## **Projeto Final de Banco de Dados**

25 de Novembro de 2020

Letícia Dias Soares - 180022059

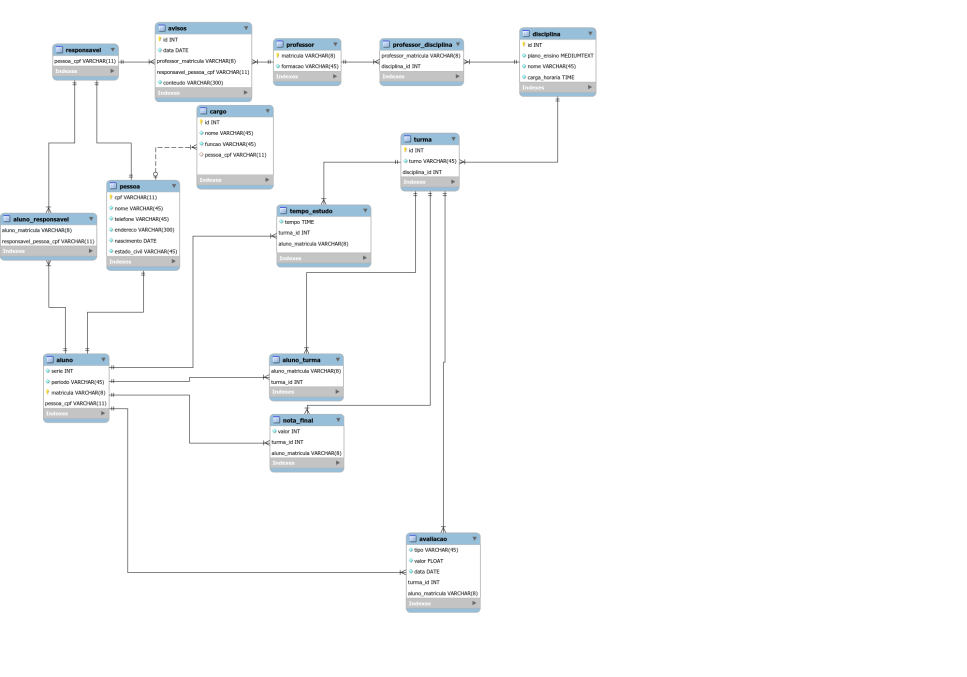
Marcos William - 160135931

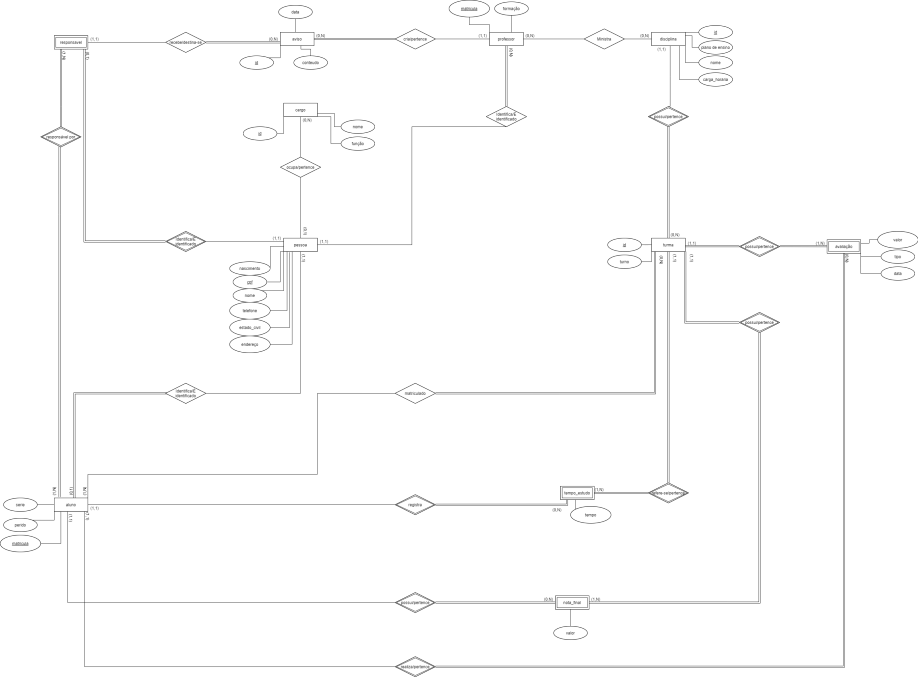
**1. Introdução:**

Nosso projeto foi realizado com o intuito de desenvolver um software para o sistema educacional onde o próprio sistema faça o monitoramento dos alunos e envie um relatório para os professores, e caso necessário, os professores enviaram uma notificação para os pais responsáveis dos alunos que não estão cumprindo as horas mínimas de estudo que são pré definidas pela escola. Foram usadas as linguagens de programação PYTHON e um banco de dados com o SGBD MySQL, não foi usada nenhuma API, foram utilizados apenas os módulos padrões do PYTHON.

**2. Modelos entidade relacionamento e Modelo Relacional :**

Foram feitos diagramas para organizar o desenvolvimento do projeto, fizemos os modelos entidade relacionamento (MER) e o modelo relacional (MR), confira os nossos modelos a seguir:

  
Figura 1: Modelo Relacional (MR)

Figura 2: Modelo Entidade Relacionamento (MER)

O modelo entidade relacionamento - MER (Figura 2), foi usado como ideia inicial do projeto, e conforme fomos pensando mais sobre o projeto esse modelo sofreu várias alterações. Já o modelo relacional (Figura 1), foi feito para deixar mais claro os atributos de cada entidade e podermos aprimorar mais nossa ideia.

**3. Álgebra Relacional:**

Para testar o funcionamento do banco de dados criado, as seguintes pesquisas foram executadas no banco de dados criado. Abaixo segue uma descrição de cada uma das pesquisas realizadas, bem como o seu correspondente em álgebra relacional.

select professor.matricula, pessoa.nome, cargo.nome as "nome cargo" from professor,pessoa,cargo  
WHERE professor.pessoa\_cpf=pessoa.cpf AND cargo.pessoa\_cpf = pessoa.cpf;

π professor.matricula, pessoa.nome, ρnome cargo(cargo.nome)(σ professor.pessoa=pessoa.cpf ∧ cargo.pessoa\_cpf = pessoa.cpf(professor x pessoa x cargo))

Essa consulta envolve as tabelas pessoa, professor e cargo. O objetivo da consulta é mostrar uma tabela que indique a relação de cargo de cada professor, destacando-se matricula, nome do professor e nome do cargo.

select responsavel.pessoa\_cpf as "CPF", pessoa.nome, COUNT(avisos.id) as "total" from responsavel, pessoa, avisos  
WHERE responsavel.pessoa\_cpf = pessoa.cpf AND pessoa.cpf = avisos.responsavel\_pessoa\_cpf;

π ρCPF(responsavel.pessoa),pessoa.nome, ρtotal(Γcount(id)(avisos))(σ responsavel.pessoa\_cpf =pessoa.cpf ∧ avisos.responsavel\_pessoa\_cpf = pessoa.cpf(responsavel x pessoa x avisos))

A consulta acima envolve as tabelas avisos, responsavel, pessoa. O objetivo é mostrar a quantidade de avisos que cada responsável cadastrado recebeu, destacando-se o CPF e nome do responsável e indicando a quantidade de avisos.

select aluno.matricula, pessoa.nome, COUNT(aluno\_turma.aluno\_matricula) as "Quantidade de turmas" from aluno, pessoa, aluno\_turma  
WHERE aluno.pessoa\_cpf = pessoa.cpf AND aluno\_turma.aluno\_matricula = aluno.matricula;

π aluno.matricula, pessoa.nome, ρQuantidade de turmas(Γcount(aluno\_matricula)(aluno\_turma))(σ aluno.pessoa\_cpf = pessoa.cpf ∧ aluno\_turma.aluno\_matricula = aluno.matricula (aluno x pessoa x aluno\_turma))

Esta consulta apresenta como resultado uma tabela contendo a matricula do aluno, nome e a quantidade de turmas em que o aluno está matriculado. Utiliza as tabelas aluno, aluno\_turma e pessoa.

select professor.pessoa\_cpf as "CPF", pessoa.nome, professor\_disciplina.disciplina\_id as "ID disciplina", disciplina.nome as "Nome disciplina" from professor, pessoa, professor\_disciplina, disciplina  
WHERE pessoa.cpf = professor.pessoa\_cpf AND disciplina.id = professor\_disciplina.disciplina\_id AND professor.matricula = professor\_disciplina.professor\_matricula;

π ρCPF(professor.pessoa\_cpf), pessoa.nome, ρID disciplina(professor\_disciplina.disciplina\_id), ρNome disciplina(disciplina.nome)(σ professor.pessoa\_cpf = pessoa.cpf ∧ disciplina.id = professor\_disciplina.disciplina\_id ∧ professor.matricula = professor\_disciplina.professor\_matricula (professor x pessoa x professor\_disciplina x disciplina))

O objetivo desta consulta é apresentar a relação entre professores e disciplinas, utilizando o CPF e nome do professor e ID e nome da disciplina. Utiliza as tabelas professor, pessoa, disciplina, professor\_disciplina.

select distintict aluno.matricula, pessoa.nome, avaliacao.tipo as "Tipo de avaliação" from aluno, pessoa, avaliacao  
WHERE pessoa.cpf = aluno.pessoa\_cpf AND avaliacao.aluno\_matricula = aluno.matricula;

π aluno.matricula, pessoa.nome, ρTipo de avaliação(avaliacao.tipo)(σ aluno.pessoa\_cpf = pessoa.cpf ∧ avaliacao.aluno\_matricula = aluno.matricula(aluno x pessoa x avaliacao))

Esta consulta lista todos os tipos de avaliação que o aluno realizou. Utiliza as tabelas aluno, pessoa e avaliacao. Destaca a matricula e nome do aluno e o tipo da avaliação.

**4. Forma Normal:**

Foi analisada a possível aplicação de até 3 formas normais em 5 tabelas. As tabelas foram: aluno, professor, cargo, disciplina e avisos. As análises foram feitas após a mudança final de estrutura do MER e consequentemente do MR. Ao analisar essas 5 tabelas foi constado que todas já possuíam as características das 3 formas normais, ou seja, para todas as tabelas analisadas: todos os atributos eram atômicos, todos os atributos não chaves eram dependentes unicamente da chave primaria e todos os atributos eram funcionalmente independentes.

**5 - Considerações finais:**

Todo o projeto desenvolvido você pode encontrar no GitHub: <https://github.com/LexRed/BancoDeDados> . Para execução do projeto basta apenas ter o PYTHON instalado na sua máquina e digitar no terminal o comando:

