**Documentação Sistema Gestão de Projetos**

**API de Gestão de Projetos**

A **API de Gestão de Projetos** é uma API Web construída em **ASP.NET WEB API** que segue o padrão **MVC** e oferece endpoints para acessar e manipular tabelas de **Projetos**, **Tarefas**, **Equipes e Membros de equipe** no banco de dados. Essa API é projetada para ser consumida por um front-end em **ASP.NET Web Forms**, permitindo a integração entre as interfaces do sistema de gestão de projetos e os dados.

**Back-end (ASP.NET Web API):**

A **API de Gestão de Projetos** é uma API RESTful implementada em C#, seguindo as melhores práticas de desenvolvimento de APIs RESTful, o que facilita a comunicação entre o front-end e o back-end. Esta API gerencia operações CRUD (Create, Read, Update, Delete) para diversos recursos essenciais à gestão de projetos.

Entre os recursos disponíveis, estão os projetos, que permitem a criação, leitura, atualização e exclusão, garantindo uma gestão eficiente de cada projeto. A API também manipula tarefas associadas a esses projetos, permitindo a atribuição de responsabilidades e o acompanhamento do progresso. Além disso, proporciona a gestão das equipes envolvidas nos projetos, facilitando a organização e a colaboração entre os membros.

Para garantir um acesso rápido e eficiente aos dados, a API utiliza ADO.NET, que estabelece a comunicação com o banco de dados SQL Server. Essa estrutura robusta assegura que as operações de manipulação de dados sejam realizadas de forma eficaz, oferecendo uma experiência fluida e responsiva aos usuários.

**Endpoints:**

Os principais endpoints da API incluem:

* **/api/v1/projeto**: Para operações relacionadas a projetos.
* **/api/v1/tarefa**: Para operações relacionadas a tarefas.
* **/api/v1/equipes**: Para operações relacionadas a equipes.
* **/api/v1/membroequipe**: Para operações relacionadas a equipes

.

### Recursos da API

#### **Projetos**

##### **-Listar Projetos**

Retorna uma lista de todos os projetos disponíveis.

* **Endpoint**: GET /api/v1/projeto
* **Parâmetros**: Nenhum
* **Resposta**:
  + **Status**: 200 OK
  + **Exemplo**:

{

"ProjetoId": 1,

"Nome": "Desenvolvimento de API",

"Descricao": "Criação de uma API para gestão de projetos.",

"DataInicio": "2024-09-01T00:00:00",

"DataFim": null,

"StatusProjeto": "Em andamento",

"EquipeId": 1

},

##### **-Buscar Projeto por ID**

Retorna os detalhes de um projeto específico, baseado no seu ID.

* **Endpoint**: GET /api/v1/projeto/{id}
* **Parâmetros**:
  + id (int): O ID do projeto que se deseja buscar.
* **Resposta**:
  + **Status**: 200 OK ou 404 Not Found
  + **Exemplo**:

"ProjetoId": 1,

"Nome": "Desenvolvimento de API",

"Descricao": "Criação de uma API para gestão de projetos.",

"DataInicio": "2024-09-01T00:00:00",

"DataFim": null,

"StatusProjeto": "Em andamento",

"EquipeId": 1

}

##### **-Buscar Projeto por Nome**

Retorna os detalhes de um projeto específico, baseado no seu nome.

* **Endpoint**: GET /api/v1/projeto/nome/{nome}
* **Parâmetros**:
  + nome (string): O nome do projeto que se deseja buscar.
* **Resposta**:
  + **Status**: 200 OK ou 404 Not Found
  + **Exemplo**:

{

"ProjetoId": 2,

"Nome": "Aplicativo Móvel",

"Descricao": "Desenvolvimento de um aplicativo móvel para Android.",

"DataInicio": "2024-09-15T00:00:00",

"DataFim": "2024-12-01T00:00:00",

"StatusProjeto": "Concluido",

"EquipeId": 2

}

##### **-Adicionar Projeto**

Cria um novo projeto com as informações fornecidas.

* **Endpoint**: POST /api/v1/projeto
* **Parâmetros**:
  + **Corpo**: Um objeto ProjetoModels
* **Resposta**:
  + **Status**: 200 OK
  + **Exemplo**:

{

"ProjetoId": 3,

"Nome": "Novo Projeto",

"Descricao": "Descrição do novo projeto.",

"DataInicio": "2024-10-01T00:00:00",

"DataFim": null,

"StatusProjeto": "Concluido",

"EquipeId": 3

}

##### **-Atualizar Projeto**

Atualiza as informações de um projeto existente.

* **Endpoint**: PUT /api/v1/projeto/{id}
* **Parâmetros**:
  + id (int): O ID do projeto que se deseja atualizar.
  + **Corpo**: Um objeto ProjetoModels
* **Resposta**:
  + **Status**: 200 OK
  + **Exemplo**:

{

"mensagem": "Projeto com ID 1 atualizado com sucesso."

}

##### **-Deletar Projeto**

Remove um projeto específico do banco de dados.

* **Endpoint**: DELETE /api/v1/projeto/{id}
* **Parâmetros**:
  + id (int): O ID do projeto que se deseja deletar.
* **Resposta**:
  + **Status**: 200 OK
  + **Exemplo**:

{

"mensagem": "Projeto com ID 1 deletado com sucesso."

}

#### **Equipes**

##### **Listar Equipes**

Retorna uma lista de todas as equipes disponíveis.

* **Endpoint:** GET /api/v1/equipe
* **Parâmetros:** Nenhum
* **Resposta:**
  + **Status:** 200 OK
* **Exemplo:**

{

"equipeId": 1,

"nome": "Desenvolvimento"

},

##### **Buscar Equipe por ID**

Retorna uma equipe específica com base no ID fornecido.

* **Endpoint:** GET /api/v1/equipe/{id}
* **Parâmetros:**
  + id (int): O ID da equipe que você deseja buscar.
* **Resposta:**
  + **Status:** 200 OK (se a equipe for encontrada)
  + **Status:** 404 Not Found (se a equipe não for encontrada)
* **Exemplo de resposta bem-sucedida:**

{

"equipeId": 1,

"nome": "Desenvolvimento"

}

* **Exemplo de resposta quando não encontrado:**

{

"message": "Equipe não encontrada."

}

##### **Buscar Equipe por Nome**

Retorna uma equipe que contém o nome fornecido.

* **Endpoint:** GET /api/v1/equipe/nome/{nome}
* **Parâmetros:**
  + nome (string): O nome da equipe que você deseja buscar.
* **Resposta:**
  + **Status:** 200 OK (se a equipe for encontrada)
  + **Status:** 404 Not Found (se a equipe não for encontrada)
* **Exemplo de resposta bem-sucedida:**

{

"equipeId": 1,

"nome": "Desenvolvimento"

}

* **Exemplo de resposta quando não encontrado:**

{

"message": "Equipe não encontrada."

}

##### **Adicionar Equipe**

Adiciona uma nova equipe ao sistema.

* **Endpoint:** POST /api/v1/equipe
* **Parâmetros:** Nenhum (os dados da equipe devem ser enviados no corpo da requisição).
* **Corpo da Requisição:**

{

"nome": "Recursos Humanos"

}

* **Resposta:**
  + **Status:** 201 Created (se a equipe for criada com sucesso)
  + **Status:** 400 Bad Request (se os dados fornecidos forem inválidos)
* **Exemplo de resposta bem-sucedida:**

{

"message": "Equipe adicionada com sucesso.",

"equipeId": 3

}

##### **Atualizar Equipe**

Atualiza os dados de uma equipe existente.

* **Endpoint:** PUT /api/v1/equipe/{id}
* **Parâmetros:**
  + id (int): O ID da equipe que você deseja atualizar.
* **Corpo da Requisição:**

{

"nome": "Desenvolvimento Avançado"

}

* **Resposta:**
  + **Status:** 200 OK (se a equipe for atualizada com sucesso)
  + **Status:** 404 Not Found (se a equipe não for encontrada)
  + **Status:** 400 Bad Request (se os dados fornecidos forem inválidos)
* **Exemplo de resposta bem-sucedida:**

{

"message": "Equipe atualizada com sucesso."

}

##### **Deletar Equipe**

Deleta uma equipe específica com base no ID fornecido.

* **Endpoint:** DELETE /api/v1/equipe/{id}
* **Parâmetros:**
  + id (int): O ID da equipe que você deseja deletar.
* **Resposta:**
  + **Status:** 200 OK (se a equipe for deletada com sucesso)
  + **Status:** 404 Not Found (se a equipe não for encontrada)
* **Exemplo de resposta bem-sucedida:**

{

"message": "Equipe deletada com sucesso."

}

#### **Membros da Equipe**

##### **Listar Todos os Membros**

Retorna uma lista de todos os membros da equipe disponíveis.

* **Endpoint:** GET /api/v1/membro
* **Parâmetros:** Nenhum
* **Resposta:**
  + **Status:** 200 OK
* **Exemplo:**

{

"MembroId": 1,

"Nome": "João Silva",

"Email": "joao@example.com",

"EquipeId": 1

},

##### **Buscar Membro por ID**

Retorna um membro específico com base no ID fornecido.

* **Endpoint:** GET /api/v1/membro/{id}
* **Parâmetros:**
  + id (int): O ID do membro que você deseja buscar.
* **Resposta:**
  + **Status:** 200 OK (se o membro for encontrado)
  + **Status:** 404 Not Found (se o membro não for encontrado)
* **Exemplo de resposta bem-sucedida:**

{

"MembroId": 1,

"Nome": "João Silva",

"Email": "joao@example.com",

"EquipeId": 1

}

* **Exemplo de resposta quando não encontrado:**

{

"message": "Membro não encontrado."

}

##### **Adicionar Membro**

Adiciona um novo membro à equipe.

* **Endpoint:** POST /api/v1/membro
* **Parâmetros:** Nenhum (os dados do membro devem ser enviados no corpo da requisição).
* **Corpo da Requisição:**

{

"Nome": "Carlos Pereira",

"Email": "carlos@example.com",

"EquipeId": 1

}

* **Resposta:**
  + **Status:** 201 Created (se o membro for criado com sucesso)
  + **Status:** 400 Bad Request (se os dados fornecidos forem inválidos)
* **Exemplo de resposta bem-sucedida:**

{

"message": "Membro adicionado com sucesso.",

"MembroId": 3

}

##### **Atualizar Membro**

Atualiza os dados de um membro existente.

* **Endpoint:** PUT /api/v1/membro/{id}
* **Parâmetros:**
  + id (int): O ID do membro que você deseja atualizar.
* **Corpo da Requisição:**

{

"Nome": "Carlos Alves",

"Email": "carlos.alves@example.com",

"EquipeId": 1

}

* **Resposta:**
  + **Status:** 200 OK (se o membro for atualizado com sucesso)
  + **Status:** 404 Not Found (se o membro não for encontrado)
  + **Status:** 400 Bad Request (se os dados fornecidos forem inválidos)
* **Exemplo de resposta bem-sucedida:**

{

"message": "Membro atualizado com sucesso."

}

##### **Deletar Membro**

Deleta um membro específico com base no ID fornecido.

* **Endpoint:** DELETE /api/v1/membro/{id}
* **Parâmetros:**
  + id (int): O ID do membro que você deseja deletar.
* **Resposta:**
  + **Status:** 200 OK (se o membro for deletado com sucesso)
  + **Status:** 404 Not Found (se o membro não for encontrado)
* **Exemplo de resposta bem-sucedida:**

{

"message": "Membro deletado com sucesso."

}

##### **Excluir Membros por Equipe**

Deleta todos os membros de uma equipe específica com base no ID da equipe.

* **Endpoint:** DELETE /api/v1/membro/equipe/{equipeId}
* **Parâmetros:**
  + equipeId (int): O ID da equipe cujos membros você deseja deletar.
* **Resposta:**
  + **Status:** 200 OK (se os membros forem deletados com sucesso)
* **Exemplo de resposta bem-sucedida:**

{

"message": "Todos os membros da equipe foram deletados com sucesso."

}

##### **Listar Membros por Equipe**

Retorna uma lista de todos os membros de uma equipe específica.

* **Endpoint:** GET /api/v1/membro/equipe/{equipeId}
* **Parâmetros:**
  + equipeId (int): O ID da equipe para buscar os membros.
* **Resposta:**
  + **Status:** 200 OK
* **Exemplo:**

{

"MembroId": 1,

"Nome": "João Silva",

"Email": "joao@example.com",

"EquipeId": 1

},

**Tarefas**

#### **Listar Todas as Tarefas**

* **Método:** GET
* **Endpoint:** /api/v1/tarefa
* **Parâmetros:** Nenhum
* **Descrição:** Retorna uma lista de todas as tarefas disponíveis.
* **Resposta:**
  + **Status:** 200 OK
  + **Exemplo de Resposta:**

{

"TarefaId": 1,

"Nome": "Desenvolver API",

"Descricao": "Criar uma API RESTful para o sistema de gestão de projetos.",

"DataCriacao": "2024-09-24T10:00:00",

"DataConclusao": null,

"Prioridade": "Alta",

"StatusTarefa": "Em Andamento",

"ProjetoId": 1,

"ResponsavelId": 1

},

#### **Buscar Tarefa por ID**

* **Método:** GET
* **Endpoint:** /api/v1/tarefa/{id}
* **Parâmetros:**
  + id (int) - O ID da tarefa que deseja buscar.
* **Descrição:** Retorna os detalhes da tarefa correspondente ao ID fornecido.
* **Resposta:**
  + **Status:** 200 OK
  + **Exemplo de Resposta:**

{

"TarefaId": 1,

"Nome": "Desenvolver API",

"Descricao": "Criar uma API RESTful para o sistema de gestão de projetos.",

"DataCriacao": "2024-09-24T10:00:00",

"DataConclusao": null,

"Prioridade": "Alta",

"StatusTarefa": "Em Andamento",

"ProjetoId": 1,

"ResponsavelId": 1

}

* + **Status:** 404 Not Found (se a tarefa não for encontrada)

#### **Adicionar Tarefa**

* **Método:** POST
* **Endpoint:** /api/v1/tarefa
* **Parâmetros:** Nenhum
* **Corpo da Requisição:**

{

"Nome": "Nome da Tarefa",

"Descricao": "Descrição da Tarefa",

"DataCriacao": "2024-09-24T10:00:00",

"DataConclusao": null,

"Prioridade": "Alta",

"StatusTarefa": "Pendente",

"ProjetoId": 1,

"ResponsavelId": 1

}

* **Descrição:** Adiciona uma nova tarefa.
* **Resposta:**
  + **Status:** 200 OK
  + **Exemplo de Resposta:**

{

"TarefaId": 3,

"Nome": "Nova Tarefa",

"Descricao": "Descrição da nova tarefa",

"DataCriacao": "2024-09-24T10:00:00",

"DataConclusao": null,

"Prioridade": "Baixa",

"StatusTarefa": "Pendente",

"ProjetoId": 2,

"ResponsavelId": null

}

#### **Atualizar Tarefa**

* **Método:** PUT
* **Endpoint:** /api/v1/tarefa/{id}
* **Parâmetros:**
  + id (int) - O ID da tarefa que deseja atualizar.
* **Corpo da Requisição:**

{

"TarefaId": 1,

"Nome": "Tarefa Atualizada",

"Descricao": "Descrição atualizada.",

"DataCriacao": "2024-09-24T10:00:00",

"DataConclusao": null,

"Prioridade": "Alta",

"StatusTarefa": "Concluída",

"ProjetoId": 1,

"ResponsavelId": 1

}

* **Descrição:** Atualiza os detalhes da tarefa correspondente ao ID fornecido.
* **Resposta:**
  + **Status:** 200 OK
  + **Exemplo de Resposta:**

json

Copiar código

"Tarefa atualizada com sucesso"

#### **Deletar Tarefa**

* **Método:** DELETE
* **Endpoint:** /api/v1/tarefa/{id}
* **Parâmetros:**
  + id (int) - O ID da tarefa que deseja deletar.
* **Descrição:** Deleta a tarefa correspondente ao ID fornecido.
* **Resposta:**
  + **Status:** 200 OK
  + **Exemplo de Resposta:**

"Tarefa deletada com sucesso"

* + **Status:** 400 Bad Request (se ocorrer um erro ao deletar)