

Especificação de Requisitos e Arquitetura

Sistema de Gestão de Clínica Odontológica (OdontoSys Lite)

Luidy Costa dos Santos

3 de dezembro de 2025

Resumo

Este documento apresenta a especificação de requisitos e a arquitetura de software para o sistema *OdontoSys Lite*, uma aplicação voltada para a gestão simplificada de clínicas odontológicas.

Conteúdo

1	Requisitos Funcionais	2
1.1	Limitações (Âmbito Negativo)	2
2	Requisitos Não Funcionais	2
3	Design e Arquitetura de Software	3
3.1	Modelo de Arquitetura	3
3.2	Tecnologias	3
3.3	Divisão de Camadas (Padrão MVC)	3
4	Módulos e Perfis de Usuário	3
4.1	Módulos do Sistema	3
4.2	Perfis de Acesso	3
5	Modelagem de Dados e Classes	3
5.1	Estrutura das Classes	4
5.2	Diagrama de Relacionamento (Conceitual)	4
6	Análise de Riscos	4

1 Requisitos Funcionais

O sistema deve fornecer as seguintes funcionalidades essenciais para a operação da clínica:

- **Gestão de Pacientes:** O sistema deve permitir registrar, visualizar, editar e eliminar pacientes.
 - *Dados obrigatórios:* Nome, CPF, Data de Nascimento e Telefone.
- **Prontuário Eletrônico:** O sistema deve permitir registrar evoluções clínicas no cadastro do paciente.
 - *Nota:* A funcionalidade de upload de imagens (Raio-X) não está incluída nesta versão (Lite).
- **Gestão de Agendamentos:** O sistema deve permitir agendar, visualizar e cancelar consultas, vinculando um paciente a uma data e horário.
- **Controle de Acesso (Login):** O sistema deve autenticar utilizadores mediante login e senha para diferenciar perfis de acesso.
- **Gestão Financeira:** O sistema deve permitir o lançamento de valores recebidos e a visualização do total faturado no período.

1.1 Limitações (Âmbito Negativo)

- O sistema não enviará notificações automáticas (WhatsApp/Email).
- O sistema não gerará gráficos visuais complexos, apenas relatórios textuais/tabulares.

2 Requisitos Não Funcionais

Estas são as restrições e qualidades esperadas do sistema:

1. **Segurança e Autenticação:** O acesso é restrito por login. As senhas não devem ser armazenadas em texto simples.
2. **Interface Amigável:** Interface limpa, intuitiva e responsiva (CSS) para desktops e tablets.
3. **Acesso via Navegador (Web-based):** Execução inteiramente no navegador, sem instalação de executáveis (.exe).
4. **Desempenho e Disponibilidade:** Resposta instantânea ($< 200\text{ms}$) e funcionamento offline (após carregado), dependendo apenas do navegador.
5. **Persistência de Dados:** Uso de `localStorage` para garantir a continuidade entre sessões no navegador.

3 Design e Arquitetura de Software

3.1 Modelo de Arquitetura

O sistema segue o modelo **Monolítico (Client-Side)** do tipo *Single Page Application* (SPA). Todos os componentes (apresentação, lógica e dados) são entregues ao navegador em um pacote único.

3.2 Tecnologias

Front-end: HTML5 (Estrutura), CSS3 (Estilo), JavaScript (Lógica).

Back-end: Simulado via JavaScript no navegador.

Base de Dados: `localStorage` (NoSQL Key-Value Store do navegador).

3.3 Divisão de Camadas (Padrão MVC)

Para organização do código JavaScript:

- **Model (Dados):** Classes responsáveis por ler e guardar JSON no `localStorage`.
- **Controller (Lógica):** Funções que validam regras de negócio (ex: senhas, agendamentos duplicados).
- **View (Apresentação):** Funções que manipulam o DOM para exibir tabelas e formulários.

4 Módulos e Perfis de Usuário

4.1 Módulos do Sistema

1. **Recepção:** Registro de pacientes e gestão visual da agenda.
2. **Clínico:** Registro médico e histórico de tratamentos.
3. **Financeiro:** Fluxo de caixa simples (lançamentos e totais).
4. **Autenticação:** Validação de credenciais e controle de acesso.

4.2 Perfis de Acesso

- **Administrador / Dentista (Dr. Ricardo):** Acesso total. Pode visualizar/editar Prontuários e relatórios financeiros completos.
- **Secretária:** Foco na organização. Pode gerir a agenda e registrar pacientes. Pode lançar pagamentos, mas **não** visualiza detalhes do prontuário.

5 Modelagem de Dados e Classes

Abaixo estão descritas as principais classes e seus atributos.

5.1 Estrutura das Classes

- **Usuario**

- Atributos: id, login, senha, perfil
- Métodos: +autenticar()

- **Paciente**

- Atributos: id, nome, cpf, telefone, historico_clinico
- Métodos: +cadastrar(), +editar()

- **Agendamento**

- Atributos: id, data_hora, paciente_id, dentista_id, status
- Métodos: +agendar(), +cancelar()

- **Pagamento**

- Atributos: id, valor, data, status
- Métodos: +registrar()

5.2 Diagrama de Relacionamento (Conceitual)

As relações entre as classes são definidas da seguinte forma:

- **Usuario (Dentista)** realiza muitos **Agendamentos** (1 para N).
- **Paciente** possui muitos **Agendamentos** (1 para N).
- **Agendamento** gera um **Pagamento** (1 para 1).

6 Análise de Riscos

Risco 1: Alto acoplamento entre HTML e JavaScript (alterações de ID no DOM podem quebrar a lógica).

Risco 2: Segurança de dados no cliente (Client-Side).

Solução Proposta: O uso do sistema deve ser restrito a computadores confiáveis da clínica, visto que a base de dados reside no navegador local.