

# INSTITUTO SUPERIOR MIGUEL TORGA

## Programação III (2024/2025)

### Relatório da Entrega – FeedzzTrab

#### Metadados do Projeto

- **Data de Entrega:** 04/04/2025
- **Título:** Sistema de Gestão de Tarefas e Competências FeedzzTrab
- **Autor:** Tiago Peixinho
- **Número de Estudante:** XXXXX
- **Curso:** Licenciatura em Engenharia Informática
- **Professor:** Prof. Nome do Professor
- **GitHub:** <https://github.com/tpeixinho/programacao3>

#### Detalhes da Base de Dados

##### Informação de Hosting

- **Sistema:** MySQL
- **Charset:** utf8mb4
- **Collation:** utf8mb4\_unicode\_ci

##### Credenciais de Acesso

```
DB_HOST=localhost  
DB_USER=[utilizador]  
DB_PASSWORD=[palavra-passe]  
DB_DATABASE=feedzztrab
```

#### Introdução

O FeedzzTrab é um sistema de gestão de tarefas e competências desenvolvido como trabalho prático para a unidade curricular de Programação III. O sistema permite a gestão eficiente de tarefas, utilizadores e suas competências, facilitando a atribuição de trabalhos baseada nas capacidades individuais.

#### Modelação da Base de Dados

##### Estrutura de Tabelas

1. **Funcoes**
  - Gestão de funções de utilizador (admin, gerente, lider\_equipa, programador)
  - Controle hierárquico de permissões
2. **Utilizadores**
  - Gestão de contas de utilizador

- Autenticação e autorização
  - Associação com funções
3. **Competencias**
    - Cadastro de competências disponíveis
    - Base para atribuição de tarefas
  4. **Tarefas**
    - Gestão de tarefas do sistema
    - Estados: nao\_atribuida, atribuida, em\_progresso, concluida
    - Rastreamento de criação e atribuição
  5. **Tarefas\_\_Atribuicoes**
    - Controle de atribuições de tarefas
    - Registro de início e fim de atividades

## Script SQL

```
CREATE TABLE Funcoes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome ENUM('admin', 'gerente', 'lider_equipa', 'programador') NOT NULL UNIQUE
);
```

```
CREATE TABLE Utilizadores (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome_utilizador VARCHAR(50) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    funcao_id INT NOT NULL,
    criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (funcao_id) REFERENCES Funcoes(id)
);
```

*-- [Demais tabelas omitidas por brevidade]*

## OpenAPI Specification

### Endpoints Principais

1. **Autenticação**
  - POST /api/utilizadores/login
  - Autenticação via JWT
2. **Gestão de Utilizadores**
  - GET/POST /api/utilizadores
  - GET/PUT/DELETE /api/utilizadores/{id}
  - PATCH /api/utilizadores/{id}/palavra\_passe
3. **Gestão de Tarefas**
  - GET/POST /api/tarefas
  - PUT/DELETE /api/tarefas/{id}

- 4. **Gestão de Atribuições**
  - POST /api/atribuicoes
  - PATCH /api/atribuicoes/{id}

## Infraestrutura Técnica

### Backend (Node.js/Express)

- Arquitetura MVC
- Autenticação JWT
- Validação de dados
- Gestão de erros centralizada

### Frontend (React)

- Context API para gestão de estado
- Rotas protegidas
- Componentes reutilizáveis
- Interface responsiva

## Detalhes de Implementação

### Sistema de Autenticação

```
const autenticacaoServico = {
  async autenticar(credenciais) {
    const resposta = await api.post('/auth/login', credenciais);
    return resposta.data;
  }
};
```

### Gestão de Permissões

- Hierarquia de funções implementada
- Middleware de autorização
- Validação por rota

### Frontend Implementation

- Páginas principais implementadas
- Sistema de rotas protegidas
- Gestão de estado global
- Feedback visual de ações

## Testes

### Testes Unitários

- Serviços de API
- Componentes React
- Funções de utilidade

### Testes de Integração

- Fluxos completos
- Comunicação API
- Gestão de estado

## Conclusão

O projeto FeedzzTrab atingiu os objetivos propostos, implementando um sistema robusto e escalável para gestão de tarefas e competências. O sistema oferece: - Autenticação segura - Interface intuitiva - API bem documentada - Código modular e manutenível - Gestão eficiente de permissões

## Bibliografia

1. React Documentation. Disponível em: <https://reactjs.org/docs>
2. Node.js Documentation. Disponível em: <https://nodejs.org/docs>
3. Express.js Guide. Disponível em: <https://expressjs.com/guide>
4. JWT.io Documentation. Disponível em: <https://jwt.io>
5. MySQL Documentation. Disponível em: <https://dev.mysql.com/doc>