

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

**Relatório Técnico de Fundamentos de Banco de Dados:
Implementação de um Inventário do Parque Computacional para o IFAP**

Discentes: Enzo Hahn Veroneze, Rafaela da Rosa Soares
Santa Maria, 11 de dezembro de 2024.

INTRODUÇÃO

Apresentação de um relatório técnico sobre o desenvolvimento do trabalho final da disciplina de Fundamentos de Banco de Dados, abordando as experiências vivenciadas ao utilizar uma das ferramentas sugeridas para a implementação de um banco de dados.

DESCRIÇÃO DO TRABALHO

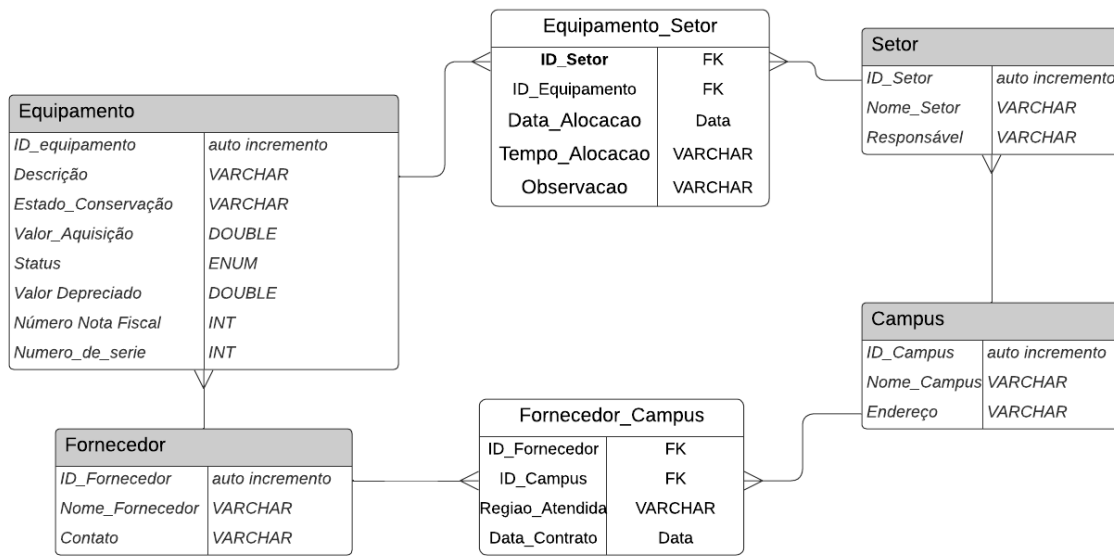
O estudo de caso selecionado para realizar o trabalho da disciplina foi o seguinte:

“Parque Computacional - Instituto Federal do Amapá (IFAP)

Dados sobre os equipamentos de TI separados por setor, campus e delimitados por seu fornecedor. Os dados foram disponibilizados em csv's e se tratava de um conjunto de dados, com 4 csv's, todos com o seguinte cabeçalho: NUMERO, STATUS, DESCRICAO, SETOR DO RESPONSÁVEL, CAMPUS DA CARGA. VALOR AQUISIÇÃO, VALOR DEPRECIADO, NÚMERO NOTA FISCAL, NÚMERO DE SÉRIE. FORNECEDOR, SALA, ESTADO DE CONSERVAÇÃO”.

DIAGRAMA DE CLASSES

A partir do estudo de caso, pudemos perceber quais seriam as entidades, os relacionamentos e os atributos de cada um. O diagrama foi feito mais de uma vez, e adaptado para possuir os requisitos do trabalho (como possuir relacionamentos n:n, triggers e outros). O diagrama de classes disponibilizado na imagem a seguir foi realizado no LucidChart.



Fonte: Autores.

METODOLOGIA

O trabalho foi implementado utilizando o banco de dados MariaDb. Dada a escolha do banco de dados, as tabelas foram criadas no seguinte padrão:

```
CREATE TABLE `campus` (  
  `ID_Campus` INT(11) NOT NULL AUTO_INCREMENT,  
  `Nome_Campus` VARCHAR(50) NOT NULL DEFAULT '' COLLATE 'utf8mb4_general_ci',  
  `Endereco` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_general_ci',  
  PRIMARY KEY (`ID_Campus`) USING BTREE  
)
```

Views para consultas específicas foram criadas para facilitar a visualização de dados que relacionam-se entre si.

View Equipamento por Setor:

```
SELECT  
  es.ID_Setor,  
  s.Nome_Setor,  
  e.ID_Equipamento,  
  e.Descricao,  
  e.Status,  
  e.Estado_Conservacao,  
  es.Data_Alocacao,  
  es.Tempo_Alocacao  
FROM equipamento_setor es  
JOIN setor s ON es.ID_Setor = s.ID_Setor  
JOIN equipamento e ON es.ID_Equipamento = e.ID_Equipamento
```

View equipamento por status:

```
1 Select  
2 Status,  
3 COUNT(*) AS Total_Equipamentos  
4 FROM equipamento  
5 GROUP BY Status |
```

View Equipamento por Estado de Conservação:

```

1  SELECT
2  e.Estado_Conservacao,
3  COUNT(*) AS Total_Equipamentos
4  FROM equipamento e
5  GROUP BY e.Estado_Conservacao

```

View Status por Campus:

```

1  SELECT
2  c.Nome_Campus,
3  esc.Sala,
4  s.Nome_Setor,
5  s.Responsavel
6  FROM edificio_setor_campus esc
7  JOIN campus c ON esc.ID_Campus = c.ID_Campus
8  JOIN setor s ON esc.ID_Setor = s.ID_Setor

```

A utilização de triggers ocorreu para identificar as seguintes situações:

1 - Para interagir com a tabela AUDITORIA, criou-se o trigger “auditar_status_equipamento” que é responsável por interceptar a mudança do status do equipamento. Quando ele fica Ativo ou Inativo, ele é adicionado à auditoria, para atestar mudança.

```

1  BEGIN
2  IF OLD.Status <> NEW.Status THEN
3  INSERT INTO auditoria_equipamento (ID_Equipamento, Status_Antigo, Status_Novo)
4  VALUES (OLD.ID_Equipamento, OLD.Status, NEW.Status);
5  END IF;
6  END

```

2 - Para interagir com a quantidade de equipamentos adicionados em um setor. Quando um equipamento é adicionado ao setor, ele incrementa na tabela “equipamento_setor”.

Execução do trigger: (Ex. "SET NEW.colunaA = TRIM(OLD.colunaA)")

```
1 BEGIN
2     UPDATE setor
3     SET Total_Equipamentos = Total_Equipamentos - 1
4     WHERE ID_Setor = OLD.ID_Setor;
5 END
```

3 - Quando um equipamento é alocado a um setor e o usuário tenta excluí-lo, ele dá um aviso.

Execução do trigger: (Ex. "SET NEW.colunaA = TRIM(OLD.colunaA)")

```
1 BEGIN
2     IF EXISTS (
3         SELECT 1 FROM equipamento_setor WHERE ID_Equipamento = OLD.ID_Equipamento
4     ) THEN
5         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Não é possível excluir um equipamento que está alocado a um setor.';
6     END IF;
7 END
```

Referente ao programa criado com a interface, a maior dificuldade foi fazer as edições de tabelas. As interações, como adição e remoção, foram simples e intuitivas. Uma das interações planejadas era que fosse possível visualizar os setores por edifícios, mas devido a complexidade alcançada nos relacionamentos ao criar essa entidade, ela foi colocada em segunda prioridade, para ser adicionada após as interações principais.

Inventário IF

Inventário do IF

Adicionar Equipamento Visualizar Equipamentos Adicionar Setor Visualizar Setores Visualizar Auditoria

Adicionar Campus Visualizar Campus Adicionar Fornecedor Visualizar Fornecedor

Adicionar Equipamento:

Descrição:

Status:

Estado de Conservação:

Setores:

Valor de Aquisição (R\$):

Valor Depreciado (R\$):

Número da Nota Fiscal:

Número de Série:

Registrar

Facilidades e dificuldades encontradas na realização do trabalho

Inicialmente, antes de iniciar as interações através da interface, a dificuldade principal estava relacionada a integrar o banco de dados às informações inseridas pelo usuário. Quando as relações ficavam co-dependentes ao extremo (uma relação como pai-filho-neto-bisneto), havia dificuldade em editar tabelas com essas necessidades.

A criação da interface foi feita com o Tkinter, utilizando Python para manipulação, pois, a partir de pesquisas feitas para idealizar a melhor ferramenta para o trabalho, constatou-se que haveriam muitos tutoriais e materiais disponíveis para criação. Além disso, a utilização do Python facilitou a manipulação do banco de dados, pois conseguimos visualizar de uma maneira mais intuitiva.

Para a realização do trabalho, a parte mais demorada foi a interação com o HeidiSQL, a interface utilizada para o banco de dados Mariadb, uma vez que se tratava de uma interação com uma aplicação nova e haviam múltiplas possibilidades para realizar cada interação com o banco de dados criado pelos integrantes. Mas, apesar do tempo gasto com a criação de tabelas, triggers, procedures e views, a interação não foi complicada e existiram poucas dificuldades com o SQL em si.