



TRABALHO FINAL DA DISCIPLINA DCC603

1. Objetivo do Sistema:

O sistema desenvolvido consiste em um ToDo List, uma aplicação web que permite ao usuário gerenciar suas tarefas diárias de forma prática e eficiente. A aplicação é composta por um front-end, integrado a um back-end que interage com um banco de dados para realizar operações de CRUD (criar, ler, atualizar e deletar). O objetivo principal do sistema é permitir que o usuário crie, visualize, edite, conclua e apague tarefas, oferecendo uma interface intuitiva e de fácil navegação.

Além das operações de CRUD, o sistema implementa uma funcionalidade adicional que permite ao usuário marcar as tarefas como concluídas, atualizando seu status visualmente, através da interface.

2. Funcionalidades:

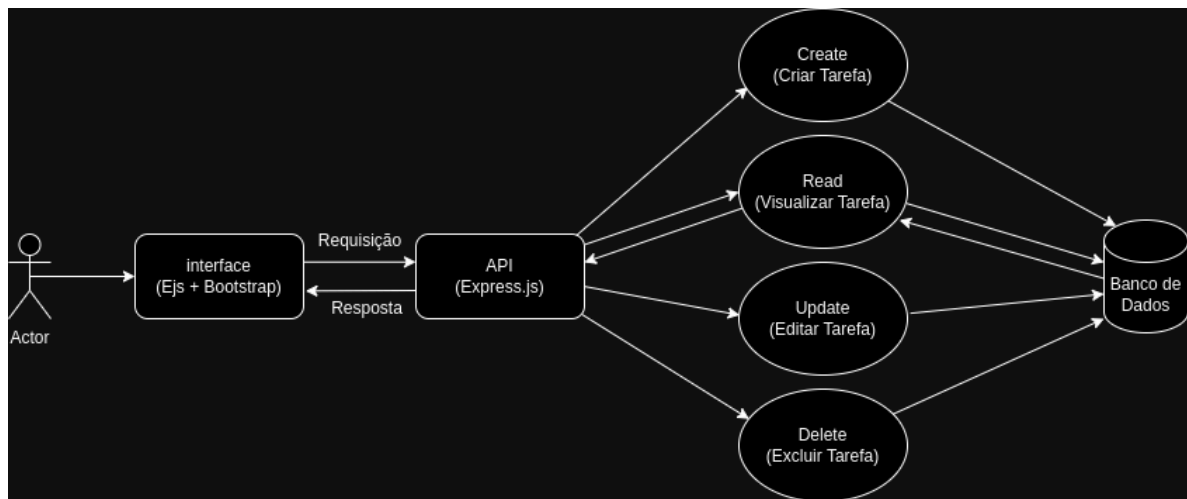
- Adicionar Tarefas: O usuário pode adicionar uma nova tarefa através de um formulário no front-end. A tarefa é inserida no banco de dados e listada na página inicial.
- Visualizar Tarefas: Todas as tarefas adicionadas pelo usuário são exibidas em uma lista. A interface permite que o usuário veja todas as tarefas registradas, incluindo o status de cada uma (pendente ou concluída).
- Editar Tarefas: O usuário pode selecionar uma tarefa para edição. A tarefa será carregada e poderá ser atualizada no banco de dados.
- Excluir Tarefas: O sistema permite ao usuário apagar uma tarefa da lista. Uma confirmação é exibida antes da exclusão, e a tarefa é removida do banco de dados.
- Marcar Tarefas como Concluídas: O usuário pode marcar uma tarefa como concluída ou reverter essa ação. A tarefa riscada na interface indica que ela foi finalizada, atualizando seu estado no banco de dados.

3. Requisitos Técnicos

- Front-end: Desenvolvido utilizando **HTML**, **CSS** e **Bootstrap** para garantir uma interface responsiva e amigável. O uso de **EJS** permite a renderização dinâmica das páginas.
- Back-end: Desenvolvido em **Node.js** utilizando o framework **Express.js**, o sistema é responsável por gerenciar todas as operações de CRUD com o banco de dados e renderizar as páginas com as informações atualizadas.

- Banco de dados: utiliza o banco de dados **MongoDB** para armazenar as tarefas criadas pelo usuário. Cada tarefa possui um ID único, além de um campo para o conteúdo da tarefa e um indicador de status (concluído ou não).
- Integração com o Front-end: O back-end e o banco de dados são integrados com o front-end através de requisições **HTTP**, permitindo que o sistema seja atualizado em tempo real conforme o usuário interage com ele.

4. Diagrama de Caso



5. Conclusão:

A aplicação **ToDo List** cumpre os requisitos definidos pela disciplina, implementando um sistema web full stack que realiza todas as operações de CRUD. O sistema foi desenvolvido de maneira a ser fácil de usar, com uma interface intuitiva e uma estrutura de back-end robusta, garantindo que os dados do usuário sejam manipulados de forma eficiente.

O projeto demonstra o uso de tecnologias modernas no desenvolvimento de sistemas web, como **Node.js**, **Express.js** e **MongoDB**, além de uma integração completa entre front-end e back-end.