ORM Agendamento

Especificação de Requisitos de Software

Para ORM Agendamento

Versão <1.0>

João Alves Cordeiro

Autor 2

Autor 3

[Observação: O template a seguir é fornecido para uso com o Rational Unified Process (RUP). O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Qualquer parágrafo inserido após esse estilo será definido automaticamente como normal (estilo=BodyText).]

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| <dd/mm/yyyy/semestre> | <x.x> | <detalhes> | <nome> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabela de Conteúdo

1. Introdução 4

1.1 Finalidade 4

1.2 Escopo 4

1.3 Definições, Acrônimos, e Abreviações 4

2. Descrição Geral 4

2.1 Perspectiva do Produto 4

2.2 Funções do Produto 4

2.3 Características do Usuário 4

2.4 Restrições 4

2.5 Suposições e pendencias 5

3. Requisitos Específicos 5

3.1 Requisitos Funcionais 5

3.1.1 Gerenciamento de Usuários 5

3.1.2 Gerenciamento de Pacientes 5

3.1.3 Agendamento de Consultas 5

3.1.4 Gerenciamento Financeiro 5

3.2 Requisitos de Performance 5

3.2.1 Tempo de resposta 5

3.2.2 Capacidade 6

3.2.3 Utilização de Recursos 6

3.3 Outros Requisitos 6

3.3.1 Requisitos de Usabilidade 6

3.3.2 Requisitos de Confiabilidade 6

4. Riscos 6

4.1.1 Riscos Identificados 6

4.1.2 Estratégias para Gerenciamento de Riscos 6

Especificação de Requisitos de Software

# Introdução

A Especificação de Requisitos de Software (SRS) descreve o comportamento externo do sistema de agendamento médico desenvolvido em Java. Esta SRS abrange todos os requisitos funcionais e não funcionais do sistema, bem como as restrições de design e outros fatores necessários para fornecer uma visão completa e abrangente dos requisitos do software.

## Finalidade

A finalidade desta SRS é descrever o comportamento completo do sistema de agendamento médico, incluindo os requisitos funcionais e não funcionais, restrições de design, e outros fatores relevantes. O documento serve como uma referência para desenvolvedores e testadores para garantir que o sistema atenda às necessidades especificadas.

## Escopo

Esta SRS se aplica ao sistema de agendamento médico desenvolvido em Java, que inclui módulos para gerenciar usuários (médicos, secretárias), pacientes, agendamentos, contas e parcelas financeiras. O sistema será utilizado por consultórios médicos para gerenciar consultas e acompanhamento financeiro.

## Definições, Acrônimos e Abreviações

 SRS: Software Requirements Specification (Especificação de Requisitos de Software)

 ORM: Object-Relational Mapping (Mapeamento Objeto-Relacional)

 CRUD: Create, Read, Update, Delete (Criar, Ler, Atualizar, Excluir)

 SQL: Structured Query Language (Linguagem de Consulta Estruturada)

 OOP: Object-Oriented Programming (Programação Orientada a Objetos)

# Descrição Geral

#### **2.1 Perspectiva do Produto**

O sistema de agendamento médico é um aplicativo de software desenvolvido em Java que utiliza um ORM baseado em OOP para gerenciar a persistência dos dados. Ele facilita o agendamento de consultas, gerenciamento de pacientes e médicos, e controle financeiro das contas e parcelas.

#### **2.2 Funções do Produto**

* **Gerenciamento de Usuários**: Adicionar, atualizar e remover usuários (médicos, secretárias).
* **Gerenciamento de Pacientes**: Adicionar, atualizar e remover pacientes.
* **Agendamento de Consultas**: Criar, atualizar e cancelar agendamentos de consultas médicas.
* **Gerenciamento Financeiro:** Gerenciar contas e parcelas associadas aos agendamentos.

#### **2.3 Características do Usuário**

* **Médicos**: Consultar agendamentos e pacientes.
* **Secretárias**: Gerenciar agendamentos, pacientes e contas.

#### **2.4 Restrições**

* O sistema deve utilizar um ORM para gerenciar o manuseio dos dados.
* O sistema deve ser realizado utilizando o paradigma de programação OOP

#### **2.5 Suposições e Dependências**

* É necessário ter ao menos um computador para atuar como servidor do sistema.
* A rede deve estar disponível para utilizar o sistema.
* Necessário se ter o conhecimento básico de Java para desenvolver o projeto.

# Requisitos Específicos

## Requisitos Funcionais

##### **3.1.1 Gerenciamento de Usuários**

* **R1**: O sistema deve permitir a criação de usuários
* **R2**: O sistema deve permitir a atualização das informações dos usuários.
* **R3**: O sistema deve permitir a inativação de usuários.

##### **3.1.2 Gerenciamento de Pacientes**

* **R4**: O sistema deve permitir a criação de perfis de pacientes com informações pessoais e de contato.
* **R5**: O sistema deve permitir a atualização das informações dos pacientes.
* **R6**: O sistema deve permitir a inativação de pacientes.

##### **3.1.3 Agendamento de Consultas**

* **R7**: O sistema deve permitir o agendamento de consultas médicas.
* **R8**: O sistema deve permitir a atualização dos agendamentos.
* **R9**: O sistema deve permitir o cancelamento de agendamentos.

##### **3.1.4 Gerenciamento Financeiro**

* **R10**: O sistema deve permitir a criação de contas associadas aos agendamentos.
* **R11**: O sistema deve permitir a criação e atualização de parcelas financeiras.
* **R12**: O sistema deve permitir o acompanhamento do status de pagamento das contas.

## Requisitos de Performance

##### **3.2.1 Tempo de Resposta**

* **R1**: O tempo de resposta para a criação, atualização e exclusão de registros deve ser inferior a 2 segundos.

##### **3.2.2 Capacidade**

* **R2**: O sistema deve suportar uma média de 5 usuários simultâneos.

##### **3.2.3 Utilização de Recursos**

* **R3**: O sistema deve utilizar no máximo 40% da capacidade da CPU e 40% da memória disponível.

## Outros Requisitos

##### **3.3.1 Requisitos de Usabilidade**

* **R1**: O tempo de treinamento necessário para que usuários normais se adequem ao básico do sistema não será superior a 1 dias.

##### **3.3.2 Requisitos de Confiabilidade**

* **R1**: O sistema deve estar disponível 100% do tempo.
* **R2**: O tempo médio para reparo deve ser de no máximo 2 hora.

# Riscos

#### **4.1.1 Riscos Identificados**

* **1**: Risco de falhas de comunicação com o banco de dados.
* **2**: Cliente perder a conexão com a rede
* **3**: Maquina que ficara alojada o servidor acabar se danificando
* **4**: Acabar a energia do local

#### **4.1.2 Estratégias para Gerenciamento de Riscos**

* **Backup Regular:** Realizar backups regulares do banco de dados para garantir que os dados possam ser recuperados em caso de falha.
* **Armazenamento Local:** Utilizar armazenamento local da base de dados para que os dados possam ser salvos localmente independentemente de haver conexão com internet.