

1. Contexto e Objetivo

ConectaMed é uma plataforma web destinada a aproximar pacientes de serviços de saúde próximos, unificando informações.

2. Público-Alvo

A solução atende três grupos principais: pacientes que desejam localizar atendimento médico com rapidez, administradores de unidades e médicos.

3. Visão Funcional do Produto

3.1 Localização Inteligente

O usuário concede ou não acesso à sua geolocalização. Caso a permissão seja fornecida, o sistema centraliza o mapa com base na localização atual.

3.2 Consulta Detalhada

Cada estabelecimento possui ficha própria com endereço completo, telefone, horário de funcionamento, descrição e avaliações.

3.3 Sistema de Filtros

Os filtros combinam busca textual, especialidades, convênios, tipo de estabelecimento e raio máximo. A aplicação filtra os resultados com base nesses critérios.

3.4 Painel em Mapa

O mapa full-screen foi implementado com Leaflet e OpenStreetMap, eliminando custos com licenciamento proprietário.

3.5 Gestão de Estabelecimentos

Administradores autenticados podem cadastrar ou atualizar dados de sua unidade, inclusive médicos e convênios.

3.6 Favoritos e Experiência do Paciente

Pacientes autenticados podem favoritar unidades de interesse, acessando-as rapidamente em visitas futuras. O sistema também permite avaliações.

4. Arquitetura Técnica

4.1 Backend

Construído em Node.js com Express, com camadas claras de rotas, controllers, models e serviços. O MongoDB é o banco de dados.

4.2 Frontend

O layout utiliza Handlebars com um template principal e parciais para cabeçalho, modais e menu de filtros. Os estilos são gerenciados com SASS.

4.3 Integrações Externas

Leaflet consome tiles do OpenStreetMap através de requisições HTTPS. Para geocodificação e cálculo de direções utiliza-se o Google Maps API.

5. Fluxo do Usuário

Passo 1: o visitante acessa a página inicial, visualiza o mapa e decide usar a barra de busca ou abrir filtros.

Passo 2: a aplicação tenta obter a localização atual e recalcula o raio padrão (10 km). Caso negado, mantém o centro do mapa em uma localização padrão.

Passo 3: filtros selecionados são enviados ao endpoint /api/estabelecimentos. A resposta alimenta os marcadores no mapa.

Passo 4: ao clicar em um marcador, o usuário pode abrir detalhes, ligar diretamente ou gerar rota externa em uma aplicação de navegação.

Passo 5: logins e cadastros ocorrem por meio de modais que sobreponham o mapa, preservando o contexto visual.

6. Estrutura de Pastas

/config contém conexão com banco e middlewares.

/controllers agrupa a lógica de domínio, como estabelecimentoController e medicoController.

/models possui os schemas Mongoose (Estabelecimento, Medico, Usuario).

/routes separa rotas públicas, administrativas e APIs REST.

/services concentra integrações externas, como GoogleMapsService.

/views guarda os templates Handlebars e layouts.

/public disponibiliza assets estáticos (CSS, JS, imagens).

/scripts contém utilitários adicionais do projeto.

7. Modelo de Dados Essencial

Estabelecimento registra nome, CNPJ, endereço completo, telefone, descrição, tipo (clínica ou órgão público), convênios e avaliações.