

Proposta de Projeto Final

ProjectFlow API — Sistema de Gestão de Projetos e Tarefas

Descrição do serviço:

A ProjectFlow API é um serviço RESTful para gerenciamento completo de projetos, tarefas e equipes. O sistema permite o controle do ciclo de vida completo de projetos, desde o planejamento até a conclusão, com funcionalidades para atribuição de tarefas, acompanhamento de progresso e gerenciamento de recursos humanos.

Casos de uso principais:

- Cadastro e autenticação de usuários (roles: MEMBER, MANAGER, ADMIN)
- Gerenciamento de projetos: com metadados (nome, descrição, prazo, orçamento)
- Equipes: agrupamento de usuários para trabalhar em projetos (relação muitos-para-muitos)
- Tarefas : unidades de trabalho dentro de projetos com status e prioridade
- Comentários: sistema de comentários para tarefas
- Anexos: arquivos associados a tarefas (URLs para armazenamento externo)
- Relatórios: progresso de projetos, tarefas por status, etc.

Tecnologias:

Java + Spring Boot, Spring Data JPA, Hibernate, Validação, JWT, Swagger, MySQL

Definição inicial das rotas (prefixo /api/v1)

Autenticação

- POST `/registro` - Registrar usuário
- POST `/login` - Login (retorna JWT)

Usuários

- POST `/usuarios` - Criar usuário
- GET `/usuarios` - Listar usuários
- GET `/usuarios/{id}` - Buscar usuário por ID
- PUT `/usuarios` - Atualizar usuário
- DELETE `/usuarios/{id}` - Excluir usuário

Projetos

- POST `/projetos` - Criar projeto
- GET `/projetos` - Listar projetos
- GET `/projetos/{id}` - Buscar projeto por ID
- PUT `/projetos` - Atualizar projeto

- PUT '/projetos/{id}/status' - Atualizar status do projeto
- DELETE '/projetos/{id}' - Excluir projeto

Equipes

- POST '/equipes' - Criar equipe
- GET '/equipes' - Listar equipes
- GET '/equipes/{id}' - Buscar equipe por ID
- PUT '/equipes' - Atualizar equipe
- DELETE '/equipes/{id}' - Remover equipe
- POST '/equipes/{equipeld}/membros/{usuarioId}' - Adicionar membro à equipe
- DELETE '/equipes/{equipeld}/membros/{usuarioId}' - Remover membro da equipe
- GET '/equipes/{equipeld}/membros' - Listar membros da equipe

Tarefas

- POST '/tarefas' - Criar tarefa
- GET '/tarefas' - Listar tarefas
- GET '/tarefas/{id}' - Buscar tarefa por ID
- GET '/tarefas/{id}/anexos' - Listar anexos da tarefa
- PUT '/tarefas' - Atualizar tarefa
- PUT '/tarefas/{id}/status' - Atualizar status da tarefa
- DELETE '/tarefas/{id}' - Remover tarefa

Comentários

- POST '/comentarios' - Criar comentário
- GET '/comentarios' - Listar comentário
- GET '/comentarios/{id}' - Buscar comentário por ID
- PUT '/comentarios' - Atualizar comentário
- DELETE '/comentarios/{id}' - Remover comentário

Anexos

- POST '/anexos' - Criar anexo
- GET '/anexos' - Listar anexos
- GET '/anexos/{id}' - Buscar anexo por ID
- PUT '/anexos' - Atualizar anexo
- DELETE '/anexos/{id}' - Remover anexo

Relatórios

- GET '/relatorios/progresso-projetos' - Progresso dos projetos

Entidades principais e relacionamentos:

Usuario

- id (Long, PK)
- nome (String)
- email (String, unique)
- senha (String)
- role (ENUM: MEMBRO, GESTOR, ADMIN)
- equipe (Set<Equipe>) (FK → Equipe)

Projeto

- id (Long, PK)
- nome (String)
- descricao (String)
- dataInicio (LocalDate)
- deadline (LocalDate)
- orcamento (BigDecimal)
- statusProjeto (ENUM: PLANEJAMENTO, EM_PROGRESSO, COMPLETO, CANCELADO)

Equipe

- id (Long, PK)
- nome (String)
- descricao (String)
- membros (Set<Usuario>) (FK → Usuario)

Tarefa

- id (Long, PK)
- titulo (String)
- descricao (String)
- prioridade (ENUM: BAIXA, MEDIA, ALTA, CRITICA)
- status (ENUM: A_FAZER, EM_PROGRESSO, REVISAO, FEITA)
- horasEstimadas (Integer)
- projeto_id (FK → Projeto)
- cessionario_id (FK → Usuario)

Comentario

- id (Long, PK)
- conteudo (String)
- criadoEm (LocalDateTime)
- autor_id (FK → Usuario)

- tarefa_id (FK → Tarefa)

Anexo

- id (Long, PK)
- nomeArquivo (String)
- URLarquivo (String)
- uploadEm (LocalDateTime)
- tarefa_id (FK → Tarefa)

Relacionamentos:

- Usuario N---N Equipe (tabela join: membros_equipe)
- Equipe 1---N Projeto
- Projeto 1---N Tarefa
- Usuario 1---N Tarefa (como cessionario)
- Tarefa 1---N Comentario
- Tarefa 1---N Anexo
- Usuario 1---N Comentario (como autor)

Mapeamento para Controllers/Services/Models

Models (Entidades JPA):

- Usuarios (@Entity)
- Projeto (@Entity)
- Equipe (@Entity)
- Tarefa (@Entity)
- Comentario (@Entity)
- Anexo (@Entity)

Repositories:

- UsuariosRepository extends JpaRepository<Usuario, Long>
- ProjetoRepository extends JpaRepository<Projeto, Long>
- EquipeRepository extends JpaRepository<Equipe, Long>
- TarefaRepository extends JpaRepository<Tarefa, Long>
- ComentarioRepository extends JpaRepository<Comentario, Long>
- AnexoRepository extends JpaRepository<Anexo, Long>

Services:

- AutenticacaoService (registro, login, JWT)
- UsuarioService (CRUD usuários)
- ProjetoService (CRUD projetos + regras de negócio)
- EquipeService (CRUD equipes + gerenciamento de membros)

- TarefaService (CRUD tarefas + atribuição + atualização status)
- ComentarioService (CRUD comentários)
- AnexoService (CRUD anexos)
- RelatorioService (geração de relatórios)

Controllers:

- AutenticacaoController
- UsuarioController
- ProjetoController
- EquipeController
- TarefaController
- ComentarioController
- AnexoController
- RelatorioController

Detalhes técnicos de implementação

Validações importantes:

- ProjetoCriadoDTO: nome (@NotBlank), deadline (@Future)
- TarefaCriadaDTO: título (@NotBlank), horasEstimadas (@PositiveOrZero)
- UsuarioCriadoDTO: email (@Email)

Regras de negócio:

- Apenas GESTOR/ADMIN pode criar projetos
- Usuários só podem ser atribuídos a tarefas se forem membros da equipe do projeto
- Atualizações de status devem seguir fluxo lógico (ex: A_FAZER → EM_PROGRESSO → REVISAO → FEITA)

Exceções customizadas:

- BusinessException
- ResourceNotFoundException

Fluxos importantes:

- Ao excluir projeto: todas as tarefas atreladas ao projeto serão excluídas