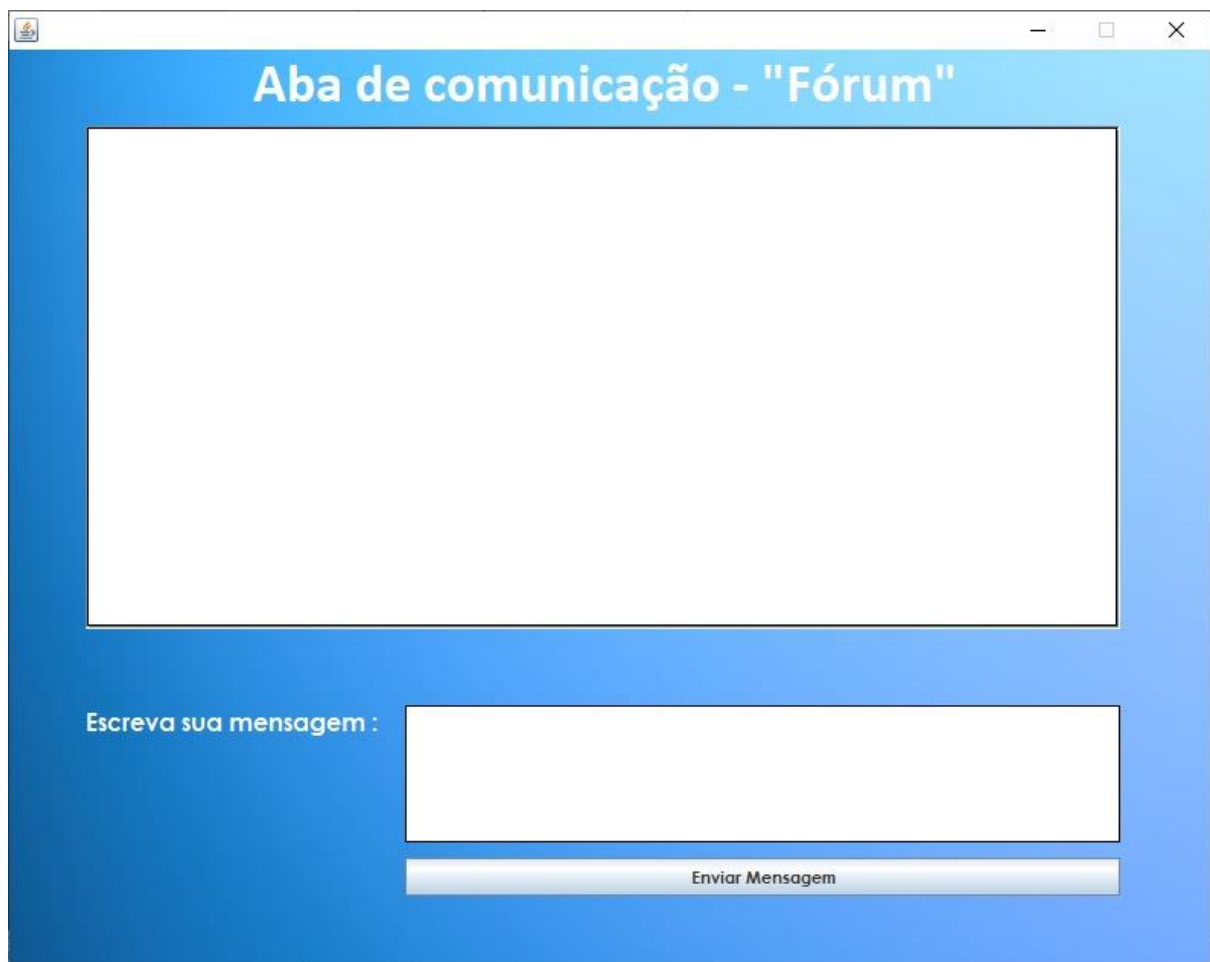


Figura 11 - Interface

- **Breve descrição :** Nesta interface, o usuário tem a possibilidade de se comunicar através de um fórum. Por meio disso, é possível digitar sua mensagem na aba de "escreva sua mensagem", permitindo enviá-la.

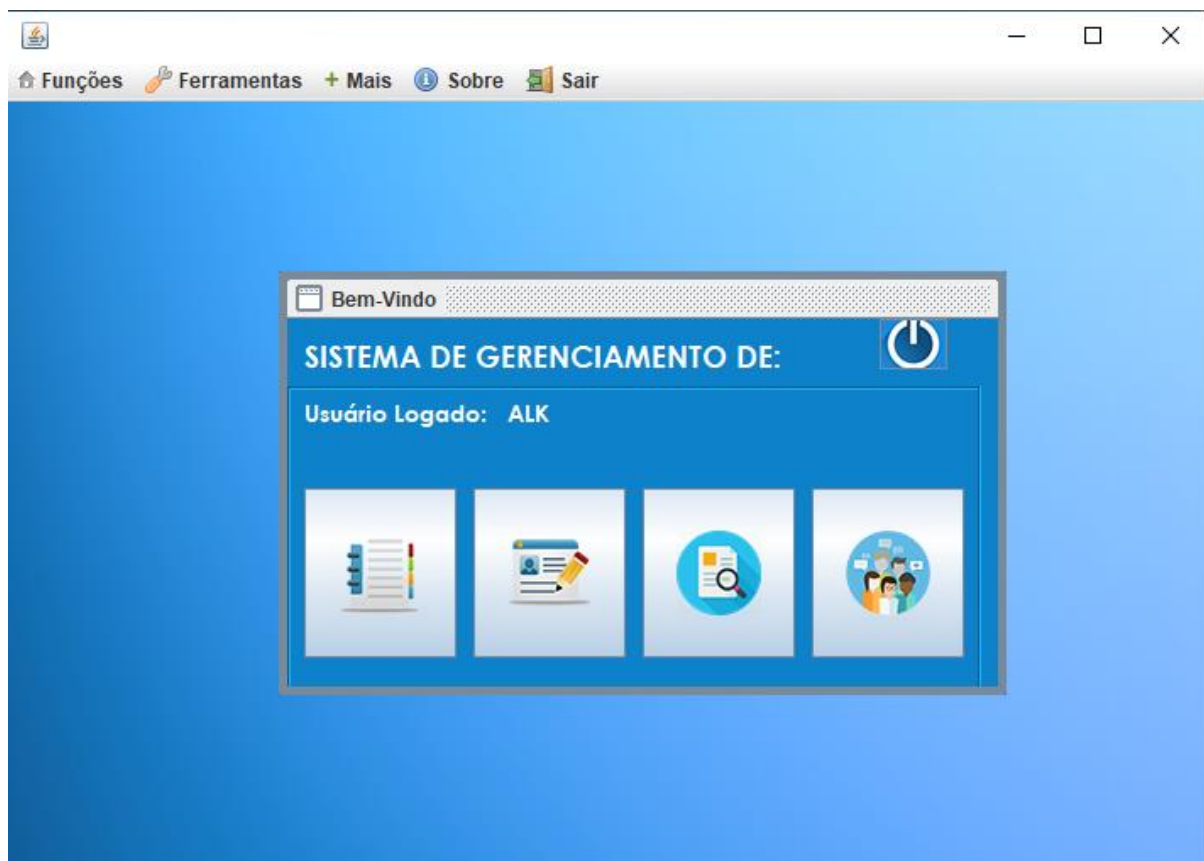


Figura 12 - Interface

- **Breve descrição :** Nesta interface de menu de gerenciamento, o usuário que será um membro da equipe de saúde pode observar em que perfil está logado o sistema. Pode também acessar as funções, as ferramentas, ter informações sobre o sistema ou fazer logout. Na próxima imagem, temos um exemplo de acesso à funções.

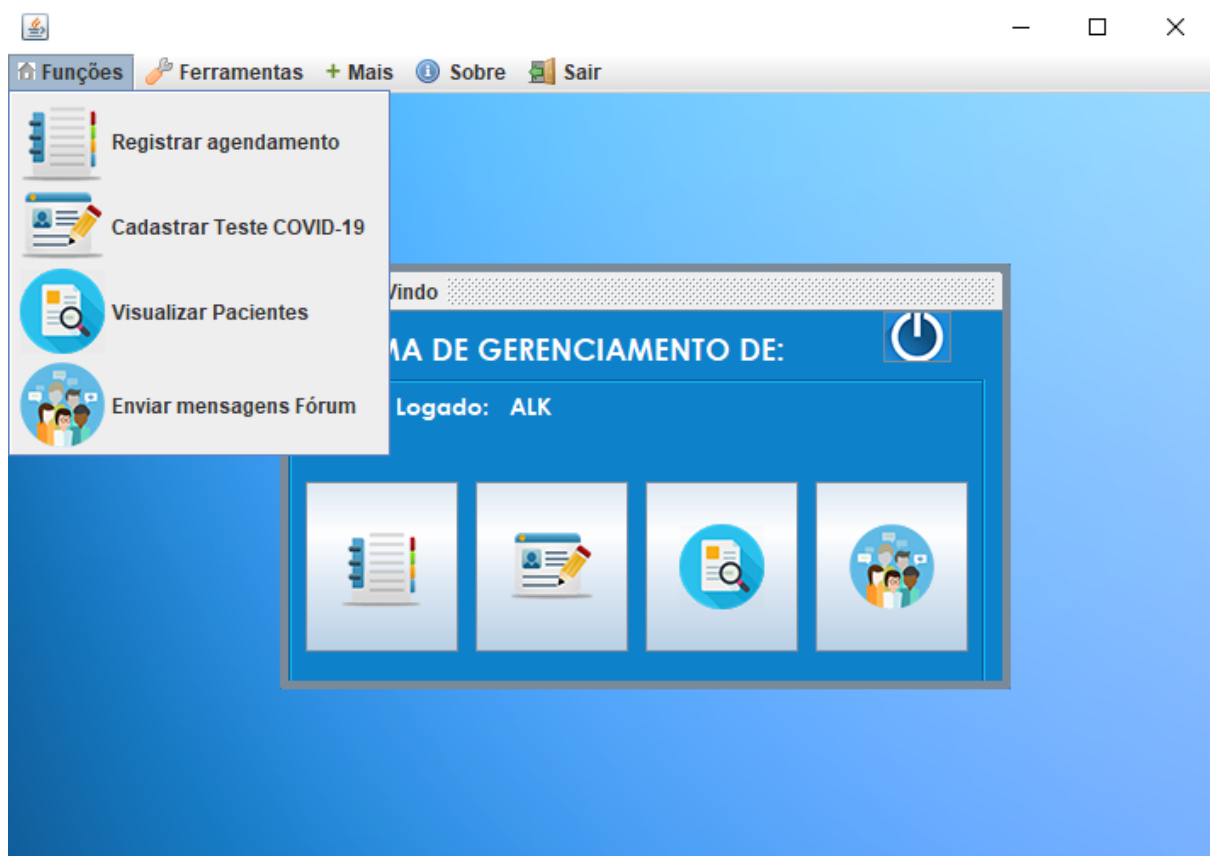


Figura 13 - Interface

- **Breve descrição :** No menu, ao selecionar funções, o usuário integrante de uma equipe de saúde obtém como retorno as funções do sistema. Pode registrar agendamento, cadastrar um teste de COVID-19 realizado, visualizar os pacientes registrados ou enviar mensagens no fórum.

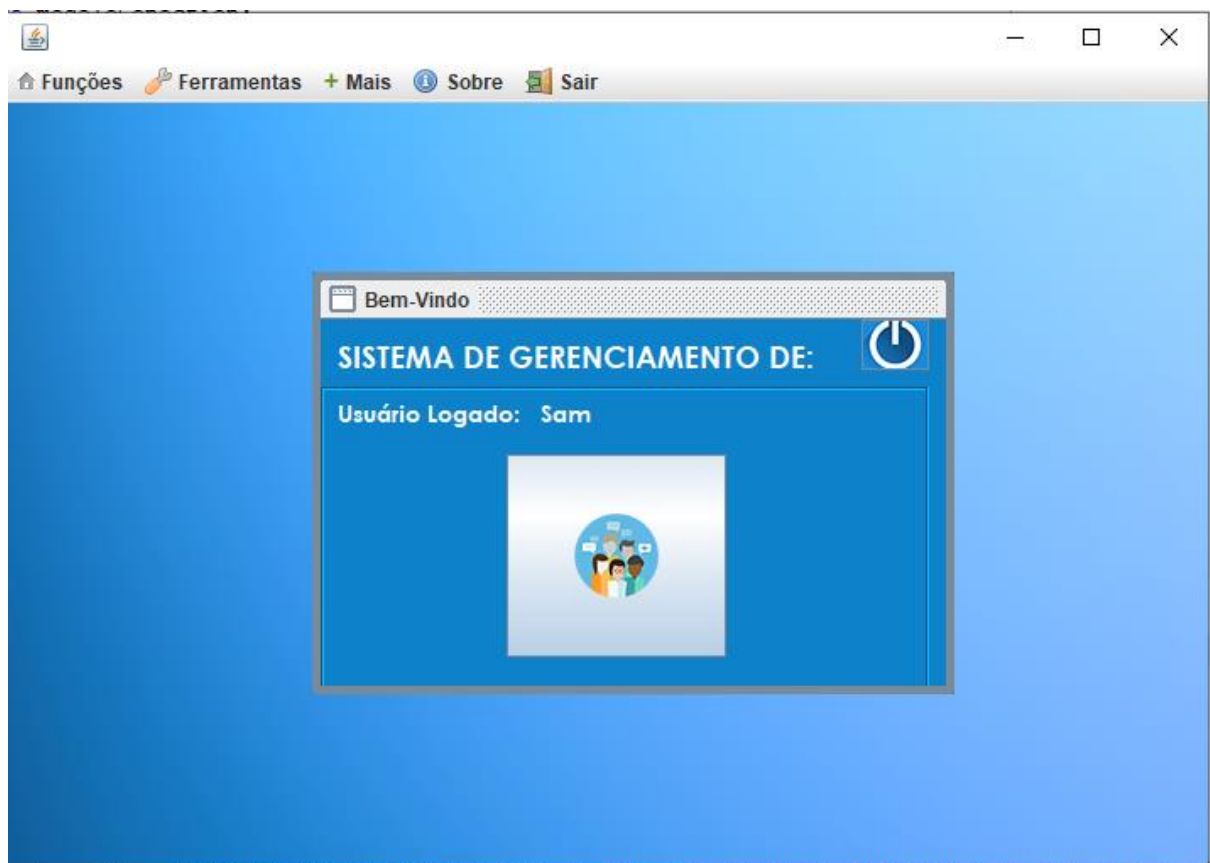


Figura 14 - Interface

- **Breve descrição :** Nesta interface de menu de gerenciamento, o usuário que será um paciente pode observar em que perfil está logado o sistema. Pode também acessar as funções, as ferramentas, ter informações sobre o sistema ou fazer logout. Na próxima imagem, temos um exemplo de acesso à funções.

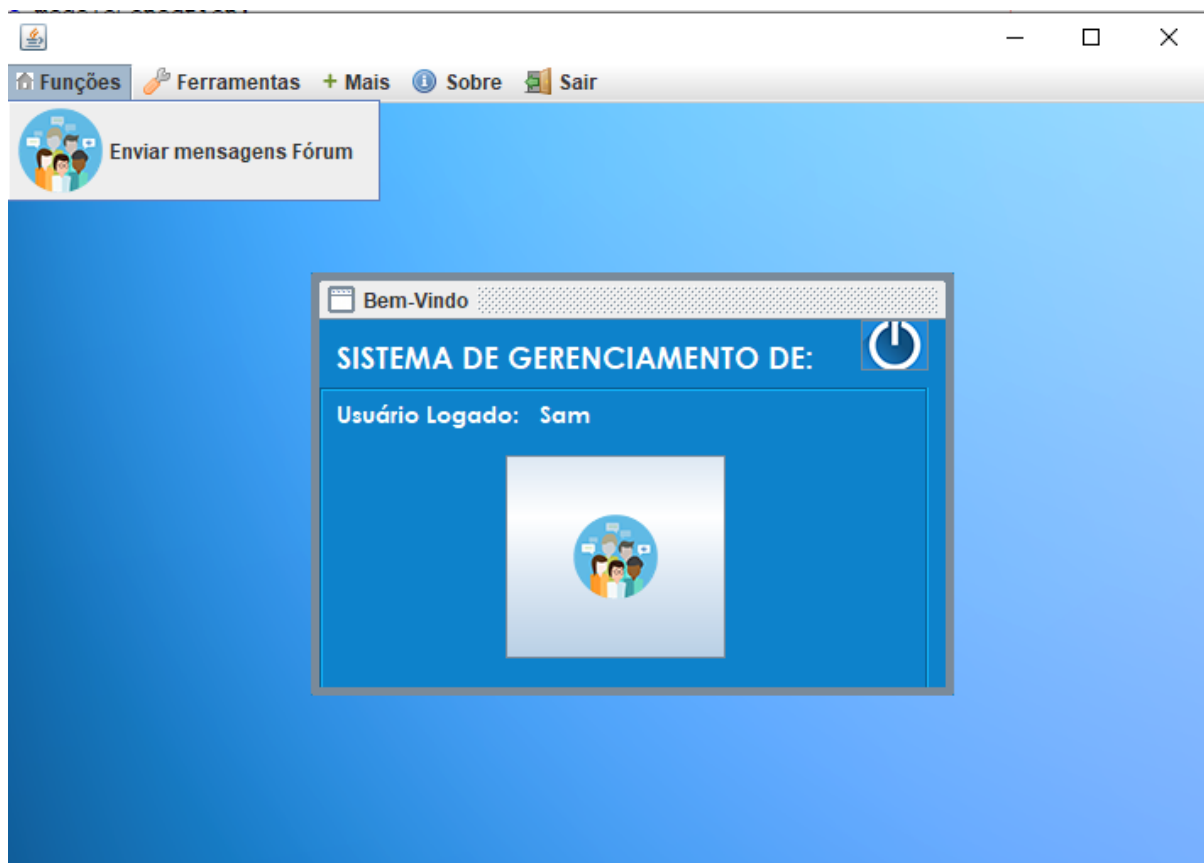
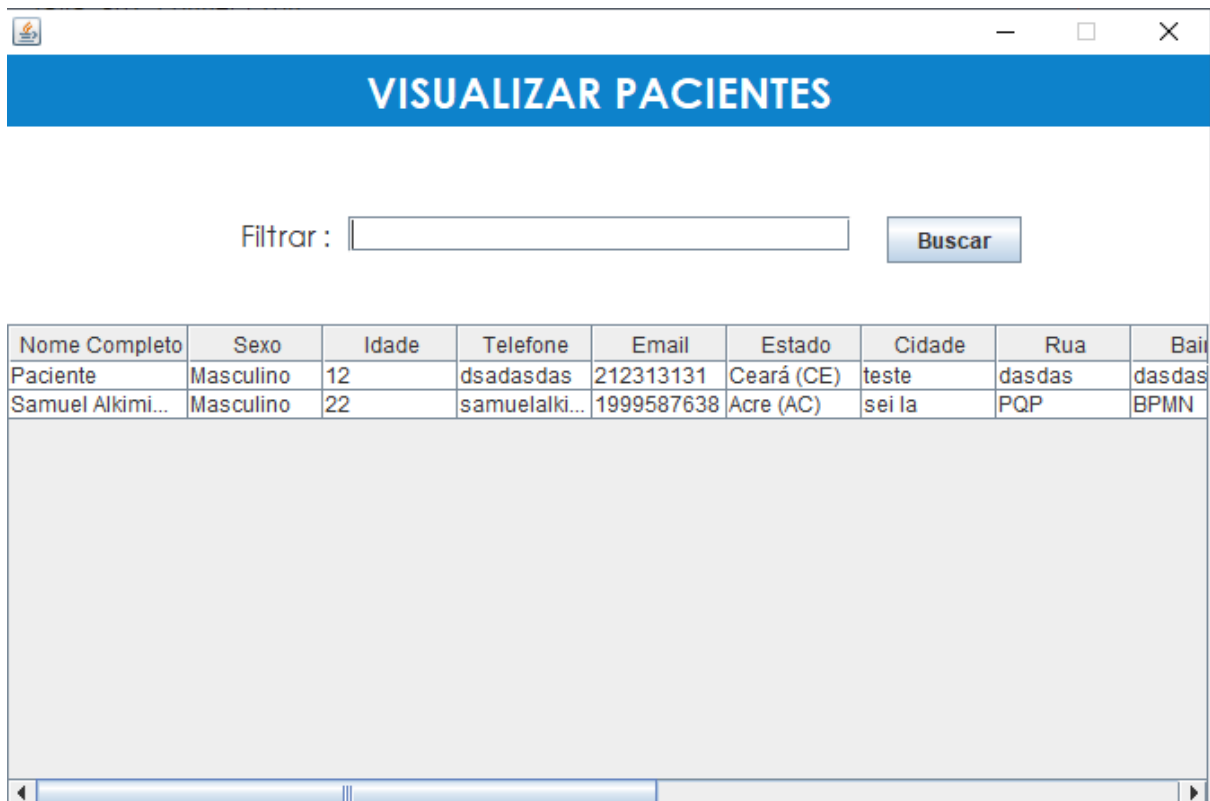


Figura 15 - Interface

- **Breve descrição :** No menu, ao selecionar funções, o usuário paciente obtém como retorno as funções do seu usuário no sistema. Podendo enviar mensagens no fórum.



Nome Completo	Sexo	Idade	Telefone	Email	Estado	Cidade	Rua	Bairro
Paciente	Masculino	12	dsadasdas	212313131	Ceará (CE)	teste	dasdas	dasdas
Samuel Alkimi...	Masculino	22	samuelalki...	1999587638	Acre (AC)	sei la	PQP	BPMN

Figura 16 - Interface

Breve descrição : Nesta interface, o membro da equipe de saúde pode visualizar todos os pacientes e seus dados cadastrados. que são “Nome Completo”, “Sexo”, “Idade”, “Telefone”, “Email”, “Estado”, “Cidade”, “Rua”, “Bairro”, “Número da casa”, “Complemento”, “CEP”, “CPF”, “RG” e “Descrição de sintomas”. Além disso, se pode filtrar os pacientes cadastrados através de uma busca pelo nome.

❏ Código

➤ Código Interfaces

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package visão;

import javax.swing.ImageIcon;

/**
 *
 * @author
 */
public class ViewPrimeiroFormMain extends javax.swing.JFrame {

    /**
     * Creates new form Iniciar
     */

    public ViewPrimeiroFormMain() {

        initComponents();

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel5 = new javax.swing.JLabel();
        jButtonSair = new javax.swing.JButton();
        jButtonEntrar = new javax.swing.JButton();
        jLabelTitulo = new javax.swing.JLabel();
        jLabelFundoView = new javax.swing.JLabel();

        jLabel5.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/imagens/Fundo3View1.png"))); //
        NOI18N
```

```

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setResizable(false);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jButtonSair.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jButtonSair.setForeground(new java.awt.Color(0, 102, 204));
        jButtonSair.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Imagens/shut-down.png"))); // NOI18N
        jButtonSair.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonSairActionPerformed(evt);
            }
        });
        getContentPane().add(jButtonSair, new
org.netbeans.lib.awtextra.AbsoluteConstraints(540, 0, 40, 40));

        jButtonEntrar.setFont(new java.awt.Font("Calibri", 1, 24)); // NOI18N
        jButtonEntrar.setForeground(new java.awt.Color(13, 130, 203));
        jButtonEntrar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/covidicon.png"))); // NOI18N
        jButtonEntrar.setText("Entrar");
        jButtonEntrar.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        jButtonEntrar.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
        jButtonEntrar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonEntrarActionPerformed(evt);
            }
        });
        getContentPane().add(jButtonEntrar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 120, 280, 200));

        jLabelTitulo.setFont(new java.awt.Font("Calibri", 1, 36)); // NOI18N
        jLabelTitulo.setForeground(new java.awt.Color(255, 255, 255));
        jLabelTitulo.setText("Aplicativo OMS - COVID-19");
        getContentPane().add(jLabelTitulo, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 50, -1, -1));

        jLabelFundoView.setFont(new java.awt.Font("Elephant", 0, 36)); // NOI18N
        jLabelFundoView.setForeground(new java.awt.Color(51, 153, 255));
        jLabelFundoView.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/FundoView1.png"))); // NOI18N
        getContentPane().add(jLabelFundoView, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 600, 380));

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

    private void jButtonEntrarActionPerformed(java.awt.event.ActionEvent evt) {
        // ENcaminhar para outra tela (classe)

        ViewListaDeSintomas ListaSintomas = new ViewListaDeSintomas();
    
```

```
        ListaSintomas.setVisible(true);
        dispose(); // fecha a interface viewPrimeiroForm e abre esta outra
    }

    private void jButtonSairActionPerformed(java.awt.event.ActionEvent evt) {
        //SAIR

        System.exit(0);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ViewPrimeiroFormMain().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButtonEntrar;
    private javax.swing.JButton jButtonSair;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabelFundoView;
    private javax.swing.JLabel jLabelTitulo;
    // End of variables declaration

}
```

```
package visão;

import java.awt.Color;
import static java.awt.Color.BLUE;
import static java.awt.Color.WHITE;

public class ViewListaDeSintomas extends javax.swing.JFrame {

    public ViewListaDeSintomas() {
        initComponents();
    }

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

        jScrollPane1 = new javax.swing.JScrollPane();
        jTextPane1 = new javax.swing.JTextPane();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jLabel2 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabelEntrarSistema = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();

        jTextPane1.setBorder(javax.swing.BorderFactory.createLineBorder(new
        java.awt.Color(0, 0, 0)));

        jTextPane1.setText("dsdddddddddddddddddddddddddddddddddddddddddddddddddd
        ddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd");
        jScrollPane1.setViewportView(jTextPane1);

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setResizable(false);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
```

```

        jTextArea1.setColumns(20);
        jTextArea1.setFont(new java.awt.Font("Monospaced", 1, 13)); // NOI18N
        jTextArea1.setLineWrap(true);
        jTextArea1.setRows(5);
        jTextArea1.setText("► Sintomas comuns:\n\n• Febre.\n• Cansaço.\n• Tosse seca.\n\n► Outros sintomas que algumas pessoas podem experimentar:\n\n• Dores e dores.\n• Congestão nasal.\n• Coriza.\n• Dor de garganta.\n• Diarréia.\n\nEm média, se leva de cinco a seis dias a partir da data em\nque alguém está infectado com o vírus para que os sintomas apareçam, no entanto, pode levar até 14 dias.\n");
        jTextArea1.setToolTipText("");
        jTextArea1.setDisabledTextColor(new java.awt.Color(0, 0, 0));
        jTextArea1.setEnabled(false);
        jScrollPane2.setViewportView(jTextArea1);

        getContentPane().add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 70, 500, 270));

        jLabel2.setFont(new java.awt.Font("Calibri", 1, 36)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(255, 255, 255));
        jLabel2.setText("Lista de Sintomas COVID-19");
        getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 10, -1, -1));

        jLabel4.setFont(new java.awt.Font("Calibri", 1, 36)); // NOI18N
        jLabel4.setForeground(new java.awt.Color(255, 255, 255));
        jLabel4.setText("Lista de Sintomas COVID-19");
        getContentPane().add(jLabel4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 10, -1, -1));

        jLabelEntrarSistema.setFont(new java.awt.Font("Century Gothic", 1, 16)); // NOI18N
        jLabelEntrarSistema.setForeground(new java.awt.Color(255, 255, 255));
        jLabelEntrarSistema.setText("<html> <u> Clique aqui caso deseje efetuar login ou se cadastrar no sistema </u> </html>");
        jLabelEntrarSistema.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jLabelEntrarSistemaMouseClicked(evt);
            }
            public void mouseEntered(java.awt.event.MouseEvent evt) {
                jLabelEntrarSistemaMouseEntered(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                jLabelEntrarSistemaMouseExited(evt);
            }
        });
        getContentPane().add(jLabelEntrarSistema, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 350, 510, -1));

        jLabel3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/FundoView1.png"))); // NOI18N
        getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 600, 380));

```

```

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

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

        ViewLogin Login = new ViewLogin();
        Login.setVisible(true);
        dispose();

    }

    private void jLabelEntrarSistemaMouseEntered(java.awt.event.MouseEvent evt) {
        jLabelEntrarSistema.setForeground(BLUE);
    }

    private void jLabelEntrarSistemaMouseExited(java.awt.event.MouseEvent evt) {
        jLabelEntrarSistema.setForeground(WHITE);
    }

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ViewListaDeSintomas().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabelEntrarSistema;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTextArea jTextArea1;
    private javax.swing.JTextPane jTextPane1;
    // End of variables declaration
}

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package visão;

```



```
import modeloBeans.ClassePaciente;
import modeloBeans.ClasseCadastro;
import modeloDao.DAOPaciente;
import modeloDao.DAOCadastro;
import modeloDao.DAOLogin;
import com.sun.glass.events.KeyEvent;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.scene.paint.Color;
import javax.swing.JOptionPane;
import static java.awt.Color.*;
import java.awt.Font;
import modeloBeans.ClassePessoa;
/**
 *
 * @author
 */
public class ViewLogin extends javax.swing.JFrame {

    /**
     * Creates new form ViewLogin
     */

    private ClasseCadastro login = null;

    private DAOPaciente userDAO = new DAOPaciente();

    public ViewLogin()
    {

        initComponents();

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel2 = new javax.swing.JPanel();
        jPanel1 = new javax.swing.JPanel();
        jLabelUsuarioLogin = new javax.swing.JLabel();
        jtfUsuario = new javax.swing.JTextField();
        jLabelSenhaLogin = new javax.swing.JLabel();
        btnEntrar = new javax.swing.JButton();
        btnCadastro = new javax.swing.JLabel();
        jpfSenha = new javax.swing.JPasswordField();
        jLabel1 = new javax.swing.JLabel();

    }
}
```

```
jLabelSouEquipeDeSaude = new javax.swing.JLabel();
jLabelSouPaciente = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setResizable(false);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel2.setBackground(new java.awt.Color(255, 255, 255));
jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 1, 1, new
java.awt.Color(0, 102, 204)));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabelUsuarioLogin.setFont(new java.awt.Font("Century Gothic", 0, 14)); // NOI18N
jLabelUsuarioLogin.setText("Usuario:");
jPanel1.add(jLabelUsuarioLogin, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 40, -1, -1));

jtfUsuario.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(jtfUsuario, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 80,
160, 30));

jLabelSenhaLogin.setFont(new java.awt.Font("Century Gothic", 0, 14)); // NOI18N
jLabelSenhaLogin.setText("Senha:");
jPanel1.add(jLabelSenhaLogin, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 130, -1, -1));

btnEntrar.setFont(new java.awt.Font("Century Gothic", 0, 11)); // NOI18N
btnEntrar.setText("ENTRAR");
btnEntrar.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnEntrarMouseClicked(evt);
    }
});
btnEntrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEntrarActionPerformed(evt);
    }
});
jPanel1.add(btnEntrar, new org.netbeans.lib.awtextra.AbsoluteConstraints(230, 60,
110, 130));

btnCadastro.setFont(new java.awt.Font("Century Gothic", 0, 11)); // NOI18N
btnCadastro.setText("Clique no que te corresponde para se cadastrar:");
btnCadastro.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
jPanel1.add(btnCadastro, new org.netbeans.lib.awtextra.AbsoluteConstraints(30,
200, -1, -1));

jpfSenha.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
```

```

jpfSenha.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jpfSenhaActionPerformed(evt);
    }
});
jpfSenha.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        jpfSenhaKeyPressed(evt);
    }
});
jPanel1.add(jpfSenha, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 160,
160, 30));
jPanel1.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 20, -1, -
1));

jLabelSouEquipeDeSaude.setFont(new java.awt.Font("Century Gothic", 0, 11)); //
NOI18N
jLabelSouEquipeDeSaude.setText("<html> <u> + Sou integrante de uma equipe de
saúde.</u></html>");
jLabelSouEquipeDeSaude.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
jLabelSouEquipeDeSaude.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jLabelSouEquipeDeSaudeMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        jLabelSouEquipeDeSaudeMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        jLabelSouEquipeDeSaudeMouseExited(evt);
    }
});
jPanel1.add(jLabelSouEquipeDeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 240, -1, -1));

jLabelSouPaciente.setFont(new java.awt.Font("Century Gothic", 0, 11)); // NOI18N
jLabelSouPaciente.setText("<html> <u> + Sou paciente. </u> </html>");
jLabelSouPaciente.setCursor(new
java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
jLabelSouPaciente.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        btnCadastroPacienteMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        jLabelSouPacienteMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        jLabelSouPacienteMouseExited(evt);
    }
});
jPanel1.add(jLabelSouPaciente, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 220, -1, -1));

```

```

        jPanel2.add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 50, 380,
270));

        jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/FundoView1.png"))); // NOI18N
        jPanel2.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(-10, 0, 880,
520));

        getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 490, 350));

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

    private void btnEntrarMouseClicked(java.awt.event.MouseEvent evt) {
        /*
        usuario = new ClasseClients();
        usuario.setUSER(jtfUsuario.getText());
        usuario.setPASS(jpfSenha.getText());
        try {
            usuario = DAOClient.Logar(usuario);
            JOptionPane.showMessageDialog(null, "Bem vindo " + usuario.getALLNAME()+ "
!!! ");
            new TelaPrincipal(usuario,jtfUsuario.getText()).setVisible(true);

        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Usuário ou senha inválidos.", "",
JOptionPane.ERROR_MESSAGE);
            jtfUsuario.setText("");
            jpfSenha.setText("");
            Logger.getLogger(ViewClientLogin.class.getName()).log(Level.SEVERE, null, ex);
        }
        */
        /* login = new ClasseCadastro();
        login.setUsuario(jtfUsuario.getText());
        login.setSenha(jpfSenha.getText());

        try {

            login = DAOLogin.checkLogin(login); // não tá funfando
            // JOptionPane.showMessageDialog(null, "Bem vindo " +
usuario.getALLNAME()+ " !!! ");
            new TelaPrincipal(login,jtfUsuario.getText()).setVisible(true); // expor nome

        } catch (Exception ex) {

            JOptionPane.showMessageDialog(null, "Usuário ou senha inválidos.", "",
JOptionPane.ERROR_MESSAGE);
            jtfUsuario.setText("");

```

```
        jpfSenha.setText("");

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

    /*ClasseCadastro nomeusuario = new ClasseCadastro();

    nomeusuario.setUsuario(jtfUsuario.getText());
    nomeusuario.setSenha(jpfSenha.getText());

    try
    {
        ClassePessoa pessoa = new ClassePessoa();
        usuario = DAOLogin.Logar(nomeusuario);
        JOptionPane.showMessageDialog(null, "Bem vindo " + pessoa.nome + " !!! ");
        new TelaPrincipal(usuario).setVisible(true);

    } catch (Exception ex)

    {

        JOptionPane.showMessageDialog(null, "Usuário ou senha inválidos.", "",
JOptionPane.ERROR_MESSAGE);
        jtfUsuario.setText("");
        jpfSenha.setText("");
        Logger.getLogger(ViewLogin.class.getName()).log(Level.SEVERE, null, ex);

    }
    */
}

private void btnEntrarActionPerformed(java.awt.event.ActionEvent evt) {
    DAOLogin dao = new DAOLogin();

    if(dao.checkLogin(jtfUsuario.getText(), jpfSenha.getText())){

        boolean paciente = false ;

        try {

            int chaveUsuario = dao.BuscaIdUsuario(jtfUsuario.getText().toString());
            paciente = dao.VerificarTipo(chaveUsuario);

        } catch (Exception ex) {
            Logger.getLogger(ViewLogin.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

```

        if( paciente == true)
        {
            new TelaPrincipalPaciente(login,jtfUsuario.getText()).setVisible(true);
            this.dispose();

        }
        else
        {
            new TelaPrincipal(login,jtfUsuario.getText()).setVisible(true);
            this.dispose();
        }
    }else{
        JOptionPane.showMessageDialog(null, "Usuário ou senha inválidos.", "",
JOptionPane.ERROR_MESSAGE);
        jtfUsuario.setText("");
        jpfSenha.setText("");

    }
}

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

}

private void jpfSenhaKeyPressed(java.awt.event.KeyEvent evt) {
    /*  if(evt.getKeyCode() == KeyEvent.VK_ENTER){
        usuario = new ClasseClients();
        usuario.setUSER(jtfUsuario.getText());
        usuario.setPASS(jpfSenha.getText());
        try {
            usuario = DAOClient.Logar(usuario);
            JOptionPane.showMessageDialog(null, "Bem vindo " + usuario.getALLNAME());
            new TelaPrincipal(usuario,jtfUsuario.getText()).setVisible(true);
            this.dispose();

        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Usuário ou senha inválidos.", "",
JOptionPane.ERROR_MESSAGE);
            jtfUsuario.setText("");
            jpfSenha.setText("");
            Logger.getLogger(ViewClientLogin.class.getName()).log(Level.SEVERE, null,
ex);
        }
    } */

    /* if(evt.getKeyCode() == KeyEvent.VK_ENTER)
    {

        login = new ClasseCadastro();

        login.setUsuario(jtfUsuario.getText());

```

```
login.setSenha(jpfSenha.getText());

try
{

    login = DAOLogin.Logar(login);
    ClassePessoa nomedousuario = new ClassePessoa();
    JOptionPane.showMessageDialog(null, "Bem vindo " +
nomedousuario.nome);
    new TelaPrincipal(login,jtfUsuario.getText()).setVisible(true);

}catch(Exception ex)
{
    JOptionPane.showMessageDialog(null, "Usuário ou senha inválidos.", "",
JOptionPane.ERROR_MESSAGE);
    jtfUsuario.setText("");
    jpfSenha.setText("");

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

}

}
/*
if(evt.getKeyCode() == KeyEvent.VK_ENTER)
{
    ClasseCadastro nomeusuario = new ClasseCadastro();

    nomeusuario.setUsuario(jtfUsuario.getText());
    nomeusuario.setSenha(jpfSenha.getText());

    try
    {

        DAOLogin.Logar(nomeusuario);
        ClassePessoa pessoa = new ClassePessoa();
        JOptionPane.showMessageDialog(null, "Bem vindo " + pessoa.nome);
        new TelaPrincipal(nomeusuario).setVisible(true);
        this.dispose();

    } catch (Exception ex)

    {

        JOptionPane.showMessageDialog(null, "Usuário ou senha inválidos.", "",
JOptionPane.ERROR_MESSAGE);
        jtfUsuario.setText("");
        jpfSenha.setText("");
        Logger.getLogger(ViewLogin.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

```
    }

    */
}

private void btnCadastroPacienteMouseClicked(java.awt.event.MouseEvent evt) {
    ViewRegistrarLoginPaciente CadastroPaciente = new ViewRegistrarLoginPaciente();
    CadastroPaciente.setVisible(true);
    dispose(); // fecha a interface viewPrimeiroForm e abre esta outra
}

private void jLabelSouPacienteMouseEntered(java.awt.event.MouseEvent evt) {
    jLabelSouPaciente.setForeground(BLUE);
}

private void jLabelSouPacienteMouseExited(java.awt.event.MouseEvent evt) {
    jLabelSouPaciente.setForeground(BLACK);
}

private void jLabelSouEquipeDeSaudeMouseEntered(java.awt.event.MouseEvent evt) {
    jLabelSouEquipeDeSaude.setForeground(BLUE);
}

private void jLabelSouEquipeDeSaudeMouseExited(java.awt.event.MouseEvent evt) {
    jLabelSouEquipeDeSaude.setForeground(BLACK);
}

private void jLabelSouEquipeDeSaudeMouseClicked(java.awt.event.MouseEvent evt) {
    ViewRegistrarLoginEquipeDeSaude CadastrarEquipeDeSaude = new
ViewRegistrarLoginEquipeDeSaude();
    CadastrarEquipeDeSaude.setVisible(true);
    dispose();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ViewLogin().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel btnCadastro;
```



```

private javax.swing.JButton btnEntrar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabelSenhaLogin;
private javax.swing.JLabel jLabelSouEquipeDeSaude;
private javax.swing.JLabel jLabelSouPaciente;
private javax.swing.JLabel jLabelUsuarioLogin;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPasswordField jpfSenha;
private javax.swing.JTextField jtfUsuario;
// End of variables declaration
}

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package visão;

import java.awt.event.ItemEvent;
import java.util.ArrayList;
import java.util.List;
import java.util.Vector;
import modeloBeans.ClasseEquipeSaude;
import modeloDao.DAOCadastro;
import modeloDao.DAOPessoa;
import modeloDao.DAOEquipeDeSaude;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ComboBoxModel;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import static javax.swing.JOptionPane.YES_NO_OPTION;
import modeloBeans.ClasseCadastro;
import modeloBeans.ClasseCidade;
import modeloBeans.ClassePessoa;
import modeloConexao.ConexaoBD;
import modeloDao.DAOCidades;

/**
 *
 * @author
 */
public class ViewRegistrarLoginEquipeDeSaude extends javax.swing.JFrame {

    public ViewRegistrarLoginEquipeDeSaude() {

```

```
initComponents();
}

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

    jPanel1 = new javax.swing.JPanel();
    jLabelNome = new javax.swing.JLabel();
    txtNomeEquipedeSaude = new javax.swing.JTextField();
    jLabelEmail = new javax.swing.JLabel();
    txtEmailEquipedeSaude = new javax.swing.JTextField();
    jLabelUsuario = new javax.swing.JLabel();
    txtUsuario = new javax.swing.JTextField();
    jLabelEstado = new javax.swing.JLabel();
    jLabelIdade = new javax.swing.JLabel();
    txtIdadeEquipedeSaude = new javax.swing.JTextField();
    jLabelCidade = new javax.swing.JLabel();
    jLabelTelefone = new javax.swing.JLabel();
    txtTelefoneEquipedeSaude = new javax.swing.JTextField();
    jLabelSenha = new javax.swing.JLabel();
    txtSenha = new javax.swing.JPasswordField();
    jButtonCadastrarPaciente = new javax.swing.JButton();
    txtCpfEquipedeSaude = new javax.swing.JTextField();
    jLabelCpf = new javax.swing.JLabel();
    txtRgEquipedeSaude = new javax.swing.JTextField();
    txtCepEquipedeSaude = new javax.swing.JTextField();
    txtNumCasaEquipedeSaude = new javax.swing.JTextField();
    txtBairroEquipedeSaude = new javax.swing.JTextField();
    txtComplementoEquipedeSaude = new javax.swing.JTextField();
    txtRuaEquipedeSaude = new javax.swing.JTextField();
    jLabelSexo = new javax.swing.JLabel();
    jLabelRg = new javax.swing.JLabel();
    jLabelRua = new javax.swing.JLabel();
    jLabelBairro = new javax.swing.JLabel();
    jLabelCep = new javax.swing.JLabel();
    jLabelComplemento = new javax.swing.JLabel();
    jLabelNumCasa = new javax.swing.JLabel();
    jSeparator = new javax.swing.JSeparator();
    jLabel3 = new javax.swing.JLabel();
    jLabel19 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    txtCodigoEspeciaEquipeDeSaude = new javax.swing.JTextField();
    txtSexoEquipedeSaude = new javax.swing.JComboBox<>();
    txtEstadoEquipedeSaude = new javax.swing.JComboBox<>();
    txtCidadeEquipedeSaude = new javax.swing.JComboBox<>();
    jPanel3 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setResizable(false);
```

```
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabelNome.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelNome.setText("Nome completo :");
jPanel1.add(jLabelNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10, 140, 20));

txtNomeEquipedeSaude.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtNomeEquipedeSaude, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 10, 270, -1));

jLabelEmail.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelEmail.setText("Email :");
jPanel1.add(jLabelEmail, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 121, -1, 30));

txtEmailEquipedeSaude.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtEmailEquipedeSaude, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 130, 330, -1));

jLabelUsuario.setFont(new java.awt.Font("Century Gothic", 0, 18)); // NOI18N
jLabelUsuario.setText("Nome de Usuário:");
jPanel1.add(jLabelUsuario, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 160, -1, 20));

txtUsuario.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtUsuario, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 200, 320, 30));

jLabelEstado.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelEstado.setText("Estado :");
jPanel1.add(jLabelEstado, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 159, -1, -1));

jLabelIdade.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelIdade.setText("Idade :");
jPanel1.add(jLabelIdade, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, -1, 20));

txtIdadeEquipedeSaude.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtIdadeEquipedeSaude, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 70, 330, -1));

jLabelCidade.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelCidade.setText("Cidade :");
jPanel1.add(jLabelCidade, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 190, -1, 20));

jLabelTelefone.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
```

```
jLabelTelefone.setText("Telefone :");
jPanel1.add(jLabelTelefone, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
100, 90, 20));

txtTelefoneEquipedeSaude.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtTelefoneEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 100, 330, -1));

jLabelSenha.setFont(new java.awt.Font("Century Gothic", 0, 18)); // NOI18N
jLabelSenha.setText("Senha:");
jPanel1.add(jLabelSenha, new org.netbeans.lib.awtextra.AbsoluteConstraints(450,
250, -1, -1));

txtSenha.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtSenha, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 290,
320, 30));

jButtonCadastrarPaciente.setText("CADASTRAR");
jButtonCadastrarPaciente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCadastrarPacienteActionPerformed(evt);
    }
});
jPanel1.add(jButtonCadastrarPaciente, new
org.netbeans.lib.awtextra.AbsoluteConstraints(-10, 440, 820, 50));
jPanel1.add(txtCpfEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 370, 360, 25));

jLabelCpf.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelCpf.setText("CPF :");
jPanel1.add(jLabelCpf, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 360, -
1, 40));
jPanel1.add(txtRgEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 400, 360, 25));
jPanel1.add(txtCepEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 340, 360, 25));
jPanel1.add(txtNumCasaEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 280, 260, 25));
jPanel1.add(txtBairroEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 250, 330, 23));
jPanel1.add(txtComplementoEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 310, 260, 25));
jPanel1.add(txtRuaEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 220, 330, 25));

jLabelSexo.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelSexo.setText("Sexo :");
jPanel1.add(jLabelSexo, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 30, -
1, 40));

jLabelRg.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelRg.setText("RG :");
```

```
jPanel1.add(jLabelRg, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 400, -1, -1));

jLabelRua.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelRua.setText("Rua :");
jPanel1.add(jLabelRua, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 211, 60, 40));

jLabelBairro.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelBairro.setText("Bairro :");
jPanel1.add(jLabelBairro, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 240, 60, 40));

jLabelCep.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelCep.setText("CEP :");
jPanel1.add(jLabelCep, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 330, 40, 40));

jLabelComplemento.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelComplemento.setText("Complemento :");
jPanel1.add(jLabelComplemento, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 300, -1, 40));

jLabelNumCasa.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelNumCasa.setText("Número da casa :");
jPanel1.add(jLabelNumCasa, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 270, -1, 40));

jSeparador.setForeground(new java.awt.Color(0, 0, 0));
jSeparador.setOrientation(javax.swing.SwingConstants.VERTICAL);
jPanel1.add(jSeparador, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 0, 20, 450));

jLabel3.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel3.setText("Agora escreva os dados que serão usados ");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 60, 340, 50));

jLabel19.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel19.setText("para seu acesso ao sistema:");
jPanel1.add(jLabel19, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 110, -1, 30));

jLabel2.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel2.setText("Código Especial:");
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 20, -1, 40));
jPanel1.add(txtCodigoEspeciaEquipeDeSaude, new org.netbeans.lib.awtextra.AbsoluteConstraints(590, 30, 160, 20));

txtSexoEquipeDeSaude.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Masculino", "Feminino", "Outro" }));
```

```

txtSexoEquipedeSaude.setSelectedIndex(-1);
jPanel1.add(txtSexoEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 40, 330, -1));

txtEstadoEquipedeSaude.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Acre (AC)", "Alagoas (AL)", "Amapá (AP)", "Amazonas (AM)", "Bahia (BA)",
"Ceará (CE)", "Distrito Federal (DF)", "Espírito Santo (ES)", "Goiás (GO)", "Maranhão
(MA)", "Mato Grosso (MT)", "Mato Grosso do Sul (MS)", "Minas Gerais (MG)", "Pará (PA)",
"Paraíba (PB)", "Paraná (PR)", "Pernambuco (PE)", "Piauí (PI)", "Rio de Janeiro (RJ)",
"Rio Grande do Norte (RN)", "Rio Grande do Sul (RS)", "Rondônia (RO)", "Roraima (RR)",
"Santa Catarina (SC)", "São Paulo (SP)", "Sergipe (SE)", "Tocantins (TO)" }));
txtEstadoEquipedeSaude.setSelectedIndex(-1);
txtEstadoEquipedeSaude.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        txtEstadoEquipedeSaude.itemStateChanged(evt);
    }
});
jPanel1.add(txtEstadoEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 160, 330, -1));

txtCidadeEquipedeSaude.setToolTipText("");
txtCidadeEquipedeSaude.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        txtCidadeEquipedeSaude.itemStateChanged(evt);
    }
});
jPanel1.add(txtCidadeEquipedeSaude, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 190, 330, -1));

getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
40, 800, 480));

jPanel3.setBackground(new java.awt.Color(13, 130, 203));
jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Century Gothic", 1, 24)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setText("CADASTRO INTEGRANTE EQUIPE DE SAÚDE");
jPanel3.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 0, -1,
40));

getContentPane().add(jPanel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 800, 40));

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

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

    ClassePessoa novoClassePessoa = new ClassePessoa(); // instanciando → objeto
    → classe Paciente

```

```

    DAOPessoa novoCadastroPessoaDAO = new DAOPessoa();

    novoClassePessoa.nome = txtNomeEquipeDeSaude.getText().trim();
    novoClassePessoa.sexo = (String) txtSexoEquipeDeSaude.getSelectedItemId();
    novoClassePessoa.idade = txtIdadeEquipeDeSaude.getText().trim();
    novoClassePessoa.email = txtEmailEquipeDeSaude.getText().trim();
    novoClassePessoa.telefone = txtTelefoneEquipeDeSaude.getText().trim();
    novoClassePessoa.estado = (String) txtEstadoEquipeDeSaude.getSelectedItemId();
    novoClassePessoa.cidade = (String) txtCidadeEquipeDeSaude.getSelectedItemId();
    novoClassePessoa.rua = txtRuaEquipeDeSaude.getText().trim();
    novoClassePessoa.bairro = txtBairroEquipeDeSaude.getText().trim();
    novoClassePessoa.numCasa = txtNumCasaEquipeDeSaude.getText().trim();
    novoClassePessoa.complemento =
txtComplementoEquipeDeSaude.getText().trim();
    novoClassePessoa.cep = txtCepEquipeDeSaude.getText().trim();

    novoClassePessoa.setCpf(txtCpfEquipeDeSaude.getText().trim());
    novoClassePessoa.setRg(txtRgEquipeDeSaude.getText().trim());

    ClasseEquipeSaude novoClasseEquipeDeSaude = new ClasseEquipeSaude();
    DAOEquipeDeSaude novoDadosEquipeDeSaudeDAO = new
DAOEquipeDeSaude(); // instanciando → objeto → classe DAOPaciente ( para banco de
dados)

novoClasseEquipeDeSaude.setCodigoEspecial(txtCodigoEspecialEquipeDeSaude.getText
().trim());

    ClasseCadastro novoCadastroEquipeDeSaude = new ClasseCadastro();
    DAOCadastro novoCadastroUsuarioAcessoEquipeDeSaudeDAO = new
DAOCadastro();

    novoCadastroEquipeDeSaude.setUsuario(txtUsuario.getText().trim());
    novoCadastroEquipeDeSaude.setSenha(txtSenha.getText().trim());

    try
    {

        novoCadastroPessoaDAO.CadastroDadosPessoa(novoClassePessoa);

        int chaveEstrangeiraPessoaCadastrada =
novoCadastroPessoaDAO.BuscaIdPessoaCadastrada(txtRgEquipeDeSaude.getText());

        novoDadosEquipeDeSaudeDAO.CadastroDadosEquipeDeSaude(novoClasseEquipeDeS
aude,chaveEstrangeiraPessoaCadastrada);

        novoCadastroUsuarioAcessoEquipeDeSaudeDAO.CadastroAcessoLoginUsuario(novoCa

```



```
        dastroEquipeDeSaude,chaveEstrangeiraPessoaCadastrada);

        }
        catch (Exception ex)
        {

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

        }

        JOptionPane.showMessageDialog(null, "" + novoClassePessoa.nome+ " foi
        cadastrado com sucesso!");

        int resposta =  JOptionPane.showConfirmDialog(null, "Deseja Volta a tela de
        Login?", "Mensagem", YES_NO_OPTION);

        if (resposta == JOptionPane.YES_OPTION)
        {

            new ViewLogin().setVisible(true);
            this.dispose();

        }

        else if(resposta == JOptionPane.NO_OPTION)
        {

            txtNomeEquipedeSaude.setText("");
            txtSexoEquipedeSaude.setEnabled(false);
            txtIdadeEquipedeSaude.setText("");
            txtEmailEquipedeSaude.setText("");
            txtTelefoneEquipedeSaude.setText("");
            txtEstadoEquipedeSaude.setEnabled(false);
            txtCidadeEquipedeSaude.setEnabled(false);
            txtRuaEquipedeSaude.setText("");
            txtBairroEquipedeSaude.setText("");
            txtComplementoEquipedeSaude.setText("");
            txtCepEquipedeSaude.setText("");
            txtNumCasaEquipedeSaude.setText("");
            txtCpfEquipedeSaude.setText("");
            txtRgEquipedeSaude.setText("");
            txtCodigoEspeciaEquipeDeSaude.setText("");
            txtUsuario.setText("");
            txtSenha.setText("");

        }

    }
}
```



```
private void txtCidadeEquipedeSaudeItemStateChanged(java.awt.event.ItemEvent evt)
{

}

private void txtEstadoEquipedeSaudeItemStateChanged(java.awt.event.ItemEvent evt)
{

    if(evt.getStateChange() == ItemEvent.SELECTED){

        DAOCidades listacidade = new DAOCidades();
        List<ClasseCidade> cidades = new ArrayList();

        try {
            txtCidadeEquipedeSaude.removeAllItems();

            cidades =
listacidade.ListarCidades(txtEstadoEquipedeSaude.getSelectedItem().toString());
            for (ClasseCidade cidade : cidades) {
                txtCidadeEquipedeSaude.addItem(cidade.nomeCidade.toString());
            }

        } catch (Exception ex){

            JOptionPane.showMessageDialog(null,"Não foi possível listar as cidades" , "",
JOptionPane.ERROR_MESSAGE);

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

    }

}

public static void main(String args[])
{
    java.awt.EventQueue.invokeLater(new Runnable()

    {
        public void run()
        {
            new ViewRegistrarLoginEquipeDeSaude().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButtonCadastrarPaciente;
```

```
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel19;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabelBairro;  
private javax.swing.JLabel jLabelCep;  
private javax.swing.JLabel jLabelCidade;  
private javax.swing.JLabel jLabelComplemento;  
private javax.swing.JLabel jLabelCpf;  
private javax.swing.JLabel jLabelEmail;  
private javax.swing.JLabel jLabelEstado;  
private javax.swing.JLabel jLabelIdade;  
private javax.swing.JLabel jLabelNome;  
private javax.swing.JLabel jLabelNumCasa;  
private javax.swing.JLabel jLabelRg;  
private javax.swing.JLabel jLabelRua;  
private javax.swing.JLabel jLabelSenha;  
private javax.swing.JLabel jLabelSexo;  
private javax.swing.JLabel jLabelTelefone;  
private javax.swing.JLabel jLabelUsuario;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JSeparator jSeparador;  
public javax.swing.JTextField txtBairroEquipedeSaude;  
public javax.swing.JTextField txtCepEquipedeSaude;  
public javax.swing.JComboBox<String> txtCidadeEquipedeSaude;  
private javax.swing.JTextField txtCodigoEspeciaEquipeDeSaude;  
public javax.swing.JTextField txtComplementoEquipedeSaude;  
private javax.swing.JTextField txtCpfEquipedeSaude;  
public javax.swing.JTextField txtEmailEquipedeSaude;  
public javax.swing.JComboBox<String> txtEstadoEquipedeSaude;  
public javax.swing.JTextField txtIdadeEquipedeSaude;  
public javax.swing.JTextField txtNomeEquipedeSaude;  
public javax.swing.JTextField txtNumCasaEquipedeSaude;  
private javax.swing.JTextField txtRgEquipedeSaude;  
public javax.swing.JTextField txtRuaEquipedeSaude;  
private javax.swing.JPasswordField txtSenha;  
public javax.swing.JComboBox<String> txtSexoEquipedeSaude;  
public javax.swing.JTextField txtTelefoneEquipedeSaude;  
private javax.swing.JTextField txtUsuario;  
// End of variables declaration  
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package visão;
```

```
import java.awt.event.ItemEvent;
import java.util.ArrayList;
import java.util.List;
import modeloBeans.ClassePaciente;
import modeloBeans.ClassePessoa;
import modeloDao.DAOPaciente;
import modeloDao.DAOCadastro;
import modeloDao.DAOPessoa;
import modeloBeans.ClasseCadastro;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import static javax.swing.JOptionPane.YES_NO_OPTION;
import modeloBeans.ClasseCidade;
import modeloDao.DAOCidades;

/**
 *
 * @author
 */
public class ViewRegistrarLoginPaciente extends javax.swing.JFrame {

    public ViewRegistrarLoginPaciente() {
        initComponents();
    }

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

        jPanel1 = new javax.swing.JPanel();
        jLabelNome = new javax.swing.JLabel();
        txtNome = new javax.swing.JTextField();
        jLabelEmail = new javax.swing.JLabel();
        txtEmail = new javax.swing.JTextField();
        jLabelUsuario = new javax.swing.JLabel();
        txtUsuario = new javax.swing.JTextField();
        jLabelEstado = new javax.swing.JLabel();
        jLabelIdade = new javax.swing.JLabel();
        txtIdade = new javax.swing.JTextField();
        jLabelCidade = new javax.swing.JLabel();
        jLabelTelefone = new javax.swing.JLabel();
        txtTelefone = new javax.swing.JTextField();
        jLabelSenha = new javax.swing.JLabel();
        txtSenha = new javax.swing.JPasswordField();
        jButtonCadastrarPaciente = new javax.swing.JButton();
        txtCpf = new javax.swing.JTextField();
        jLabelCpf = new javax.swing.JLabel();
        txtRg = new javax.swing.JTextField();
        txtCep = new javax.swing.JTextField();
    }
}
```

```
txtNumCasa = new javax.swing.JTextField();
txtBairro = new javax.swing.JTextField();
txtComplemento = new javax.swing.JTextField();
txtRua = new javax.swing.JTextField();
jLabelSexo = new javax.swing.JLabel();
jLabelRg = new javax.swing.JLabel();
jLabelRua = new javax.swing.JLabel();
jLabelBairro = new javax.swing.JLabel();
jLabelCep = new javax.swing.JLabel();
jLabelComplemento = new javax.swing.JLabel();
jLabelNumCasa = new javax.swing.JLabel();
jSeparador = new javax.swing.JSeparator();
jLabel3 = new javax.swing.JLabel();
jLabel19 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
txtDescricaoSintomas = new javax.swing.JTextArea();
jLabel2 = new javax.swing.JLabel();
txtSexo = new javax.swing.JComboBox<>();
txtEstado = new javax.swing.JComboBox<>();
txtCidade = new javax.swing.JComboBox<>();
jPanel3 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setResizable(false);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabelNome.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelNome.setText("Nome completo :");
jPanel1.add(jLabelNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10,
140, 20));

txtNome.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 10,
270, -1));

jLabelEmail.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelEmail.setText("Email :");
jPanel1.add(jLabelEmail, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 121,
-1, 30));

txtEmail.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtEmail, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 130,
330, -1));

jLabelUsuario.setFont(new java.awt.Font("Century Gothic", 0, 18)); // NOI18N
jLabelUsuario.setText("Nome de Usuário:");
jPanel1.add(jLabelUsuario, new org.netbeans.lib.awtextra.AbsoluteConstraints(450,
250, -1, 20));
```

```
txtUsuario.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtUsuario, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 280,
320, 30));

jLabelEstado.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelEstado.setText("Estado :");
jPanel1.add(jLabelEstado, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
159, -1, -1));

jLabelIdade.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelIdade.setText("Idade :");
jPanel1.add(jLabelIdade, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, -
1, 20));

txtIdade.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtIdade, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 70, 330,
-1));

jLabelCidade.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelCidade.setText("Cidade :");
jPanel1.add(jLabelCidade, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
190, -1, 20));

jLabelTelefone.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelTelefone.setText("Telefone :");
jPanel1.add(jLabelTelefone, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
100, 90, 20));

txtTelefone.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtTelefone, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 100,
330, -1));

jLabelSenha.setFont(new java.awt.Font("Century Gothic", 0, 18)); // NOI18N
jLabelSenha.setText("Senha:");
jPanel1.add(jLabelSenha, new org.netbeans.lib.awtextra.AbsoluteConstraints(450,
320, -1, -1));

txtSenha.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jPanel1.add(txtSenha, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 350,
320, 30));

jButtonCadastrarPaciente.setText("CADASTRAR");
jButtonCadastrarPaciente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCadastrarPacienteActionPerformed(evt);
    }
});
jPanel1.add(jButtonCadastrarPaciente, new
org.netbeans.lib.awtextra.AbsoluteConstraints(-10, 440, 820, 50));
jPanel1.add(txtCpf, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 370, 360,
25));
```

```
jLabelCpf.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelCpf.setText("CPF :");
jPanel1.add(jLabelCpf, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 360, -
1, 40));
jPanel1.add(txtRg, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 400, 360,
25));
jPanel1.add(txtCep, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 340, 360,
25));
jPanel1.add(txtNumCasa, new org.netbeans.lib.awtextra.AbsoluteConstraints(160,
280, 260, 25));
jPanel1.add(txtBairro, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 250,
330, 23));
jPanel1.add(txtComplemento, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 310, 260, 25));
jPanel1.add(txtRua, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 220, 330,
25));

jLabelSexo.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelSexo.setText("Sexo :");
jPanel1.add(jLabelSexo, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 30, -
1, 40));

jLabelRg.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelRg.setText("RG :");
jPanel1.add(jLabelRg, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 400, -
1, -1));

jLabelRua.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelRua.setText("Rua :");
jPanel1.add(jLabelRua, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 211,
60, 40));

jLabelBairro.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelBairro.setText("Bairro :");
jPanel1.add(jLabelBairro, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
240, 60, 40));

jLabelCep.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelCep.setText("CEP :");
jPanel1.add(jLabelCep, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 330,
40, 40));

jLabelComplemento.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelComplemento.setText("Complemento :");
jPanel1.add(jLabelComplemento, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 300, -1, 40));

jLabelNumCasa.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelNumCasa.setText("Número da casa :");
jPanel1.add(jLabelNumCasa, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
270, -1, 40));
```

```
jSeparador.setForeground(new java.awt.Color(0, 0, 0));
jSeparador.setOrientation(javax.swing.SwingConstants.VERTICAL);
jPanel1.add(jSeparador, new org.netbeans.lib.awtextra.AbsoluteConstraints(440, 0,
20, 450));

jLabel3.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel3.setText("Agora escreva os dados que serão usados ");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 160,
340, 50));

jLabel19.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel19.setText("para seu acesso ao sistema:");
jPanel1.add(jLabel19, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 200, -
1, 30));

txtDescricaoSintomas.setColumns(20);
txtDescricaoSintomas.setLineWrap(true);
txtDescricaoSintomas.setRows(5);
txtDescricaoSintomas.setWrapStyleWord(true);
jScrollPane1.setViewportView(txtDescricaoSintomas);

jPanel1.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(450,
50, 330, 110));

jLabel2.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel2.setText("Descrição sintomas :");
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(450, 20, -1,
-1));

txtSexo.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Masculino", "Feminino", "Outro" }));
txtSexo.setSelectedIndex(-1);
jPanel1.add(txtSexo, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 40, 330,
-1));

txtEstado.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Acre
(AC)", "Alagoas (AL)", "Amapá (AP)", "Amazonas (AM)", "Bahia (BA)", "Ceará (CE)",
"Distrito Federal (DF)", "Espírito Santo (ES)", "Goiás (GO)", "Maranhão (MA)", "Mato
Grosso (MT)", "Mato Grosso do Sul (MS)", "Minas Gerais (MG)", "Pará (PA)", "Paraíba
(PB)", "Paraná (PR)", "Pernambuco (PE)", "Piauí (PI)", "Rio de Janeiro (RJ)", "Rio Grande
do Norte (RN)", "Rio Grande do Sul (RS)", "Rondônia (RO)", "Roraima (RR)", "Santa
Catarina (SC)", "São Paulo (SP)", "Sergipe (SE)", "Tocantins (TO)" }));
txtEstado.setSelectedIndex(-1);
txtEstado.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        txtEstadoItemStateChanged(evt);
    }
});
jPanel1.add(txtEstado, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 160,
330, -1));
```



```

txtCidade.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { " " }));
txtCidade.setSelectedIndex(-1);
jPanel1.add(txtCidade, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 190,
330, -1));

getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
40, 800, 480));

jPanel3.setBackground(new java.awt.Color(13, 130, 203));
jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Century Gothic", 1, 24)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setText("CADASTRO PACIENTE");
jPanel3.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(250, 0, -1,
40));

getContentPane().add(jPanel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 800, 40));

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

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

    ClassePessoa novoClassePessoa = new ClassePessoa(); // instanciando → objeto
    → classe Paciente

    novoClassePessoa.nome = txtNome.getText().trim();
    novoClassePessoa.sexo = (String) txtSexo.getSelectedItem();
    novoClassePessoa.idade = txtIdade.getText().trim();
    novoClassePessoa.email = txtEmail.getText().trim();
    novoClassePessoa.telefone = txtTelefone.getText().trim();
    novoClassePessoa.estado = (String) txtEstado.getSelectedItem();
    novoClassePessoa.cidade = (String) txtCidade.getSelectedItem();
    novoClassePessoa.rua = txtRua.getText().trim();
    novoClassePessoa.bairro = txtBairro.getText().trim();
    novoClassePessoa.numCasa = txtNumCasa.getText().trim();
    novoClassePessoa.complemento = txtComplemento.getText().trim();
    novoClassePessoa.cep = txtCep.getText().trim();

    novoClassePessoa.setCpf(txtCpf.getText().trim());
    novoClassePessoa.setRg(txtRg.getText().trim());

    ClassePaciente novoClassePaciente = new ClassePaciente();
    DAOPaciente novoDadosPacienteDAO = new DAOPaciente(); // instanciando →
objeto → classe DAOPaciente ( para banco de dados)

    novoClassePaciente.descricaoSintomas = (txtDescricaoSintomas.getText().trim());

```



```
ClasseCadastro novoCadastro = new ClasseCadastro();
DAOCadastro novoCadastroUsuarioAcessoPacienteDAO = new DAOCadastro();

novoCadastro.setUsuario(txtUsuario.getText().trim());
novoCadastro.setSenha(txtSenha.getText().trim());

    DAOPessoa novoCadastroPessoaDAO = new DAOPessoa();

    try
    {

        novoCadastroPessoaDAO.CadastroDadosPessoa(novoClassePessoa);

        int chaveEstrangeiraPessoaCadastrada =
novoCadastroPessoaDAO.BuscaIdPessoaCadastrada(txtRg.getText());

novoDadosPacienteDAO.CadastroDadosUsuario(novoClassePaciente,chaveEstrangeiraPessoaCadastrada);

novoCadastroUsuarioAcessoPacienteDAO.CadastroAcessoLoginUsuario(novoCadastro,c
haveEstrangeiraPessoaCadastrada);

    }
    catch (Exception ex)
    {

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

    }

        JOptionPane.showMessageDialog(null, "" + novoClassePessoa.nome+ " foi
cadastrado com sucesso!");

        int resposta =  JOptionPane.showConfirmDialog(null, "Deseja Volta a tela de
Login?", "Mensagem", YES_NO_OPTION);

        if (resposta == JOptionPane.YES_OPTION)
        {

            new ViewLogin().setVisible(true);
            this.dispose();

        }

        else if(resposta == JOptionPane.NO_OPTION)
        {
```

```
txtNome.setText("");
txtSexo.setEnabled(false);
txtIdade.setText("");
txtEmail.setText("");
txtTelefone.setText("");
txtEstado.setEnabled(false);
txtCidade.setEnabled(false);
txtRua.setText("");
txtBairro.setText("");
txtComplemento.setText("");
txtCep.setText("");
txtNumCasa.setText("");
txtDescricaoSintomas.setText("");
txtCpf.setText("");
txtRg.setText("");
txtUsuario.setText("");
txtSenha.setText("");

}

}

private void txtEstadoItemStateChanged(java.awt.event.ItemEvent evt) {
    if(evt.getStateChange() == ItemEvent.SELECTED){

        DAOCidades listacidade = new DAOCidades();
        List<ClasseCidade> cidades = new ArrayList();

        try {
            txtCidade.removeAllItems();

            cidades = listacidade.ListarCidades(txtEstado.getSelectedItem().toString());

            for (ClasseCidade cidade : cidades)
            {
                txtCidade.addItem(cidade.nomeCidade.toString());
            }

        } catch (Exception ex){

            JOptionPane.showMessageDialog(null,"Não foi possível listar as cidades" , "",
            JOptionPane.ERROR_MESSAGE);

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

    }

}
```

```
public static void main(String args[])
{
    java.awt.EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            new ViewRegistrarLoginPaciente().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButtonCadastrarPaciente;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabelBairro;
private javax.swing.JLabel jLabelCep;
private javax.swing.JLabel jLabelCidade;
private javax.swing.JLabel jLabelComplemento;
private javax.swing.JLabel jLabelCpf;
private javax.swing.JLabel jLabelEmail;
private javax.swing.JLabel jLabelEstado;
private javax.swing.JLabel jLabelIdade;
private javax.swing.JLabel jLabelNome;
private javax.swing.JLabel jLabelNumCasa;
private javax.swing.JLabel jLabelRg;
private javax.swing.JLabel jLabelRua;
private javax.swing.JLabel jLabelSenha;
private javax.swing.JLabel jLabelSexo;
private javax.swing.JLabel jLabelTelefone;
private javax.swing.JLabel jLabelUsuario;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSeparator jSeparador;
public javax.swing.JTextField txtBairro;
public javax.swing.JTextField txtCep;
private javax.swing.JComboBox<String> txtCidade;
public javax.swing.JTextField txtComplemento;
private javax.swing.JTextField txtCpf;
public javax.swing.JTextArea txtDescricaoSintomas;
public javax.swing.JTextField txtEmail;
public javax.swing.JComboBox<String> txtEstado;
public javax.swing.JTextField txtIdade;
public javax.swing.JTextField txtNome;
public javax.swing.JTextField txtNumCasa;
private javax.swing.JTextField txtRg;
public javax.swing.JTextField txtRua;
```

```
private javax.swing.JPasswordField txtSenha;  
public javax.swing.JComboBox<String> txtSexo;  
public javax.swing.JTextField txtTelefone;  
private javax.swing.JTextField txtUsuario;  
// End of variables declaration  
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package visão;  
  
import modeloBeans.ClasseAgendamento;  
import modeloBeans.ClassePaciente;  
import modeloBeans.ClasseCadastro;  
import modeloConexao.ConexaoBD;  
  
public class TelaPrincipal extends javax.swing.JFrame {  
  
    ClasseCadastro userLogado = null;  
    String usuarioLogado;  
    ConexaoBD conecta = new ConexaoBD();//Instanciando a classe conexão  
  
    public TelaPrincipal(ClasseCadastro user, String usuario) {  
        initComponents();  
        userLogado = user;  
        usuarioLogado = usuario;  
        conecta.getConnection();//Chama todo o método conexão da classe ConexaoBD  
  
        jLabelUsuariologadoEquipeDeSaude.setText(usuario);  
    }  
  
    /**  
     * This method is called from within the constructor to initialize the form.  
     * WARNING: Do NOT modify this code. The content of this method is always  
     * regenerated by the Form Editor.  
     */  
    @SuppressWarnings("unchecked")  
    // <editor-fold defaultstate="collapsed" desc="Generated Code">  
    private void initComponents() {  
  
        jLabel1 = new javax.swing.JLabel();  
        jInternalFrameBemVindo = new javax.swing.JInternalFrame();  
        jPanelInternal = new javax.swing.JPanel();  
        jButtonAgenda = new javax.swing.JButton();  

```

```
jLabel2 = new javax.swing.JLabel();
jLabelUsuariologadoEquipeDeSaude = new javax.swing.JLabel();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jButton3 = new javax.swing.JButton();
jLabelGerenciamento = new javax.swing.JLabel();
jButtonSairBemVindo = new javax.swing.JButton();
jLabelFundoPrincipal = new javax.swing.JLabel();
jMenuBar1 = new javax.swing.JMenuBar();
jMenuCadastro = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenuItem2 = new javax.swing.JMenuItem();
jMenuItem4 = new javax.swing.JMenuItem();
jMenuItem5 = new javax.swing.JMenuItem();
jMenuFerramentas = new javax.swing.JMenu();
jMenuItemTelaBemVindo = new javax.swing.JMenuItem();
jMenu3 = new javax.swing.JMenu();
jMenuItem6 = new javax.swing.JMenuItem();
jMenu1 = new javax.swing.JMenu();
jMenuItem3 = new javax.swing.JMenuItem();
jMenuSair = new javax.swing.JMenu();
jMenuItemSair = new javax.swing.JMenuItem();

jLabel1.setText("jLabel1");

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(null);

jInternalFrameBemVindo.setBackground(new java.awt.Color(13, 130, 203));
jInternalFrameBemVindo.setTitle("Bem-Vindo");
jInternalFrameBemVindo.setVisible(true);
jInternalFrameBemVindo.getContentPane().setLayout(null);

jPanelInternal.setBackground(new java.awt.Color(13, 130, 203));
jPanelInternal.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jPanelInternal.setLayout(null);

jButtonAgenda.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Agendatesteste.png"))); // NOI18N
jButtonAgenda.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonAgendaActionPerformed(evt);
    }
});
jPanelInternal.add(jButtonAgenda);
jButtonAgenda.setBounds(10, 60, 90, 100);

jLabel2.setFont(new java.awt.Font("Century Gothic", 1, 14)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setText("Usuário Logado:");
jPanelInternal.add(jLabel2);
jLabel2.setBounds(10, -10, 120, 50);
```

```
jLabelUsuariologadoEquipeDeSaude.setFont(new java.awt.Font("Century Gothic", 1,
14)); // NOI18N
jLabelUsuariologadoEquipeDeSaude.setForeground(new java.awt.Color(255, 255,
255));
jPanelInternal.add(jLabelUsuariologadoEquipeDeSaude);
jLabelUsuariologadoEquipeDeSaude.setBounds(130, 0, 110, 30);

jButton1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/agendateste2.png"))); //
NOI18N
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jPanelInternal.add(jButton1);
jButton1.setBounds(110, 60, 90, 100);

jButton2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/PESQ.png"))); // NOI18N
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
jPanelInternal.add(jButton2);
jButton2.setBounds(210, 60, 90, 100);

jButton3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Forum.png"))); // NOI18N
jPanelInternal.add(jButton3);
jButton3.setBounds(310, 60, 90, 100);

jInternalFrameBemVindo.getContentPane().add(jPanelInternal);
jPanelInternal.setBounds(0, 40, 410, 180);

jLabelGerenciamento.setFont(new java.awt.Font("Century Gothic", 1, 18)); // NOI18N
jLabelGerenciamento.setForeground(new java.awt.Color(255, 255, 255));
jLabelGerenciamento.setText("SISTEMA DE GERENCIAMENTO DE: ");
jInternalFrameBemVindo.getContentPane().add(jLabelGerenciamento);
jLabelGerenciamento.setBounds(10, 10, 300, 23);

jButtonSairBemVindo.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/shut-down.png"))); // NOI18N
jButtonSairBemVindo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonSairBemVindoActionPerformed(evt);
    }
});
jInternalFrameBemVindo.getContentPane().add(jButtonSairBemVindo);
jButtonSairBemVindo.setBounds(350, 0, 40, 30);
```

```
getContentPane().add(jInternalFrameBemVindo);
jInternalFrameBemVindo.setBounds(160, 100, 430, 250);

jLabelFundoPrincipal.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/FundoView1.png"))); // NOI18N
getContentPane().add(jLabelFundoPrincipal);
jLabelFundoPrincipal.setBounds(0, -20, 710, 480);

jMenuCadastro.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Menu_1.png"))); // NOI18N
jMenuCadastro.setText("Funções");

jMenuItem1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Agendateste.png"))); // NOI18N
jMenuItem1.setText("Registrar agendamento");
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem1ActionPerformed(evt);
    }
});
jMenuCadastro.add(jMenuItem1);

jMenuItem2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/agendateste2.png"))); //
NOI18N
jMenuItem2.setText("Cadastrar Teste COVID-19");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenuCadastro.add(jMenuItem2);

jMenuItem4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/PESQ.png"))); // NOI18N
jMenuItem4.setText("Visualizar Pacientes");
jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem4ActionPerformed(evt);
    }
});
jMenuCadastro.add(jMenuItem4);

jMenuItem5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Forum.png"))); // NOI18N
jMenuItem5.setText("Enviar mensagens Fórum");
jMenuItem5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem5ActionPerformed(evt);
    }
});
```

```
jMenuCadastro.add(jMenuItem5);

jMenuBar1.add(jMenuCadastro);

jMenuFerramentas.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Ferramenta.png"))); // NOI18N
jMenuFerramentas.setText("Ferramentas");

jMenuItemTelaBemVindo.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.ev
ent.KeyEvent.VK_M, java.awt.event.InputEvent.CTRL_MASK));
jMenuItemTelaBemVindo.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Menu inicial.gif"))); // NOI18N
jMenuItemTelaBemVindo.setText("Bem-Vindo");
jMenuItemTelaBemVindo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItemTelaBemVindoActionPerformed(evt);
    }
});
jMenuFerramentas.add(jMenuItemTelaBemVindo);

jMenuBar1.add(jMenuFerramentas);

jMenu3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Teste_2.png"))); // NOI18N
jMenu3.setText("Mais");

jMenuItem6.setText("Tela Login");
jMenuItem6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem6ActionPerformed(evt);
    }
});
jMenu3.add(jMenuItem6);

jMenuBar1.add(jMenu3);

jMenu1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/information_finish.png"))); //
NOI18N
jMenu1.setText("Sobre");

jMenuItem3.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.
VK_C, java.awt.event.InputEvent.CTRL_MASK));
jMenuItem3.setText("Criadores");
jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem3ActionPerformed(evt);
    }
});
jMenu1.add(jMenuItem3);
```



```
jMenuBar1.add(jMenu1);

jMenuSair.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Sair.png"))); // NOI18N
jMenuSair.setText("Sair");

jMenuItemSair.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S, java.awt.event.InputEvent.CTRL_MASK));
jMenuItemSair.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Sair.png"))); // NOI18N
jMenuItemSair.setText("Sair");
jMenuItemSair.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItemSairActionPerformed(evt);
    }
});
jMenuSair.add(jMenuItemSair);

jMenuBar1.add(jMenuSair);

setJMenuBar(jMenuBar1);

setSize(new java.awt.Dimension(726, 514));
setLocationRelativeTo(null);
} // </editor-fold>

private void jButtonSairBemVindoActionPerformed(java.awt.event.ActionEvent evt) {
    jInternalFrameBemVindo.dispose();
}

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

    System.exit(0);
}

private void jMenuItemTelaBemVindoActionPerformed(java.awt.event.ActionEvent evt) {
    jInternalFrameBemVindo.setVisible(true);
}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {

    ViewCriadores mostra = new ViewCriadores();
    mostra.setVisible(true);
}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {

    ViewRegistrarAgendamento agendar = new
```

```
ViewRegistrarAgendamento(usuarioLogado);
    agendar.setVisible(true);

}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    ViewRegistrarTesteDoAgendamento registrarTeste = new
ViewRegistrarTesteDoAgendamento(usuarioLogado);
    registrarTeste.setVisible(true);

}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
    ViewVisualizarPacientes visualizar = new ViewVisualizarPacientes(usuarioLogado);
    visualizar.setVisible(true);
}

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    ViewEnviarMensagens enviarMensagens = new
ViewEnviarMensagens(usuarioLogado);
    enviarMensagens.setVisible(true);
}

private void jButtonAgendaActionPerformed(java.awt.event.ActionEvent evt) {
    ViewRegistrarAgendamento mostra = new
ViewRegistrarAgendamento(usuarioLogado);
    mostra.setVisible(true);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    ViewRegistrarTesteDoAgendamento mostra = new
ViewRegistrarTesteDoAgendamento(usuarioLogado);
    mostra.setVisible(true);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    ViewVisualizarPacientes visualizar = new ViewVisualizarPacientes(usuarioLogado);
    visualizar.setVisible(true);
}

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
    ViewLogin login = new ViewLogin();
    login.setVisible(true);
    login.dispose();
}

/**
 * @param args the command line arguments
 */
```

```
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButtonAgenda;
private javax.swing.JButton jButtonSairBemVindo;
private javax.swing.JFrame jInternalFrameBemVindo;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabelFundoPrincipal;
private javax.swing.JLabel jLabelGerenciamento;
public javax.swing.JLabel jLabelUsuariologadoEquipeDeSaude;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenu jMenuCadastro;
private javax.swing.JMenu jMenuFerramentas;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JMenuItem jMenuItem5;
private javax.swing.JMenuItem jMenuItem6;
private javax.swing.JMenuItem jMenuItemSair;
private javax.swing.JMenuItem jMenuItemTelaBemVindo;
private javax.swing.JMenu jMenuSair;
private javax.swing.JPanel jPanelInternal;
// End of variables declaration
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package visão;

import modeloBeans.ClasseAgendamento;
import modeloBeans.ClassePaciente;
import modeloBeans.ClasseCadastro;
import modeloConexao.ConexaoBD;

public class TelaPrincipalPaciente extends javax.swing.JFrame {

    ClasseCadastro userLogado = null;
    String usuarioLogado;
```

```
ConexaoBD conecta = new ConexaoBD();//Instanciando a classe conexão

public TelaPrincipalPaciente(ClasseCadastro user, String usuario) {
    initComponents();
    userLogado = user;
    usuarioLogado = usuario;
    conecta.getConnection();//Chama todo o método conexao da classe ConexaoBD

    jLabelUsuariologadoEquipeDeSaude.setText(usuario);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jInternalFrameBemVindo = new javax.swing.JInternalFrame();
    jPanelInternal = new javax.swing.JPanel();
    jButtonAgenda = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    jLabelUsuariologadoEquipeDeSaude = new javax.swing.JLabel();
    jLabelGerenciamento = new javax.swing.JLabel();
    jButtonSairBemVindo = new javax.swing.JButton();
    jLabelFundoPrincipal = new javax.swing.JLabel();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenuCadastro = new javax.swing.JMenu();
    jMenuItem5 = new javax.swing.JMenuItem();
    jMenuFerramentas = new javax.swing.JMenu();
    jMenuItemTelaBemVindo = new javax.swing.JMenuItem();
    jMenu3 = new javax.swing.JMenu();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenu1 = new javax.swing.JMenu();
    jMenuItem3 = new javax.swing.JMenuItem();
    jMenuSair = new javax.swing.JMenu();
    jMenuItemSair = new javax.swing.JMenuItem();

    jLabel1.setText("jLabel1");

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    jInternalFrameBemVindo.setBackground(new java.awt.Color(13, 130, 203));
    jInternalFrameBemVindo.setTitle("Bem-Vindo");
    jInternalFrameBemVindo.setVisible(true);
    jInternalFrameBemVindo.getContentPane().setLayout(null);
```

```
jPanelInternal.setBackground(new java.awt.Color(13, 130, 203));
jPanelInternal.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jPanelInternal.setLayout(null);

jButtonAgenda.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Forum.png"))); // NOI18N
jButtonAgenda.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonAgendaActionPerformed(evt);
    }
});
jPanelInternal.add(jButtonAgenda);
jButtonAgenda.setBounds(110, 40, 130, 120);

jLabel2.setFont(new java.awt.Font("Century Gothic", 1, 14)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setText("Usuário Logado:");
jPanelInternal.add(jLabel2);
jLabel2.setBounds(10, -10, 120, 50);

jLabelUsuariologadoEquipeDeSaude.setFont(new java.awt.Font("Century Gothic", 1,
14)); // NOI18N
jLabelUsuariologadoEquipeDeSaude.setForeground(new java.awt.Color(255, 255,
255));
jPanelInternal.add(jLabelUsuariologadoEquipeDeSaude);
jLabelUsuariologadoEquipeDeSaude.setBounds(130, 0, 110, 30);

jInternalFrameBemVindo.getContentPane().add(jPanelInternal);
jPanelInternal.setBounds(0, 40, 360, 180);

jLabelGerenciamento.setFont(new java.awt.Font("Century Gothic", 1, 18)); // NOI18N
jLabelGerenciamento.setForeground(new java.awt.Color(255, 255, 255));
jLabelGerenciamento.setText("SISTEMA DE GERENCIAMENTO DE: ");
jInternalFrameBemVindo.getContentPane().add(jLabelGerenciamento);
jLabelGerenciamento.setBounds(10, 10, 300, 23);

jButtonSairBemVindo.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/shut-down.png"))); // NOI18N
jButtonSairBemVindo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonSairBemVindoActionPerformed(evt);
    }
});
jInternalFrameBemVindo.getContentPane().add(jButtonSairBemVindo);
jButtonSairBemVindo.setBounds(320, 0, 40, 30);

getContentPane().add(jInternalFrameBemVindo);
jInternalFrameBemVindo.setBounds(180, 100, 380, 250);

jLabelFundoPrincipal.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/FundoView1.png"))); // NOI18N
getContentPane().add(jLabelFundoPrincipal);
```

```
jLabelFundoPrincipal.setBounds(0, -20, 710, 480);

jMenuCadastro.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Menu_1.png"))); // NOI18N
jMenuCadastro.setText("Funções");

jMenuItem5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Forum.png"))); // NOI18N
jMenuItem5.setText("Enviar mensagens Fórum");
jMenuItem5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem5ActionPerformed(evt);
    }
});
jMenuCadastro.add(jMenuItem5);

jMenuBar1.add(jMenuCadastro);

jMenuFerramentas.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Ferramenta.png"))); // NOI18N
jMenuFerramentas.setText("Ferramentas");

jMenuItemTelaBemVindo.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.ev
ent.KeyEvent.VK_M, java.awt.event.InputEvent.CTRL_MASK));
jMenuItemTelaBemVindo.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Menu inicial.gif"))); // NOI18N
jMenuItemTelaBemVindo.setText("Bem-Vindo");
jMenuItemTelaBemVindo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItemTelaBemVindoActionPerformed(evt);
    }
});
jMenuFerramentas.add(jMenuItemTelaBemVindo);

jMenuBar1.add(jMenuFerramentas);

jMenu3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Teste_2.png"))); // NOI18N
jMenu3.setText("Mais");

jMenuItem1.setText("Tela login");
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem1ActionPerformed(evt);
    }
});
jMenu3.add(jMenuItem1);

jMenuBar1.add(jMenu3);

jMenu1.setIcon(new
```

```

javax.swing.ImageIcon(getClass().getResource("/imagens/information_finish.png")))); //
NOI18N
    jMenuItem1.setText("Sobre");

jMenuItem3.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.
VK_C, java.awt.event.InputEvent.CTRL_MASK));
    jMenuItem3.setText("Criadores");
    jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItem3ActionPerformed(evt);
        }
    });
    jMenuItem1.add(jMenuItem3);

    jMenuItemBar1.add(jMenuItem1);

    jMenuItemSair.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Sair.png")))); // NOI18N
    jMenuItemSair.setText("Sair");

jMenuItemSair.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEv
ent.VK_S, java.awt.event.InputEvent.CTRL_MASK));
    jMenuItemSair.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/Sair.png")))); // NOI18N
    jMenuItemSair.setText("Sair");
    jMenuItemSair.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItemSairActionPerformed(evt);
        }
    });
    jMenuItemSair.add(jMenuItemSair);

    jMenuItemBar1.add(jMenuItemSair);

    setJMenuBar(jMenuBar1);

    setSize(new java.awt.Dimension(726, 514));
    setLocationRelativeTo(null);
} // </editor-fold>

private void jButtonSairBemVindoActionPerformed(java.awt.event.ActionEvent evt) {
    jInternalFrameBemVindo.dispose();
}

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

    System.exit(0);
}

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

```

```
jInternalFrameBemVindo.setVisible(true);

}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {

    ViewCriadores mostra = new ViewCriadores();
    mostra.setVisible(true);

}

private void jButtonAgendaActionPerformed(java.awt.event.ActionEvent evt) {
    ViewEnviarMensagens enviarMensagens = new
ViewEnviarMensagens(usuarioLogado);
    enviarMensagens.setVisible(true);
}

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {
    ViewEnviarMensagens enviarMensagens = new
ViewEnviarMensagens(usuarioLogado);
    enviarMensagens.setVisible(true);
}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    ViewLogin login = new ViewLogin();
    login.setVisible(true);
    login.dispose();
}

/**
 * @param args the command line arguments
 */

// Variables declaration - do not modify
private javax.swing.JButton jButtonAgenda;
private javax.swing.JButton jButtonSairBemVindo;
private javax.swing.JInternalFrame jInternalFrameBemVindo;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabelFundoPrincipal;
private javax.swing.JLabel jLabelGerenciamento;
public javax.swing.JLabel jLabelUsuariologadoEquipeDeSaude;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenu jMenuCadastro;
private javax.swing.JMenu jMenuFerramentas;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JMenuItem jMenuItem5;
```



```
private javax.swing.JMenuItem jMenuItemSair;  
private javax.swing.JMenuItem jMenuItemTelaBemVindo;  
private javax.swing.JMenu jMenuSair;  
private javax.swing.JPanel jPanelInternal;  
// End of variables declaration  
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package visão;
```

```
import static com.sun.org.apache.xalan.internal.xsltc.compiler.util.Type.Int;  
import java.awt.Color;  
import java.awt.event.ItemEvent;  
import java.sql.Date;  
import java.sql.Time;  
import java.sql.Timestamp;  
import java.text.DateFormat;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Vector;  
import modeloBeans.ClasseEquipeSaude;  
import modeloDao.DAOCadastro;  
import modeloDao.DAOPessoa;  
import modeloDao.DAOWebservice;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.ComboBoxModel;  
import javax.swing.DefaultComboBoxModel;  
import javax.swing.DefaultListModel;  
import javax.swing.JOptionPane;  
import static javax.swing.JOptionPane.YES_NO_OPTION;  
import static jdk.nashorn.internal.objects.NativeString.substring;  
import modeloBeans.ClasseAgendamento;  
import modeloBeans.ClasseCadastro;  
import modeloBeans.ClasseCidade;  
import modeloBeans.ClassePessoa;  
import modeloConexao.ConexaoBD;  
import modeloDao.DAOWebservice;  
import modeloDao.DAOWebservice;
```

```
/**
```

```
*
* @author
*/
public class ViewAlterarAgendamento extends javax.swing.JFrame {

    int Enter = 0;
    String usuarioLogado;
    public List<ClasseAgendamento> resultado = new ArrayList();
    public ViewAlterarAgendamento(String userLogado ) {

        initComponents();

        usuarioLogado = userLogado;
        txtNomeAgendado.setEditable(false);
        txtPacienteIdAgendado.setEditable(false);
        txtHoraVisita.setEditable(false);
        txtDataVisita.setEditable(false);
        jListAgendados.setVisible(false);

    }

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

        jLabel2 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        jButtonAlterarAgendado = new javax.swing.JButton();
        txtPesquisaPacientesAgendados = new javax.swing.JTextField();
        jLabelPesquisa = new javax.swing.JLabel();
        txtNomeAgendado = new javax.swing.JTextField();
        txtPacienteIdAgendado = new javax.swing.JTextField();
        txtDataVisita = new javax.swing.JTextField();
        txtHoraVisita = new javax.swing.JTextField();
        jLabelNome = new javax.swing.JLabel();
        jLabelCodigoPaciente = new javax.swing.JLabel();
        jLabelDataVisita = new javax.swing.JLabel();
        jLabelHoraVisita = new javax.swing.JLabel();
        jListAgendados = new javax.swing.JList<>();
        txtStatus = new javax.swing.JComboBox<>();
        jLabel3 = new javax.swing.JLabel();
        jSeparator2 = new javax.swing.JSeparator();
        jSeparator3 = new javax.swing.JSeparator();
        jSeparator4 = new javax.swing.JSeparator();
        jPanel3 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();

        jLabel2.setText("jLabel2");

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setResizable(false);
    }
}
```

```
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jButtonAlterarAgendado.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jButtonAlterarAgendado.setText("ALTERAR");
jButtonAlterarAgendado.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
jButtonAlterarAgendado.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonAlterarAgendamentoActionPerformed(evt);
    }
});
jPanel1.add(jButtonAlterarAgendado, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 390, 700, 50));

txtPesquisaPacientesAgendados.setBorder(javax.swing.BorderFactory.createLineBorder(
new java.awt.Color(0, 0, 0)));
txtPesquisaPacientesAgendados.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPesquisaPacientesAgendadosActionPerformed(evt);
    }
});
txtPesquisaPacientesAgendados.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtPesquisaPacientesAgendadosKeyReleased(evt);
    }
});
jPanel1.add(txtPesquisaPacientesAgendados, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 20, 380, 20));

jLabelPesquisa.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelPesquisa.setText("Pesquisa :");
jPanel1.add(jLabelPesquisa, new org.netbeans.lib.awtextra.AbsoluteConstraints(80,
20, -1, -1));
jPanel1.add(txtNomeAgendado, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 210, 260, -1));
jPanel1.add(txtPacienteIdAgendado, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 210, 270, -1));
jPanel1.add(txtDataVisita, new org.netbeans.lib.awtextra.AbsoluteConstraints(160,
300, 190, -1));
jPanel1.add(txtHoraVisita, new org.netbeans.lib.awtextra.AbsoluteConstraints(170,
340, 180, -1));

jLabelNome.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelNome.setText("Nome paciente agendado :");
jPanel1.add(jLabelNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
180, 250, 20));
```

```
jLabelCodigoPaciente.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelCodigoPaciente.setText("Código Agendamento :");
jPanel1.add(jLabelCodigoPaciente, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 180, -1, -1));

jLabelDataVisita.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelDataVisita.setText("Data agendada :");
jPanel1.add(jLabelDataVisita, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
300, 160, -1));

jLabelHoraVisita.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelHoraVisita.setText("Horário agendado :");
jPanel1.add(jLabelHoraVisita, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
340, -1, -1));

jListAgendados.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
jListAgendados.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jListAgendadosMouseClicked(evt);
    }
});
jPanel1.add(jListAgendados, new org.netbeans.lib.awtextra.AbsoluteConstraints(170,
40, 380, 90));

txtStatus.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Pendente", "Concluída", "Cancelada" }));
txtStatus.setSelectedIndex(-1);
jPanel1.add(txtStatus, new org.netbeans.lib.awtextra.AbsoluteConstraints(420, 330,
250, -1));

jLabel3.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel3.setText("Estado da visita agendada:");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(420, 291, -1,
30));
jPanel1.add(jSeparator2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 270,
710, -1));
jPanel1.add(jSeparator3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 150, -
1, -1));
jPanel1.add(jSeparator4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 150,
710, -1));

getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
40, 700, 440));

jPanel3.setBackground(new java.awt.Color(13, 130, 203));
jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Century Gothic", 1, 24)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setText("ALTERAR AGENDAMENTO");
jPanel3.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 0, -1,
```

```
40));

    getContentPane().add(jPanel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 700, 40));

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

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

    ClasseAgendamento novoAgendamento = new ClasseAgendamento(); //
instanciando → objeto → classe Paciente
    DAOAgendamento novoCadastroAgendamento = new DAOAgendamento();

    novoAgendamento.nomePaciente = txtNomeAgendado.getText().trim();
    novoAgendamento.agendamentoid =
Integer.parseInt(txtPacienteldAgendado.getText().trim());
    novoAgendamento.Hora = txtHoraVisita.getText().trim();
    novoAgendamento.Data = txtDataVisita.getText().trim();
    novoAgendamento.Status = (String) txtStatus.getSelectedItemAt();

    String formato = "dd/MM/yyyy";
    java.util.Date date = null;

    try {
        date = new SimpleDateFormat(formato).parse(novoAgendamento.Data);

    } catch (ParseException ex) {
        Logger.getLogger(ViewAlterarAgendamento.class.getName()).log(Level.SEVERE,
null, ex);
    }

    try
    {

novoCadastroAgendamento.AlterarAgendamento(novoAgendamento, usuarioLogado, date)
;

        JOptionPane.showMessageDialog(null, " O agendamento foi alterado com
sucesso!");

    }
    catch (Exception ex)
    {

        JOptionPane.showMessageDialog(null, " O agendamento não foi alterado com
sucesso!");
    }
}
```

```
}

txtNomeAgendado.setText("");
txtPacientIdAgendado.setText("");
txtDataVisita.setText("");
txtHoraVisita.setText("");
txtStatus.setSelectedIndex(-1);

}

private void
txtPesquisaPacientesAgendadosActionPerformed(java.awt.event.ActionEvent evt) {
    jListAgendados.setVisible(false);
    Enter = 1;
}

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

    int linhaClicada = jListAgendados.getSelectedIndex();

    DAOAgendamento itemClicado = new DAOAgendamento();

    try {
        List <ClasseAgendamento> setarValores = new ArrayList();

        setarValores =
itemClicado.ListarItensAgendadosClicado(linhaClicada,txtPesquisaPacientesAgendados.g
etText(),resultado.get(linhaClicada).codigo);

        String texto = String.valueOf(setarValores.get(0).agendamentId);

        txtPacientIdAgendado.setText(texto);

        txtNomeAgendado.setText(setarValores.get(0).nomePaciente);

        txtHoraVisita.setText(setarValores.get(0).Hora);

        txtStatus.setSelectedItem(setarValores.get(0).Status);

        //2020-05-28
        String ano;
        String barra = "/";
        String mes;
        String dia;
```

```
        ano = setarValores.get(0).Data.substring ( 0 , 4 );
        mes = setarValores.get(0).Data.substring ( 5 , 7 );
        dia = setarValores.get(0).Data.substring ( 8 , 10 );

        String data1 = dia.concat(barra);
        String data2 = data1.concat(mes);
        String data3 = data2.concat(barra);
        String datacompleta = data3.concat(ano);

        txtDataVisita.setText(datacompleta);

        //txtDataVisita.setText(setarValores.get(0).Data);

        txtHoraVisita.setEditable(true);
        txtDataVisita.setEditable(true);

    } catch (Exception ex) {
        Logger.getLogger(ViewAlterarAgendamento.class.getName()).log(Level.SEVERE,
        null, ex);
    }

}

private void txtPesquisaPacientesAgendadosKeyReleased(java.awt.event.KeyEvent
evt) {
    jListAgendados.setVisible(false);

    DefaultListModel listModel = new DefaultListModel();

    if (Enter == 0) {

        DAOAgendamento agendamento = new DAOAgendamento();

        try {

            resultado =
            agendamento.ListarNomesPacientesAgendadosSistema(txtPesquisaPacientesAgendados
```

```
.getText());

        if (agendamento.mostraMax != 5) {
            for (ClasseAgendamento listapesquisada : resultado) {
                listModel.addElement(listapesquisada.nomePaciente);
            }

            jListAgendados.setModel(listModel);

            jListAgendados.setVisible(true);

        } else {

            jListAgendados.setVisible(false);

        }
    } catch (Exception ex) {

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

    } else {

        Enter = 0;

    }

}

/* public static void main(String args[])
{
    java.awt.EventQueue.invokeLater(new Runnable()

    {
        public void run()
        {
            new ViewRegistrarAgendamento().setVisible(true);
        }
    });
}*/

// Variables declaration - do not modify
private javax.swing.JButton jButtonAlterarAgendado;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
```



```

private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabelCodigoPaciente;
private javax.swing.JLabel jLabelDataVisita;
private javax.swing.JLabel jLabelHoraVisita;
private javax.swing.JLabel jLabelNome;
private javax.swing.JLabel jLabelPesquisa;
public javax.swing.JList<String> jListAgendados;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel3;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JSeparator jSeparator4;
public javax.swing.JTextField txtDataVisita;
public javax.swing.JTextField txtHoraVisita;
public javax.swing.JTextField txtNomeAgendado;
public javax.swing.JTextField txtPacienteldAgendado;
private javax.swing.JTextField txtPesquisaPacientesAgendados;
private javax.swing.JComboBox<String> txtStatus;
// End of variables declaration
}

```

```
package visão;
```

```
public class ViewCriadores extends javax.swing.JFrame {
```

```

    public ViewCriadores() {
        initComponents();
    }

```

```

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

```

```

        jLabel2 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();

```

```
        jLabel2.setIcon(new
```

[illegible]

```
public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ViewCriadores().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea jTextArea1;
// End of variables declaration
}
```

```
package visão;

import java.awt.Color;
import static java.awt.Color.BLUE;
import static java.awt.Color.WHITE;
import java.sql.Date;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;
import java.util.TimeZone;
import java.util.logging.Level;
import java.util.logging.Logger;
import modeloBeans.ClasseForum;
import modeloDao.DAOForum;

public class ViewEnviarMensagens extends javax.swing.JFrame {

    String usuarioLogado;

    public ViewEnviarMensagens(String userLogado) {
        initComponents();
        usuarioLogado = userLogado;
        txtForum.setEditable(false);
        DAOForum forum = new DAOForum();
    }
}
```

```

List <ClasseForum> listadosforum = new ArrayList();
try {

    listadosforum = forum.ListarDados();

    for (ClasseForum listardados : listadosforum){

        txtForum.append(listardados.userMensagem);
        txtForum.append(" ");
        txtForum.append(listardados.tipouser);
        txtForum.append(" ) : ");
        txtForum.append(listardados.mensagem);
        txtForum.append("\n");
    }
} catch (Exception ex) {
    Logger.getLogger(ViewEnviarMensagens.class.getName()).log(Level.SEVERE,
null, ex);
}

}

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

    jScrollPane1 = new javax.swing.JScrollPane();
    jTextPane1 = new javax.swing.JTextPane();
    jLabel1 = new javax.swing.JLabel();
    txtMensagem = new javax.swing.JTextArea();
    jButton1 = new javax.swing.JButton();
    jScrollPane2 = new javax.swing.JScrollPane();
    txtForum = new javax.swing.JTextArea();
    jLabel4 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jScrollPane3 = new javax.swing.JScrollPane();

    jTextPane1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

    jTextPane1.setText("dsdddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd");
    jScrollPane1.setViewportView(jTextPane1);

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setResizable(false);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel1.setFont(new java.awt.Font("Century Gothic", 1, 16)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(255, 255, 255));
    jLabel1.setText("Escreva sua mensagem :");
    getContentPane().add(jLabel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 430, -1, -1));

```

```
txtMensagem.setColumns(20);
txtMensagem.setLineWrap(true);
txtMensagem.setRows(5);
txtMensagem.setWrapStyleWord(true);
txtMensagem.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
txtMensagem.setDisabledTextColor(new java.awt.Color(0, 0, 0));
getContentPane().add(txtMensagem, new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 430, 470, 90));

jButton1.setFont(new java.awt.Font("Century Gothic", 1, 11)); // NOI18N
jButton1.setText("Enviar Mensagem");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonEnviarMensagem(evt);
    }
});
getContentPane().add(jButton1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 530, 470, -1));

txtForum.setColumns(20);
txtForum.setFont(new java.awt.Font("Monospaced", 1, 13)); // NOI18N
txtForum.setLineWrap(true);
txtForum.setRows(5);
txtForum.setToolTipText("");
txtForum.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
txtForum.setDisabledTextColor(new java.awt.Color(0, 0, 0));
txtForum.setEnabled(false);
jScrollPane2.setViewportView(txtForum);

getContentPane().add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 50, 680, 330));

jLabel4.setFont(new java.awt.Font("Calibri", 1, 36)); // NOI18N
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setText("Aba de comunicação - \"Fórum\"");
getContentPane().add(jLabel4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 0, -1, 50));

jLabel3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/FundoView1.png"))); // NOI18N
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 790, 600));
getContentPane().add(jScrollPane3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(350, 380, 370, 90));

pack();
setLocationRelativeTo(null);
} // </editor-fold>
```

```
private void jButtonEnviarMensagem(java.awt.event.ActionEvent evt) {

    ClasseForum enviarmensagens = new ClasseForum();

    enviarmensagens.userMensagem = usuarioLogado;
    enviarmensagens.mensagem = txtMensagem.getText();

    Calendar calendar = Calendar.getInstance(TimeZone.getTimeZone("Brazil/East"));
    int hora = calendar.get(Calendar.HOUR_OF_DAY);
    int minuto = calendar.get(Calendar.MINUTE);

    String horarioEnvioMensagem = hora + ":" + minuto;

    Date data = new Date(System.currentTimeMillis());

    enviarmensagens.data = data;
    enviarmensagens.hora = horarioEnvioMensagem;

    DAOForum forum = new DAOForum();
    boolean paciente = false;

    int chaveEstrangeiraUsuarioMensagem = 0;

    try {

        List <ClasseForum> listaultimodado = new ArrayList();

        chaveEstrangeiraUsuarioMensagem =
forum.BuscaIdUsuarioMensagem(enviarmensagens.userMensagem);

        paciente = forum.VerificarTipo(chaveEstrangeiraUsuarioMensagem);

        if(paciente == true){

            String tipo = "Paciente";

forum.CadastroMensagem(enviarmensagens,chaveEstrangeiraUsuarioMensagem,tipo);

            listaultimodado = forum.ListarUltimoDadoescrito();

            for (ClasseForum listarultimodado : listaultimodado){

                txtForum.append(listarultimodado.userMensagem);
                txtForum.append(" ");
                txtForum.append(listarultimodado.tipouser);
                txtForum.append(" ) : ");
                txtForum.append(listarultimodado.mensagem);
                txtForum.append("\n");
            }
        }
    }
}
```

```
    }

    }else{

        String tipo = "Equipe de Saúde";

forum.CadastroMensagem(enviarmensagens,chaveEstrangeiraUsuarioMensagem,tipo);

        listaultimodado= forum.ListarDados();

        for (ClasseForum listarultimodado : listaultimodado){

            txtForum.append(listarultimodado.userMensagem);
            txtForum.append(" (");
            txtForum.append(listarultimodado.tipouser);
            txtForum.append(" ) : ");
            txtForum.append(listarultimodado.mensagem);
            txtForum.append("\n");
        }

    }

    } catch (Exception ex) {

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

    txtMensagem.setText("");

}

/* public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ViewEnviarMensagens().setVisible(true);
        }
    });
}*/

// Variables declaration - do not modify
```

```
private javax.swing.JButton jButton1;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JScrollPane jScrollPane2;  
private javax.swing.JScrollPane jScrollPane3;  
private javax.swing.JTextPane jTextPane1;  
public javax.swing.JTextArea txtForum;  
public javax.swing.JTextArea txtMensagem;  
// End of variables declaration  
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package visão;  
  
import static com.sun.org.apache.xalan.internal.xsltc.compiler.util.Type.Int;  
import java.awt.Color;  
import java.awt.event.ItemEvent;  
import java.sql.Date;  
import java.sql.Time;  
import java.sql.Timestamp;  
import java.text.DateFormat;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Vector;  
import modeloBeans.ClasseEquipeSaude;  
import modeloDao.DAOCadastro;  
import modeloDao.DAOPessoa;  
import modeloDao.DAOWebsite;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.ComboBoxModel;  
import javax.swing.DefaultComboBoxModel;  
import javax.swing.DefaultListModel;  
import javax.swing.JOptionPane;  
import static javax.swing.JOptionPane.YES_NO_OPTION;  
import modeloBeans.ClasseAgendamento;  
import modeloBeans.ClasseCadastro;  
import modeloBeans.ClasseCidade;  
import modeloBeans.ClassePessoa;
```



```
import modeloBeans.ClasseTesteCovid;
import modeloConexao.ConexaoBD;
import modeloDao.DAOAgendamento;
import modeloDao.DAOCidades;
import modeloDao.DAOTesteCovid;

/**
 *
 * @author
 */
public class ViewRegistrarTesteDoAgendamento extends javax.swing.JFrame {

    int Enter = 0;
    String usuarioLogado;
    public List<ClasseTesteCovid> resultado = new ArrayList();
    public ViewRegistrarTesteDoAgendamento(String userLogado ) {

        initComponents();

        usuarioLogado = userLogado;
        txtNomePaciente.setEditable(false);
        txtPacienteIdAgendado.setEditable(false);
        txtCodAgendamentoPaciente.setEditable(false);
        txtStatus.setEnabled(false);
        jListPacientesAgendados.setVisible(false);

    }

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

        jLabel2 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        jButtonCadastrarTestePaciente = new javax.swing.JButton();
        txtPesquisa = new javax.swing.JTextField();
        jLabelPesquisa = new javax.swing.JLabel();
        txtNomePaciente = new javax.swing.JTextField();
        txtPacienteIdAgendado = new javax.swing.JTextField();
        jLabelNome = new javax.swing.JLabel();
        jLabelCodigoPaciente = new javax.swing.JLabel();
        jListPacientesAgendados = new javax.swing.JList<>();
        txtStatus = new javax.swing.JComboBox<>();
        jLabel3 = new javax.swing.JLabel();
        jSeparator2 = new javax.swing.JSeparator();
        jSeparator3 = new javax.swing.JSeparator();
        jSeparator4 = new javax.swing.JSeparator();
        jLabel4 = new javax.swing.JLabel();
        txtCodAgendamentoPaciente = new javax.swing.JTextField();
        jPanel3 = new javax.swing.JPanel();
    }
}
```

```
jLabel1 = new javax.swing.JLabel();

jLabel2.setText("jLabel2");

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setResizable(false);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jButtonCadastrarTestePaciente.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jButtonCadastrarTestePaciente.setText("CADASTRAR");

jButtonCadastrarTestePaciente.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
jButtonCadastrarTestePaciente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonCadastrarTestePacienteActionPerformed(evt);
    }
});
jPanel1.add(jButtonCadastrarTestePaciente, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 400, 700, 50));

txtPesquisa.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));
txtPesquisa.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPesquisaActionPerformed(evt);
    }
});
txtPesquisa.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        txtPesquisaKeyReleased(evt);
    }
});
jPanel1.add(txtPesquisa, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 20, 380, 20));

jLabelPesquisa.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelPesquisa.setText("Pesquisa :");
jPanel1.add(jLabelPesquisa, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 20, -1, -1));
jPanel1.add(txtNomePaciente, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 170, 240, -1));
jPanel1.add(txtPacienteIdAgendado, new org.netbeans.lib.awtextra.AbsoluteConstraints(630, 170, 60, -1));

jLabelNome.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelNome.setText("Nome Paciente :");
```

```
jPanel1.add(jLabelNome, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
170, 140, 20));

jLabelCodigoPaciente.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabelCodigoPaciente.setText("Código Pessoa Paciente :");
jPanel1.add(jLabelCodigoPaciente, new
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 160, -1, 40));

jListPacientesAgendados.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));
jListPacientesAgendados.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jListPacientesAgendadosMouseClicked(evt);
    }
});
jPanel1.add(jListPacientesAgendados, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 40, 380, 90));

txtStatus.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Positivo", "Negativo" }));
txtStatus.setSelectedIndex(-1);
jPanel1.add(txtStatus, new org.netbeans.lib.awtextra.AbsoluteConstraints(380, 340,
140, -1));

jLabel3.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel3.setText("Resultado teste covid-19 :");
jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 340, -1,
-1));
jPanel1.add(jSeparator2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 300,
710, 20));
jPanel1.add(jSeparator3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 150, -
1, -1));
jPanel1.add(jSeparator4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 150,
710, -1));

jLabel4.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
jLabel4.setText("Código do Agendamento do Paciente :");
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 210, -1,
20));
jPanel1.add(txtCodAgendamentoPaciente, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 240, 310, -1));

getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
40, 700, 450));

jPanel3.setBackground(new java.awt.Color(13, 130, 203));
jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Century Gothic", 1, 24)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setText("CADASTRAR TESTE COVID-19");
```

```
jPanel3.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 0, -1, 40));

getContentPane().add(jPanel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 700, 40));

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

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

    ClasseTesteCovid novoCadastroTesteCovid = new ClasseTesteCovid(); //
    instanciando → objeto → classe Paciente
    DAOTesteCovid novoCadastroTeste = new DAOTesteCovid();

    novoCadastroTesteCovid.nomePaciente = txtNomePaciente.getText().trim();
    novoCadastroTesteCovid.personPacienteId =
    Integer.parseInt(txtPacienteIdAgendado.getText().trim());
    novoCadastroTesteCovid.Idagendamento =
    Integer.parseInt(txtCodAgendamentoPaciente.getText().trim());

    if (txtStatus.getSelectedItem().toString().equals("Positivo")){

        novoCadastroTesteCovid.resultado = true;

    }else{

        novoCadastroTesteCovid.resultado = false;

    }

    try
    {

        novoCadastroTeste.CadastrarTesteCovid(novoCadastroTesteCovid, usuarioLogado);

        JOptionPane.showMessageDialog(null, " O teste do COVID-19 foi cadastrado com
        sucesso!");

    }
    catch (Exception ex)
    {

        JOptionPane.showMessageDialog(null, " O teste do COVID-19 não foi
        cadastrado com sucesso!");

    }

}
```

```
txtPesquisa.setText("");
txtNomePaciente.setText("");
txtPacienteIdAgendado.setText("");
txtStatus.setSelectedIndex(-1);
txtCodAgendamentoPaciente.setText("");

}

private void txtPesquisaActionPerformed(java.awt.event.ActionEvent evt) {
    jListPacientesAgendados.setVisible(false);
    Enter = 1;
}

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

    int linhaClicada = jListPacientesAgendados.getSelectedIndex();

    DAOTesteCovid itemClicado = new DAOTesteCovid();

    try {

        List <ClasseTesteCovid> setarValores = new ArrayList();

        setarValores =
itemClicado.ListarItemClicadoPacientesAgendados(linhaClicada,txtPesquisa.getText(),res
ultado.get(linhaClicada).codigo);

        String agendamentold = String.valueOf(setarValores.get(0).Idagendamento);
        String idpaciente = String.valueOf(setarValores.get(0).personPacienteId);

        txtNomePaciente.setText(setarValores.get(0).nomePaciente);
        txtCodAgendamentoPaciente.setText(agendamentold);
        txtPacienteIdAgendado.setText(idpaciente);

        txtStatus.setEnabled(true);

    } catch (Exception ex) {

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

}

private void txtPesquisaKeyReleased(java.awt.event.KeyEvent evt) {
```

```
jListPacientesAgendados.setVisible(false);

DefaultListModel listModel = new DefaultListModel();

if (Enter == 0) {

    DAOTesteCovid teste = new DAOTesteCovid();

    try {

        resultado =
teste.ListarNomesPacientesAgendadosSistema(txtPesquisa.getText());

        if (teste.mostraMax != 5) {

            for (ClasseTesteCovid listapesquisada : resultado) {

                listModel.addElement(listapesquisada.nomePaciente);

            }
jListPacientesAgendados.setModel(listModel);

jListPacientesAgendados.setVisible(true);

        } else {

            jListPacientesAgendados.setVisible(false);

        }

    } catch (Exception ex) {

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

    }

    } else {

        Enter = 0;

    }

}

/* public static void main(String args[])
```

```

{
    java.awt.EventQueue.invokeLater(new Runnable()
    {
        public void run()
        {
            new ViewRegistrarAgendamento().setVisible(true);
        }
    });
}*/

// Variables declaration - do not modify
private javax.swing.JButton jButtonCadastrarTestePaciente;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabelCodigoPaciente;
private javax.swing.JLabel jLabelNome;
private javax.swing.JLabel jLabelPesquisa;
private javax.swing.JList<String> jListPacientesAgendados;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel3;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JSeparator jSeparator4;
private javax.swing.JTextField txtCodAgendamentoPaciente;
public javax.swing.JTextField txtNomePaciente;
public javax.swing.JTextField txtPacienteIdAgendado;
private javax.swing.JTextField txtPesquisa;
private javax.swing.JComboBox<String> txtStatus;
// End of variables declaration
}

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package visão;

import static com.sun.org.apache.xalan.internal.xsltc.compiler.util.Type.Int;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ItemEvent;
import java.awt.event.KeyAdapter;

```

```
import java.awt.event.KeyEvent;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Time;
import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;
import java.util.Vector;
import modeloBeans.ClasseEquipeSaude;
import modeloDao.DAOCadastro;
import modeloDao.DAOPessoa;
import modeloDao.DAOEquipeDeSaude;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ComboBoxModel;
import javax.swing.DefaultComboBoxModel;
import javax.swing.DefaultListModel;
import javax.swing.JOptionPane;
import static javax.swing.JOptionPane.YES_NO_OPTION;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;
import modeloBeans.ClasseAgendamento;
import modeloBeans.ClasseCadastro;
import modeloBeans.ClasseCidade;
import modeloBeans.ClassePessoa;
import modeloConexao.ConexaoBD;
import modeloDao.DAOAgendamento;
import modeloDao.DAOCidades;
import javax.swing.RowFilter;
import javax.swing.table.TableModel;

/**
 *
 * @author
 */

public class ViewVisualizarPacientes extends javax.swing.JFrame {

    int Enter = 0;
    String usuarioLogado;
    public List<ClasseAgendamento> resultado = new ArrayList();

    //DefaultTableModel modelo;

    public ViewVisualizarPacientes(String userLogado ) {
```



```

initComponents();

usuarioLogado = userLogado;
ActionEvent evt = null;
jButtonBuscarActionPerformed(evt);

}

//TableRowSorter tr;

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

    jLabel2 = new javax.swing.JLabel();
    jPanel1 = new javax.swing.JPanel();
    jSeparator3 = new javax.swing.JSeparator();
    jScrollPane1 = new javax.swing.JScrollPane();
    tbDatosPaciente = new javax.swing.JTable();
    txtFiltrar = new javax.swing.JLabel();
    txtFiltro = new javax.swing.JTextField();
    jButtonBuscar = new javax.swing.JButton();
    jPanel3 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();

    jLabel2.setText("jLabel2");

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setResizable(false);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jPanel1.setBackground(new java.awt.Color(255, 255, 255));
    jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
    jPanel1.add(jSeparator3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 150, -
1, -1));

    tbDatosPaciente.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, null, null, null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null, null, null, null, null, null, null, null},
            {null, null, null, null, null, null, null, null, null, null, null, null, null, null, null}
        },
        new String[] {
            "ID", "NOMBRE", "APELLIDO", "FECHA", "HORA", "LUGAR", "MOTIVO", "DIAGNOSTICO", "TRATAMIENTO", "FARMACOS", "EVALUACION", "OBSERVACIONES", "FECHA", "HORA", "LUGAR"
        }
    ));
}

```

[illegible]


```
        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
    tbDadosPaciente.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_OFF);
    jScrollPane1.setViewportViewView(tbDadosPaciente);
    if (tbDadosPaciente.getColumnModel().getColumnCount() > 0) {
        tbDadosPaciente.getColumnModel().getColumn(0).setResizable(false);
        tbDadosPaciente.getColumnModel().getColumn(0).setPreferredWidth(100);
        tbDadosPaciente.getColumnModel().getColumn(14).setResizable(false);
        tbDadosPaciente.getColumnModel().getColumn(14).setPreferredWidth(150);
    }

    jPanel1.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 110,
670, 270));

    txtFiltrar.setFont(new java.awt.Font("Century Gothic", 0, 16)); // NOI18N
    txtFiltrar.setText("Filtrar :");
    jPanel1.add(txtFiltrar, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 50,
60, -1));

    txtFiltro.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtFiltroKeyReleased(evt);
        }
        public void keyTyped(java.awt.event.KeyEvent evt) {
            txtFiltroKeyTyped(evt);
        }
    });
    jPanel1.add(txtFiltro, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 50,
280, -1));

    jButtonBuscar.setText("Buscar");
    jButtonBuscar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonBuscarActionPerformed(evt);
        }
    });
    jPanel1.add(jButtonBuscar, new org.netbeans.lib.awtextra.AbsoluteConstraints(490,
50, -1, -1));

    getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
40, 670, 380));

    jPanel3.setBackground(new java.awt.Color(13, 130, 203));
    jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel1.setFont(new java.awt.Font("Century Gothic", 1, 24)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(255, 255, 255));
    jLabel1.setText("VISUALIZAR PACIENTES");
    jPanel3.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 0, 300,
```

```
40));

    getContentPane().add(jPanel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 670, 40));

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

private void txtFiltroKeyReleased(java.awt.event.KeyEvent evt) {

}

private void txtFiltroKeyTyped(java.awt.event.KeyEvent evt) {

    /*
    txtFiltro.addKeyListener(new KeyAdapter(){
    @Override
    public void keyReleased(KeyEvent ke) {
        tr.setRowFilter(RowFilter.regexFilter(txtFiltro.getText(), 1));
    }

    });

    tr = new TableRowSorter(modelo);

    tbDadosPaciente.setRowSorter(tr);
    */
}

private void jButtonBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String sql = null;
        if(txtFiltro.getText().equals("") || txtFiltro.getText() == null){
            sql = "SELECT * from public.pessoa as p "
                + "INNER JOIN public.paciente "
                + "as pac on p.personid = pac.personid";
        } else {
            sql = "SELECT * from public.pessoa as p "
                + "INNER JOIN public.paciente "
                + "as pac on p.personid = pac.personid "
                + "where p.nome like '" + txtFiltro.getText() + "%' ORDER BY p.nome";
        }

        ConexaoBD con = new ConexaoBD();

        PreparedStatement ps;

        ps = con.getConnection().prepareStatement(sql);

        ResultSet rs;
```

```
rs = ps.executeQuery();

DefaultTableModel modelo = (DefaultTableModel) tbDadosPaciente.getModel();
modelo.setNumRows(0);

while(rs.next()){

    modelo.addRow(new Object[]{rs.getString("nome"),rs.getString("sexo"),
        rs.getString("idade"),rs.getString("email"),rs.getString("telefone"),
        rs.getString("estado"),rs.getString("cidade"),rs.getString("rua"),
        rs.getString("bairro"),rs.getString("numcasa"),rs.getString("complemento"),
        rs.getString("cep"),
        rs.getString("cpf"),rs.getString("rg"),rs.getString("descricao Sintomas")});

}

rs.close();
ps.close();

} catch (SQLException ex) {

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

}
}

// Variables declaration - do not modify
private javax.swing.JButton jButtonBuscar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JSeparator jSeparator3;
private javax.swing.JTable tbDadosPaciente;
private javax.swing.JLabel txtFiltrar;
private javax.swing.JTextField txtFiltro;
// End of variables declaration
}
```

> DAO'S

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloDao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
import modeloBeans.ClasseAgendamento;
import modeloBeans.ClassePessoa;
import modeloConexao.ConexaoBD;

/**
 *
 * @author
 */
public class DAOAgendamento {

    public int mostraMax = 5;

    int[] codigo = new int[5];

    public List<ClasseAgendamento> ListarNomesPacientesCadastradosSistema(String
nomePaciente) throws Exception {

        List<ClasseAgendamento> listaPesquisaNomes = new ArrayList();

        Connection con = ConexaoBD.getConnection();
        PreparedStatement pstm = null; //Responsavel por realizar uma pesquisa no BD
        ResultSet rs = null; //Armazenar resultado da pesquisa
        String resultado;
        int i = 0;

        try {

            if(!nomePaciente.equals("")){

                pstm = con.prepareStatement("select * from pessoa where nome like '" +
nomePaciente + "%' ORDER BY nome");
                rs = pstm.executeQuery();
```

```
while(rs.next() && mostraMax != 0){
    ClasseAgendamento agend = new ClasseAgendamento();
    agend.codigo[i] = rs.getInt("personid");
    i++;

    agend.nomePaciente = rs.getString("nome");

    listaPesquisaNomes.add(agend);

    mostraMax--;

}

}
} catch (SQLException exception) {

    JOptionPane.showMessageDialog(null, "Erro ao tentar listar dados da pesquisa" +
exception);

}finally{
    con.close();
}

return listaPesquisaNomes;
}

public List<ClasseAgendamento> ListarItemClicked(int linha,String nomePaciente, int
[] resultado) throws Exception {

    int resultadold = resultado[linha];

    List<ClasseAgendamento> listaSetarPesquisa = new ArrayList();

    Connection con = ConexaoBD.getConnection();
    PreparedStatement pstm = null; //Responsavel por realizar uma pesquisa no BD
    ResultSet rs = null; //Armazenar resultado da pesquisa

    try {

        pstm = con.prepareStatement("select * from pessoa where personid =
"+resultadold+""");
        rs = pstm.executeQuery();

        if(rs.next()){
```



```
        ClasseAgendamento agendamento = new ClasseAgendamento();

        agendamento.personId = (rs.getInt("personid"));
        agendamento.nomePaciente = (rs.getString("nome"));

        listaSetarPesquisa.add(agendamento);
    }

} catch (SQLException exception) {

    JOptionPane.showMessageDialog(null, "Erro ao tentar listar dados da pesquisa" +
exception);

    finally{
        con.close();
    }

    return listaSetarPesquisa;
}

public void CadastroAgendamento (ClasseAgendamento novoAgendamento, String
usuarioLogado, java.util.Date date) throws Exception{

    ConexaoBD con = new ConexaoBD();
    java.sql.Date sqlDate = new java.sql.Date(date.getTime());

    String sql = "insert into agendamento ( usuarioregistrador, nomepaciente , datavisita,
hora ,status , personid) values (?, ?, ?, ?, ?, ?)";

    try (PreparedStatement ps = con.getConnection().prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {

        ps.setString(1, usuarioLogado);
        ps.setString(2, novoAgendamento.nomePaciente);
        ps.setDate(3, sqlDate);
        ps.setString(4, novoAgendamento.Hora);
        ps.setString(5, novoAgendamento.Status);
        ps.setInt(6, novoAgendamento.personId); // chave estrangeira do paciente

        ps.executeUpdate();
    }
}
```

```
}

}

////////// ALTERAR AGENDAMENTO
//////////

    public List<ClasseAgendamento> ListarNomesPacientesAgendadosSistema(String
nomePaciente) throws Exception {

        List<ClasseAgendamento> listaPesquisaNomes = new ArrayList();

        Connection con = ConexaoBD.getConnection();
        PreparedStatement pstm = null; //Responsavel por realizar uma pesquisa no BD
        ResultSet rs = null; //Armazenar resultado da pesquisa
        String resultado;
        int i = 0;

        try {

            if(!nomePaciente.equals("")){

                pstm = con.prepareStatement("select * from agendamento where nomepaciente
like '" + nomePaciente + "%' ORDER BY nomepaciente");
                rs = pstm.executeQuery();

                while(rs.next() && mostraMax != 0){

                    ClasseAgendamento agend = new ClasseAgendamento();

                    agend.codigo[i] = rs.getInt("agendamentoid");

                    i++;

                    agend.nomePaciente = rs.getString("nomepaciente");

                    listaPesquisaNomes.add(agend);

                    mostraMax--;

                }

            }

        } catch (SQLException exception) {

            JOptionPane.showMessageDialog(null, "Erro ao tentar listar dados da pesquisa" +
exception);

        }finally{
```

```
        con.close();

    }

    return listaPesquisaNomes;
}

public List<ClasseAgendamento> ListarItensAgendadosClicado(int linha,String
nomePaciente, int [] resultado) throws Exception {

    int resultadold = resultado[linha];

    List<ClasseAgendamento> listaSetarPesquisa = new ArrayList();

    Connection con = ConexaoBD.getConnection();
    PreparedStatement pstmt = null; //Responsavel por realizar uma pesquisa no BD
    ResultSet rs = null; //Armazenar resultado da pesquisa

    try {

        pstmt = con.prepareStatement("select * from agendamento where agendamentoid
= '"+resultadold+"'");
        rs = pstmt.executeQuery();

        if(rs.next()){

            ClasseAgendamento agendamento = new ClasseAgendamento();

            agendamento.agendamentoid = (rs.getInt("agendamentoid"));
            agendamento.nomePaciente = (rs.getString("nomepaciente"));
            agendamento.Data = (rs.getString("datavisita"));
            agendamento.Hora = (rs.getString("hora"));
            agendamento.Status = (rs.getString("status"));

            listaSetarPesquisa.add(agendamento);

        }

    } catch (SQLException exception) {

        JOptionPane.showMessageDialog(null, "Erro ao tentar listar dados da pesquisa" +
exception);

    }finally{
        con.close();
    }
```

```
    }

    return listaSetarPesquisa;
}

public void AlterarAgendamento (ClasseAgendamento alterarAgendamento, String
usuarioLogado, java.util.Date date) throws Exception{

    ConexaoBD con = new ConexaoBD();
    java.sql.Date sqlDate = new java.sql.Date(date.getTime());

    String sql = "update agendamento set datavisita = ?, hora = ? ,status = ? ,
usuarioalterador = ? "
        + "where agendamentoid = '"+alterarAgendamento.agendamentoid+"'";

    try (PreparedStatement ps = con.getConnection().prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {

        ps.setDate(1,sqlDate);
        ps.setString(2, alterarAgendamento.Hora);
        ps.setString(3, alterarAgendamento.Status);
        ps.setString(4, usuarioLogado);

        ps.executeUpdate();

    }

}

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloDao;
```

```

import modeloBeans.ClasseCadastro;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import modeloConexao.ConexaoBD;

/**
 *
 * @author
 */
public class DAOCadastro {

    ///////////////////////////////////////////////////
    public void CadastroAcessoLoginUsuario (ClasseCadastro novoUsuario, int
foreignKeyPessoa) throws Exception{

        ConexaoBD con = new ConexaoBD();

        String sql = "insert into cadastro_usuario (usuario, senha,personid) values (?, ?, ?)";

        try (PreparedStatement ps = con.getConnection().prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {

            ps.setString(1, novoUsuario.getUsuario());
            ps.setString(2, novoUsuario.getSenha());
            ps.setInt(3, foreignKeyPessoa);

            ps.executeUpdate();

        }
    }

    ///////////////////////////////////////////////////
}

```

```

/** To change this license header, choose License Headers in Project Properties.* To
change this template file, choose Tools | Templates* and open the template in the editor.
*/package modeloDao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
import modeloBeans.ClasseCidade;
import modeloConexao.ConexaoBD;
import visão.ViewRegistrarLoginEquipeDeSaude;

/**
 * ** @author
 */
public class DAOCidades {

    public List<ClasseCidade> ListarCidades(String uf) throws Exception {

        List<ClasseCidade> listaCidades = new ArrayList<>();

        Connection con = ConexaoBD.getConnection();
        PreparedStatement pstm = null; //Responsavel por realizar uma pesquisa no BD
        ResultSet rs = null; //Armazenar resultado da pesquisa

        try {
            pstm = con.prepareStatement("select * from cidades where estado =" + uf + "");
            rs = pstm.executeQuery();

            while (rs.next()) {

                ClasseCidade listaCidade = new ClasseCidade();
                listaCidade.nomeCidade = (rs.getString(3));

                listaCidades.add(listaCidade);

            }

        } catch (SQLException exception) {

            JOptionPane.showMessageDialog(null, "Erro ao tentar buscar a pesquisa" +
            exception);

        } finally {
            con.close();
        }

        return listaCidades;
    }
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloDao;

import java.sql.PreparedStatement;
import java.sql.Statement;
import modeloBeans.ClasseEquipeSaude;
import modeloConexao.ConexaoBD;

/**
 *
 * @author
 */
public class DAOEquipeDeSaude {

    public void CadastroDadosEquipeDeSaude (ClasseEquipeSaude
    novoEquipeDeSaude, int chaveEstrangeiraPessoa) throws Exception{

        ConexaoBD con = new ConexaoBD();

        String sql = "insert into equipedesauade (codigospecial,personid) values (?,?)";
        con.getConnection();

        try (PreparedStatement ps = con.getConnection().prepareStatement(sql,
        Statement.RETURN_GENERATED_KEYS)) {

            ps.setString(1, novoEquipeDeSaude.getCodigoEspecial());
            ps.setInt(2, chaveEstrangeiraPessoa);

            ps.executeUpdate();

        }

    }

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloDao;
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
import modeloBeans.ClasseForum;
import modeloConexao.ConexaoBD;

/**
 *
 * @author
 */
public class DAOForum {

    public int BuscaIdUsuarioMensagem(String usuarioMensagem) throws Exception {

        String sql = ("select * from cadastro_usuario where usuario =" + usuarioMensagem +
        "");

        ConexaoBD con = new ConexaoBD();

        PreparedStatement ps = con.getConnection().prepareStatement(sql);

        ResultSet rs = ps.executeQuery();

        int guardaforeignkeyUsuarioMensagem = 0;

        if (rs.next()) {

            guardaforeignkeyUsuarioMensagem = (rs.getInt(1));

        }

        rs.close();
        ps.close();

        return guardaforeignkeyUsuarioMensagem;
    }

    public boolean VerificarTipo(int chaveusuarioMensagem) throws Exception {

        String sql = ("SELECT * from public.cadastro_usuario as cad\n" +
        "INNER JOIN public.paciente as pac\n" +
        "on pac.personid = cad.personid and cad.usuarioid =" +
        chaveusuarioMensagem + "");
    }
}
```



```
ConexaoBD con = new ConexaoBD();

PreparedStatement ps = con.getConnection().prepareStatement(sql);

ResultSet rs = ps.executeQuery();

boolean check = false;

if (rs.next()) {

    check = true;

}

rs.close();
ps.close();

return check;
}

public void CadastroMensagem(ClasseForum novamensagem, int
foreignkeyUsuarioMensagem , String tipo) throws Exception {

    ConexaoBD con = new ConexaoBD();

    //java.sql.Date sqlDate = new java.sql.Date(novamensagem.data.getTime());
    String sql = "insert into forum ( usuariomensagem, mensagem , datamensagem,
horamensagem , usuarioid, tipousuario) values (?, ?, ?, ?, ?, ?)";

    try (PreparedStatement ps = con.getConnection().prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {

        ps.setString(1, novamensagem.userMensagem);
        ps.setString(2, novamensagem.mensagem);
        ps.setDate(3, novamensagem.data);
        ps.setString(4, novamensagem.hora);
        ps.setInt(5, foreignkeyUsuarioMensagem);
        ps.setString(6, tipo);

        ps.executeUpdate();

    }
}

public List<ClasseForum> ListarDados() throws Exception {

    List<ClasseForum> listadados = new ArrayList();

    Connection con = ConexaoBD.getConnection();
    PreparedStatement pstm = null; //Responsavel por realizar uma pesquisa no BD
```

```
ResultSet rs = null; //Armazenar resultado da pesquisa

try {
    pstmt = con.prepareStatement("select * from forum order by forumid");
    rs = pstmt.executeQuery();

    while (rs.next()) {

        ClasseForum listadadosforum = new ClasseForum();
        listadadosforum.userMensagem = (rs.getString("usuariomensagem"));
        listadadosforum.tipouser = (rs.getString("tipousuario"));
        listadadosforum.mensagem = (rs.getString("mensagem"));

        listadados.add(listadadosforum);

    }

} catch (SQLException exception) {

    JOptionPane.showMessageDialog(null, "Erro ao tentar buscar os dados" +
exception);

    } finally {
        con.close();
    }

}

return listadados;
}

public List<ClasseForum> ListarUltimoDadoescrito() throws Exception {

    List<ClasseForum> listaUltimoDadoEnviado = new ArrayList();

    Connection con = ConexaoBD.getConnection();
    PreparedStatement pstmt = null; //Responsavel por realizar uma pesquisa no BD
    ResultSet rs = null; //Armazenar resultado da pesquisa

    try {
        pstmt = con.prepareStatement("select * from forum where forumid = (select
max(forumid) from forum)");
        rs = pstmt.executeQuery();

        while (rs.next()) {

            ClasseForum listaultimodado = new ClasseForum();
            listaultimodado.userMensagem = (rs.getString("usuariomensagem"));
            listaultimodado.tipouser = (rs.getString("tipousuario"));
            listaultimodado.mensagem = (rs.getString("mensagem"));

        }

    } catch (SQLException exception) {

        JOptionPane.showMessageDialog(null, "Erro ao tentar buscar os dados" +
exception);

    } finally {
        con.close();
    }

}

return listaultimodado;
}
```

```
        listaUltimoDadoEnviado.add(listaUltimoDado);  
  
    }  
  
    } catch (SQLException exception) {  
  
        JOptionPane.showMessageDialog(null, "Erro ao tentar buscar os dados" +  
exception);  
  
        } finally {  
            con.close();  
  
        }  
  
        return listaUltimoDadoEnviado;  
    }  
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package modeloDao;  
  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import modeloBeans.ClasseCadastro;  
import modeloConexao.ConexaoBD;  
  
import modeloConexao.ConexaoBD;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
  
/**  
 *  
 * @author  
 */  
public class DAOLogin {  
  
    public boolean checkLogin(String usuario, String senha) {  
  
        Connection con = ConexaoBD.getConnection();
```

```
PreparedStatement stmt = null;
ResultSet rs = null;

boolean check = false;

try {

    stmt = con.prepareStatement("SELECT * FROM public.cadastro_usuario WHERE
usuario = ? and senha = ?");
    stmt.setString(1, usuario);
    stmt.setString(2, senha);

    rs = stmt.executeQuery();

    if (rs.next()) {

        check = true;
    }

} catch (SQLException ex) {
    Logger.getLogger(DAOLogin.class.getName()).log(Level.SEVERE, null, ex);
}

return check;
}

public int BuscaldUsuario(String usuario) throws Exception {

    String sql = ("select * from cadastro_usuario where usuario =" + usuario + "");

    ConexaoBD con = new ConexaoBD();

    PreparedStatement ps = con.getConnection().prepareStatement(sql);

    ResultSet rs = ps.executeQuery();

    int guardakeyUsuario= 0;

    if (rs.next()) {

        guardakeyUsuario = (rs.getInt("personid"));

    }

    rs.close();
    ps.close();

    return guardakeyUsuario;
}

public boolean VerificarTipo(int chaveUsuario) throws Exception {
```

```
String sql = ("SELECT * FROM public.paciente where personid = " + chaveUsuario +
""");

ConexaoBD con = new ConexaoBD();

PreparedStatement ps = con.getConnection().prepareStatement(sql);

ResultSet rs = ps.executeQuery();

boolean check = false;

if (rs.next()) {

    check = true;

}

rs.close();
ps.close();

return check;
}

}

/**
 *
 * @author
 */

/*
public class DAOLogin {

    public class DAOPaciente {

        public static ClasseCadastro Logar(ClasseCadastro login) throws Exception{

            ConexaoBD con = new ConexaoBD();

            String sql = "insert into paciente (descricaoSintomas) values (?)";

            PreparedStatement ps = con.getConnection().prepareStatement(sql);

            try (
```

```
}

// String sql = "SELECT * FROM public.cadastro_usuario where usuario
="+login.getUsuario()+"";

ps.setString(1, login.getUsuario());
ps.setString(2, login.getSenha());

ResultSet rs = ps.executeQuery();

if(rs.next()){

    ClasseCadastro user = new ClasseCadastro();

    user.setUsuario(rs.getString(1));
    user.setSenha(rs.getString(2));

    return user;
}
rs.close();
ps.close();

return null;
}
}

/*
public static ClasseCadastro Logar(ClasseCadastro login) throws Exception{

    String sql = "select * from cadastro_usuario where usuario = ? and senha = ? ";

    ConexaoBD con = new ConexaoBD();

    PreparedStatement ps = con.getConnection().prepareStatement(sql);

    ps.setString(1, login.getUsuario());
    ps.setString(2, login.getSenha());

    ResultSet rs = ps.executeQuery();

    if(rs.next()){

        ClasseCadastro usuario = new ClasseCadastro();
```

```
        usuario.setUsuario(rs.getString(1));
        usuario.setSenha(rs.getString(2));

        return usuario;
    }

    rs.close();
    ps.close();

    return null;
}

}*/
/*

    public static ClasseCadastro Logar(ClasseCadastro login) throws Exception{
        String sql = "select * from public.cadastro_usuario where usuario = ? and senha = ? ";

        ConexaoBD con = new ConexaoBD();
        PreparedStatement ps = con.getConnection().prepareStatement(sql);
        ps.setString(1, login.getUsuario());
        ps.setString(2, login.getSenha());

        ResultSet rs = ps.executeQuery();

        if(rs.next()){

            ClasseCadastro user = new ClasseCadastro();

            user.setCadastroId(rs.getInt(1));
            user.setUsuario(rs.getString(2));
            user.setSenha(rs.getString(3));

            return user;
        }
        rs.close();
        ps.close();

        return null;
    }
}*/
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloDao;

import modeloBeans.ClassePaciente;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import modeloBeans.ClasseCadastro;
import modeloConexao.ConexaoBD;

/**
 *
 * @author
 */
public class DAOPaciente {

    public void CadastroDadosUsuario (ClassePaciente novoPaciente,int
chaveEstrangeiraPessoa) throws Exception{

        ConexaoBD con = new ConexaoBD();

        String sql = "insert into paciente (descricaoSintomas,personid) values (?,?)";

        try (PreparedStatement ps = con.getConnection().prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {

            ps.setString(1, novoPaciente.descricaoSintomas);
            ps.setInt(2, chaveEstrangeiraPessoa);

            ps.executeUpdate();

        }

    }

    /*
    public static ClassePaciente verificaLogin(ClassePaciente login) throws Exception{

        String sql = "select * from cadastro_usuario where usuario = ? and senha = ? ";

        ConexaoBD con = new ConexaoBD();
```



```
PreparedStatement ps = con.getConnection().prepareStatement(sql);

ps.setString(1, login.getUser());
ps.setString(2, login.getPass());

ResultSet rs = ps.executeQuery();

if(rs.next()){
    ClassePaciente user = new ClassePaciente();

    user.setCOD_IMOVEL(rs.getInt(1));
    user.setALLNAME(rs.getString(2));
    user.setEMAIL(rs.getString(3));
    user.setIDADE(rs.getString(4));
    user.setESTADO(rs.getString(5));
    user.setALLCITY(rs.getString(6));
    user.setTELEFONE(rs.getString(7));
    user.setUser(rs.getString(8));
    user.setPASS(rs.getString(9));

    return user;
}
rs.close();
ps.close();

return null;
}
/*
public ArrayList<ClassePaciente> listarTodos() throws Exception {
    ConexaoBD con = new ConexaoBD();
    String sql = "select * from Cadastro_Usuario";
    PreparedStatement ps = con.getConnection().prepareStatement(sql);

    ResultSet rs = ps.executeQuery();

    ArrayList ListaAnuncio = new ArrayList();
    while (rs.next()) {
        ClassePaciente anuncio = new ClassePaciente();

        anuncio.setCOD_IMOVEL(rs.getInt(1));
        anuncio.setALLNAME(rs.getString(2));
        anuncio.setEMAIL(rs.getString(3));
        anuncio.setIDADE(rs.getString(4));
        anuncio.setESTADO(rs.getString(5));
        anuncio.setALLCITY(rs.getString(6));
        anuncio.setTELEFONE(rs.getString(7));
        anuncio.setUser(rs.getString(8));
        anuncio.setPASS(rs.getString(9));
```

```
        ListaAnuncio.add(anuncio);
    }

    return ListaAnuncio;
}
*/
/*
public static ClasseCadastro verificaLogin(ClasseCadastro login) throws Exception{

    String sql = "select * from cadastro_usuario where usuario = ? and senha = ? ";

    ConexaoBD con = new ConexaoBD();

    PreparedStatement ps = con.getConnection().prepareStatement(sql);

    ps.setString(1, login.getUsuario());
    ps.setString(2, login.getSenha());

    ResultSet rs = ps.executeQuery();

    if(rs.next()){

        ClasseCadastro usuario = new ClasseCadastro();

        usuario.setUsuario(rs.getString(1));
        usuario.setSenha(rs.getString(2));

        return usuario;
    }

    rs.close();
    ps.close();

    return null;
}

*/

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloDao;
```

```
import modeloBeans.ClassePessoa;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import javax.swing.JTextField;
import modeloConexao.ConexaoBD;

/**
 *
 * @author
 */
public class DAOPessoa {

    public void CadastroDadosPessoa (ClassePessoa novaPessoa) throws Exception{

        ConexaoBD con = new ConexaoBD();

        // String sql = "insert into tbpessoa
        (nome,sexo,idade,email,telefone,estado,cidade,rua,bairro,complemento,cep,numCasa,cpf
        ,rg) values (?,?,?,?,?,?,?,?,?,?,?,?,?)";

        String sql = "insert into pessoa
        (nome,sexo,idade,email,telefone,estado,cidade,rua,bairro,numcasa,complemento,cep,cpf,
        rg) values (?,?,?,?,?,?,?,?,?,?,?,?,?)";

        try (PreparedStatement ps = con.getConnection().prepareStatement(sql,
        Statement.RETURN_GENERATED_KEYS)) {

            ps.setString(1, novaPessoa.nome);
            ps.setString(2, novaPessoa.sexo);
            ps.setString(3, novaPessoa.idade);
            ps.setString(4, novaPessoa.email);
            ps.setString(5, novaPessoa.telefone);
            ps.setString(6, novaPessoa.estado);
            ps.setString(7, novaPessoa.cidade);
            ps.setString(8, novaPessoa.rua);
            ps.setString(9, novaPessoa.bairro);
            ps.setString(10, novaPessoa.numCasa);
            ps.setString(11, novaPessoa.complemento);
            ps.setString(12, novaPessoa.cep);
            ps.setString(13, novaPessoa.getCpf());
            ps.setString(14, novaPessoa.getRg());

            ps.executeUpdate();

        }
    }
}
```

```
}

public int BuscaIdPessoaCadastrada(String valorRg) throws Exception{

    String sql = ("select * from pessoa where rg='"+valorRg+"'");

    ConexaoBD con = new ConexaoBD();

    PreparedStatement ps = con.getConnection().prepareStatement(sql);

    ResultSet rs = ps.executeQuery();

    int guardaforeignkeyPacienteCadastradoAtual = 0;

    if(rs.next()){

        guardaforeignkeyPacienteCadastradoAtual = (rs.getInt(1));

    }

    rs.close();
    ps.close();

    return guardaforeignkeyPacienteCadastradoAtual;
}

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloDao;
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
import modeloBeans.ClasseTesteCovid;
import modeloConexao.ConexaoBD;

/**
 *
 * @author
 */
public class DAOTesteCovid {

    public int mostraMax = 5;

    public List<ClasseTesteCovid> ListarNomesPacientesAgendadosSistema(String
nomePaciente) throws Exception {

        List<ClasseTesteCovid> listaPesquisaNomes = new ArrayList();

        Connection con = ConexaoBD.getConnection();

        PreparedStatement pstm = null; //Responsavel por realizar uma pesquisa no BD

        ResultSet rs = null; //Armazenar resultado da pesquisa

        String resultado;

        int i = 0;

        try {

            if (!nomePaciente.equals("")) {

                pstm = con.prepareStatement("select * from agendamento where status =
'Concluída' and nomepaciente like '" + nomePaciente + "%' ORDER BY nomepaciente");

                rs = pstm.executeQuery();

                while (rs.next() && mostraMax != 0) {

                    ClasseTesteCovid mostrar = new ClasseTesteCovid();

                    mostrar.codigo[i] = rs.getInt("agendamentoid");
                    i++;

                    mostrar.nomePaciente = rs.getString("nomepaciente");
```

```
        listaPesquisaNomes.add(mostrar);

        mostraMax--;
    }

}

} catch (SQLException exception) {

    JOptionPane.showMessageDialog(null, "Erro ao tentar listar dados da pesquisa" +
exception);

    } finally {
        con.close();
    }

    return listaPesquisaNomes;
}

public List<ClasseTesteCovid> ListarItemClickedPacientesAgendados(int linha,
String nomePaciente, int [] resultado) throws Exception {

    int resultadold = resultado[linha];

    List<ClasseTesteCovid> listaSetarPesquisa = new ArrayList();

    Connection con = ConexaoBD.getConnection();
    PreparedStatement pstm = null; //Responsavel por realizar uma pesquisa no BD
    ResultSet rs = null; //Armazenar resultado da pesquisa

    try {

        pstm = con.prepareStatement("select * from agendamento where agendamentoid
= '"+resultadold+"'");
        rs = pstm.executeQuery();

        if(rs.next()){

            ClasseTesteCovid dadospacienteagendado = new ClasseTesteCovid();

            dadospacienteagendado.personPacienteld = (rs.getInt("personid"));
            dadospacienteagendado.Idagendamento = (rs.getInt("agendamentoid"));
            dadospacienteagendado.nomePaciente = (rs.getString("nomepaciente"));

            listaSetarPesquisa.add(dadospacienteagendado);

        }

    }
```

```
        } catch (SQLException exception) {

            JOptionPane.showMessageDialog(null, "Erro ao tentar listar dados da pesquisa" +
exception);

        }finally{
            con.close();
        }

        return listaSetarPesquisa;
    }

    public void CadastroTesteCovid (ClasseTesteCovid novoRegistroTeste, String
usuarioLogado) throws Exception{

        ConexaoBD con = new ConexaoBD();

        String sql = "insert into testecovid ( usuarioregistrador, nomepaciente ,resultado,
agendamentoid) values (?, ?, ?, ?)";

        try (PreparedStatement ps = con.getConnection().prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {

            ps.setString(1, usuarioLogado);
            ps.setString(2, novoRegistroTeste.nomePaciente);
            ps.setBoolean(3, novoRegistroTeste.resultado);
            ps.setInt(4, novoRegistroTeste.Idagendamento); // chave estrangeira do paciente

            ps.executeUpdate();

        }
    }
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
package modeloDao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.table.DefaultTableModel;
import modeloConexao.ConexaoBD;

/**
 *
 * @author
 */
public class DAOVisualizarPacientes {

    /**
     public int ListarDadosPaciente () throws Exception{

        String sql = ("SELECT * from public.pessoa as p\n" +
        "INNER JOIN public.paciente as pac\n" +
        "on p.personid = pac.personid ");

        ConexaoBD con = new ConexaoBD();

        PreparedStatement ps = con.getConnection().prepareStatement(sql);

        ResultSet rs = ps.executeQuery();

        // DefaultTableModel modelo = ( DefaultTableModel) tbpaciente.getModel();
        // modelo.setNumRows(0);

        while(rs.next()){

            //modelo.addRow(new Object[](){rs.getString("Nome
            Completo")getClass,rs.getString("Sexo")});

            DAOPessoa pessoapaciente = new DAOPessoa();

        }

    }

    rs.close();
    ps.close();
```



```
        return guardaforeignkeyPacienteCadastradoAtual;
    }

    */
}
```

➤ Classes Beans

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloBeans;

import java.sql.Date;

/**
 *
 * @author
 */
public class ClasseAgendamento {
    public int codigo[] = new int[5];
    public int agendamentoid;
    public String usuarioLogado;
    public int personId;
    public String nomePaciente;
    public String Data;
    public String Hora;
    public String Status;
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloBeans;
```

```
/**
 *
 * @author
 */
public class ClasseCadastro {

    public int getCadastroId() {
        return cadastroId;
    }

    public void setCadastroId(int cadastroId) {
        this.cadastroId = cadastroId;
    }

    public int cadastroId;
    // public String userLogado;
    private String usuario; // getters and setters
    private String senha; // getters and setters

    public String getUsuario() {
        return usuario;
    }

    public void setUsuario(String usuario) {
        this.usuario = usuario;
    }

    public String getSenha() {
        return senha;
    }

    public void setSenha(String senha) {
        this.senha = senha;
    }
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloBeans;

/**
 *
```

```
* @author
*/
public class ClasseCidade {

    public int idCidade;
    public String nomeCidade;

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloBeans;

/**
 *
 * @author
 */
public class ClasseEquipeSaude extends ClassePaciente{

    public int integranteEquipeld;
    private String codigoEspecial;

    public String getCodigoEspecial() {
        return codigoEspecial;
    }

    public void setCodigoEspecial(String codigoEspecial) {
        this.codigoEspecial = codigoEspecial;
    }

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloBeans;

import java.sql.Date;

/**
 *
```

```
* @author
*/
public class ClasseForum {

    public String userMensagem;
    public String tipouser;
    public Date data;
    public String hora;
    public String mensagem;
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloBeans;

/**
 *
 * @author
 */
public class ClassePaciente extends ClassePessoa {

    public int pacienteld;
    public String descricaoSintomas;
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloBeans;

/**
 *
 * @author
 */
public class ClassePessoa {

    public String nome;
```

```
public String sexo;
public String idade; // Adicionar no diagrama de classes
public String email;
public String telefone;
public String estado;
public String cidade;
public String rua;
public String bairro;
public String complemento;
public String cep;
public String numCasa;
private String cpf; // adicionar getters e setters diagrama de classes
private String rg; // adicionar getters e setters diagrama de classes

public String getCpf() {
    return cpf;
}

public void setCpf(String cpf) {
    this.cpf = cpf;
}

public String getRg() {
    return rg;
}

public void setRg(String rg) {
    this.rg = rg;
}
}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modeloBeans;

/**
 *
 * @author
 */
public class ClasseTesteCovid {

    public int codigo[] = new int[5];
    public int Idagendamento;
    public String nomePaciente;
    public boolean resultado;
```

```
public int personPacienteld;  
public int integranteEquipeld;  
  
}
```

➤ Conexão Banco de Dados (SGBD PostgreSQL)

```
package modeloConection;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.*;  
import java.sql.SQLException;  
import javax.swing.JOptionPane;  
  
/**  
 *  
 * @author  
 */  
public class ConexaoBD {  
  
    public Statement stm; //Responsavel por realizar uma pesquisa no BD  
    public ResultSet rs; //Armazenar resultado da pesquisa  
    public Connection con;  
  
    public static Connection getConnection()  
    {  
  
        Connection con = null;  
  
        try{  
            Class.forName("org.postgresql.Driver");  
            con =  
DriverManager.getConnection("jdbc:postgresql://127.0.0.1:5434/postgres","postgres","123  
");  
            //JOptionPane.showMessageDialog(null, "Conectado.");  
        }  
        catch(ClassNotFoundException | SQLException e){  
            System.out.println("Erro na conexão.");  
        }  
        return con;  
    }  
}
```

```
public void executaSql(String sql) {
    try {
        stm = con.createStatement(rs.TYPE_SCROLL_INSENSITIVE,
rs.CONCUR_READ_ONLY);
        rs = stm.executeQuery(sql);
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Erro ExecutaSQL " + ex.getMessage());
    }
}

public void confirmar() throws SQLException {
    try {
        con.commit();
    } catch (SQLException e) {
        throw new SQLException("Problemas no SQL.\n" + e.getMessage());
    } finally {
        con.close();
    }
}
}
```

➤ Códigos SQL:

```
-- Table: public.cadastro_usuario

-- DROP TABLE public.cadastro_usuario;

CREATE TABLE public.cadastro_usuario
(
    usuarioid integer NOT NULL DEFAULT
nextval('cadastro_usuario_usuarioid_seq'::regclass),
    usuario character varying(50) COLLATE pg_catalog."default",
    senha character varying(50) COLLATE pg_catalog."default",
    personid integer,
    CONSTRAINT cadastro_usuario_pkey PRIMARY KEY (usuarioid),
    CONSTRAINT fk_person_id_pk_person_id FOREIGN KEY (personid)
REFERENCES public.pessoa (personid) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
)
```

```
)  
WITH (  
    OIDS = FALSE  
)  
TABLESPACE pg_default;  
  
ALTER TABLE public.cadastro_usuario  
    OWNER to postgres;
```

```
-- Table: public.agendamento  
  
-- DROP TABLE public.agendamento;  
  
CREATE TABLE public.agendamento  
(  
    agendamentoid integer NOT NULL DEFAULT  
nextval('agendamento_testeid_seq'::regclass),  
    usuarioregistrador character varying(50) COLLATE pg_catalog."default" NOT  
NULL,  
    nomepaciente character varying(50) COLLATE pg_catalog."default" NOT NULL,  
    datavisita date NOT NULL,  
    hora character varying(50) COLLATE pg_catalog."default" NOT NULL,  
    status character varying(50) COLLATE pg_catalog."default" NOT NULL,  
    personid integer,  
    usuarioalterador character varying(50) COLLATE pg_catalog."default",  
    CONSTRAINT agendamento_pkey PRIMARY KEY (agendamentoid),  
    CONSTRAINT fk_person_id_pk_person_id FOREIGN KEY (personid)  
        REFERENCES public.pessoa (personid) MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
)  
WITH (  
    OIDS = FALSE  
)  
TABLESPACE pg_default;  
  
ALTER TABLE public.agendamento  
    OWNER to postgres;
```

```
-- Table: public.cidades  
  
-- DROP TABLE public.cidades;  
  
CREATE TABLE public.cidades
```



```
(
  codigo_cidade integer NOT NULL DEFAULT
nextval('cidades_codigo_cidade_seq'::regclass),
  estado character varying(30) COLLATE pg_catalog."default" NOT NULL,
  nomecidade character varying(30) COLLATE pg_catalog."default" NOT NULL,
  CONSTRAINT cidades_pkey PRIMARY KEY (codigo_cidade)
)
WITH (
  OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.cidades
  OWNER to postgres;
```

```
-- Table: public.equipedesaude

-- DROP TABLE public.equipedesaude;

CREATE TABLE public.equipedesaude
(
  integranteequipedesaudeid integer NOT NULL DEFAULT
nextval('equipedesaude_integranteequipedesaudeid_seq'::regclass),
  codigospecial character varying(100) COLLATE pg_catalog."default" NOT
NULL,
  personid integer,
  CONSTRAINT equipedesaude_pkey PRIMARY KEY
(integranteequipedesaudeid),
  CONSTRAINT fk_person_id_pk_person_id FOREIGN KEY (personid)
REFERENCES public.pessoa (personid) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
)
WITH (
  OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.equipedesaude
  OWNER to postgres;
```

```
-- Table: public.forum

-- DROP TABLE public.forum;
```

```
CREATE TABLE public.forum
(
    forumid integer NOT NULL DEFAULT nextval('forum_forumid_seq'::regclass),
    usuariomensagem character varying(50) COLLATE pg_catalog."default" NOT
    NULL,
    mensagem character varying(250) COLLATE pg_catalog."default" NOT NULL,
    datamensagem date NOT NULL,
    horamensagem character varying(50) COLLATE pg_catalog."default" NOT
    NULL,
    tipousuario character varying(50) COLLATE pg_catalog."default" NOT NULL,
    usuarioid integer,
    CONSTRAINT forum_pkey PRIMARY KEY (forumid),
    CONSTRAINT fk_cadastro_usuario_id_pk_cadastro_usuario_id FOREIGN KEY
    (usuarioid)
        REFERENCES public.cadastro_usuario (usuarioid) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.forum
    OWNER to postgres;
```

```
-- Table: public.pesquisa

-- DROP TABLE public.pesquisa;

CREATE TABLE public.pesquisa
(
    codigo integer NOT NULL DEFAULT nextval('pesquisa_codigo_seq'::regclass),
    nome character varying(50) COLLATE pg_catalog."default" NOT NULL,
    telefone character varying(50) COLLATE pg_catalog."default" NOT NULL,
    preco integer NOT NULL,
    tipo character varying(45) COLLATE pg_catalog."default" NOT NULL,
    hora character varying(45) COLLATE pg_catalog."default" NOT NULL,
    email character varying(100) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT pesquisa_pkey PRIMARY KEY (codigo)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.pesquisa
```

```
OWNER to postgres;
```

```
-- Table: public.pessoa

-- DROP TABLE public.pessoa;

CREATE TABLE public.pessoa
(
    personid integer NOT NULL DEFAULT
nextval('pessoa_personid_seq'::regclass),
    nome character varying(50) COLLATE pg_catalog."default",
    sexo character varying(50) COLLATE pg_catalog."default",
    idade character varying(100) COLLATE pg_catalog."default",
    email character varying(100) COLLATE pg_catalog."default",
    telefone character varying(100) COLLATE pg_catalog."default",
    estado character varying(100) COLLATE pg_catalog."default",
    cidade character varying(100) COLLATE pg_catalog."default",
    rua character varying(100) COLLATE pg_catalog."default",
    bairro character varying(100) COLLATE pg_catalog."default",
    numcasa character varying(100) COLLATE pg_catalog."default",
    complemento character varying(100) COLLATE pg_catalog."default",
    cep character varying(100) COLLATE pg_catalog."default",
    cpf character varying(100) COLLATE pg_catalog."default",
    rg character varying(100) COLLATE pg_catalog."default",
    CONSTRAINT pessoa_pkey PRIMARY KEY (personid)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.pessoa
    OWNER to postgres;
```

```
-- Table: public.testecovid

-- DROP TABLE public.testecovid;

CREATE TABLE public.testecovid
(
    testeid integer NOT NULL DEFAULT nextval('testecovid_testeid_seq'::regclass),
    usuarioregistrador character varying(50) COLLATE pg_catalog."default" NOT
NULL,
    nomepaciente character varying(50) COLLATE pg_catalog."default" NOT NULL,
```

```
    resultado boolean NOT NULL,  
    agendamentoid integer,  
    CONSTRAINT testecovid_pkey PRIMARY KEY (testeid),  
    CONSTRAINT fk_agendamento_id_pk_agendamento_id FOREIGN KEY  
    (agendamentoid)  
    REFERENCES public.agendamento (agendamentoid) MATCH SIMPLE  
    ON UPDATE NO ACTION  
    ON DELETE NO ACTION  
)  
WITH (  
    OIDS = FALSE  
)  
TABLESPACE pg_default;  
  
ALTER TABLE public.testecovid  
    OWNER to postgres;
```

```
-- Table: public.paciente  
  
-- DROP TABLE public.paciente;  
  
CREATE TABLE public.paciente  
(  
    pacienteid integer NOT NULL DEFAULT  
nextval('paciente_pacienteid_seq'::regclass),  
    descricao sintomas character varying(50) COLLATE pg_catalog."default",  
    personid integer,  
    CONSTRAINT paciente_pkey PRIMARY KEY (pacienteid),  
    CONSTRAINT fk_person_id_pk_person_id FOREIGN KEY (personid)  
    REFERENCES public.pessoa (personid) MATCH SIMPLE  
    ON UPDATE NO ACTION  
    ON DELETE NO ACTION  
)  
WITH (  
    OIDS = FALSE  
)  
TABLESPACE pg_default;  
  
ALTER TABLE public.paciente  
    OWNER to postgres;
```

➤ Indicação de demissão 01 :

➤ Vitor Roberto Kogawa de Moraes

- Motivo : Não participou ativamente do trabalho.

➤ **Indicação de demissão 01 :**

- Vinícius Ito

- Motivo : Não participou ativamente do trabalho.