

1. Justificativa do segmento e problema central a ser resolvido

Escolhemos desenvolver um mini-ERP voltado para uma **clínica de saúde** porque é um ambiente com processos administrativos repetitivos e críticos que impactam diretamente a qualidade do atendimento e possibilidade de erros operacionais. O problema central que o sistema se propõe a resolver é a **falta de um fluxo confiável e centralizado para gerenciamento de pessoas (pacientes/clientes e funcionários) e para o controle de agendas e atendimentos**, que hoje gera: perda de informação, conflitos de agendamento, retrabalho entre secretária e médico e dificuldade para gerir compras e estoque de insumos.

2. Justificativa para a Tecnologia

Nossa equipe resolveu utilizar Java com Spring Boot por conta da robustez e confiabilidade para sistemas empresariais, além dos conceitos de orientação a objetos para organização e modularização do nosso código. Para o desenvolvimento das views, optamos por utilizar o framework Vaadin, para centralizarmos tanto o front-end quanto o back-end com a linguagem Java, focando na simplicidade.

Na escolha do banco de dados optamos por utilizar o PostgreSQL, um SGDB popular e open source, com integração compatível entre o nosso banco de dados e o ORM do Spring JPA.

3. Entidades

Paciente

Representa um cliente da clínica, cadastrado pelas entidades, secretária e médico.

Secretária

Representa a secretária da clínica de saúde, responsável pelos agendamentos e cadastros dos pacientes.

Médico

Representa o médico da clínica de saúde, responsável pelo preenchimento da consulta do cliente.

Agendamento

Representa o agendamento de uma consulta do paciente, feito pela secretária.

Consulta

Representa a consulta de um paciente, que foi agendado pela secretária e foi atendido pelo médico.

Usuário

Representa a entidade de login e acesso ao sistema dos que possuem permissão.

Controllers

UserController

O UserController é responsável por gerenciar as operações relacionadas ao acesso e autenticação dos usuários do sistema. Ações como login, logout, criação de novos usuários, gerenciamento de permissões específicas, alteração de senha e gerenciamento de sessões.

AppointmentController

O AppointmentController é responsável pelo gerenciamento das informações da consulta. Pode criar, atualizar e consultar dados de consultas, validar se a consulta já foi atendida, anexar observações do médico e vincular a consulta ao paciente e ao agendamento correspondente. Também expõe operações para que a view possa buscar histórico de consultas por paciente.

ScheduleController

O ScheduleController é responsável por gerenciar as operações relacionadas ao agendamento de um paciente em determinado horário. Como por exemplo mostrar horários disponíveis, bloquear horários já ocupados, marcar consulta para um paciente em determinado horário e cancelar um agendamento. Também coordena checagens de conflito e comunicação com o AgendamentoService.

PatientController

O PatientController é responsável por gerenciar todas as operações relacionadas aos pacientes dentro do sistema. Ações como criação de pacientes, edição de dados, listar pacientes, pesquisar por CPF ou nome, e fornecer dados do histórico para visualização nas consultas.

Views

LoginView

UserView

PatientView

ScheduleView

AppointmentView

Diagrama

