

FIAP

LARISSA ARAÚJO GAMA ALVARENGA – 96496 - 2TDSPS

LARISSA LOPES OLIVEIRA – 552628 - 2TDSPC

LUNA FAUSTINO LIMA – 552473 - 2TDSPS

SPRINT 4 – JAVA ADVANCED

São Paulo

2025

OBJETIVO DA SOLUÇÃO

Este projeto tem como objetivo desenvolver um sistema de gestão para clínicas odontológicas, permitindo o cadastro, consulta, atualização e remoção de informações sobre clínicas, dentistas e pacientes. O sistema visa facilitar o gerenciamento dos dados e relacionamentos entre estas entidades, proporcionando uma interface web intuitiva para administração.

LINK VÍDEO YOUTUBE

<https://www.youtube.com/watch?v=myVk3CoMQ90>

LINK REPOSITÓRIO

<https://github.com/LunaFaustino/Sprints-JavaAdvanced>

ARQUITETURA DA SOLUÇÃO

O projeto segue uma arquitetura baseada no padrão MVC, implementada com Spring Boot e Thymeleaf. A estrutura do projeto está organizada da seguinte forma:

1. Camada Model:

- Classes de domínio que representam as entidades do sistema (Clinica, Dentista, Paciente, Endereco, Usuario, Perfil)
- Enums que representam estados e categorias (Status, Genero, Porte)
- Responsável pela lógica de negócio e validações específicas do domínio

2. Camada View:

- Templates Thymeleaf para renderização das páginas HTML
- Formulários para entrada de dados
- Interfaces de listagem e visualização de registros
- Integração com Bootstrap para estilização responsiva
- Suporte a internacionalização (i18n) com múltiplos idiomas (português e inglês)

3. Camada Controller:

- Controllers Spring MVC para tratamento das requisições HTTP
- Mapeamento de rotas e endpoints REST
- Validação de entrada de dados
- Redirecionamento e gerenciamento de mensagens flash
- Endpoints para autenticação e recursos protegidos

4. Camada Service:

- Implementa regras de negócio mais complexas
- Orquestra operações entre múltiplos repositories
- Gerencia transações
- Realiza validações adicionais
- Serviços para autenticação (implementando UserDetailsService)
- Serviços para integração com IA (OpenAI)
- Serviços para envio e consumo de mensagens (RabbitMQ)

5. Camada Repository:

- Interfaces Spring Data JPA para operações com o banco de dados
- Métodos customizados para consultas específicas

6. Camada DTO:

- Classes para transferência de dados entre camadas
- Conversores entre DTOs e entidades
- Validação de dados de entrada com anotações Jakarta

7. Segurança:

- Configuração de autenticação e autorização com Spring
- Gerenciamento de perfis e permissões de acesso
- Proteção de rotas com base em perfis de usuário

8. Mensageria:

- Configuração de filas, exchanges e bindings com RabbitMQ
- Produtores para envio de notificações
- Consumidores para processamento de mensagens

9. Monitoramento:

- Endpoints de monitoramento com Spring Boot Actuator
- Métricas personalizadas para o sistema
- Indicadores de saúde (health checks)

10. Inteligência Artificial:

- Integração com Azure OpenAI através do Spring AI
- Assistente virtual para resposta a perguntas frequentes
- Interface de chat para interação com o assistente

FUNCIONAMENTO DA APLICAÇÃO

O sistema permite:

1. Gestão de Clínicas:

- Cadastro de novas clínicas com dados como CNPJ, razão social, nome fantasia, contato e endereço
- Listagem de todas as clínicas cadastradas
- Visualização detalhada de uma clínica específica
- Edição dos dados de uma clínica existente
- Exclusão de clínicas

2. Gestão de Dentistas:

- Cadastro de dentistas com CRO, nome, gênero e associação a uma clínica
- Listagem de dentistas
- Visualização de detalhes de cada dentista
- Edição e exclusão de dentistas

3. Gestão de Pacientes:

- Cadastro de pacientes com CPF, dados pessoais, contato e endereço
- Vinculação de pacientes a clínicas específicas
- Visualização e edição de dados dos pacientes
- Exclusão de pacientes

4. Gestão de Usuários e Autenticação:

- Autenticação por username e senha
- Diferentes perfis de acesso (ADMIN e USER)
- Usuários com permissões diferenciadas
- Proteção de rotas com base em perfis
- Gerenciamento de sessões

5. Internacionalização:

- Suporte para múltiplos idiomas (português e inglês)
- Seleção de idioma pela interface do usuário
- Mensagens e textos traduzidos para diferentes idiomas

6. Mensageria e Notificações:

- Notificações para eventos do sistema (cadastro, atualização, exclusão)
- Filas específicas para diferentes tipos de entidades (clínicas, dentistas, pacientes)
- Produtores e consumidores para processamento de mensagens

7. Monitoramento e Métricas:

- Endpoints para monitoramento do sistema
- Métricas de contagem de cadastros
- Verificação de saúde de componentes (RabbitMQ)
- Informações sobre a aplicação

8. Assistente Virtual com IA:

- Interface de chat para interação com assistente virtual
- Respostas contextualizadas sobre serviços odontológicos
- Integração com Azure OpenAI para processamento de linguagem natural

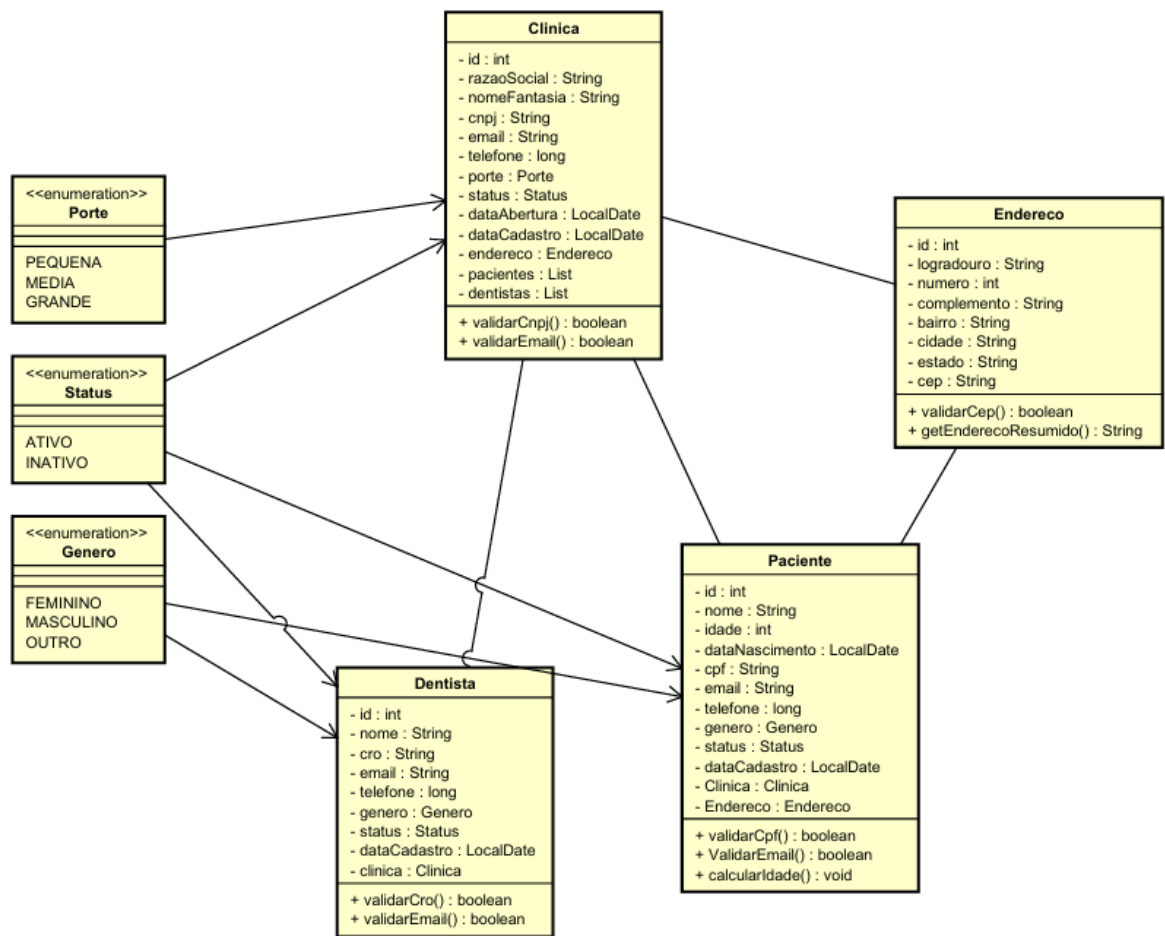
O fluxo de navegação permite acessar as diferentes seções através de um menu principal, e cada seção apresenta operações CRUD completas para a respectiva entidade.

DIFICULDADES ENCONTRADAS ATÉ O MOMENTO

Durante o desenvolvimento, enfrentamos algumas dificuldades:

1. **Validação de Formulários:** Encontramos desafios na validação de formulários com objetos aninhados, especialmente para os endereços. A falta de setters ou a não inicialização de objetos aninhados nos DTOs causou problemas de binding durante a submissão de formulários.
2. **Conversão de Enums em Formulários:** Tivemos dificuldades na conversão entre strings e enums nos formulários, principalmente quando o valor do campo era a descrição do enum e não o nome da constante.
3. **Cálculo Automático de Valores:** A automatização do cálculo da idade do paciente a partir da data de nascimento apresentou desafios na validação e persistência.
4. **Configuração de Segurança:** O balanceamento entre segurança e usabilidade foi um desafio, principalmente ao definir regras de acesso granulares e manter a experiência do usuário fluida.
5. **Integração com RabbitMQ:** A configuração inicial do RabbitMQ e a estruturação adequada de filas e exchanges exigiu um estudo para garantir o correto funcionamento da mensageria.
6. **Configuração do Azure OpenAI:** A integração com a API do Azure OpenAI através do Spring AI exigiu estudos e ajustes para adequar o formato de prompts e respostas ao contexto da aplicação.

DIAGRAMA DE CLASSES



DER

