

Documentação dos Endpoints da API

Backend

Esta seção descreve os endpoints da API RESTful expostos pelo backend do sistema de gerenciamento de TCCs. A API permite a interação com os recursos do sistema, como autenticação, gerenciamento de usuários, alunos, orientadores, TCCs, categorias, subcategorias e cursos.

Autenticação (/auth)

Endpoints responsáveis pelo processo de autenticação, registro e gerenciamento de senhas dos usuários.

Login de Usuário

- **Método HTTP:** POST
- **URL:** /auth/login
- **Descrição:** Autentica um usuário no sistema utilizando email ou matrícula e senha.
- **Parâmetros de Corpo (Request Body):**
 - `LoginRequestDTO` : Objeto JSON contendo `email` (String, opcional), `matricula` (String, opcional) e `password` (String, obrigatório). Apenas um entre `email` e `matricula` deve ser fornecido.
- **Respostas:**
 - `200 OK` : Autenticação bem-sucedida. Retorna um `LoginResponseDTO` com o token JWT (`token`) ou um `LoginResponseMustChangeDTO` se o usuário precisar alterar a senha (`token` , `mustChangePassword: true`).
 - `400 Bad Request` : Se ambos `email` e `matricula` forem fornecidos, ou nenhum deles. Retorna `LoginErroDTO` .
 - `401 Unauthorized` : Credenciais inválidas. Retorna `LoginErroDTO` .

Registro de Novo Usuário

- **Método HTTP:** POST
- **URL:** /auth/register
- **Descrição:** Registra um novo usuário no sistema. Envia um email de boas-vindas com link para definição de senha.

- **Parâmetros de Corpo (Request Body):**
 - RegisterUserDTO : Objeto JSON contendo nomeCompleto , email , matricula .
- **Respostas:**
 - 200 OK : Registro bem-sucedido. Corpo da resposta vazio.
 - 400 Bad Request : Se o email ou matrícula já estiverem em uso. Retorna LoginErroDTO .

Alteração de Senha

- **Método HTTP:** PATCH
- **URL:** /auth/changePassword
- **Descrição:** Permite que um usuário autenticado altere sua própria senha.
- **Parâmetros de Corpo (Request Body):**
 - ChangePasswordRequestDTO : Objeto JSON contendo currentPassword , newPassword , confirmPassword .
- **Autenticação:** Requer token JWT válido no cabeçalho Authorization .
- **Respostas:**
 - 200 OK : Senha alterada com sucesso. Retorna LoginResponseDTO com um novo token JWT.
 - 400 Bad Request (implícito via IllegalStateException): Senha atual incorreta ou novas senhas não coincidem.

Envio de Token para Redefinição de Senha

- **Método HTTP:** POST
- **URL:** /auth/sendPasswordReset
- **Descrição:** Envia um email para o usuário contendo um link com token para redefinição de senha.
- **Parâmetros de Corpo (Request Body):**
 - SendMailResetRequestDTO : Objeto JSON contendo email .
- **Respostas:**
 - 200 OK : Email enviado (ou o usuário não foi encontrado, para evitar enumeração de usuários). Corpo da resposta vazio.

Redefinição de Senha

- **Método HTTP:** PATCH
- **URL:** /auth/reset-password={token}
- **Descrição:** Redefine a senha do usuário utilizando o token recebido por email.

- **Parâmetros de Rota:**
 - `token` : String contendo o token de redefinição.
- **Parâmetros de Corpo (Request Body):**
 - `ResetPasswordDTO` : Objeto JSON contendo `newPassword` , `confirmPassword` .
- **Respostas:**
 - `200 OK` : Senha redefinida com sucesso. Retorna `LoginResponseDTO` com um novo token JWT.
 - `400 Bad Request` : Token inválido, token já utilizado ou senhas não coincidem. Retorna `LoginErroDTO` ou mensagem de erro.

Gerenciamento de Usuários (/users)

Endpoints para visualização e gerenciamento de contas de usuário no sistema.

Buscar Usuário por ID

- **Método HTTP:** `GET`
- **URL:** `/users/{id}`
- **Descrição:** Retorna os detalhes de um usuário específico com base no seu UUID.
- **Parâmetros de Rota:**
 - `id` : UUID do usuário.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna `UserForListDTO` com os dados do usuário.
 - `404 Not Found` : Usuário não encontrado.

Listar Todos os Usuários

- **Método HTTP:** `GET`
- **URL:** `/users`
- **Descrição:** Retorna uma lista com informações resumidas de todos os usuários cadastrados.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna uma lista de `UserForListDTO` .

Atualizar Usuário

- **Método HTTP:** `PUT`
- **URL:** `/users/{id}`

- **Descrição:** Atualiza os dados de um usuário existente.
- **Parâmetros de Rota:**
 - `id` : UUID do usuário a ser atualizado.
- **Parâmetros de Corpo (Request Body):**
 - `UserDTO` : Objeto JSON com os dados atualizados do usuário.
- **Autenticação:** Requer token JWT válido e permissões adequadas.
- **Respostas:**
 - `200 OK` : Usuário atualizado com sucesso. Retorna o `UserDTO` atualizado.
 - `404 Not Found` : Usuário não encontrado.
 - `400 Bad Request` : Dados inválidos.

Excluir Usuário

- **Método HTTP:** `DELETE`
- **URL:** `/users/{id}`
- **Descrição:** Remove um usuário do sistema.
- **Parâmetros de Rota:**
 - `id` : UUID do usuário a ser excluído.
- **Autenticação:** Requer token JWT válido e permissões adequadas.
- **Respostas:**
 - `204 No Content` : Usuário excluído com sucesso.
 - `404 Not Found` : Usuário não encontrado.

Gerenciamento de Alunos (`/alunos`)

Endpoints dedicados ao gerenciamento dos registros de alunos. Requerem autenticação e autorização (`USER` , `ADMIN` , `MODERATOR`).

Buscar Aluno por ID

- **Método HTTP:** `GET`
- **URL:** `/alunos/{id}`
- **Descrição:** Retorna os detalhes completos de um aluno específico.
- **Parâmetros de Rota:**
 - `id` : UUID do aluno.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna `AlunoDTO` com os dados do aluno.
 - `404 Not Found` : Aluno não encontrado.

Listar Todos os Alunos

- **Método HTTP:** GET
- **URL:** /alunos
- **Descrição:** Retorna uma lista com informações resumidas de todos os alunos.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Retorna uma lista de AlunoMinDTO .

Criar Aluno

- **Método HTTP:** POST
- **URL:** /alunos
- **Descrição:** Cria um novo registro de aluno.
- **Parâmetros de Corpo (Request Body):**
 - AlunoDTO : Objeto JSON com os dados do novo aluno.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Aluno criado com sucesso. Retorna o AlunoDTO criado.
 - 400 Bad Request : Dados inválidos.

Atualizar Aluno

- **Método HTTP:** PUT
- **URL:** /alunos/{id}
- **Descrição:** Atualiza os dados de um aluno existente.
- **Parâmetros de Rota:**
 - id : UUID do aluno a ser atualizado.
- **Parâmetros de Corpo (Request Body):**
 - AlunoDTO : Objeto JSON com os dados atualizados do aluno.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Aluno atualizado com sucesso. Retorna o AlunoDTO atualizado.
 - 404 Not Found : Aluno não encontrado.
 - 400 Bad Request : Dados inválidos.

Excluir Aluno

- **Método HTTP:** DELETE
- **URL:** /alunos/{id}
- **Descrição:** Remove um registro de aluno.

- **Parâmetros de Rota:**
 - `id` : UUID do aluno a ser excluído.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `204 No Content` : Aluno excluído com sucesso.
 - `404 Not Found` : Aluno não encontrado.

Importar Alunos via Planilha

- **Método HTTP:** `POST`
- **URL:** `/alunos/import`
- **Descrição:** Importa múltiplos alunos a partir de um arquivo Excel (.xlsx).
- **Parâmetros de Formulário (multipart/form-data):**
 - `file` : Arquivo .xlsx contendo os dados dos alunos.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna um inteiro indicando o número de alunos importados com sucesso.
 - `400 Bad Request` : Arquivo vazio (`-1`) ou tipo de arquivo inválido (`-2`).
 - `500 Internal Server Error` : Erro de leitura do arquivo (`-3`) ou outro erro interno (`-4`).

Gerenciamento de Orientadores (`/orientadores`)

Endpoints para gerenciar os registros de orientadores.

Buscar Orientador por ID

- **Método HTTP:** `GET`
- **URL:** `/orientadores/{id}`
- **Descrição:** Retorna os detalhes completos de um orientador específico.
- **Parâmetros de Rota:**
 - `id` : UUID do orientador.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna `OrientadorDTO` com os dados do orientador.
 - `404 Not Found` : Orientador não encontrado.

Listar Todos os Orientadores

- **Método HTTP:** `GET`

- **URL:** /orientadores
- **Descrição:** Retorna uma lista com informações resumidas de todos os orientadores.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Retorna uma lista de OrientadorMinDTO .

Criar Orientador

- **Método HTTP:** POST
- **URL:** /orientadores
- **Descrição:** Cria um novo registro de orientador.
- **Parâmetros de Corpo (Request Body):**
 - OrientadorDTO : Objeto JSON com os dados do novo orientador.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Orientador criado com sucesso. Retorna o OrientadorDTO criado.
 - 400 Bad Request : Dados inválidos.

Atualizar Orientador

- **Método HTTP:** PUT
- **URL:** /orientadores/{id}
- **Descrição:** Atualiza os dados de um orientador existente.
- **Parâmetros de Rota:**
 - id : UUID do orientador a ser atualizado.
- **Parâmetros de Corpo (Request Body):**
 - OrientadorDTO : Objeto JSON com os dados atualizados do orientador.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Orientador atualizado com sucesso. Retorna o OrientadorDTO atualizado.
 - 404 Not Found : Orientador não encontrado.
 - 400 Bad Request : Dados inválidos.

Excluir Orientador

- **Método HTTP:** DELETE
- **URL:** /orientadores/{id}
- **Descrição:** Remove um registro de orientador.

- **Parâmetros de Rota:**
 - `id` : UUID do orientador a ser excluído.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `204 No Content` : Orientador excluído com sucesso.
 - `404 Not Found` : Orientador não encontrado.

Gerenciamento de TCCs (`/tcc`)

Endpoints para gerenciar os Trabalhos de Conclusão de Curso.

Buscar TCC por ID

- **Método HTTP:** `GET`
- **URL:** `/tcc/{id}`
- **Descrição:** Retorna os detalhes completos de um TCC específico.
- **Parâmetros de Rota:**
 - `id` : UUID do TCC.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna `TCCDTO` com os dados do TCC.
 - `404 Not Found` : TCC não encontrado.

Listar Todos os TCCs

- **Método HTTP:** `GET`
- **URL:** `/tcc`
- **Descrição:** Retorna uma lista com informações resumidas de todos os TCCs.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna uma lista de `TCCMinDTO` .

Buscar Meu TCC

- **Método HTTP:** `GET`
- **URL:** `/tcc/my`
- **Descrição:** Retorna os detalhes do TCC pertencente ao usuário autenticado (presumivelmente um aluno).
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna `TCCDTO` com os dados do TCC do usuário.

- 404 Not Found : TCC não encontrado para o usuário.

Criar TCC

- **Método HTTP:** POST
- **URL:** /tcc
- **Descrição:** Cria um novo registro de TCC.
- **Parâmetros de Corpo (Request Body):**
 - TCCDTO : Objeto JSON com os dados do novo TCC.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : TCC criado com sucesso. Retorna o TCCDTO criado.
 - 400 Bad Request : Dados inválidos.

Atualizar TCC

- **Método HTTP:** PUT
- **URL:** /tcc/{id}
- **Descrição:** Atualiza os dados de um TCC existente.
- **Parâmetros de Rota:**
 - id : UUID do TCC a ser atualizado.
- **Parâmetros de Corpo (Request Body):**
 - TCCUpdateDTO : Objeto JSON com os dados a serem atualizados no TCC.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : TCC atualizado com sucesso. Retorna o TCCDTO atualizado.
 - 404 Not Found : TCC não encontrado.
 - 400 Bad Request : Dados inválidos.

Excluir TCC

- **Método HTTP:** DELETE
- **URL:** /tcc/{id}
- **Descrição:** Remove um registro de TCC.
- **Parâmetros de Rota:**
 - id : UUID do TCC a ser excluído.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 204 No Content : TCC excluído com sucesso.
 - 404 Not Found : TCC não encontrado.

Gerenciamento de Categorias (/categorias)

Endpoints para gerenciar as categorias de TCC.

Buscar Categoria por ID

- **Método HTTP:** GET
- **URL:** /categorias/{id}
- **Descrição:** Retorna os detalhes de uma categoria específica.
- **Parâmetros de Rota:**
 - id : UUID da categoria.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Retorna CategoriaDTO .
 - 404 Not Found : Categoria não encontrada.

Listar Todas as Categorias

- **Método HTTP:** GET
- **URL:** /categorias
- **Descrição:** Retorna uma lista de todas as categorias.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Retorna uma lista de CategoriaDTO .

Criar Categoria

- **Método HTTP:** POST
- **URL:** /categorias
- **Descrição:** Cria uma nova categoria.
- **Parâmetros de Corpo (Request Body):**
 - CategoriaDTO : Objeto JSON com os dados da nova categoria.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Categoria criada com sucesso. Retorna a CategoriaDTO criada.
 - 400 Bad Request : Dados inválidos.

Atualizar Categoria

- **Método HTTP:** PUT
- **URL:** /categorias/{id}

- **Descrição:** Atualiza os dados de uma categoria existente.
- **Parâmetros de Rota:**
 - `id` : UUID da categoria a ser atualizada.
- **Parâmetros de Corpo (Request Body):**
 - `CategoriaDTO` : Objeto JSON com os dados atualizados.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Categoria atualizada com sucesso. Retorna a `CategoriaDTO` atualizada.
 - `404 Not Found` : Categoria não encontrada.
 - `400 Bad Request` : Dados inválidos.

Excluir Categoria

- **Método HTTP:** DELETE
- **URL:** `/categorias/{id}`
- **Descrição:** Remove uma categoria.
- **Parâmetros de Rota:**
 - `id` : UUID da categoria a ser excluída.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `204 No Content` : Categoria excluída com sucesso.
 - `404 Not Found` : Categoria não encontrada.
 - `400 Bad Request` : Categoria possui subcategorias associadas.

Gerenciamento de Subcategorias (`/subcategorias`)

Endpoints para gerenciar as subcategorias de TCC.

Buscar Subcategoria por ID

- **Método HTTP:** GET
- **URL:** `/subcategorias/{id}`
- **Descrição:** Retorna os detalhes de uma subcategoria específica.
- **Parâmetros de Rota:**
 - `id` : UUID da subcategoria.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - `200 OK` : Retorna `SubcategoriaDTO` .
 - `404 Not Found` : Subcategoria não encontrada.

Listar Todas as Subcategorias

- **Método HTTP:** GET
- **URL:** /subcategorias
- **Descrição:** Retorna uma lista de todas as subcategorias. Pode ser filtrada por categoria.
- **Parâmetros de Query:**
 - `idCategoria` (UUID, opcional): Filtra as subcategorias pela categoria pai.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Retorna uma lista de `SubcategoriaDTO` .

Criar Subcategoria

- **Método HTTP:** POST
- **URL:** /subcategorias
- **Descrição:** Cria uma nova subcategoria.
- **Parâmetros de Corpo (Request Body):**
 - `SubcategoriaDTO` : Objeto JSON com os dados da nova subcategoria (incluindo o ID da categoria pai).
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Subcategoria criada com sucesso. Retorna a `SubcategoriaDTO` criada.
 - 400 Bad Request : Dados inválidos ou categoria pai não encontrada.

Atualizar Subcategoria

- **Método HTTP:** PUT
- **URL:** /subcategorias/{id}
- **Descrição:** Atualiza os dados de uma subcategoria existente.
- **Parâmetros de Rota:**
 - `id` : UUID da subcategoria a ser atualizada.
- **Parâmetros de Corpo (Request Body):**
 - `SubcategoriaDTO` : Objeto JSON com os dados atualizados.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Subcategoria atualizada com sucesso. Retorna a `SubcategoriaDTO` atualizada.
 - 404 Not Found : Subcategoria não encontrada.
 - 400 Bad Request : Dados inválidos.

Excluir Subcategoria

- **Método HTTP:** DELETE
- **URL:** /subcategorias/{id}
- **Descrição:** Remove uma subcategoria.
- **Parâmetros de Rota:**
 - id : UUID da subcategoria a ser excluída.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 204 No Content : Subcategoria excluída com sucesso.
 - 404 Not Found : Subcategoria não encontrada.
 - 400 Bad Request : Subcategoria possui TCCs associados.

Gerenciamento de Cursos (/cursos)

Endpoints relacionados aos cursos oferecidos.

Listar Todos os Cursos

- **Método HTTP:** GET
- **URL:** /cursos
- **Descrição:** Retorna uma lista de todos os cursos cadastrados.
- **Autenticação:** Requer token JWT válido.
- **Respostas:**
 - 200 OK : Retorna uma lista de CursoDTO .