



Relatório do Desafio Técnico - Meisters Solutions

- Autor: Daniel Zaki Sommer
- E-mail: danielsommer@alunos.utfpr.edu.br
- Telefone: +55 (41) 99708-5707
- GitHub: github.com/danisommer
- LinkedIn: linkedin.com/in/danisommer

1. Levantamento de Requisitos

Objetivo

O objetivo principal deste projeto é desenvolver um sistema completo de gerenciamento de tarefas, utilizando como tecnologias: Java (com Spring Boot) para o backend, MySQL para armazenamento de dados, e HTML/JS/CSS para o frontend.

Requisitos Funcionais

Configuração do ambiente: Inicialmente, foi necessário configurar um ambiente de desenvolvimento integrado que incluía um projeto web Java utilizando Spring Boot para o backend, um banco de dados MySQL para armazenar as tarefas e um projeto client-side em JavaScript para o frontend. Esta configuração permitiu a comunicação fluida entre as diferentes camadas da aplicação.

Backend (Java): Para o backend, foi criado um modelo de dados que representa as tarefas a serem gerenciadas pelo sistema. Cada tarefa será caracterizada por um título descritivo, uma descrição mais detalhada e um status que pode ser pendente, em progresso ou concluída. Foram implementadas operações CRUD (Create, Read, Update, Delete) para permitir a criação, leitura, atualização e exclusão de tarefas através de APIs RESTful, assegurando um sistema flexível e escalável.

Além das operações básicas, foram introduzidas lógicas de negócio específicas para garantir a integridade dos dados. Por exemplo, novas tarefas só podem ser criadas em dias úteis, tarefas só podem ser atualizadas ou deletadas se estiverem no status pendente e somente é permitida a exclusão de tarefas criadas há mais de 5 dias. Essas restrições ajudam a manter a consistência e a relevância das informações no sistema.

Frontend (HTML/CSS/JavaScript): No frontend, foi desenvolvida uma interface de usuário intuitiva utilizando HTML para estruturar o conteúdo, CSS para estilização e JavaScript para interatividade. A página inicial do aplicativo inclui uma barra de pesquisa e filtragem, uma lista dinâmica de tarefas e um formulário interativo para adicionar novas tarefas ao sistema. A interface permite que os usuários visualizem, adicionem, editem, marquem como concluídas e excluam tarefas de maneira eficiente e amigável.

Integração Front-End e Back-End: A integração entre o frontend e o backend foi realizada através de chamadas assíncronas utilizando Fetch API ou bibliotecas similares em JavaScript. Essas chamadas são responsáveis por enviar requisições HTTP ao servidor backend para realizar as operações CRUD nas tarefas. Essa abordagem não apenas promove uma comunicação eficiente, mas também garante uma resposta rápida do servidor, melhorando a experiência do usuário final.

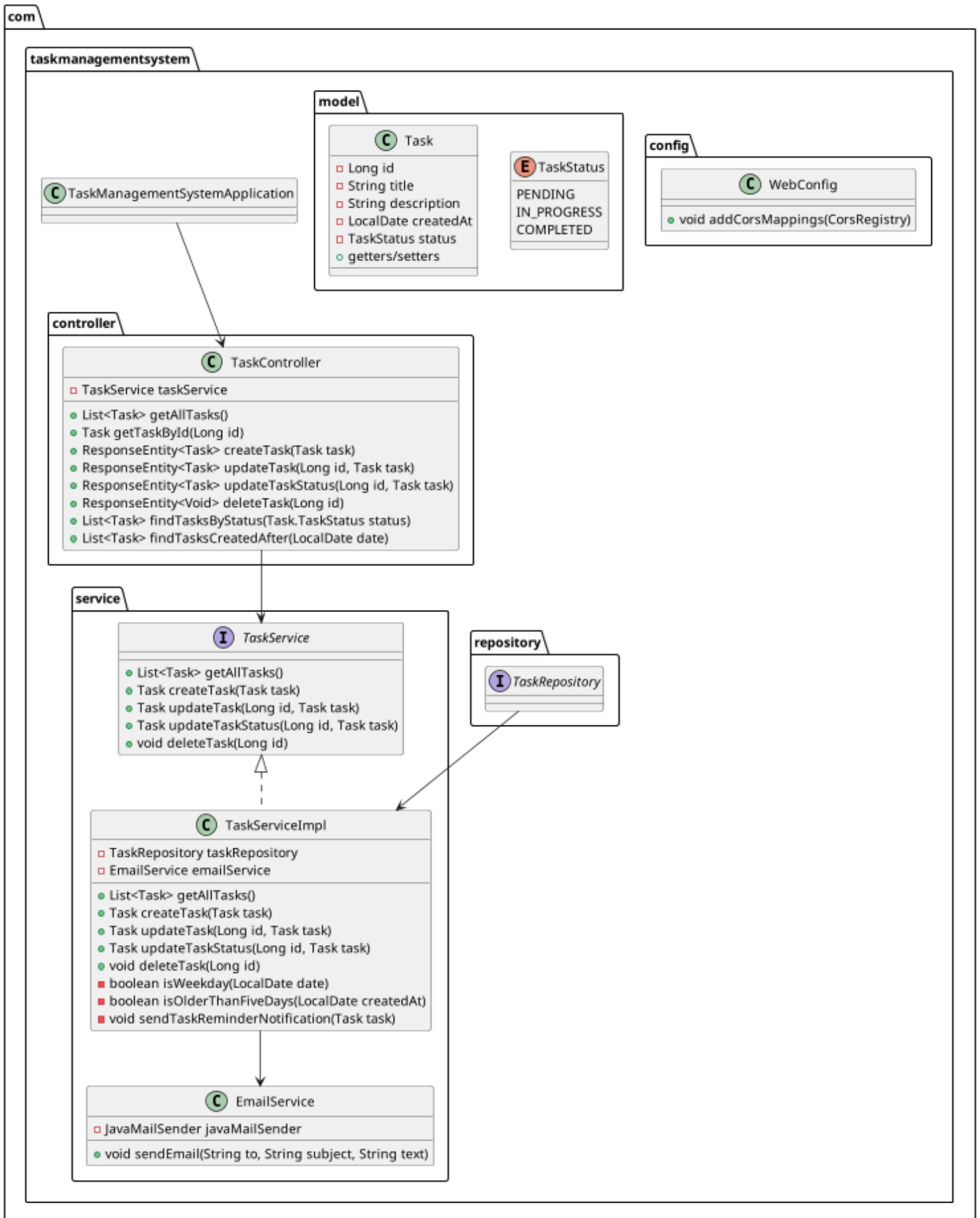
Extras: Para enriquecer ainda mais a funcionalidade do sistema, foram implementadas funcionalidades adicionais como busca e filtragem de tarefas, permitindo aos usuários encontrar rapidamente as informações desejadas. A adição de paginação na lista de tarefas ajuda a gerenciar grandes volumes de dados de forma organizada e acessível. Melhorias na interface do usuário foram feitas através de animações suaves e transições elegantes para tornar a experiência de navegação mais agradável e intuitiva. Um recurso de notificação por e-mail foi implementado para enviar lembretes automáticos aos usuários sobre tarefas pendentes, proporcionando uma maneira eficaz de manter todos os envolvidos informados e atualizados sobre tarefas pendentes.

2. Implementação

Esquema do banco de dados

Tasks		
id	bigint	auto_increment
title	varchar(255)	
description	varchar(255)	
status	enum('PENDING', 'IN_PROGRESS', 'COMPLETED')	
created_at	date	

Diagrama de Classes UML



3. Resultados

Interface do site com diferentes tasks:

Task Manager

1

2

3

4

Task 5 - Task 5 Description
Created on: 24/06/2024
Status: IN_PROGRESS

Task 4 - Task 4 Description
Created on: 24/06/2024
Status: COMPLETED

Task 3 - Task 3 Description
Created on: 24/06/2024
Status: COMPLETED

Task 2 - Task 2 Description
Created on: 24/06/2024
Status: PENDING

Mecanismo de busca de tasks em funcionamento:

Task Manager

task 1

Filter

Enter task title

Enter task description

Add Task

Task 1 - Task 1 Description

Created on: 24/06/2024

Status: IN_PROGRESS

Task 1.1 - Task 1.1 Description

Created on: 24/06/2024

Status: COMPLETED

Task 1.2 - Task 1.2 Description

Created on: 24/06/2024

Status: PENDING

Start Task

Edit

Delete

Mecanismo de filtragem de tasks em funcionamento:

Task Manager

Search tasks...

Filter

Status: Completed

Start Date: 16/06/2024

End Date: 24/06/2024

Apply

Task 1 - Task 1 Description

Created on: 24/06/2024

Status: COMPLETED

Task 3 - Task 3 Description

Created on: 24/06/2024

Status: COMPLETED

Task 1.1 - Task 1.1 Description

Created on: 24/06/2024

Status: COMPLETED

qwewqeqwe - wqeqweqw

Created on: 24/06/2024

Status: COMPLETED

Task 3 - Task 3 Description

Created on: 24/06/2024

Status: COMPLETED

Task 4 - Task 4 Description

Created on: 24/06/2024

Status: COMPLETED

4. Conclusões

Avaliação

Após a implementação e testes completos, conclui-se que o sistema atende integralmente aos requisitos funcionais estabelecidos. Ele proporciona um gerenciamento eficiente e intuitivo de tarefas, com validações robustas e interações fluidas entre os usuários e as funcionalidades oferecidas pelo sistema.

Decisões Tomadas

As tecnologias utilizadas durante o desenvolvimento, como o Spring Boot para o backend e JavaScript para o frontend, mostraram-se acertadas. O Spring Boot ofereceu a base necessária para um backend estável e seguro, enquanto o JavaScript possibilitou a criação de uma interface de usuário moderna e responsiva. Essas escolhas não apenas garantiram a funcionalidade esperada, mas também contribuíram significativamente para a qualidade e usabilidade do sistema como um todo.