



**ADS – ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ARQUITETURA DE SISTEMAS**

**PROFESSOR(A): RENATO WILIAM**

**NOME: ALISSA GARCIA MOREIRA**

**SISTEMA DE GESTÃO DE CLÍNICA**

**BOA VIAGEM-CE**

**2025**

# Entrevista Simulada — Levantamento de Requisitos para um Sistema de Gestão de Clínica Odontológica

## Proposta do Sistema

O SmileLab é um sistema odontológico desenvolvido para gerenciar clínicas dentárias, oferecendo funcionalidades específicas para diferentes perfis de usuário: administrador, secretárias e dentistas. O sistema tem como objetivo principal centralizar e otimizar todas as operações de uma clínica odontológica, desde o cadastro de pacientes até o agendamento de consultas, gestão financeira e acompanhamento de tratamentos.

O sistema busca proporcionar uma experiência intuitiva e eficiente, permitindo que os profissionais da saúde concentrem-se no atendimento aos pacientes enquanto o SmileLab gerencia toda a parte administrativa e operacional. Com interfaces específicas para cada tipo de usuário, o sistema garante que cada profissional tenha acesso apenas às funcionalidades relevantes para seu trabalho, mantendo a segurança e organização dos dados.

## 1. Requisitos Funcionais

**RF001** - Login de Usuários: Permite que usuários pré-cadastrados façam login no sistema utilizando credenciais de usuário e senha.

**RF002** - Controle de Acesso por Perfil: Direciona automaticamente o usuário para a interface correspondente ao seu perfil (administrador, secretária ou dentista)

**RF003** - Logout: Permite que usuários autenticados façam logout do sistema.

**RF013** - Calendário de Agendamentos: O Sistema exibe um calendário mensal interativo para visualização e seleção de datas

**RF014** - Seleção de Horários: O Sistema mostra horários disponíveis para agendamento naquela data.

**RF015** - Agendamento de Consultas Específicas: O Sistema permite agendar consultas selecionando paciente, dentista, data, horário e especialidade.

**RF016** - Controle de Conflitos: O Sistema impede agendamentos em horários já ocupados para o mesmo dentista.

**RF017** - Visualização de Agenda: Exibe agenda semanal organizada por atendimento mais próximo.

**RF018** - Dashboard Financeiro: O Sistema apresenta resumo financeiro com entradas, saídas e saldo do mês

**RF019** - Registro de Pagamentos: Permite visualizar pagamentos registrados no sistema

**RF020** - Relatórios : O Sistema gera relatórios (Financeiro, Pacientes Cadastrados e Dentistas Cadastrados) em formato CSV para análise externa.

## 2. Requisitos Não Funcionais

**RNF001** - Autenticação Básica: Sistema utiliza autenticação simples com credenciais armazenadas no código.

**RNF002** - Controle de Acesso: Diferentes níveis de acesso baseados no perfil e credenciais do usuário.

**RNF003** - Isolamento de Dados: Cada perfil acessa apenas as funcionalidades e dados relevantes.

**RNF004** - Interface Amigável: Design intuitivo com uso de emojis e ícones para facilitar a identificação das funcionalidades

**RNF005** - Layout Responsivo: Interface adaptável para diferentes tamanhos de tela e dispositivos

**RNF006** - Navegação Intuitiva: Sistema de navegação lateral com categorização clara das funcionalidades

**RNF007** - Feedback Visual: Animações e transições suaves para melhor experiência do usuário.

**RNF008** - Armazenamento Local: Utiliza localStorage do navegador para persistência de dados

**RNF009** - Acesso Remoto: Pode ser acessado via GitHub Pages

**RNF010** - Código Monolítico: Sistema desenvolvido em arquivo único concentrando todas as funcionalidades e interações.

**RNF011** - Organização Modular: Apesar do código monolítico, a lógica é organizada em módulos JavaScript

**RNF012** - Compatibilidade com Navegadores: Funciona em navegadores modernos que suportam ES6+ e localStorage

**RNF013** - Tecnologias Web Padrão: Utiliza HTML5, CSS3 e JavaScript.

### 3. Módulos

#### 3.1 Administrador

Dashboard: Visão geral do sistema com resumo financeiro e próximos agendamentos

Gestão de Pacientes: Cadastro, edição e remoção completa de pacientes

Gestão de Dentistas: Cadastro e gerenciamento de dentistas e suas especialidades

Agendamentos: Calendário completo para agendamento de consultas

Financeiro: Controle de entradas, saídas e saldo financeiro

Relatórios: Geração de relatórios em CSV para todos os módulos

#### 3.2 Secretaria

Dashboard: Visualização dos próximos agendamentos

Cadastro de Pacientes: Interface para cadastro de novos pacientes

Agendamentos: Sistema completo de agendamento de consultas

Visualização de Agendamentos: Lista de todos os agendamentos registrados

Financeiro: Lançamento de Pagamentos dos clientes.

### 3.3 Dentista

Minha Agenda: Visualização semanal da agenda de consultas

Meus Pacientes: Lista de pacientes atendidos pelo dentista

Detalhes do Paciente: Acesso ao prontuário completo do paciente

## 4. Arquitetura do Software

Arquitetura Monolítica: Todo o sistema está contido em um único arquivo HTML.

Frontend Completo: Interface, lógica de negócio e dados no cliente.

Armazenamento Local: Persistência de dados via localStorage do navegador.

Complexidade de Manutenção: Código extenso e difícil de navegar

Performance no Desenvolvimento: Travamentos em máquinas menos potentes, demora no carregamento de arquivos extensos que não serão usados no momento

Dificuldade de Debug: Localização de problemas complexa devido ao código agregado.

## 5. Tecnologias Utilizadas

HTML5: Estruturação da aplicação e semântica

CSS3: Estilização avançada com Grid, Flexbox e animações

JavaScript ES6+: Lógica de aplicação com módulos e classes

LocalStorage API: Persistência de dados no navegador

GitHub Pages: Hospedagem e deploy da aplicação

## 6. Divisão das Camadas

### 6.1 Camada de Apresentação (HTML)

Estrutura de todas as telas e componentes

Formulários de cadastro e edição

Tabelas de listagem de dados

Modal de detalhes do paciente

### 6.2 Camada de Estilos (CSS)

Estilos globais e reset

Estilos específicos por tela

Layout responsivo com media queries

Animações e transições

### 6.3 Camada de Lógica (JavaScript)

Módulo de utilitários

Módulo de autenticação

Módulos específicos por perfil

Gestão de estado e dados

## 7. Atributos e variáveis por Seção

### 7.1. Módulo de Utilitários

```
const Utilitarios = {
```

```
    // Sistema de armazenamento
```

```
    armazenamento: {
```

```
        obter(chave, valorPadrao) - Acessa dados do localStorage
```

```
        definir(chave, valor) - Salva dados no localStorage
```

```
    },
```

```
// Geração de identificadores  
gerarId() - Cria IDs únicos para registros  
  
// Formatação de dados  
formatarMoeda(valor) - Formata valores monetários  
formatarData(data) - Formata datas para exibição
```

```
// Seletores DOM  
seletor(seletor, raiz) - Busca elemento único  
seletores(seletor, raiz) - Busca múltiplos elementos  
}
```

## 7.2. Módulo de Login

```
const ModuloLogin = {  
    inicializar() - Configura eventos de login  
    abrirTela(tipo) - Redireciona para tela específica  
}
```

## 7.3. Módulo de Pacientes

```
const ModuloPacientes = {  
    // Variáveis de estado  
    pacientes: [] // Array de pacientes cadastrados
```

```
// Métodos principais  
inicializar() - Configura formulário e tabela  
salvarPaciente() - Adiciona novo paciente  
renderizarPacientes() - Atualiza lista na interface  
editarPaciente(id) - Preenche formulário para edição
```

```
removerPaciente(id) - Remove paciente do sistema  
abrirModalPaciente(id) - Exibe detalhes completos  
}
```

#### 7.4. Módulo de Agendamentos

```
const ModuloAgendamentos = {
```

Variáveis de estado

dataSelecionada: null // Data atualmente selecionada

dataAtual: new Date() // Data de referência do calendário

Métodos principais

inicializar() - Configura calendário e seletores

renderizarCalendario() - Gera interface do calendário

renderizarHorariosDisponiveis() - Mostra horários livres

confirmarAgendamento() - Registra novo agendamento

```
}
```

#### 7.5. Módulo Administrativo

javascript

```
const ModuloAdmin = {
```

Métodos de navegação

configurarNavegacao() - Gerencia troca entre abas

mostrarAba(aba) - Exibe conteúdo da aba selecionada

Gestão de dentistas

inicializarDentistas() - Configura formulário de dentistas

salvarDentista() - Adiciona novo dentista

renderizarDentistas() - Atualiza lista de dentistas

Relatórios

```
    exportarCSV(dados, campos, nomeArquivo) - Gera arquivos  
    CSV  
}
```

## 8. Funções

### 8.1. Seção do Administrador

#### 8.1.1. Dashboard

renderizarDashboard() - Atualiza resumo financeiro e agendamentos

renderizarFinanceiro() - Calcula e exibe dados financeiros

#### 8.1.2. Gestão de Dentistas

adicionarCampoEspecialidade() - Dinamicamente adiciona campos de especialidade

removerDentista(id) - Remove dentista do sistema

#### 8.1.3. Relatórios

exportarRelatorioPacientes() - Gera CSV com dados de pacientes

exportarRelatorioDentistas() - Gera CSV com dados de dentistas

exportarRelatorioFinanceiro() - Gera CSV com dados financeiros

### 8.2. Seção da Secretaria

#### 8.2.1. Dashboard

renderizarDashboard() - Exibe próximos agendamentos

#### 8.2.2. Agendamentos

renderizarAgendamentosSecretaria() - Lista todos os agendamentos

### 8.3. Seção do Dentista

#### 8.3.1. Agenda

carregarAgendaDentista() - Exibe agenda semanal do dentista

`carregarPacientesDentista()` - Lista pacientes atendidos

### 8.3.2. Pacientes

`abrirPaciente(id)` - Abre modal com detalhes do paciente

### 8.4. Funções Globais

`fecharModalPaciente()` - Fecha modal de detalhes

`fazerLogout()` - Encerra sessão do usuário

`configurarBotoesLogout()` - Configura eventos de logout

### 8.5 Entidades de Dados:

Paciente: Dados completos do paciente

Dentista: Informações dos dentistas e especialidades

Agendamento: Registros de consultas agendadas

Pagamento: Novo - Registro de pagamentos

AtendimentoPago: Controle de atendimentos pagos

### 8.6 Componentes de UI:

ModalPaciente: Modal de detalhes do paciente

ModalPagamento: Novo - Modal para registro de pagamentos

### 8.7 Relacionamentos Principais:

Composição: Utilitarios → Armazenamento

Dependência: Todos os módulos → Utilitarios

Associação: Módulos ↔ Entidades

Uso: ModuloSecretaria → ModalPagamento