

Portal Transparência



2024.2

Sumário

1. Introdução

2. Finalidade

3. Objetivo

4. Escopo

5. Visão Geral

6. Visão de Casos de Uso

7. Documentação API

7.1 Objetivo

7.2 Tags

7.3 Endpoints

Equipe: Sync

1. Introdução

Este documento descreve o desenvolvimento do Portal de Transparência da Fundação de Apoio à Pesquisa de Pós- Graduandos (FAPG). O portal será utilizado para fornecer informações detalhadas sobre os projetos apoiados pela fundação, permitindo o acompanhamento público de suas atividades e promovendo a transparência nas informações. A plataforma será desenvolvida utilizando uma arquitetura de microsserviços, visando flexibilidade, e facilidade de manutenção. Além disso, o sistema permitirá que administradores gerenciem projetos e dados, enquanto os usuários poderão consultar e visualizar relatórios detalhados sobre os projetos.

2. Finalidade

A finalidade deste documento é apresentar uma descrição detalhada do desenvolvimento do Portal de Transparência da Fundação de Apoio à Pesquisa de Pós-Graduandos (FAPG). O documento visa fornecer uma visão clara das funcionalidades, requisitos e estrutura do sistema

3. Objetivo

O principal objetivo deste portal é proporcionar um meio transparente e eficiente para a disponibilização das informações dos projetos apoiados pela FAPG. A plataforma permitirá a inserção, edição e visualização de projetos, bem como a geração de relatórios interativos. Através do portal, a FAPG buscará atender às necessidades de transparência e disponibilização de dados, tanto para o público em geral quanto para os dministradores que gerenciam os projetos.

4. Escopo

O sistema será desenvolvido para abranger as seguintes funcionalidades:

- Cadastro de projetos;
- Busca avançada de projetos com diversos filtros (coordenador, status, empresa, data etc.);
- Visualização de detalhes dos projetos e anexos;
- Relatórios/dashboards interativos com gráficos sobre o status dos projetos;
- Exportação de dados para relatórios externos;
- Interface intuitiva e responsiva, adequada para uso em múltiplos dispositivos;

5. Visão Geral

O Portal de Transparência da FAPG será composto por diferentes módulos, divididos entre funcionalidades de administração e de consulta pública. Administradores terão acesso a um painel exclusivo onde poderão gerenciar projetos e dados. Usuários públicos poderão realizar consultas, visualizar detalhes e dos projetos.

6. Visão de Casos de Uso

Este sistema será utilizado por dois tipos principais de usuários: administradores e usuários públicos. Cada um desses grupos de usuários terá acesso a diferentes funcionalidades dentro do portal.

Administrador:

- Gerenciamento completo de projetos: cadastro, edição, exclusão e importação de dados.
- Visualização de dashboards com estatísticas dos projetos.
- Exportação de dados de projetos e dos relatórios.

Usuário Público:

- Visualização dos projetos listados no portal.
- Pesquisa avançada de projetos por coordenador, status, empresa, data, classificação e palavras-chave.
- Visualização dos detalhes completos de cada projeto.

7. Documentação API

7.1 Objetivo

O objetivo deste documento tem como informar ao leitor sobre as mecânicas que foram desenvolvidas em nossa aplicação.

7.2 Tags

Abaixo estão listadas quais foram as seguintes tags utilizadas:

- **projects**
- **auth**
- **documents**

7.3 Endpoints

- **projects/create**
método: **POST**
objetivo: Criar um projeto.
parâmetros:

```
{
  "projectReference": "string",
  "nameCoordinator": "string",
  "projectCompany": "string",
  "projectObjective": "string",
  "projectDescription": "string",
  "projectValue": 0,
  "projectEndDate": "2024-09-29",
  "projectStartDate": "2024-09-29",
  "projectClassification": "string"
}
```

- **response:** (200, OK)

Equipe: Sync

- projects/get/{id}
método: GET
objetivo: Buscar dados de um projeto específico.
parâmetros: Nenhum.
response: (200, OK)
- projects/getAll
método: GET
objetivo: Buscar todos os projetos.
parâmetros: Nenhum.
response: (200, OK)
- auth/login
método: POST
objetivo: Autenticar o administrador ao sistema.
parâmetros:


```
{  
  "email": "string",  
  "password": "string"  
}
```


response: (200, OK)
- documents/get/{documentId}
método: GET
objetivo: Buscar os documentos dos projetos.
parâmetros: Nenhum.
response: (200, OK)
- documents/create
método: POST

Equipe: Sync

objetivo: Criar um documento para o projeto.

parâmetros:

```
{
  "projectId": {
    "type": "string",
    "in": "query",
    "required": true,
    "description": "ID do projeto"
  },
  "typeFile": {
    "type": "string",
    "in": "query",
    "required": true,
    "description": "Tipo do arquivo",
    "enum": [
      "CONTRATO",
      "TERMO_ADITIVO",
      "PLANO_DE_TRABALHO",
      "OUTROS"
    ]
  }
}
```

response: (200, OK)

Equipe: Sync

- /projects/update/{id}

método: **PUT**

objetivo: Atualizar um projeto pelo ID.

Parâmetros:

```
{
  "id": "string",
  "projectReference": "string",
  "nameCoordinator": "string",
  "projectCompany": "string",
  "projectObjective": "string",
  "projectDescription": "string",
  "projectValue": 0,
  "projectEndDate": "2024-10-19",
  "projectStartDate": "2024-10-19",
  "projectClassification": "OUTROS",
  "projectStatus": "NAO_INICIADOS"
}
```

Response: (200, OK)

- /projects/delete/{id}

método: **DELETE**

objetivo: Deletar um projeto pelo ID.

Parâmetros: Nenhum.

Response: (200, OK)

- /projects/filter

método: **GET**

objetivo: Filtrar projetos por critérios.

Parâmetros:

```
{
  "projectReference": "string",
  "projectCompany": "string",
  "nameCoordinator": "string",
  "projectClassification": "OUTROS",
  "projectStatus": "NAO_INICIADOS",
  "projectStartDate": "2024-09-29",
  "projectEndDate": "2024-09-29"
}
```

Response: (200, OK)

Equipe: Sync

- /projects/get/coordinators
método: GET
objetivo: Obter a lista de coordenadores de projetos.
parâmetros: Nenhum.
response: (200, OK)
- /projects/get/companies
método: GET
objetivo: Obter a lista de empresas envolvidas em projetos.
parâmetros: Nenhum.
response: (200, OK)
- projects/get/all/near-end
método: GET
objetivo: Obter projetos próximos ao término.
parâmetros: Nenhum.
response: (200, OK)