

Portal Transparência



2024.2

Equipe: Sync

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 administradores que gerenciam os projetos.

Equipe: Sync

4. Escopo

O sistema será desenvolvido para abranger as seguintes funcionalidades:

- Cadastro, listagem, edição e exclusão de projetos;
- Cadastro, listagem, edição e exclusão de bolsistas;
- Cadastro, listagem, edição e exclusão de tipo de bolsa;
- Cadastro, listagem, edição e exclusão de coordenador;
- Cadastro, listagem, edição e exclusão de empresa;
- Pesquisa avançada de projetos por coordenador, status, empresa, data, classificação e palavras-chave;
- Visualização de detalhes dos projetos e anexos;
- Relatórios/dashboards interativos com gráficos sobre o status dos projetos;
- Exportação de dados para relatórios externos;
- Geração de plano de trabalho e contratos;
- 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.

6. Visão de Casos de Uso

Este sistema será utilizado por dois tipos principais de usuários: administradores e usuários públicos.

Administrador:

- Gerenciamento completo de projetos: cadastro, edição, exclusão e importação de dados.
- Gerenciamento completo de bolsistas: cadastro, edição e exclusão.
- Geração de plano de trabalho e contratos.
- Visualização de dashboards com estatísticas dos projetos.
- Exportação de dados de projetos e dos relatórios.
- Pesquisa avançada de projetos por coordenador, status, empresa, data, classificação e palavras-chave.

Usuário Público:

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

Equipe: Sync

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**
- **dashboard**
- **scholar**
- **grant**
- **coordinators**
- **company**
- **plano-de-trabalho**
- **contract**

7.3 Endpoints

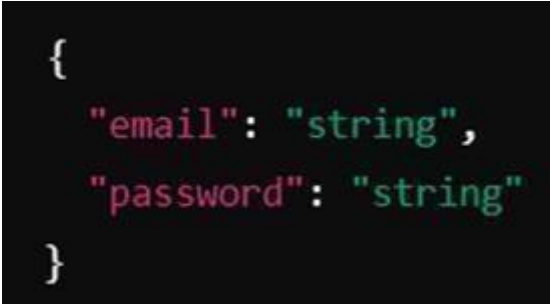
- **/projects/create**
método: **POST**
objetivo: Criar um projeto.
parâmetros:

Equipe: Sync

```
{
  "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)

Equipe: Sync

- documents/create

método: **POST**

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)
- /dashboard/export/{format}
método: GET
objetivo: Exportar os dados do dashboard.
parâmetros: Nenhum.
response: (200, OK)
- /dashboard/count/status/coordinator
método: GET
objetivo: Dados para gerar gráfico de status dos projetos por coordenador.
parâmetros: Nenhum.
response: (200, OK)

Equipe: Sync

- /dashboard/count/status/company
método: GET
objetivo: Dados para gerar o grafico de status dos projetos por empresa.
parâmetros: Nenhum.
response: (200, OK)
- /dashboard/count/month/coordinator
método: GET
objetivo: Dados para gerar grafico de projetos de coordenadores por mês.
parâmetros: Nenhum.
response: (200, OK)
- /dashboard/count/month/company
método: GET
objetivo: Dados para gerar grafico de projetos das empresas por mês.
parâmetros: Nenhum.
response: (200, OK)
- /dashboard/count/investment/company
método: GET
objetivo: Dados para gerar o gráfico de investimento de empresa.
parâmetros: Nenhum.
response: (200, OK)
- /dashboard/count/classification/coordinator
método: GET
objetivo: Dados para gerar o grafico de classificação do coordenador.
parâmetros: Nenhum.
response: (200, OK)

Equipe: Sync

- /dashboard/count/classifications/company
método: GET
objetivo: Dados para gerar grafico de classificação de empresas.
parâmetros: Nenhum.
response: (200, OK)
- /documents/removed
método: PUT
objetivo: Documentos removidos.
parâmetros: Nenhum.
response: (200, OK)
- /documents/add
método: PUT
objetivo: Documentos adicionados.
parâmetros: Nenhum.
response: (200, OK)
- /projects/get/history-projects/{id}
método: GET
objetivo: Buscar o histórico de alterações do projeto.
parâmetros: Nenhum.
response: (200, OK)
- /projects/export/{id}/{format}
método: GET
objetivo: Exportar um projeto no seu determinado formato.
parâmetros: Nenhum.
response: (200, OK)
- /scholar-ship-holder-controller/update/
método: PUT
objetivo: Atualiza os dados de um bolsista.

Equipe: Sync

parâmetros: Nenhum.

response: (200, OK)

- /scholar-ship-holder-controller/create/

método: **POST**

objetivo: Cria um bolsista.

parâmetros:

```
{
  "name": "string",
  "email": "string",
  "cpf": "string",
  "rg": "string",
  "nationality": "string",
  "address": {
    "street": "string",
    "number": "string",
    "neighborhood": "string",
    "city": "string",
    "state": "string",
    "zipCode": "string"
  },
  "grantId": "string",
  "projectId": "string"
}
```

response: (200, OK)

- /scholar-ship-holder-controller/buscar/

método: **GET**

objetivo: Lista todos os bolsistas.

parâmetros: Nenhum.

response: (200, OK)

- /scholar-ship-holder-controller/bucsar/{id}/

método: **GET**

objetivo: Busca os dados de um bolsista específico.

parâmetros: Nenhum.

response: (200, OK)

- /scholar-ship-holder-controller/delete/{id}/

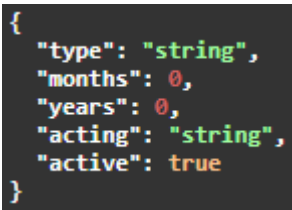
método: **DELETE**

objetivo: Deleta o bolsista.

parâmetros: Nenhum.

Equipe: Sync

response: (200, OK)

- /grant/update/
método: **PUT**
objetivo: Atualiza os dados de uma bolsa.
parâmetros: Nenhum.
response: (200, OK)
- /grant/deactivate/{id}/
método: **PATCH**
objetivo: Desativa os status da bolsa.
parâmetros: Nenhum.
response: (200, OK)
- /grant/create/
método: **POST**
objetivo: Cria uma bolsa.
parâmetros:


```
{
  "type": "string",
  "months": 0,
  "years": 0,
  "acting": "string",
  "active": true
}
```


response: (200, OK)
- /grant/getAll/
método: **GET**
objetivo: Lista de todas as bolsas.
parâmetros: Nenhum.
response: (200, OK)
- /grant/get/{id}/
método: **GET**
objetivo: Pega os dados de uma bolsa específica.
parâmetros: Nenhum.

Equipe: Sync

response: (200, OK)

- /coordinators/update/

método: PUT

objetivo: Atualiza os dados de um coordenador.

parâmetros: Nenhum.

response: (200, OK)

- /coordinators/create/

método: POST

objetivo: Cria um coordenador.

parâmetros:

```
{
  "coordinatorName": "string",
  "coordinatorCPF": "string",
  "coordinatorTelefone": "string",
  "coordinatorEconomicActivity": "string",
  "coordinatorRG": "string",
  "coordinatorNationality": "string",
  "coordinatorMaritalStatus": "string",
  "AddressStreet": "string",
  "AddressNumber": "string",
  "AddressNeighborhood": "string",
  "AddressCity": "string",
  "AddressState": "string",
  "AddressZipCode": "string",
  "addressNeighborhood": "string",
  "addressNumber": "string",
  "addressState": "string",
  "addressCity": "string",
  "addressStreet": "string",
  "addressZipCode": "string"
}
```

response: (200, OK)

- /coordinators/filter/

método: GET

objetivo: Filtra os coordenadores.

parâmetros: Nenhum.

response: (200, OK)

- /coordinators/buscar/

método: GET

objetivo: Lista todos os coordenadores.

parâmetros: Nenhum.

Equipe: Sync

response: (200, OK)

- /coordinators/buscar/{id}/

método: GET

objetivo: Busca os dados de um coordenador específico.

parâmetros: Nenhum.

response: (200, OK)

- /coordinators/delete/{id}/

método: DELETE

objetivo: Deleta os dados de um coordenado.

parâmetros: Nenhum.

response: (200, OK)

- /plano-de-trabalho/gerar/

método: POST

objetivo: Gera um plano de trabalho.

parâmetros:

```
{
  "projectId": "string",
  "coordinatorCPF": "string",
  "coordinatorAddress": "string",
  "coordinatorCity": "string",
  "coordinatorUF": "string",
  "coordinatorCEP": "string",
  "coordinatorTelefone": "string",
  "coordinatorEconomicActivity": "string",
  "coordinatorPeriod": "string",
  "companyRazaoSocial": "string",
  "companyCNPJ": "string",
  "companyResponsavelTecnico": "string",
  "companyTelefone": "string",
  "companyEndereco": "string",
  "companyEmpresaPrivada": true,
  "projetoJustificativa": "string",
  "projetoResultadosEsperados": "string",
  "fases": [
    {
      "fase": "string",
      "descricao": "string"
    }
  ],
  "cronograma": [
    {
      "atividade": "string",
      "indicadorFisico": "string",

```

Equipe: Sync

```
{
  "dataInicio": "string",
  "dataFim": "string"
},
"equipe": [
  {
    "nome": "string",
    "instituicao": "string",
    "competencia": "string"
  }
],
"planoAplicacao": [
  {
    "descricao": "string",
    "total": "string"
  }
],
"cronogramaFinanceiro": [
  {
    "valor": "string"
  }
],
"contratanteNome": "string",
"contratanteCargo": "string",
"dataAssinatura": "string"
}
```

response: (200, OK)

- /plano-de-trabalho/download/{projectid}/
método: GET
objetivo: Baixa o plano de trabalho.
parâmetros: Nenhum.
response: (200, OK)
- /company/update/
método: PUT
objetivo: Atualiza os dados de uma empresa.
parâmetros: Nenhum.
response: (200, OK)
- /company/create/
método: POST
objetivo: Adiciona uma nova empresa no sistema.
parâmetros:

Equipe: Sync

```
{
  "corporateName": "string",
  "cnpj": "string",
  "phone": "string",
  "privateCompany": true,
  "AddressStreet": "string",
  "AddressNumber": "string",
  "AddressNeighborhood": "string",
  "AddressCity": "string",
  "AddressState": "string",
  "AddressZipCode": "string",
  "addressNeighborhood": "string",
  "addressNumber": "string",
  "addressState": "string",
  "addressCity": "string",
  "addressStreet": "string",
  "addressZipCode": "string"
}
```

response: (200, OK)

- /company/filter/
método: GET
objetivo: Filtra as empresas.
parâmetros: Nenhum.
response: (200, OK)
- /company/buscar/
método: GET
objetivo: Faz uma listagem de todas as empresas.
parâmetros: Nenhum.
response: (200, OK)
- /company/buscar/{id}/
método: GET
objetivo: Busca os dados de uma empresa específica.
parâmetros: Nenhum.
response: (200, OK)
- /company/delete/{id}/
método: DELETE
objetivo: Deleta os dados de uma empresa.
parâmetros: Nenhum.
response: (200, OK)

Equipe: Sync

- /contracts/generate/{projectId}/
método: GET
objetivo: Gerar um contrato.
parâmetros: Nenhum.
response: (200, OK)