UNIVERSIDADE DO OESTE DE SANTA CATARINA

ANA LUIZA EPPING, BRUNA ZIMMERMANN TREGNAGO, ÉVELYN PERONDI FILIMBERTI, ISADORA GAMBATTO BOCALON

PANCORA: Sistema de Monitoramento de Casos e Estatísticas do Corona vírus

São Miguel do Oeste – SC

2021

ANA LUIZA EPPING, BRUNA ZIMMERMANN TREGNAGO, ÉVELYN PERONDI FILIMBERTI, ISADORA GAMBATTO BOCALON

PANCORA: Sistema de Monitoramento de Casos e Estatísticas do Corona vírus

Relatório de formação apresentado ao Curso de Ciência da Computação, Área das Ciências Exatas, da Universidade do Oeste de Santa Catarina como requisito parcial à obtenção de nota para trabalho final nos componentes de Banco de dados II, Programação II e Engenharia de Software I.

Orientadores: Prof. Roberson J. F. Alves, Prof. Otília D. Barbosa e Prof. Francieli C. Petry.

São Miguel do Oeste – SC

2021

**SUMÁRIO**

1. **INTRODUÇÃO**...................................................................................................................3
2. **MODELO RELACIONAL**................................................................................................4
3. **BANCO DE DADOS**..........................................................................................................8

3.1 RELATÓRIOS.....................................................................................................................8

1. **CONSIDERAÇÕES FINAIS**...........................................................................................10

**REFERÊNCIAS**................................................................................................................11

1. **INTRODUÇÃO**

Com o avanço da pandemia do novo Corona Vírus, a população vem sofrendo mudanças evidentes na forma de viver. A necessidade de permanecer em casa e em segurança, evitando assim o contágio, possibilitou um grande avanço no setor tecnológico e também uma maior aceitação nas novas formas de trabalho, como home office.

Em meio a esse cenário foi proposto, para as disciplinas de Banco de dados II, Programação II e Engenharia de Software I, a elaboração de um sistema para o monitoramento dos casos de COVID-19.

1. **MODELO RELACIONAL**

Como parte inicial do trabalho, para fins de desenvolvimento do modelo relacional, se utilizou a plataforma de modelagem Visual Paradigm. Priorizando a aplicação das informações repassadas pelos professores orientadores, tais como dados gerais de pessoas e de empresas, informações geográficas e diferenciação entre usuários, também foram utilizadas informações fornecidas pelos órgãos de saúde nacional e pela OMS (Organização Mundial da Saúde).

No Diagrama 1, a seguir, é apresentado o Modelo Relacional.

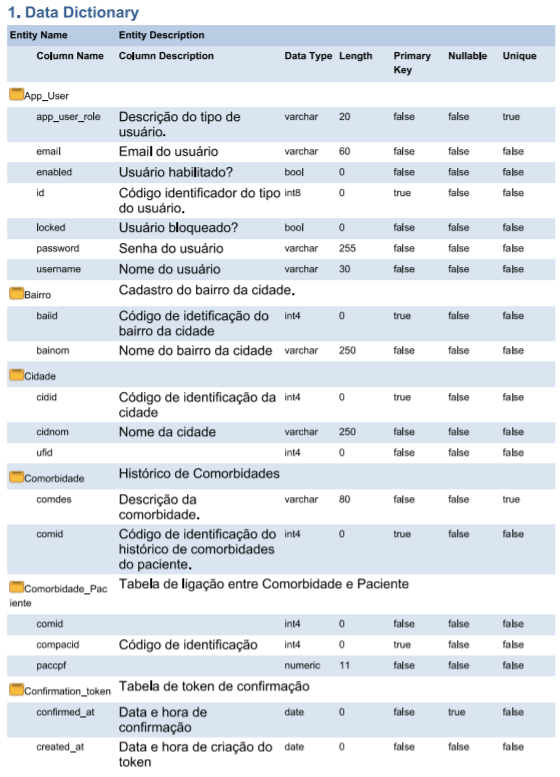
Diagrama 1 - Modelo Relacional

|  |
| --- |
|  |

Fonte: Os autores.

O padrão utilizado para a nomenclatura de tabelas e colunas foi a abreviação das 3 primeiras letras de cada palavra, exceto para tabelas geradas através de classes Java, como é o caso de App\_User e Confirmation\_token. Para uma melhor compreensão, observe o dicionário de dados a seguir:

Figura 1 - Dicionário de dados



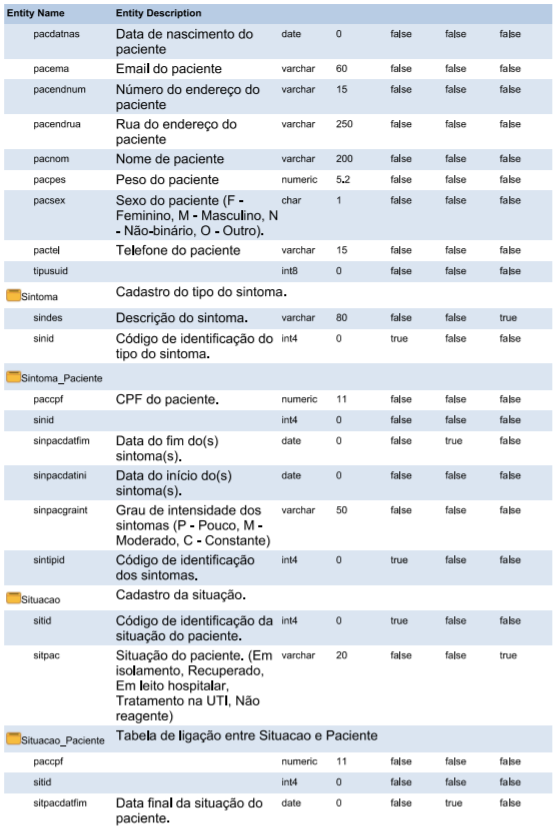
Fonte: Os autores.

Figura 2 - Dicionário de Dados



Fonte: Os autores.

Figura 3 - Dicionário de Dados



Fonte: Os autores.

Figura 4 - Dicionário de Dados

|  |
| --- |
|  |
|  |

Fonte: Os autores.

1. **ENGENHARIA DE SOFTWARE**

Com a existência do modelo relacional, foram levantados os requisitos do sistema e em seguida realizada a montagem do Modelo de Caso de Uso.

Posteriormente, foi desenvolvido o Fluxo de Casos de Uso, e os Diagramas de Sequência, Atividade, Estado e Classes.

No apêndice podemos ver os Digramas desenvolvidos.

1. **BANCO DE DADOS**

Após a finalização do modelo relacional, modelo de caso de uso e diagramas para entendimento do funcionamento do sistema, iniciou-se as alterações do banco de dados utilizando a ferramenta multiplataforma DBeaver, a linguagem SQL (Structured Query Language, ou Linguagem de Consulta Estruturada) e o sistema de gerenciamento de banco de dados PostgreSQL.

Na sequência, foi realizada a edição dos dados e elaborados selects, views, procedures e triggers. Ao decorrer do projeto foram adicionados dados à base e realizados ajustes conforme necessário.

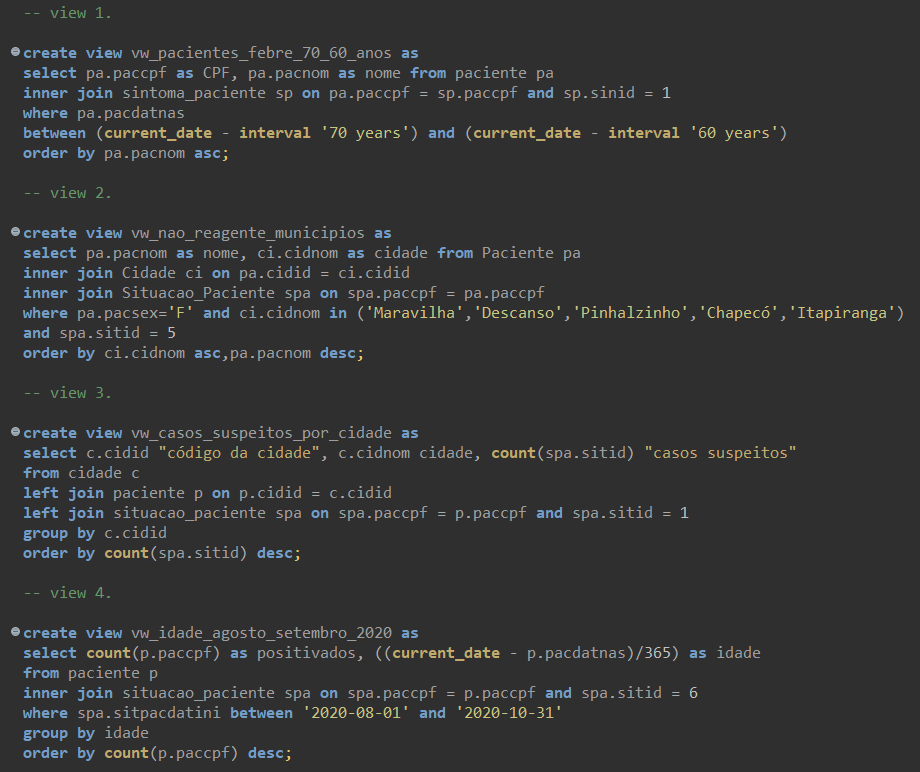
## RELATÓRIOS

Desenvolveram-se quatro views para consulta, seguindo as seguintes propostas:

1. Relacione o código e nome de pacientes com idades entre 60 e 70 anos, que apresentaram febre. Relacione a consulta em ordem ascendente de nome;
2. Relacione o nome do paciente, nome da cidade de residência de pacientes do sexo feminino, residentes nos municípios de Maravilha, Descanso, Pinhalzinho, Chapecó e Itapiranga que apresentaram sintomas e não foram positivados com covid. Relacione o relatório pelo nome da cidade ascendente e o nome do paciente descendente;
3. Relacione o código da cidade, nome da cidade, quantidade de casos suspeitos de covid para todas as cidades. Ordene o relatório da cidade com mais casos suspeitos para a cidade com menos casos suspeitos;
4. Relacione a idade e quantidade de casos positivos de covid por idade, registrados no período agosto a outubro de 2020. Ordene o relatório pela idade com mais casos para a idade com menos casos.

Na figura 5, é possível visualizar a estrutura e construção das views de relatório.

Figura 5 – Views propostas



Fonte: Os autores.

Para cada view, fez-se uso das cláusulas inner join, where e group by para cruzamento de tabelas, filtragem e agrupamento de resultados, a fim de que o retorno apresentado contivesse apenas as tuplas correspondentes aos seus respectivos requisitos. Uma exceção a ser notada, contudo, diz respeito a view 3, onde foi necessário utilizar o método left join para alcançar o objetivo proposto. Desse modo, os pacientes suspeitos de covid são agrupados por cidade, enquanto as cidades com nenhum paciente suspeito têm retorno valor 0.

1. **PROGRAMAÇÃO**

Com a estrutura base do sistema montada, foi iniciado o desenvolvimento das telas e funcionalidades do sistema. Utilizando a linguagem JAVA e o ambiente de desenvolvimento integrado IntelliJ IDEA Community Edition 2021.2.3, foi elaborado o código da aplicação web, considerando os requisitos solicitados.

Uma das classes desenvolvidas foi a classe EmailService, que implementa um método para envio do e-mail de confirmação de cadastro para novos usuários, exibida abaixo. Para conhecer outros trechos do código, confira o apêndice.

package br.edu.unoesc.springboot.pancora.email;  
  
import lombok.AllArgsConstructor;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.mail.javamail.JavaMailSender;  
import org.springframework.mail.javamail.MimeMessageHelper;  
import org.springframework.scheduling.annotation.Async;  
import org.springframework.stereotype.Service;  
  
import javax.mail.MessagingException;  
import javax.mail.internet.MimeMessage;  
  
*/\*\*  
 \* Classe de serviços para envio de email<br>  
 \* Esta classe foi construída e adaptada com base no projeto login-registration-backend de amigoscode  
 \** ***@see*** *<a href="https://github.com/amigoscode/login-registration-backend">login-registration-backend</a>  
 \*/*@Service  
@AllArgsConstructor  
public class EmailService implements EmailSender {  
  
 private final static Logger *LOGGER* = LoggerFactory.getLogger(EmailService.class);  
  
 @Autowired  
 private final JavaMailSender mailSender;  
  
 */\*\*  
 \* Sobreescrita do método send para envio de email  
 \** ***@param*** *to endereço de email  
 \** ***@param*** *email corpo do email  
 \*/* @Override  
 @Async  
 public void send(String to, String email) {  
 try {  
 MimeMessage mimeMessage = mailSender.createMimeMessage();  
 MimeMessageHelper helper = new MimeMessageHelper(mimeMessage, "utf-8");  
 helper.setText(email, true);  
 helper.setTo(to);  
 helper.setSubject("Confirme seu email");  
 helper.setFrom("pancora2021@gmail.com");  
 mailSender.send(mimeMessage);  
 }catch (MessagingException e) {  
 *LOGGER*.error("falha ao enviar email", e);  
 throw new IllegalStateException("falha ao enviar email");  
 }  
 }  
}

1. **CONSIDERAÇÕES FINAIS**

Pode-se perceber através do desenvolvimento deste trabalho a grande importância de uma base e um planejamento bem estruturados antes de iniciar o desenvolvimento de um banco de dados, bem como a programação de um sistema, tendo em vista a facilidade de gerar redundâncias e erros que podem impactar a segurança e bom funcionamento do mesmo.

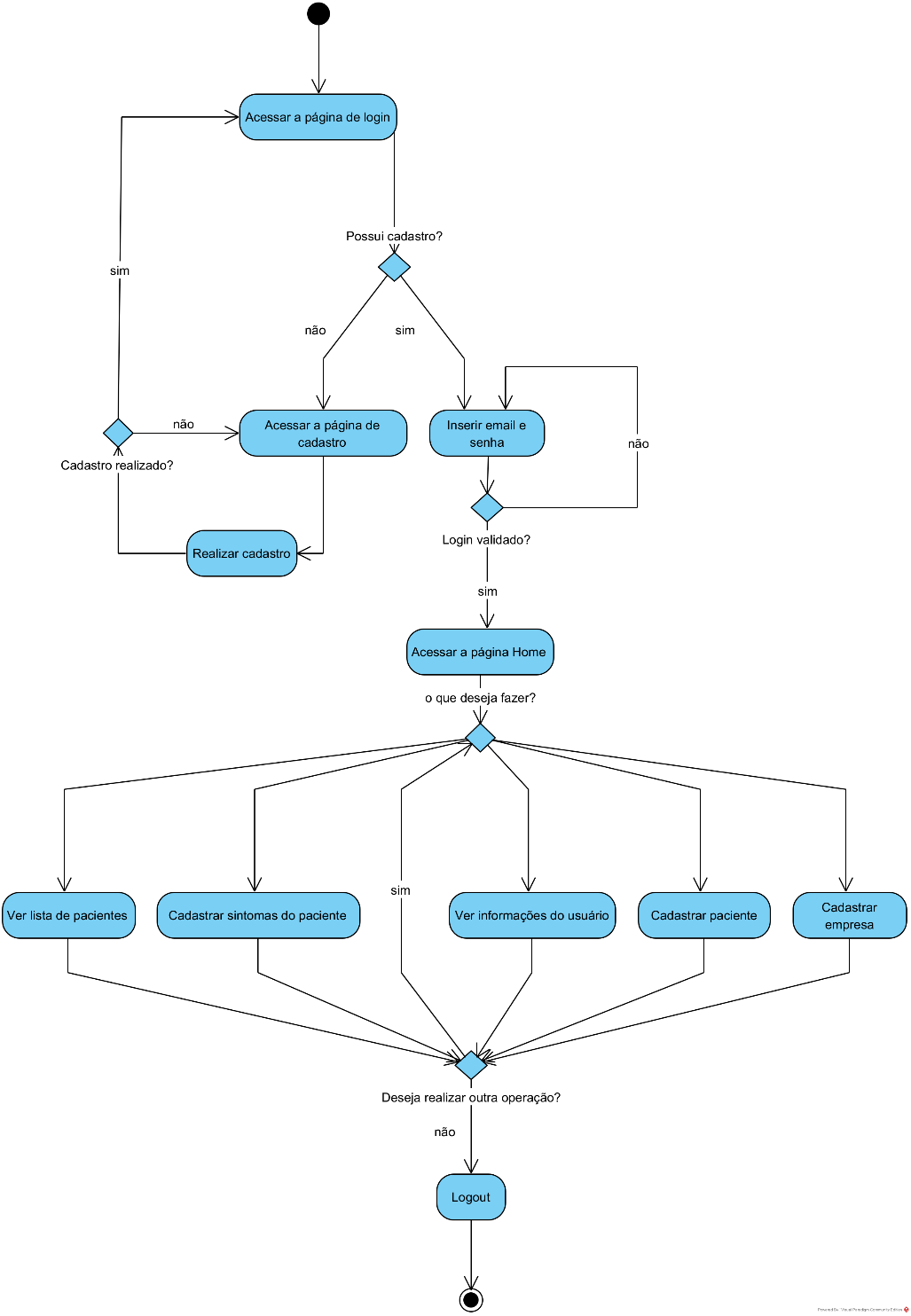
O trabalho completo pode ser visualizado na plataforma GithHub, disponível através do link <https://github.com/BrunaZimmermannTregnago/Pancora>.

**REFERÊNCIAS**

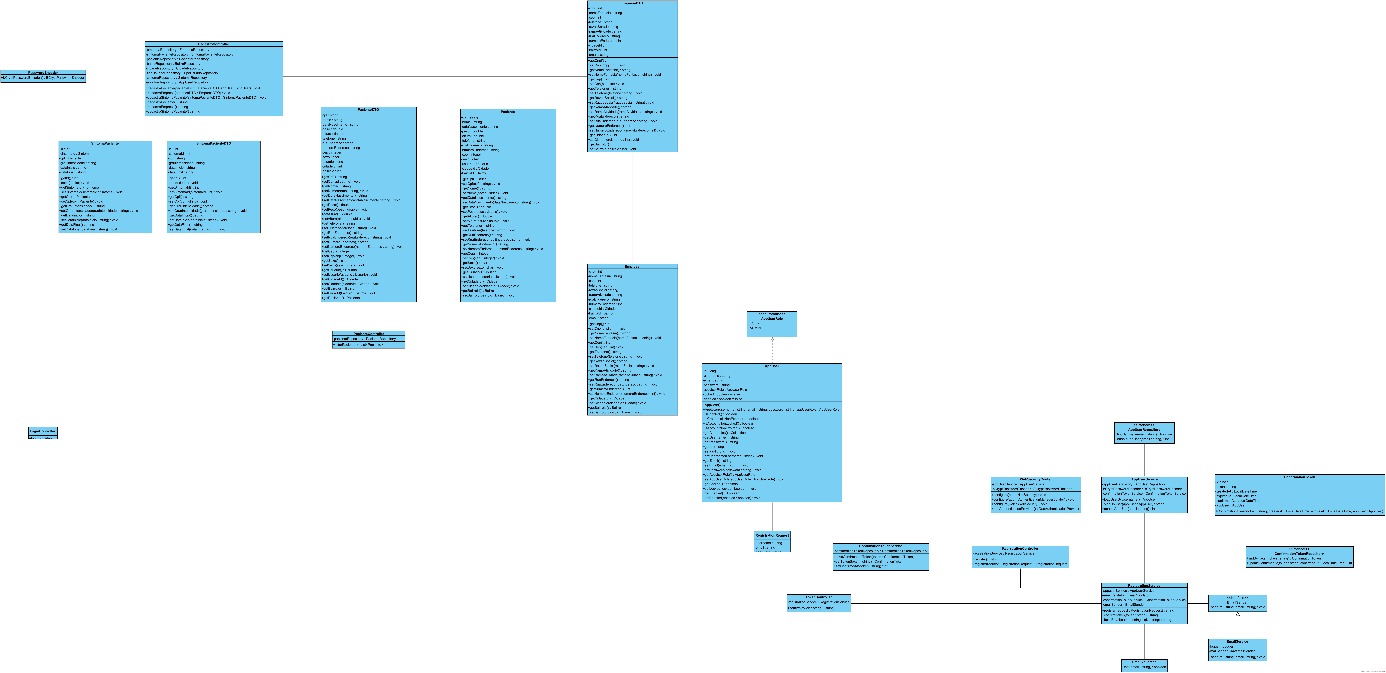
ALVES, Roberson J. F. **Apostila de Banco de Dados**. São Miguel do Oeste: Unoesc, 2021. Material didático.

FOLHA informativa sobre COVID-19. [21 jun. 2021] [*S. l.*]. Disponível em: https://www.paho.org/pt/covid19. Acesso em: 21 jun. 2021.

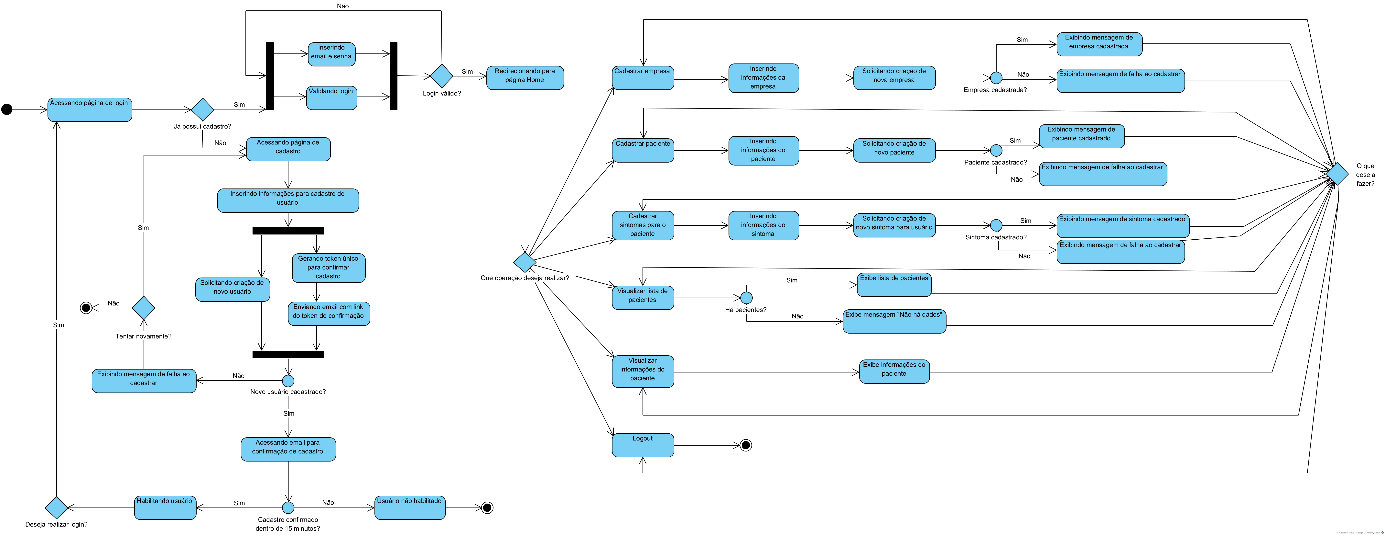
# APÊNDICE

Diagrama de Atividades:

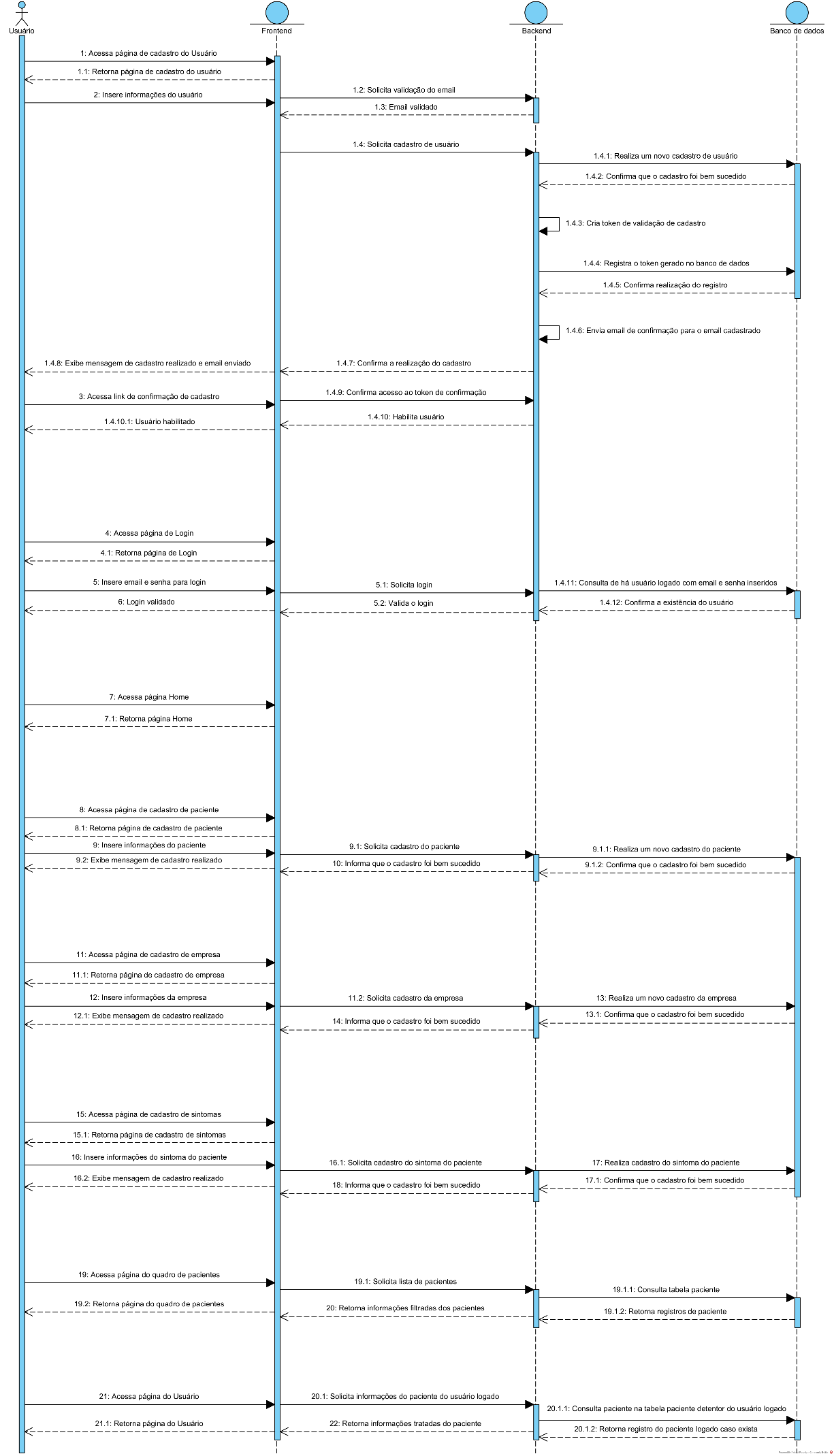
Fonte: Os autores.

Diagrama de Classes:

Fonte: Os autores.

Diagrama de Máquina de Estados:

Fonte: Os autores.

Diagrama de Sequência:

Fonte: Os autores.

PROGRAMAÇÃO:

Interface para envio de e-mail:

package br.edu.unoesc.springboot.pancora.email;  
  
*/\*\*  
 \* Interface com método para envio de email<br>  
 \* Esta interface foi construída com base no projeto login-registration-backend de amigoscode  
 \** ***@see*** *<a href="https://github.com/amigoscode/login-registration-backend">login-registration-backend</a>  
 \*/*public interface EmailSender {  
 void send(String to, String email);  
}

Classe da entidade comorbidade:

package br.edu.unoesc.springboot.pancora.entities;  
  
import lombok.\*;  
  
import javax.persistence.\*;  
import java.io.Serializable;  
  
@Table(name = "comorbidade", indexes = {  
 @Index(name = "comorbidade\_comdes\_key", columnList = "comdes", unique = true)  
})  
@Entity(name = "Comorbidade")  
@NoArgsConstructor  
@EqualsAndHashCode  
@Getter  
public class Comorbidade implements Serializable {  
 private static final long *serialVersionUID* = -1848373027810511848L;  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "comid", nullable = false)  
 private Integer id;  
  
 @Column(name = "comdes", nullable = false, length = 80)  
 @Setter private String descricao;  
  
 public Comorbidade(String descricao) {  
 this.descricao = descricao;  
 }  
}

Arquivo html para página de Login:

<!DOCTYPE html>  
<html lang="pt-br" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <title>Login</title>  
 <link rel="stylesheet/less" type="text/css" href="/css/style.less">  
</head>  
<body>  
  
<div class="wrapper">  
 <div class="container">  
 <h1>Pancora</h1>  
 <form class="form" name="f" th:action="@{/login}" method="post">  
 <input type="text" id="username" name="username" placeholder="Email">  
 <input type="password" id="password" name="password" placeholder="Senha">  
 <button type="submit" id="login-button">Login</button>  
 </form>  
 <p>Não possui cadastro? <a href="/registration" style="text-decoration:none"><font class="login" color="white">Cadastrar-se</font></a> </p>  
 </div>  
  
 <ul class="bg-bubbles">  
 <li></li>  
 <li></li>  
 <li></li>  
 <li></li>  
 <li></li>  
 <li></li>  
 <li></li>  
 <li></li>  
 <li></li>  
 <li></li>  
 </ul>  
</div>  
<script src="https://cdn.jsdelivr.net/npm/less@4.1.1" ></script>  
</body>  
</html>

Arquivo html para página de cadastro de paciente:

<!DOCTYPE html>  
<html lang="pt-br">  
 <head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <title>Cadastro de paciente</title>  
 <link rel="stylesheet" href="/css/cadastro\_style.css">  
 </head>  
  
 <header>  
 <nav class="navbar">  
 <a href="/" class="nav-logo">Pancora</a>  
 <ul class="nav-menu">  
 <li class="nav-item">  
 <a href="/cadastro-paciente" class="nav-link">Cadastro de paciente</a>  
 </li>  
 <li class="nav-item">  
 <a href="/cadastro-empresa" class="nav-link">Cadastro de empresa</a>  
 </li>  
 <li class="nav-item">  
 <a href="/cadastro-sintoma" class="nav-link">Cadastro de sintoma</a>  
 </li>  
 <li class="nav-item">  
 <a href="/lista-paciente" class="nav-link">Quadro de Pacientes</a>  
 </li>  
 <li class="nav-item">  
 <a href="/usuario" class="nav-link">Usuário</a>  
 </li>  
 <li class="nav-item">  
 <a href="/logout" class="nav-link">Sair</a>  
 </li>  
 </ul>  
 <div class="hamburger">  
 <span class="bar"></span>  
 <span class="bar"></span>  
 <span class="bar"></span>  
 </div>  
 </nav>  
 </header>  
  
 <body>  
 <div class="container">  
  
 <h1>Cadastro de Paciente</h1>  
  
 <section class="form-section">  
 <form class="form" onsubmit="return false">  
  
 <h2 class="form-section-heading">Informações pessoais</h2>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="nome">Nome Completo</label>  
 <input class="input" id="nome" name="nome" type="text" required>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="cpf">CPF</label>  
 <input class="input" id="cpf" name="cpf" type="number" required>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="telefone">Telefone</label>  
 <input class="input" id="telefone" name="telefone" type="text" required>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="datanascimento">Data de nascimento</label>  
 <input class="input" id="datanascimento" name="datanascimento" type="date" required>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="peso">Peso</label>  
 <input class="input" id="peso" name="peso" type="number" required>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="altura">Altura</label>  
 <input class="input" id="altura" name="altura" type="number" step="0.01" min="0" max="3.00" required>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="sexo">Sexo</label>  
 <select class="input" id="sexo" required>  
 <option value="0">Clique para selecionar</option>  
 <option value="F">Feminino</option>  
 <option value="M">Masculino</option>  
 <option value="O">Outro</option>  
 </select>  
 </div>  
 </div>  
  
 <h2 class="form-section-heading">Endereço</h2>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="estado">Estado</label>  
 <select class="input" id="estado" required>  
 <option value="0">Clique para selecionar</option>  
 <option value="1">Acre</option>  
 <option value="2">Alagoas</option>  
 <option value="3">Amazonas</option>  
 <option value="4">Amapá</option>  
 <option value="5">Bahia</option>  
 <option value="6">Ceará</option>  
 <option value="7">Distrito Federal</option>  
 <option value="8">Espírito Santo</option>  
 <option value="9">Goiás</option>  
 <option value="10">Maranhão</option>  
 <option value="11">Minas Gerais</option>  
 <option value="12">Mato Grosso do Sul</option>  
 <option value="13">Mato Grosso</option>  
 <option value="14">Pará</option>  
 <option value="15">Paraíba</option>  
 <option value="16">Pernambuco</option>  
 <option value="17">Piauí</option>  
 <option value="18">Paraná</option>  
 <option value="19">Rio de Janeiro</option>  
 <option value="20">Rio Grande do Norte</option>  
 <option value="21">Rondônia</option>  
 <option value="22">Roraima</option>  
 <option value="23">Rio Grande do Sul</option>  
 <option value="24">Santa Catarica</option>  
 <option value="25">Sergipe</option>  
 <option value="26">São Paulo</option>  
 <option value="27">Tocantins</option>  
 </select>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="cidade">Cidade</label>  
 <select class="input" id="cidade" required>  
 <option value="0">Clique para selecionar</option>  
 <option value="1">Afonso Cláudio</option>  
 <option value="2">Água Doce do Norte</option>  
 <option value="3">Água Branca</option>  
 <option value="4">Alegre</option>  
 <option value="5">Alfredo Chaves</option>  
 <option value="6">Alto Rio Novo</option>  
 <option value="7">Anchieta</option>  
 <option value="8">Apiacá</option>  
 <option value="9">Aracruz</option>  
 <option value="10">Atilio Vivacqua</option>  
 </select>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="bairro">Bairro</label>  
 <select class="input" id="bairro" required>  
 <option value="0">Clique para selecionar</option>  
 <option value="1">Agostini</option>  
 <option value="2">Centro</option>  
 <option value="3">Salete</option>  
 <option value="4">Progresso</option>  
 <option value="5">Estrela</option>  
 <option value="6">Santa Rita</option>  
 <option value="7">São Luiz</option>  
 <option value="8">São Jorge</option>  
 <option value="9">São Sebastião</option>  
 </select>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="rua">Rua</label>  
 <input class="input" id="rua" name="rua" type="text" required>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="numero">Número</label>  
 <input class="input" id="numero" name="numero" type="text" required>  
 </div>  
 </div>  
  
 <div class="field-wrap">  
 <div class="field">  
 <label class="label" for="cep">Cep</label>  
 <input class="input" id="cep" name="cep" type="number" required>  
 </div>  
 </div>  
  
 <div class="frame">  
 <button type="button" class="btn" id="cadastro-button" onclick="cadastrarPaciente()">Cadastrar</button>  
 <a id="voltar-button" href="/index" class="btn">  
 Voltar  
 </a>  
 </div>  
 </form>  
 </section>  
 </div>  
 <canvas class="background"></canvas>  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/particlesjs/2.2.2/particles.min.js"></script>  
 </script>  
 <script  
 src="https://code.jquery.com/jquery-3.6.0.min.js"  
 integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="  
 crossorigin="anonymous">  
 </script>  
 <script src="/js/particlesBackground.js"></script>  
 <script src="/js/navbar.js"></script>  
 <script src="/js/cadastrarPaciente.js"></script>  
 </body>  
  
 <footer className="footer mt-auto py-3 bg-dark">  
 <div className="footer container">  
 <p className="text-light">App desenvolvido por <a class="link-github" href="https://github.com/Ana-Epping" target="\_blank" rel="noreferrer">Ana Luiza Epping</a>,  
 <a class="link-github" href="https://github.com/BrunaZimmermannTregnago" target="\_blank" rel="noreferrer">Bruna Zimmermann Tregnago</a>,  
 <a class="link-github" href="https://github.com/evelynperondi" target="\_blank" rel="noreferrer">Évelyn Perondi Filimberti</a> e  
 <a class="link-github" href="https://github.com/isadoragbocalon" target="\_blank" rel="noreferrer">Isadora Gambatto Bocalon</a></p>  
 <p className="text-light"><a class="link-unoesc" href="https://www.unoesc.edu.br/" target="\_blank" rel="noreferrer"><strong>Ciência da Computação - UNOESC SMO</strong></a><br/>  
 </div>  
 </footer>  
</html>