ATIVIDADE PARCIAL

DOCUMENTAÇÃO DE REQUISITOS - APLICATIVO DE LISTA DE TAREFAS

Equipe:

Maria Emelly da Silva - 2318766
Nilton Fernandes Vieira - 2318113
Robério Leal de Sousa - 2314798
Marcelo Falcão de Oliveira - 2318131
João Rômulo Martins Alves - 2317039
Samira Moura Sousa - 2316027

FORTALEZA – CE

MARÇO 2025

- **Objetivo do documento**

O documento tem como objetivo descrever os requisitos funcionais e não funcionais do aplicativo de Lista de Tarefas, além de mapear como cada requisito está implementado no código, como a criação, edição, exclusão e organização de tarefas, além da implementação de filtros dinâmicos para facilitar a busca por informações. Também apresenta os conceitos técnicos aplicados, como funções de alta ordem, closures e expressões lambda, garantindo um desenvolvimento padronizado e de fácil manutenção.

- **Escopo**

O aplicativo permitirá aos usuários criar, editar, excluir e organizar tarefas, utilizando conceitos de Programação Funcional. Além disso, contará com filtros dinâmicos para busca personalizada de tarefas por status, nome, ID ou descrição. A aplicação fará uso de funções de alta ordem, closures, expressões lambda e list comprehension para otimizar a manipulação dos dados. Também será implementado um sistema de categorização e priorização das tarefas, possibilitando a organização eficiente do fluxo de trabalho dos usuários.

- **Requisitos**
- **Requisitos Funcionais**

D	**Descrição**	**Implementação**
F01	O usuário deve poder criar uma nova tarefa.	Função: const createTarefa = async (req, res) => { try { const novaTarefa = await tarefaModel.createTarefa(req.body);
F02	O usuário deve poder excluir uma tarefa.	Função: const deleteTarefa = async (req, res) => { try {

		const removeTarefa = await tarefaModel.deleteTarefa(parseInt(req.params.id)); if (removeTarefa) {
F03	O usuário deve poder editar uma tarefa.	Função: const updateTarefa = async (id, Tarefa) => { const {nometarefa, descricao, status} = Tarefa;
F04	O usuário deve poder listar todas as tarefas.	Funçã : const getTarefaById = async (id) => { const result = await pool.query('SELECT * FROM tarefas_tb WHERE id = $1', [id]);

- **Requisitos Não Funcionais**

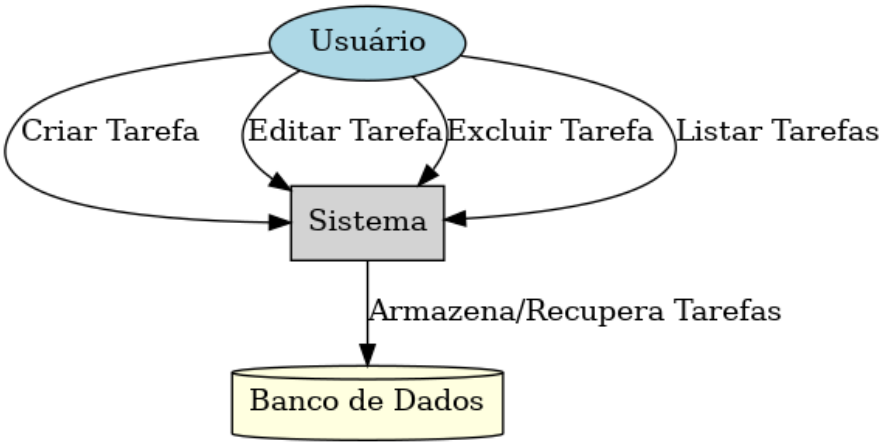
ID	Descrição
RNF01	O sistema deve ser implementado utilizando a linguagem JavaScript.
RNF02	O código deve seguir princípios da Programação Funcional.
RNF03	O sistema deve garantir a integridade dos dados das tarefas.

- **Programação Funcional no Código**

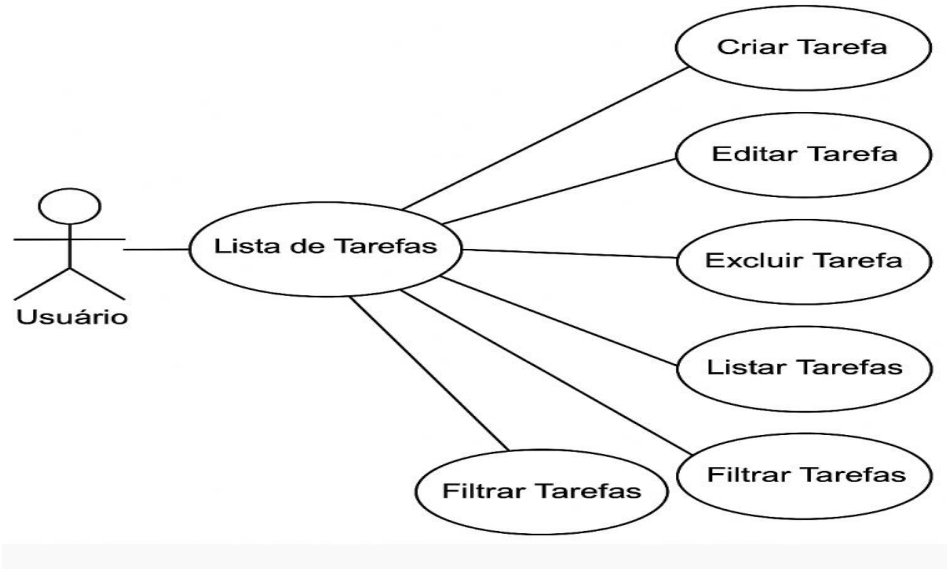
Conceito	Implementação
Função Lambda	Utilizada em “const getTarefas = async (req, res) => { try {

	const tarefas = await tarefaModel.getAllTarefas();”
List Comprehension	Usada para gerar listas de tarefas filtradas: const filtroTermo = await tarefaModel.filterByTermo(buscarTermo);
Closure	const filterByTermo = createFilterDn((buscarTermo) => (tarefa) => tarefa.descricao.toLowerCase().includes(buscarTermo.toLowerCase()));
Função de Alta Ordem	Utilizada na função “const filterByTermo = createFilterDn((buscarTermo) => (tarefa) => tarefa.descricao.toLowerCase().includes(buscarTermo.toLowerCase())); ” que recebe outra função como parâmetro.

- **Diagrama do Sistema**



- **Diagrama de Casos de Uso**



- **Divisão de tarefas do grupo**

Tarefa-01: Estudos e escolha da linguagem e Implementação do código**.**
Participantes:(Robério, Marcelo e Nailton).

Tarefa-02: Pesquisa e Documentação de requisitos**.**
Participantes:(Samira, Emelly e Romulo).

Tarefa-03: Redigir documentação de requisitos e compartilhar via Google driver**.**
Participantes:(Samira e Robério)

Tarefa-04: Compartilhar implementação do código via github**.**
Participantes:(Nailton).
Link GitHub: https://github.com/nailton-vieira/ativ-unifor-n704