**Documentação do Software [Nome do Software][Versão]**

**1. Introdução**

**1.1. Objetivo**

Este documento tem como finalidade descrever detalhadamente o software desenvolvido, incluindo requisitos, estrutura do código-fonte, testes realizados e demais informações relevantes.

**1.2. Escopo**

O software foi desenvolvido para [descrever a finalidade do software] e será utilizado por [descrever o público-alvo].

**1.3. Definições, Acrônimos e Abreviações**

* **RF**: Requisito Funcional
* **RNF**: Requisito Não Funcional
* **API**: Interface de Programação de Aplicação
* **CI/CD**: Integração Contínua e Entrega Contínua

**1.4. Referências**

[Listar normas, manuais ou referências técnicas utilizadas no projeto]

**Sites\*:**

SOBRENOME, Nome. **Título da matéria.** Nome do site, ano. Disponível em: . Acesso em: dia, mês e ano.

**Livro online\*:**

AUTOR OU ORGANIZAÇÃO. **Título**. Cidade: Responsável pela publicação, ano. Total de páginas. Disponível em: URL. Acesso em: dia, mês e ano.

**Livro físico:**

Nome do autor. Título do livro: Subtítulo (se houver). Edição (se houver). Local de publicação: Editora, Ano de publicação.

\*Caso não tenha data, indicar s.d.

**2. Requisitos do Software**

**2.1. Requisitos Funcionais**

| **ID** | **Descrição** | **Prioridade** |
| --- | --- | --- |
| RF001 | O sistema deve permitir o cadastro de usuários. | Alta |
| RF002 | O sistema deve permitir login e autenticação de usuários. | Alta |
| RF003 | O sistema deve permitir a geração de relatórios. | Média |

**2.2. Requisitos Não Funcionais**

| **ID** | **Descrição** | **Prioridade** |
| --- | --- | --- |
| RNF001 | O sistema deve responder às solicitações em menos de 2 segundos. | Alta |
| RNF002 | O sistema deve estar disponível 99,9% do tempo. | Alta |
| RNF003 | O sistema deve utilizar criptografia para armazenamento de senhas. | Alta |

**3. Arquitetura e Código-Fonte**

**3.1. Tecnologias Utilizadas**

* Linguagem de programação: [Ex: Python, Java, JavaScript]
* Banco de dados: [Ex: MySQL, PostgreSQL]
* Frameworks: [Ex: Django, Spring Boot, React]
* Controle de versão: [Ex: GitHub, GitLab]

**3.2. Estrutura do Código-Fonte**

/project

│── src/ # Código-fonte principal

│── tests/ # Testes automatizados

│── docs/ # Documentação do projeto

│── config/ # Configurações do sistema

│── README.md # Documentação inicial

│── requirements.txt # Dependências do projeto

**3.3. Principais Componentes**

* **Módulo de Autenticação:** Gerencia usuários e permissões.
* **Módulo de Relatórios:** Gera relatórios personalizados para os usuários.
* **Módulo de API:** Responsável por comunicação entre serviços.

**4. Testes Realizados**

**4.1. Tipos de Testes**

* **Testes Unitários:** Validam funcionalidades individuais do sistema.
* **Testes de Integração:** Verificam a comunicação entre módulos.
* **Testes de Aceitação:** Avaliam se o software atende aos requisitos do cliente.

**4.2. Ferramentas Utilizadas**

* [Ex: JUnit, Selenium, Postman]

**4.3. Cobertura de Testes**

* Cobertura de código: [Ex: 85%]
* Número de testes executados: [Ex: 150]
* Erros encontrados e corrigidos: [Descrever brevemente]

**5. Implantação e Manutenção**

**5.1. Processo de Deploy**

* Ambiente de produção: [Ex: AWS, Azure, Heroku]
* Pipeline CI/CD configurado: [Sim/Não]
* Backup e recuperação de dados: [Descrever estratégia]

**5.2. Monitoramento e Suporte**

* Ferramentas de monitoramento: [Ex: Prometheus, Grafana]
* Política de manutenção: [Ex: Atualizações trimestrais]

**6. Conclusão**

Este documento serve como guia para desenvolvimento, manutenção e evolução do software, garantindo sua qualidade e continuidade. Para mais detalhes, consulte a documentação técnica e código-fonte disponível no repositório do projeto.

**7. Anexos**

[Listar diagramas, fluxogramas e outros documentos complementares]

**Fotos do produto funcionando (teste de protótipo)**

[]

Figura 1: protótipo montado.

[]

Figura 2: protótipo em funcionamento.