

1. Introdução

Este documento descreve os requisitos do sistema para o Task Flow, um aplicativo web para gerenciar tarefas de forma pessoal. O sistema será composto por um frontend em React e um backend em Java (Spring Boot). O objetivo do sistema é permitir a criação, organização, e acompanhamento de tarefas de uma forma intuitiva e eficiente.

2. Objetivo

O objetivo do Gestor de Tarefas é fornecer uma plataforma para que os usuários possam:

Criar tarefas.

Organizar tarefas em listas (To Do) e em KanBan.

Atualizar o status das tarefas (por exemplo, "Em andamento", "Concluído") no método Kanban.

Visualizar o progresso de um conjunto de tarefas.

Criação de áreas de trabalhos para organização de diferentes listas.

3. Requisitos Funcionais

3.1. Usuário

O sistema deve permitir login utilizando credenciais (email e senha).

O sistema deve permitir login com plataformas credenciadas (Google, Facebook).

O sistema deve permitir que os usuários vejam sua própria lista de tarefas.

O sistema deve permitir que os usuários acessem as áreas de trabalho do KanBan.

O sistema deve permitir os usuários a acessarem o fluxo das áreas de trabalho, dentro de cada área de trabalho ele deve ter acesso aos processos, e por si só acesso as tarefas dentro da lista.

3.2. Tarefas

O sistema deve permitir que os usuários criem tarefas.

Cada tarefa deve ter os seguintes campos obrigatórios:

Título: Texto curto que descreve a tarefa.

Descrição: Detalhes sobre a tarefa.

Data de vencimento (opcional).

Prioridade: Indicação de prioridade (por exemplo, alta, média, baixa).

O sistema deve permitir que os usuários editem ou excluam tarefas.

O sistema deve permitir que os usuários movam tarefas entre listas (ex: "A Fazer", "Em andamento", "Concluído").

3.3. Listas de Tarefas

O sistema deve permitir que os usuários criem listas para organizar as tarefas.

Cada lista deve ter um título, como "A Fazer", "Em andamento", "Concluído".

O sistema deve permitir a adição, edição e exclusão de listas.

O sistema deve permitir que os usuários movam as tarefas entre as listas.

3.4. Funcionalidades Avançadas

Arrastar e Soltar: O sistema deve permitir que os usuários arrastem e soltem tarefas entre listas.

Notificações: O sistema deve enviar notificações por email quando as tarefas estiverem perto do prazo de vencimento.

Calendário: O sistema deve permitir o usuário conectar suas tarefas ao Google Calendar e enviar notificações quando a tarefa estiver perto do dia definido.

Busca e Filtros: O sistema deve permitir que os usuários filtrem as tarefas por prioridade e data de vencimento.

3.5. Interface do Usuário

A interface do usuário deve ser responsiva, funcionando bem em dispositivos móveis e desktops.

O layout deve ser intuitivo para todos os usuários.

A interface deve permitir fácil navegação entre tarefas, listas e quadros.

4. Requisitos Não Funcionais

4.1. Performance

O sistema deve ser capaz de lidar com um número de tarefas em tempo real sem degradação perceptível de performance.

O sistema deve ser capaz de realizar requisições ao backend de forma rápida, com baixa latência.

4.2. Escalabilidade

O sistema deve ser projetado para ser escalável, permitindo adicionar mais funcionalidades no futuro (por exemplo, integração com outros serviços, como Slack, etc.).

O sistema deve ser capaz de suportar múltiplos usuários simultaneamente.

4.3. Segurança

O sistema deve garantir que os dados dos usuários sejam mantidos em segurança.

O sistema deve permitir autenticação segura com JWT (JSON Web Tokens) ou OAuth.

A comunicação com o backend deve ser feita através de HTTPS.

O sistema deve validar todos os dados fornecidos pelo usuário (ex: validar que os campos obrigatórios da tarefa sejam preenchidos corretamente).

4.4. Usabilidade

O sistema deve ter uma interface de fácil utilização, com feedback claro para o usuário, como confirmações de ações e mensagens de erro.

O sistema deve ter um design consistente em todas as páginas e seções.

O sistema deve ter uma curva de aprendizado baixa, para que qualquer usuário possa usar sem necessidade de treinamentos complexos.

5. Tecnologias e Ferramentas

5.1. Frontend

React: Biblioteca JavaScript para construir a interface do usuário.

Axios: Para realizar requisições HTTP ao backend.

TailwindCSS: Framework CSS para estilizar a interface de forma responsiva.

React Router: Para navegação entre diferentes páginas do aplicativo.

5.2. Backend (Java)

Spring Boot: Framework Java para criar APIs RESTful.

Spring Security: Para gerenciamento de autenticação e autorização.

JWT: Para autenticação baseada em token.

PostgreSQL (ou outra base de dados relacional): Para armazenar dados do sistema (usuários, tarefas, listas).

JPA/Hibernate: Para a interação com a base de dados.

6. Requisitos de Integração

O sistema deve ser capaz de se comunicar com o backend por meio de uma API RESTful utilizando os seguintes endpoints principais:

GET /dashboard: Para mostrar a navegação de KanBan e ToDo.

GET /workspace-kanban: Para mostrar as áreas de trabalho do KanBan.

GET /workspace-todo: Para mostrar as áreas de trabalho do ToDo.

POST /workspace-kanban: Para criar uma área de trabalho do KanBan.

POST /workspace-todo: Para criar uma área de trabalho do ToDo.

GET /flow-kanban: Para mostrar os processos do KanBan.

GET /flow-todo: Para mostrar as listas do ToDo.

POST /flow-kanban: Para criar os processos do KanBan.

POST /flow-todo: Para criar as listas do ToDo.

GET /tasks-kanban: Para mostrar as tarefas do KanBan.

GET /tasks-todo: Para mostrar as tarefas do ToDo.

POST /tasks-kanban: Para criar as tarefas do KanBan.

POST /tasks-todo: Para criar as tarefas do ToDo.

GET /tasks-kanban/{id}: Para mostrar as especificações de uma tarefa do KanBan.

GET /tasks-todo/{id}: Para mostrar as especificações de uma tarefa do ToDo.

PUT /tasks-kanban/{id}: Para alterar as especificações de uma tarefa do KanBan.

PUT /tasks-todo/{id}: Para alterar as especificações de uma tarefa do ToDo.

DELETE /tasks-kanban/{id}: Para deletar uma tarefa do KanBan.

DELETE /tasks-todo/{id}: Para deletar uma tarefa do ToDo.

DELETE /flow-kanban/{id}: Para deletar os processos do KanBan.

DELETE /flow-todo/{id}: Para deletar as listas do ToDo.

DELETE /workspace-kanban/{id}: Para deletar uma área de trabalho do KanBan.

DELETE /workspace-todo/{id}: Para deletar uma área de trabalho do ToDo.