

RELATÓRIO DE ATIVIDADES

CONSOLIDAÇÃO DE INFORMAÇÕES EM UM BANCO DE
DADOS NA ÁREA DE GESTÃO DE CONTRATOS VALE

PRO 752 - ENGENHARIA DA INFORMAÇÃO

DANILO LAFETÁ CABRAL - 23.1.1310

1. GitHub

Primeiro, foi criado um repositório no GitHub para organizar o trabalho e deixar tudo versionado (com histórico de alterações). O repositório foi clonado no computador e estruturado com pastas para separar os materiais.

The screenshot shows a GitHub repository named 'Trabalho-Final-Danilo-23.1.1310'. The repository has 1 branch and 0 tags. Recent commits include:

- CabralDL: Concertei o teste que estava errado (b154ad8 - 2 hours ago)
- Documento Análise e Modelos: Danilo Cabral (6 hours ago)
- sql: Concertei o teste que estava errado (2 hours ago)
- README.md: Update README.md (2 hours ago)

The README file contains the following text:

Trabalho-Final-Danilo-23.1.1310

Uma ferramenta para concentrar a visualização de ações/tarefas do time e também trazer insights sobre o desempenho da equipe

2. Análise

Contexto

Este trabalho é baseado em um projeto que eu já estou elaborando na **Vale**, durante meu estágio. A ideia surgiu por uma necessidade prática e pela alta vontade em automatizar este processo. Hoje existem **três sistemas diferentes** que registram ações/tarefas e informações relacionadas, e isso deixa a visão do todo bem fragmentada:

- **SAP (Ações SAP)**: registra ações operacionais com ID, título, descrição, prazo, status e executante.
- **Strawts**: também registra ações/tarefas, com campos parecidos e alguns dados adicionais (ex.: gestor, elaborador e gerência).
- **MoC (Management of Change)**: registra tarefas ligadas a mudanças/planos, com campos próprios como código do plano, título da mudança, status do plano e status do MoC.

Como cada sistema tem seu jeito de registrar e organizar, acompanhar tudo junto vira um desafio moroso. Existem também várias outras fontes de informação, mas para facilitar o desenvolvimento do trabalho vamos seguir apenas com esses três.

Problema

- As informações ficam **espalhadas** (um pedaço no SAP, outro no Strawts e outro no MoC).
- Não existe uma forma simples de ter uma **visão consolidada** para:
 - acompanhar tarefas por responsável;
 - comparar volume e prazos entre os sistemas;
 - identificar atrasos e prioridades.
- Alguns campos mudam bastante de um sistema para outro (principalmente no MoC), o que dificulta padronizar.

Objetivo

O objetivo é montar um **modelo de dados integrado** para:

1. carregar os dados das três fontes em um **banco de dados único**;
2. exibir tudo em um **painel no Power BI**, com filtros e indicadores de acompanhamento.

Escopo

- Integração das ações/tarefas dos sistemas **SAP, Strawts e MoC**.
- Criação de uma entidade central **Ação/Tarefa** para consolidar os campos comuns.
- Considerar apenas **um papel de pessoa** no modelo integrado: o **Colaborador responsável por executar a tarefa/ação**.
- Manter claro no modelo que são sistemas diferentes por meio de:
 - entidade **Fonte/Sistema**, e
 - **subtipos (ISA)** com atributos específicos de cada sistema.

Regras de negócio

1. Toda **Ação/Tarefa** vem de **exatamente uma Fonte/Sistema**.
2. Toda **Ação/Tarefa** é atribuída a **exatamente um Colaborador responsável**.
3. Uma **Fonte/Sistema** pode registrar **zero ou muitas** ações/tarefas.

4. Um **Colaborador** pode ser responsável por **zero ou muitas** ações/tarefas.
5. Existem atributos comuns (no supertipo) e atributos específicos por sistema (nos subtipos).
6. A especialização ISA é **Total e Disjunta**:
 - toda Ação/Tarefa pertence a um subtipo (SAP, Strawts ou MoC);
 - Uma Ação/Tarefa não pertence a mais de um subtipo ao mesmo tempo.

Requisitos de informação (campos)

Atributos comuns (Supertipo Ação/Tarefa)

- identificador
- titulo
- descricao
- prazo
- status

Atributos específicos (Subtipos)

- **Ação_SAP**: (conforme a fonte; se não houver campos exclusivos, apenas o identificador original do SAP)
- **Ação_Strawts**:
 - id_da_acao
 - elaborador
 - gestor
 - gerencia_do_elaborador
 - executante
- **Ação_MoC**:

- responsavel_do_moc
- gerencia_do_responsavel_da_tarefa
- descricao_do_plano
- unidade_organizadora
- prazo_da_tarefa
- status_do_plano
- descricao_da_tarefa
- status_moc
- responsavel_do_plano
- codigo_do_plano
- titulo_da_mudanca
- codigo_da_tarefa

Perguntas que o painel no Power BI deve responder

- Quantas ações/tarefas existem por **Fonte/Sistema**.
- Quantas ações estão com cada **Colaborador**.
- Quais ações estão **em atraso** (prazo < data atual) por responsável e por sistema.
- Distribuição de ações por **status** (geral e por sistema).
- Volume de ações por período (ex.: mês/semana).

Decisões de modelagem

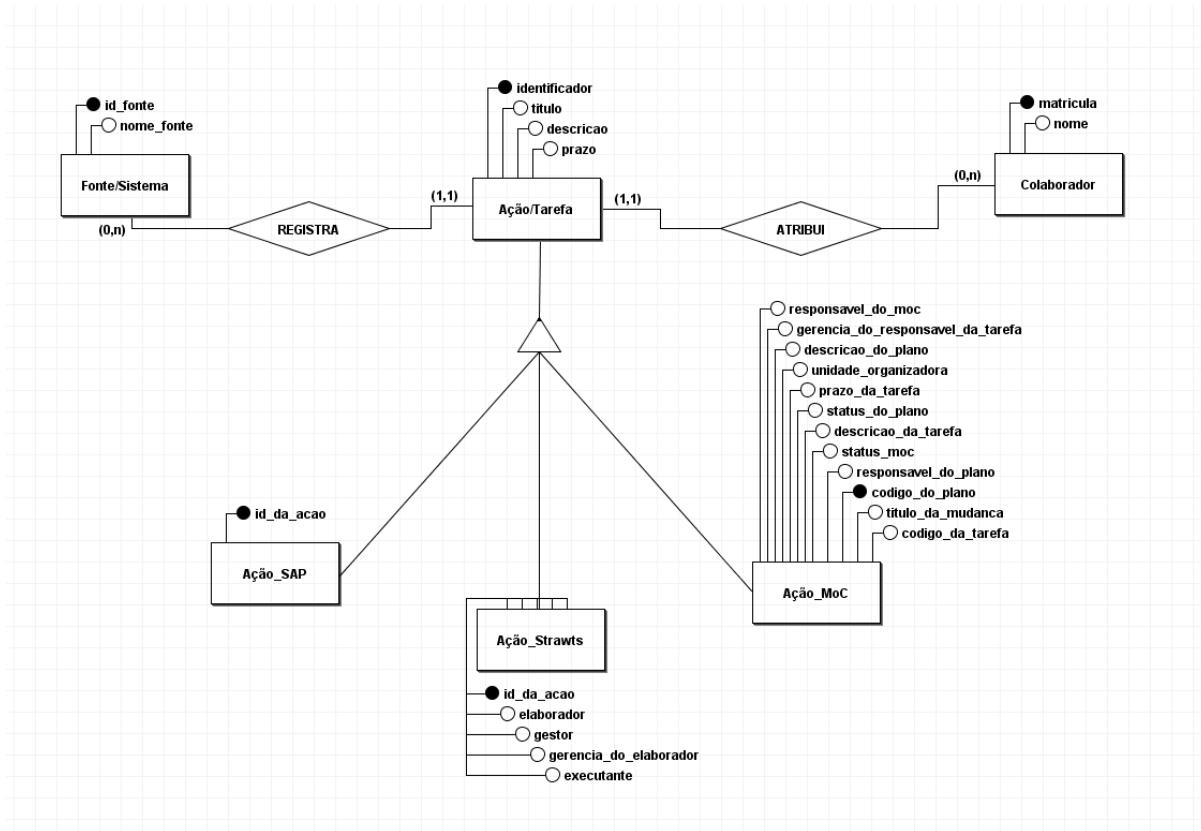
- O foco do modelo é **integração para BI**, então foi criada uma entidade central única (**Ação/Tarefa**) para facilitar consulta e painel.
- O MoC foi mantido com atributos mais “agrupados” (sem separar Plano/Mudança em entidades) para reduzir complexidade e manter o escopo do trabalho mais direto.
- A diferenciação entre sistemas fica explícita pela **Fonte/Sistema** e pelos **subtipos (ISA)**.

3. Modelo Conceitual

Depois da análise, foi feito o modelo conceitual no BRModelo. A base do modelo é a entidade central **Ação/Tarefa**, que representa o item principal que vai ser integrado e analisado no BI.

Para deixar claro que existem 3 sistemas diferentes, foi criada a entidade **Fonte/Sistema**, ligada à **Ação/Tarefa** com a regra de que cada ação vem de exatamente uma fonte, e uma fonte pode registrar várias ações.

Também foi criada a entidade **Colaborador**, porque no escopo do trabalho foi considerado apenas um papel de pessoa: o responsável por executar a tarefa/ação. A relação foi definida de forma que toda ação tem um responsável (obrigatório para a ação), mas um colaborador pode existir sem ter uma tarefa atribuída (opcional para o colaborador).

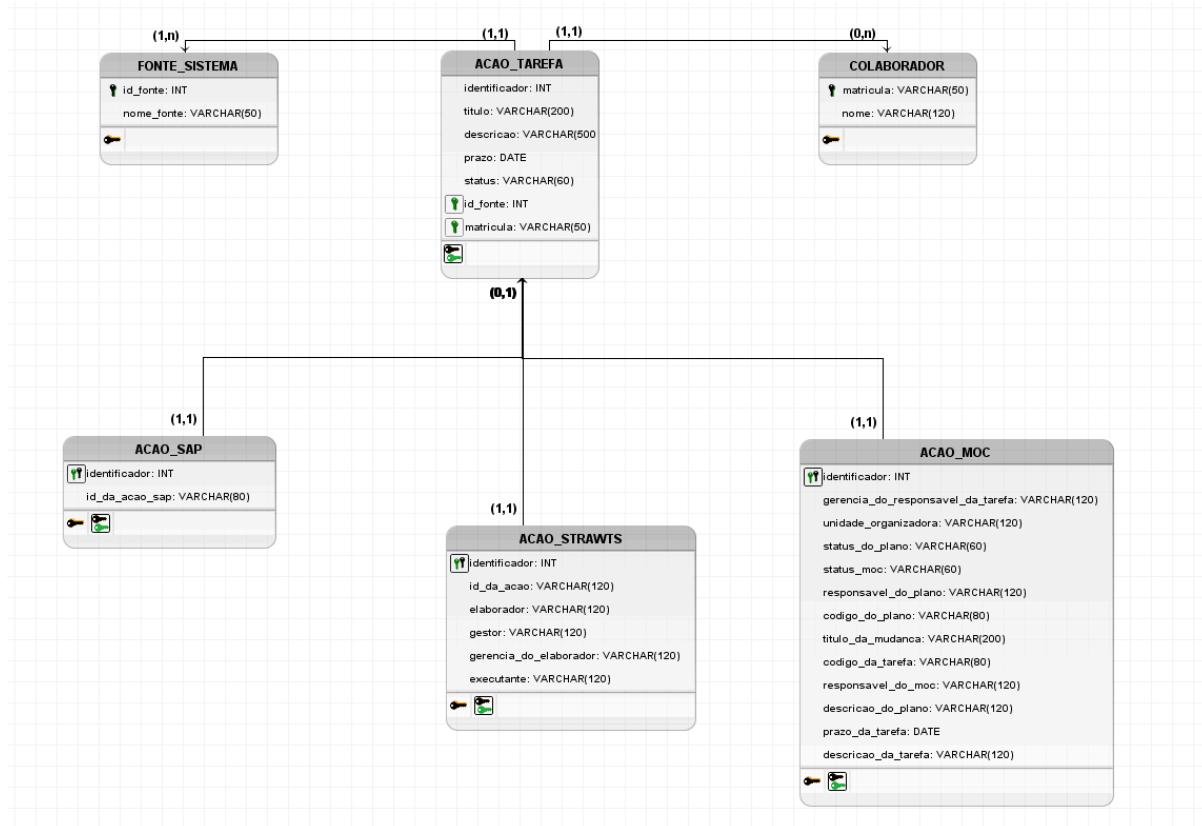


Para representar que cada sistema tem campos próprios, foi usada a ideia de especialização/generalização (ISA), com os subtipos:

- **Ação_SAP**
- **Ação_Strawts**
- **Ação_MoC**

Os atributos comuns ficaram no supertipo (Ação/Tarefa) e os atributos específicos ficaram nos subtipos, deixando explícito que SAP, Strawts e MoC têm diferenças.

4. Modelo Lógico



A partir do conceitual, foi montado o modelo lógico (relacional). Ou seja, o modelo virou tabelas com definição de chaves e relacionamentos.

As tabelas principais ficaram assim:

- **FONTE_SISTEMA** (cadastro das origens)
- **COLABORADOR** (responsáveis)
- **ACAO_TAREFA** (tabela central / supertipo)

A tabela **ACAO_TAREFA** ficou com os atributos comuns (identificador, título, descrição, prazo e status) e com as chaves estrangeiras:

- **id_fonte** → **FONTE_SISTEMA**
- **matricula** → **COLABORADOR**

Os subtipos do ISA foram implementados como tabelas 1:1 com o supertipo (o **identificador** do subtipo é ao mesmo tempo PK e FK para a **ACAO_TAREFA**), resultando nas tabelas:

- **ACAO_SAP**
- **ACAO_STRAWTS**
- **ACAO_MOC**

Cada subtipo guarda apenas os campos específicos daquele sistema.

5. Modelo Físico

Por fim, foi produzido o modelo físico em SQL para PostgreSQL. Esse script cria as tabelas definidas no lógico e inclui as chaves primárias e estrangeiras. O modelo físico contempla:

- tabelas base (**fonte_sistema**, **colaborador**, **acao_tarefa**)
- tabelas de subtipos (**acao_sap**, **acao_strawts**, **acao_moc**)
- constraints de PK/FK para manter integridade
- alguns índices simples (por fonte, responsável e prazo) para facilitar consulta

-- Trabalho Eng. da Informação

-- Modelo físico (PostgreSQL) - integração SAP / Strawts / MoC

-- Observação: regras do ISA (total/disjunto) ficam garantidas na carga/ETL.

```
CREATE TABLE fonte_sistema (
    id_fonte SERIAL PRIMARY KEY,
    nome_fonte VARCHAR(51) NOT NULL
);
```

```
CREATE TABLE colaborador (
    matricula VARCHAR(50) PRIMARY KEY,
    nome     VARCHAR(120) NOT NULL
);
```

-- Supertipo (tabela principal)

```
CREATE TABLE acao_tarefa (
    identificador SERIAL PRIMARY KEY,
    titulo      VARCHAR(200),
    descricao   TEXT,
    prazo       DATE,
    status      VARCHAR(60),
```

-- origem do registro

```
    id_fonte    INT NOT NULL,
    -- responsável
    matricula   VARCHAR(50) NOT NULL,
```

```
CONSTRAINT fk_acao_fonte
```

```
FOREIGN KEY (id_fonte) REFERENCES fonte_sistema(id_fonte),
```

```
CONSTRAINT fk_acao_colaborador
```

```
FOREIGN KEY (matricula) REFERENCES colaborador(matricula)
```

```
);
```

-- Subtipo SAP (1:1 com acao_tarefa)

```
CREATE TABLE acao_sap (
    identificador INT PRIMARY KEY,
    id_da_acao_sap VARCHAR(80),
```

```
CONSTRAINT fk_sap_acao
```

```
FOREIGN KEY (identificador) REFERENCES acao_tarefa(identificador)
```

);

-- Subtipo Strawts (1:1 com acao_tarefa)

CREATE TABLE acao_strawts (

identificador INT PRIMARY KEY,

id_da_acao VARCHAR(80),

elaborador VARCHAR(120),

gestor VARCHAR(120),

gerencia_do_elaborador VARCHAR(120),

executante VARCHAR(120),

CONSTRAINT fk_strawts_acao

FOREIGN KEY (identificador) REFERENCES acao_tarefa(identificador)

);

-- Subtipo MoC (1:1 com acao_tarefa)

CREATE TABLE acao_moc (

identificador INT PRIMARY KEY,

gerencia_do_responsavel_da_tarefa VARCHAR(120),

unidade_organizadora VARCHAR(120),

status_do_plano VARCHAR(60),

status_moc VARCHAR(60),

responsavel_do_plano VARCHAR(120),

codigo_do_plano VARCHAR(80),

titulo_da_mudanca VARCHAR(200),

```
codigo_da_tarefa VARCHAR(80),  
responsavel_do_moc VARCHAR(120),  
  
descricao_do_plano TEXT,  
prazo_da_tarefa DATE,  
descricao_da_tarefa TEXT,  
  
CONSTRAINT fk_moc_acao  
FOREIGN KEY (identificador) REFERENCES acao_tarefa(identificador)  
);  
  
-- índices simples pra ajudar consulta (opcional)  
CREATE INDEX idx_acao_id_fonte ON acao_tarefa(id_fonte);  
CREATE INDEX idx_acao_matricula ON acao_tarefa(matricula);  
CREATE INDEX idx_acao_prazo ON acao_tarefa(prazo);
```

Esse script foi preparado para ser colocado no GitHub dentro da pasta [/sql1/](#), deixando a entrega reproduzível e fácil de validar.