Engenharia e Projeto de Software

Turma B - 0724 - Virtual - GV 2.0
Professor Dr. Romes Heriberto
Aluno(s):
Helam da Costa Sobrinho

DOCUMENTOS DE REQUISITO DE SISTEMA

Sistema de Gerenciamento de Tarefas

Versão 0.0.5

Brasília-DF,
2025

Histórico de alterações

DATA	VERSÃO	DESCRIÇÃO	AUTOR
20/09/2025	0.0.1	Criação do documento de requisitos, estruturando os elementos do arquivo	Helam Sobrinho
21/09/2025	0.0.2	Desenvolvimento do Sistema e suas necessidades para funcionamento real	Helam Sobrinho
21/09/2025	0.0.3	Desenvolvimento da opção de prioridade	Helam Sobrinho
21/09/2025	0.0.4	Desenvolvimento das subtasks para maior controle de demandas.	Helam Sobrinho
21/09/2025	0.0.5	Desenvolvimento das opções de mover os cards	Helam Sobrinho

Índice

Documento de Requisitos de Sistema

1. Introdução

Este documento especifica os requisitos para a criação de um protótipo de Sistema de Gerenciamento de Tarefas. O objetivo é entregar um protótipo funcional (front-end) e a documentação que descreve os requisitos e o processo de desenvolvimento. O sistema auxilia usuários a organizar tarefas diárias, priorizar atividades e acompanhar prazos e foi criado como uma forma de organizar demandas e tarefas diárias, evitando o uso de bloco de notas e notas adesivas no desktop do usuário.

2. Descrição Geral do Sistema

O sistema proposto é uma aplicação web leve que permite ao usuário adicionar, editar, marcar como concluída e remover tarefas. Público-alvo: estudantes, profissionais autônomos e pequenas equipes que precisam de uma ferramenta simples para gerenciar atividades. Cenário de aplicação: gerenciamento pessoal de estudos, lista de tarefas do dia a dia e acompanhamento de pequenas entregas.

3. Requisitos Funcionais

RF001 - Adicionar Tarefa

- **Prioridade:** Essencial
- **Descrição:** O sistema deve permitir ao usuário adicionar uma tarefa com título, descrição, prioridade e data de vencimento.
- **Atores:** Usuário
- **Fluxo de Eventos:**

1. O usuário interage com a interface;
2. o sistema valida os campos e atualiza o armazenamento local;
3. O resultado é exibido na lista.

RF002 - Editar Tarefa

- **Prioridade:** Essencial
- **Descrição:** O sistema deve permitir editar informações de uma tarefa existente.
- **Atores:** Usuário
- **Fluxo de Eventos:**
    1. O usuário interage com a interface;
    2. O sistema valida os campos e atualiza o armazenamento local;
    3. O resultado é exibido na lista.

RF003 - Remover Tarefa

- **Prioridade:** Essencial
- **Descrição:** O sistema deve permitir excluir tarefas da lista.
- **Atores:** Usuário
- **Fluxo de Eventos:**
    1. O usuário interage com a interface;
    2. O sistema valida os campos e atualiza o armazenamento local;
    3. O resultado é exibido na lista.

RF004 - Marcar Concluída

- **Prioridade:** Essencial
- **Descrição:** O sistema deve permitir marcar ou desmarcar uma tarefa como concluída.
- **Atores:** Usuário
- **Fluxo de Eventos:**
    1. O usuário interage com a interface;
    2. O sistema valida os campos e atualiza o armazenamento local;
    3. O resultado é exibido na lista.

RF005 - Listar Tarefas

- **Prioridade:** Essencial
- **Descrição:** O sistema deve exibir a lista de tarefas, com filtros e pesquisa.
- **Atores:** Usuário
- **Fluxo de Eventos:**
    1. O usuário interage com a interface;
    2. O sistema valida os campos e atualiza o armazenamento local;
    3. O resultado é exibido na lista

RF005 – Adicionar Subtarefas

- **Prioridade:** Essencial
- **Descrição:** O sistema permite criação de subtarefas com opções de completar e deletar.
- **Atores:** Usuário
- **Fluxo de Eventos:**
    1. O usuário interage com a interface;
    2. O sistema valida os campos e atualiza o armazenamento local;
    3. O resultado é exibido na lista

RF005 – Mover cards

- **Prioridade:** Essencial
- **Descrição:** O sistema permite mover cards para facilitar a organização das demandas.
- **Atores:** Usuário
- **Fluxo de Eventos:**

1. O usuário interage com a interface;
2. O sistema valida os campos e atualiza o armazenamento local;
3. O resultado é exibido na lista

4. Requisitos Não Funcionais

NF001 - Usabilidade

- **Descrição:** A interface deve ser intuitiva e responsiva (funcionar em desktop e mobile).

NF002 - Desempenho

- **Descrição:** A aplicação deve responder rapidamente (operações locais sem tempo de espera perceptível).

NF003 - Persistência

- **Descrição:** Dados devem ser persistidos no navegador utilizando localStorage para protótipo.

NF004 - Persistência

- **Descrição:** Validar entradas e evitar execução de código por meio de inserção de scripts; proteger contra XSS simples.

5. Arquitetura do Sistema (Implementação do Protótipo)

Arquitetura: Aplicação web (front-end) single-page, sem back-end para o protótipo.

Linguagens e ferramentas: HTML5, CSS3 e JavaScript (ES6). Armazenamento: localStorage do navegador.

O protótipo foi implementado com foco em usabilidade e demonstração das funcionalidades essenciais. Para produção, a arquitetura deve evoluir para um back-end com API REST, autenticação e armazenamento em banco de dados.

6. Modelagem do Sistema (Diagrama UML)

Diagrama de Casos de Uso:

- **Atores:** Usuário
- Adicionar Tarefa;
- Editar Tarefa;
- Remover Tarefa;
- Marcar/Desmarcar Concluída;
- Mover card;
- Adicionar subtarefa.

Diagrama de Classes:

- Classe Tarefa { id, title, description, priority, due, done, created }
- Classe ListaDeTarefas { tarefas[], métodos: add, edit, remove, toggleDone }.

Diagrama de Sequência (Adicionar Tarefa):

Usuário -> Interface -> Validação -> Armazenamento(localStorage) -> Interface atualiza lista.

7. Testes e Garantia de Qualidade

Os testes podem ser executados ou apenas descritos. A ideia é ter uma Tabela com casos de teste aplicados.

Entrada → Ação → Resultado esperado → Resultado obtido.

E, também, a Indicação de quais requisitos foram atendidos e quais ainda precisam evoluir (caso tenha rodado algum teste).

Testes Unitários:

Casos de teste (exemplos):

1) Adicionar tarefa válida

Entrada: título = 'Estudar para prova'

Ação: clicar em Adicionar

Resultado esperado: tarefa aparece na lista com dados corretos.

Resultado obtido: (quando executar, preencher aqui).

2) Adicionar tarefa sem título

Entrada: título = ''

Ação: clicar em Adicionar

Resultado esperado: alerta solicitando título.

Resultado obtido: (quando executar, preencher aqui).

3) Marcar tarefa como concluída

4) Adicionar Subtasks

5) Mover cards

8. Documentação do Desenvolvimento

Relato das etapas de construção:

1. Levantamento de requisitos e definição do escopo.

2. Protótipo de interface (HTML/CSS) e implementação das funcionalidades em JavaScript.

3. Testes manuais no navegador e ajustes de usabilidade.

Dificuldades encontradas: Limitação de persistência no protótipo (localStorage) e falta de autenticação de usuário. Como foram superadas: mantivemos foco em UX e em funcionalidades essenciais.

Próximos passos: Implementar back-end, autenticação, sincronização entre dispositivos e notificações push..

9. Entrega

Arquivos entregues com o protótipo funcional (client-side):

- index.html

- styles.css

- app.js

- README.txt

O protótipo pode ser executado localmente abrindo o arquivo index.html em um navegador.

Sugere-se publicar o projeto em repositório público (GitHub) e, se necessário, publicar como extensão ou PWA.

10. Conclusão

A atividade permitiu aplicar conceitos de Engenharia de Software na prática, do levantamento de requisitos até a entrega de um protótipo funcional. O desenvolvimento demonstrou a importância de priorizar funcionalidades essenciais e considerar a evolução arquitetural para um sistema em produção.

11. Referências

DOCUMENTAÇÃO oficial da API de Extensões do Chrome. Disponível em: https://developer.chrome.com/docs/extensions/mv3/getstarted/. Acesso em: 2025.

MDN Web Docs - WebExtensions. Disponível em: https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Your_first_WebExtension. Acesso em: 2025.

OWASP Top 10. Disponível em: https://owasp.org/Top10/pt_BR/. Acesso em: 2025.

W3Schools, Tutorials e artigos sobre desenvolvimento front-end.

12. Anexos

https://github.com/HelaaoDev/sistematizacao_gerenciamento_tarefas_helam_sobrinho