# DESCRIÇÃO

A tabela "projetos" armazena projetos salvos pelos usuários, contendo os parâmetros de entrada, rota selecionada e resultados de cálculo de TPE.

Permite que usuários salvem configurações específicas para reutilização futura, facilitando recálculos e análises comparativas.

* Armazena configurações de cálculo (parâmetros + rota)
* Permite reutilização de cálculos anteriores
* Mantém histórico de projetos calculados
* Facilita análise e comparação de diferentes cenários
* Salva resultado completo do cálculo (roteiro calculado)
* Permite reprocessar roteiros de materiais já preenchidos

# OBJETIVO

* Permitir salvamento de projetos de cálculo
* Facilitar reutilização de configurações
* Manter histórico de cálculos realizados
* Suportar análise comparativa entre projetos
* Integrar com sistema de rotas e centros de produção

# LÓGICA TÉCNICA DE AQUISIÇÃO DE DADOS

**Fontes**

* Interface administrativa do widget (criação/edição de parâmetros);
* Chamadas de API diretamente (criação/edição de parâmetros).

**Script para criar tabela**

CREATE TABLE projetos (

id INT IDENTITY(1,1) PRIMARY KEY,

nome NVARCHAR(255) NOT NULL UNIQUE,

parametros NVARCHAR(MAX) NOT NULL,

roteiro NVARCHAR(MAX),

data\_criacao DATETIME2 DEFAULT GETDATE(),

data\_modificacao DATETIME2 DEFAULT GETDATE(),

centro NVARCHAR(100),

rota NVARCHAR(MAX),

idrota NVARCHAR(20)

);

-- Índices para performance

CREATE INDEX idx\_projetos\_nome ON projetos(nome);

CREATE INDEX idx\_projetos\_centro ON projetos(centro);

CREATE INDEX idx\_projetos\_idrota ON projetos(idrota);

CREATE INDEX idx\_projetos\_data\_criacao ON projetos(data\_criacao);

-- Constraint para validar JSON

ALTER TABLE projetos

ADD CONSTRAINT chk\_projetos\_parametros\_json

CHECK (parametros IS NULL OR ISJSON(parametros) = 1);

ALTER TABLE projetos

ADD CONSTRAINT chk\_projetos\_roteiro\_json

CHECK (roteiro IS NULL OR ISJSON(roteiro) = 1);

# FILTROS

Recomendações de filtros no consumo (API/relatórios/SQL):

* **Por nome**
* **Por centro de produção**
* **Por rota (nome ou ID)**
* **Por data de criação**
* **Por data de modificação**
* **Busca textual em nome**

# RECORRÊNCIA DE ATUALIZAÇÃO

* **Widget/API** → atualização **em tempo real** (transacional).

# CAMPO CHAVE PARA ATUALIZAÇÃO

**Chave primária**: id (INT IDENTITY)

**Chave natural (UNIQUE):** nome - garante que não existam projetos com nomes duplicados

**Estratégia de operações:**

**CREATE (POST)**

- Valida se nome já existe (UNIQUE constraint)

- Insere novo registro com ID auto-gerado

**UPDATE (PUT)**

- Identificação por id enviado na URL (/orbis/projetos/:id)

- Atualiza campos enviados no body

- Atualiza data\_modificacao = CURRENT\_TIMESTAMP

- Valida unicidade de nome se este campo for alterado

**DELETE (DELETE)**

- Quando usuário excluir via widget: DELETE FROM projetos WHERE id = @id

- Remove permanentemente o registro do banco

- Usado quando o projeto foi criada por engano ou não é mais necessária

**DESATIVAR (PATCH)**

- Quando usuário desativar via widget: UPDATE projetos SET ativo = 0 WHERE id = @id

- Mantém o registro no banco com ativo = 0

- Usado quando o projeto não deve mais aparecer, mas precisa ser mantida para histórico/auditoria

# API – DADOS DE ENTRADA

**POST /orbis/projetos - Criar novo projeto**

* **Descrição:** Salva um novo projeto com parâmetros e rota.
* **Body (JSON):**

{

"nome": "Chapa\_Dobrada\_500x300",

"parametros": {

"espessura\_chapa": "0.35",

"material": "ALUZINC",

"perimetro": "505",

"peso\_peca": "15",

"pokayoke": "NÃO",

"qtd\_aletas": "1",

"qtd\_dobras": "2",

"qtd\_recortes\_laserchapa": "1",

"tamanho\_peca": "240"

},

"centro": "MP10",

"rota": "LASER CHAPA, DESTAQUE, DOBRADEIRA",

"idrota": "2",

"roteiro": {

"etapas": [

{

"nome": "CORTE LASER",

"operacoes": [

{

"id": 12,

"descricao": "Nao cortar - peça sai na laser",

"tempo\_homem": 0.059,

"tempo\_maquina": 0.059,

"tempo\_preparacao": 1.05,

"unidade": "MIN"

}

]

},

{

"nome": "DESTAQUE",

"operacoes": [

{

"id": 13,

"descricao": "Destacar peça",

"tempo\_homem": 0.12,

"tempo\_maquina": 0,

"tempo\_preparacao": 0,

"unidade": "MIN"

}

]

}

],

"tempo\_total\_homem": 0.179,

"tempo\_total\_maquina": 0.059,

"tempo\_total\_preparacao": 1.05

}

}

* **Retorno 201:**

{

"success": true,

"data": {

"id": 35,

"nome": "Chapa\_Dobrada\_500x300",

"parametros": {

"espessura\_chapa": "0.35",

"material": "ALUZINC",

"perimetro": "505",

"peso\_peca": "15",

"pokayoke": "NÃO",

"qtd\_aletas": "1",

"qtd\_dobras": "2",

"qtd\_recortes\_laserchapa": "1",

"tamanho\_peca": "240"

},

"roteiro": {

"etapas": [

{

"nome": "CORTE LASER",

"operacoes": [

{

"id": 12,

"descricao": "Nao cortar - peça sai na laser",

"tempo\_homem": 0.059,

"tempo\_maquina": 0.059,

"tempo\_preparacao": 1.05,

"unidade": "MIN"

}

]

}

],

"tempo\_total\_homem": 0.179,

"tempo\_total\_maquina": 0.059,

"tempo\_total\_preparacao": 1.05

},

"data\_criacao": "2025-10-09T14:30:00Z",

"data\_modificacao": "2025-10-09T14:30:00Z",

"centro": "MP10",

"rota": "LASER CHAPA, DESTAQUE, DOBRADEIRA",

"idrota": "2"

}

}

**PUT /orbis/projetos/:id - Atualizar projeto existente**

* **Descrição:** Atualiza dados de um projeto existente identificado pelo ID
* **URL Parameter:** id (obrigatório)
* **Body (JSON):**

{

"nome": "Chapa\_Dobrada\_500x300\_v2",

"parametros": {

"espessura\_chapa": "0.50",

"material": "ALUZINC",

"perimetro": "505",

"peso\_peca": "18",

"pokayoke": "NÃO",

"qtd\_aletas": "1",

"qtd\_dobras": "3",

"qtd\_recortes\_laserchapa": "2",

"tamanho\_peca": "240"

}

}

* **Retorno 200:**

{

"success": true,

"data": {

"id": 35,

"nome": "Chapa\_Dobrada\_500x300\_v2",

"parametros": {

"espessura\_chapa": "0.50",

"material": "ALUZINC",

"perimetro": "505",

"peso\_peca": "18",

"pokayoke": "NÃO",

"qtd\_aletas": "1",

"qtd\_dobras": "3",

"qtd\_recortes\_laserchapa": "2",

"tamanho\_peca": "240"

},

"data\_criacao": "2025-10-09T14:30:00Z",

"data\_modificacao": "2025-10-09T16:15:00Z",

"centro": "MP10",

"rota": "LASER CHAPA, DESTAQUE, DOBRADEIRA",

"idrota": "2"

}

}

* **Retorno** **404**: {"success": false, "error": "Projeto não encontrado"}
* **Retorno** **409**: {"success": false, "error": "Já existe um projeto com este nome"}

**DELETE /orbis/projetos/:id - Excluir projeto**

* **Descrição:** DELETE físico - remove registro do banco definitivamente
* **URL Parameter:** id (obrigatório)
* **Retorno 200:**

{

"success": true,

"message": "Projeto excluído permanentemente"

}

* **Retorno 404:** {"success": false, "error": "Projeto não encontrado"}

# API – DADOS DE SAÍDA

**GET /orbis/projetos – Listar projeto**

* **Descrição**: Retorna lista de todos os projetos
* **Parâmetros**: filtros opcionais
* **Retorno**: coleção de projetos.

**GET /orbis/projetos/:id – Buscar projeto por ID**

* **Descrição**: Retorna um projeto específico pelo ID
* **Retorno** **200**: Objeto único do projeto
* **Retorno** **404**: {"success": false, "error": "Projeto não encontrado"}

**GET /orbis/projetos/nome/:nome – Buscar projeto por nome**

* **Descrição**: Retorna um projeto específico pelo nome
* **Retorno** **200**: Objeto único do projeto
* **Retorno** **404**: {"success": false, "error": "Projeto não encontrado "}

# CATEGORIZAÇÃO DO DHIVE

# ÁREAS IMPACTADAS PELA INFORMAÇÃO

Engenharia de Processos

# VALOR AGREGADO PARA A COMPANHIA

* **Automação**: Elimina necessidade de intervenção manual em ajustes pós-cálculo
* **Produtividade**: Reutilização de cálculos anteriores economiza tempo
* **Rastreabilidade**: Histórico completo de projetos calculados
* **Análise**: Facilita comparação entre diferentes cenários
* **Padronização**: Projetos salvos servem como referência
* **Integração**: Base para análises de tempos e custos
* **Migração** **para** **3DEXPERIENCE**: Histórico digital de cálculos realizados
* **Conhecimento**: Preserva configurações de produtos/processos

# INTEGRAÇÃO COM SISTEMA DE AUDITORIA

Todas as operações de CREATE, UPDATE, DELETE, ACTIVATE e DEACTIVATE nesta tabela DEVEM registrar logs na tabela "logs\_auditoria".

1. RECEBER user\_id no Body

Todas as requisições POST, PUT, PATCH e DELETE devem receber o campo user\_id:

Body exemplo:

{

"nome": "...",

"descricao": "...",

"user\_id": "eduardo.silva" ← OBRIGATÓRIO

}

2. CHAMAR Função de Log

Após cada operação bem-sucedida, chamar:

registrarLog(

tabela: 'nome\_da\_tabela',

registro\_id: id\_do\_registro,

operacao: 'CREATE' | 'UPDATE' | 'DELETE' | 'ACTIVATE' | 'DEACTIVATE',

valores\_anteriores: objeto\_com\_valores\_antigos (NULL para CREATE),

valores\_novos: objeto\_com\_valores\_novos (NULL para DELETE),

user\_id: user\_id\_recebido\_no\_body

);

3. EXEMPLOS POR OPERAÇÃO

**CREATE** (POST):

* Capturar user\_id do body
* Executar INSERT na tabela
* Chamar: registrarLog('tabela', novo\_id, 'CREATE', null, dados\_inseridos, user\_id)

**UPDATE** (PUT):

* Capturar user\_id do body
* Buscar valores anteriores: SELECT \* FROM tabela WHERE id = @id
* Executar UPDATE na tabela
* Buscar valores novos: SELECT \* FROM tabela WHERE id = @id
* Chamar: registrarLog('tabela', id, 'UPDATE', valores\_anteriores, valores\_novos, user\_id)

**DELETE** (DELETE):

* Capturar user\_id do body
* Buscar valores anteriores: SELECT \* FROM tabela WHERE id = @id
* Executar DELETE na tabela
* Chamar: registrarLog('tabela', id, 'DELETE', valores\_anteriores, null, user\_id)

**ACTIVATE** (PATCH /:id/ativar):

* Capturar user\_id do body
* Buscar valor anterior: SELECT ativo FROM tabela WHERE id = @id
* Executar: UPDATE tabela SET ativo = 1 WHERE id = @id
* Chamar: registrarLog('tabela', id, 'ACTIVATE', {ativo: 0}, {ativo: 1}, user\_id)

**DEACTIVATE (PATCH /:id/desativar):**

* Capturar user\_id do body
* Buscar valor anterior: SELECT ativo FROM tabela WHERE id = @id
* Executar: UPDATE tabela SET ativo = 0 WHERE id = @id
* Chamar: registrarLog('tabela', id, 'DEACTIVATE', {ativo: 1}, {ativo: 0}, user\_id)