

Levantamento de Requisitos do Software

Referências:

IEEE Std. 830 – 1993. IEEE Recommended for Software Requirements Specifications

Filho, W.P.P. Engenharia de Software: Fundamentos, Métodos e Padrões. LTC: Rio de Janeiro, 2001.

INTERGRANTES DO GRUPO

Arthur Souza Lima (RA: 12723112788)

Aurea Reis (RA: 1272313156)

Alfredo Ruas Neto (RA: 1272326487)

Daniel Santos Lopes (RA: 12723130024)

Enrique Silva dos Reis (RA: 12724149018)

Elvis Oliveira dos Reis (RA: 1272318921)

Edioelson Júnior A. B. Teixeira (RA: 1272318423)

Gabriel Silva Magalhães (RA: 1272313274)

Gabriel Moreira de Oliveira (RA: 1272311981)

Hanspeter Dietiker (RA: 1272313332)

Natan Oliveira da Silva (RA: 12723211400)

➤ **Introdução**

➤ **Objetivo**

O objetivo deste documento é fornecer uma descrição detalhada dos requisitos do sistema de gerenciamento de Ferramentas de Software, incluindo suas funcionalidades, dentre elas listagem de Ferramentas de Software, na qual haverá filtragem a partir dos Tipos de Ferramentas. Este documento destina-se ao grupo de desenvolvedores do trabalho da A3, das disciplinas Gestão e qualidade de software e Modelos, métodos e técnicas da engenharia de software.

➤ **Escopo do Produto**

O sistema de gerenciamento de Ferramentas de Software é uma aplicação web que permite aos usuários, realizar operações de Listagem a partir do tipo de Ferramentas e Nome de Ferramentas de Software.

➤ **Material de Referência**

Para o desenvolvimento desse sistema utilizamos a metodologia ágil Kanban: <https://www.atlassian.com/br/agile/kanban>;

Para a implementação de commits usamos o conventional commits: <https://www.conventionalcommits.org/en/v1.0.0/> ;

Para boas práticas seguimos uma linha de raciocínio parecida com o livro Código Limpo de Robert Cecil Martin: <https://www.amazon.com.br/Código-Limpo-Robert CMartinebook/dp/B085Q2K632> ;

Para inspiração, utilizamos o site: <https://theresanaiforthat.com/>

➤ **Visão Geral do Sistema**

Este documento está estruturado para fornecer uma visão detalhada dos requisitos do sistema de listagem de Ferramentas de Software. Ele inclui seções que descrevem os Objetivos, Arquitetura, Estimativas do Projeto, Escopo, Materiais de referência, bem como uma visão geral das funcionalidades do sistema.

➤ **Descrição Geral do Sistema**

Interfaces de Usuário

- Funcionará no Windows 7, 10, 11;
- Preparação do ambiente com HTML5, CSS3, Java Script, VSCode.

➤ **Requisitos de Adaptação ao Ambiente**

- Configuração do Front End com HTML5, CSS, Java Script e VS Code;
- Automação do CI/CD com GitActions;
- Gerenciamento de projeto e colaboração com Jira Software, utilizando a Metodologia Ágil Kanban;
- Manutenção atualizada das dependências durante o desenvolvimento do projeto.

➤ **StakeHolders do Projeto**

- Desenvolvedores e Equipe de Desenvolvimento: São responsáveis pela criação e implementação do sistema.
- Usuários Finais: Usuários que utilizarão o sistema para buscar e gerenciar ferramentas de software. Estes podem ser desenvolvedores de software, engenheiros de qualidade, gerentes de projeto, e outros profissionais que necessitam de ferramentas de software específicas para suas tarefas diárias.
- Professores e Orientadores: Professores das disciplinas de Gestão e Qualidade de Software e Modelos, Métodos e Técnicas da Engenharia de Software, que supervisionam e avaliam o projeto, fornecendo orientação e feedback aos desenvolvedores.
- Organização Acadêmica (Unifacs): A instituição de ensino (Unifacs) que fornece os recursos, infraestrutura e suporte necessário para a realização do projeto. A universidade também pode usar o projeto como exemplo para futuros estudantes ou como base para projetos similares.
- Equipe de Suporte Técnico: Equipe responsável por manter a infraestrutura técnica necessária para o desenvolvimento do projeto, incluindo servidores, ferramentas de CI/CD (GitActions), e plataformas de colaboração (Jira Software).
- Comunidade de Software Livre e de Desenvolvimento: Comunidades que podem se beneficiar dos resultados do projeto, seja através do uso do sistema ou da adoção de práticas e tecnologias recomendadas. A documentação e as melhores práticas utilizadas no projeto podem ser compartilhadas com essas comunidades para fins educacionais e de desenvolvimento contínuo.

➤ **Funções do Produto**

- Permite aos usuários buscar Ferramentas de Software específicos pelo filtro Tipo de Ferramentas, retornando detalhes completos incluindo site (documentação da Ferramenta), Nome e tipo de Ferramenta.

➤ **Características dos Usuários**

- **Usuários Padrão:** podem consultar tipos de Ferramentas de Software, dentre o filtro disponível a partir da aplicação (Nome, Tipo de Ferramenta).

➤ **Representação Arquitetural do Sistema de Gerenciamento de Ferramentas de Software**

Baseada no modelo "4+1", a seguir está a representação arquitetural do projeto de Sistema de Gerenciamento de Ferramentas de Software:

➤ **Visão de Caso de Uso (Caso de Uso)**

- Público: Todos os envolvidos no projeto, incluindo desenvolvedores, analistas, integradores e gerentes.
- Área: Requisitos funcionais.
- Artefato da MDS: Realização dos Casos de Uso.
- Descrição: Esta visão identifica e descreve os casos de uso principais do sistema, como a busca de ferramentas de software por nome ou tipo, a listagem de ferramentas disponíveis, e o acesso às informações detalhadas de cada ferramenta, incluindo nome, tipo e link para a documentação.

➤ **Visão Lógica (Realização dos Casos de Uso)**

- Público: Analistas e desenvolvedores.
- Área: Implementação da lógica de negócios.
- Artefato da MDS: Realização dos Casos de Uso.
- Descrição: Esta visão detalha a estrutura lógica do sistema, mostrando como os casos de uso são realizados por meio de componentes de software. Inclui a representação dos principais fluxos de dados e funcionalidades do sistema, como cadastro, busca, atualização e exclusão de ferramentas de software, além da consulta detalhada das informações das ferramentas.

➤ **Visão de Processo (Performance, Escalabilidade, Concorrência)**

- Público: Integradores e desenvolvedores.
- Área: Aspectos de desempenho, escalabilidade e concorrência.
- Artefato da MDS: Não especificado.
- Descrição: Esta visão aborda aspectos relacionados ao desempenho e à eficiência do sistema, incluindo sua capacidade de lidar com um grande volume de usuários, transações e dados. Também considera questões de escalabilidade e concorrência para garantir que o sistema possa crescer e funcionar de forma eficaz em diferentes cenários, suportando múltiplos usuários consultando e atualizando informações simultaneamente.

➤ **Visão de Implementação (Componentes de Software)**

- Público: Programadores e desenvolvedores.
- Área: Implementação do sistema em termos de componentes de software.
- Artefato da MDS: Componentes de Software.
- Descrição: Nesta visão, são identificados e detalhados os componentes de software que compõem o sistema, incluindo:
 - Interface do Usuário: Desenvolvida com HTML5 e CSS3, Java Script permitindo que os usuários interajam com o sistema de forma intuitiva.
 - Lógica de Negócios: Implementada para gerenciar as operações principais, como busca e filtragem de ferramentas de software.
 - Acesso a Dados: Componentes responsáveis por acessar e manipular os dados armazenados.
 - Segurança: Mecanismos para garantir a segurança dos dados e operações.

➤ **Visão de Implementação (Nodos físicos)**

- Público: Gerência de Configuração, Administradores de Sistemas.
- Área: Implantação física e distribuição do sistema.
- Artefato da MDS: Nodos físicos.
- Descrição: Esta visão descreve a infraestrutura física necessária para hospedar e implantar o sistema, incluindo:
 - Servidores: Para hospedagem do sistema e banco de dados.

- Redes: Configuração da rede para garantir a conectividade e a comunicação eficiente entre os componentes do sistema.
- Bancos de Dados: Armazenamento das informações das ferramentas de software.
- Recursos de Hardware e Software: Necessários para a operação e manutenção do sistema.
- Escalabilidade e Disponibilidade: Planejamento para suportar crescimento e garantir a disponibilidade contínua do sistema.
- Segurança da Infraestrutura: Considerações para proteger o sistema contra ameaças externas e garantir a integridade dos dados.

Essa representação arquitetural fornece uma visão abrangente do sistema de gerenciamento de Ferramentas de Software, abordando diferentes aspectos desde os requisitos funcionais até a implementação e implantação física do sistema.

➤ Cronograma Projetado do Produto

O cronograma do projeto será dividido em etapas principais, incluindo análise de requisitos, design, implementação, testes e lançamento. O tempo estimado para cada etapa pode variar, mas uma projeção inicial sugere o seguinte:

- Análise de Requisitos: 2 semanas
- Implementação: 4 semanas
- Lançamento e Entrega: 2 semanas

➤ Riscos e Considerações

Alguns dos principais riscos e considerações a serem levados em conta durante o desenvolvimento do projeto incluem:

- Complexidade dos Requisitos: Requisitos adicionais podem surgir durante o desenvolvimento, o que pode impactar o cronograma e os recursos necessários.
- Dependências Externas: O projeto pode depender de tecnologias ou serviços externos, cuja disponibilidade ou confiabilidade pode afetar o progresso do projeto.
- Mudanças de Escopo: Mudanças no escopo do projeto podem ocorrer durante o desenvolvimento, exigindo ajustes no cronograma e nos recursos.

➤ Conclusão da Estimativa do Produto

Essas estimativas fornecem uma visão preliminar do escopo, recursos necessários e cronograma projetado para o desenvolvimento do site de Busca de Ferramentas de Software. À medida que o projeto avança, essas estimativas serão refinadas e atualizadas para refletir com mais precisão os requisitos e o progresso real do projeto.

➤ **Requisitos Específicos**

Visão Geral

O sistema foi desenvolvido para facilitar a gestão de Ferramentas de Software, oferecendo funcionalidade importante. Os usuários podem buscar tipos de Ferramentas que necessitam no momento, através do nome ou pelo tipo da ferramenta. Sendo assim, facilitador para a busca de ferramentas de software em websites.

➤ **Requisitos Funcionais**

- **Read (busca) de Ferramentas de Software**
 - Buscar Ferramentas de Software
 - Filtrar através Tipo de Ferramenta de Software

➤ **Requisitos não-funcionais**

- **Tecnologias Utilizadas:**
 - HTML5;
 - CSS3;
 - Java Script;
 - Visual Studio Code;
 - Git Actions;
 - Jira Software;

➤ Estudo de Viabilidade

Viabilidade Técnica

- **Tecnologias e Ferramentas Utilizadas:** O projeto utiliza uma variedade de tecnologias bem estabelecidas e amplamente adotadas no desenvolvimento de software.

Viabilidade Econômica:

- **Benefícios Potenciais:** O sistema de gerenciamento de Ferramentas de Software pode trazer benefícios significativos, como melhorar a eficiência na busca e gestão de ferramentas de software, o que pode resultar em uma economia de tempo e recurso para os usuários.

Viabilidade Operacional:

- **Manutenção e Suporte:** A manutenção atualizada das dependências durante o desenvolvimento do projeto e a utilização de práticas como automação CI/CD com GitActions podem contribuir para a operacionalidade contínua do sistema.
- **Adoção pelo Usuário:** A integração de metodologias ágeis como Kanban e o uso de tecnologias familiares aos usuários, como VSCode, podem facilitar a adoção e a utilização do sistema.

➤ Informações de Suporte

- Documentação Spring Boot: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/> ;
- Repositório do Projeto: <https://github.com/hanspeterdietiker/A3-ProjectUnifacs> .
- KRU41: The “4+1” view model of software architecture, Philippe Kruchten, November 1995, <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>