



CENTRO UNIVERSITÁRIO UNIFECAF
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GABRIEL MASCARENHAS DE LIMA

PROJETO : CRIANDO UMA BASE DE DADOS PARA QUE ATENDA UMA
UNIVERSIDADE

GABRIEL MASCARENHAS DE LIMA

TÍTULO: PROJETO

Subtítulo: criando uma base de dados para que atenda uma universidade.

Trabalho apresentado como requisito parcial de avaliação da disciplina Database Modeling & SQL do Curso de Graduação em Análise e Desenvolvimento de Sistemas do Centro Universitário UniFECAF. Tutor(a): Vitor Jansen

TABOÃO DA SERRA, SP
2024

SUMÁRIO

1 INTRODUÇÃO	4
2 MODELOS	5
2.1 Modelo MER	5
2.2 Modelo DER	6
2.3 Modelo físico	7
4 CONCLUSÃO	12

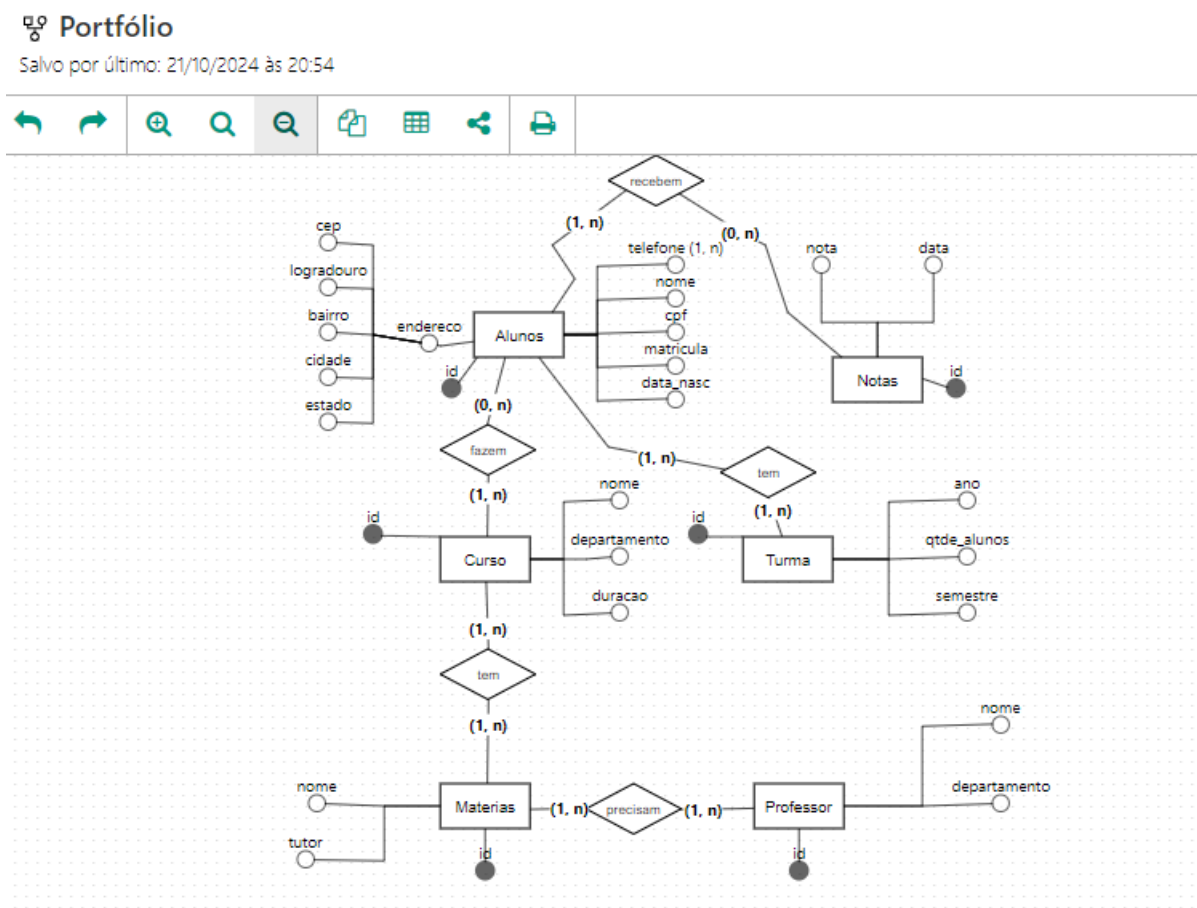
1 INTRODUÇÃO

Os proprietários de uma faculdade precisam de um sistema que viabilize o armazenamento de informações sobre seus alunos, cursos, matérias e professores para que seja possível realizar controles básicos como montar turmas e realizar o armazenamento de notas dos alunos.

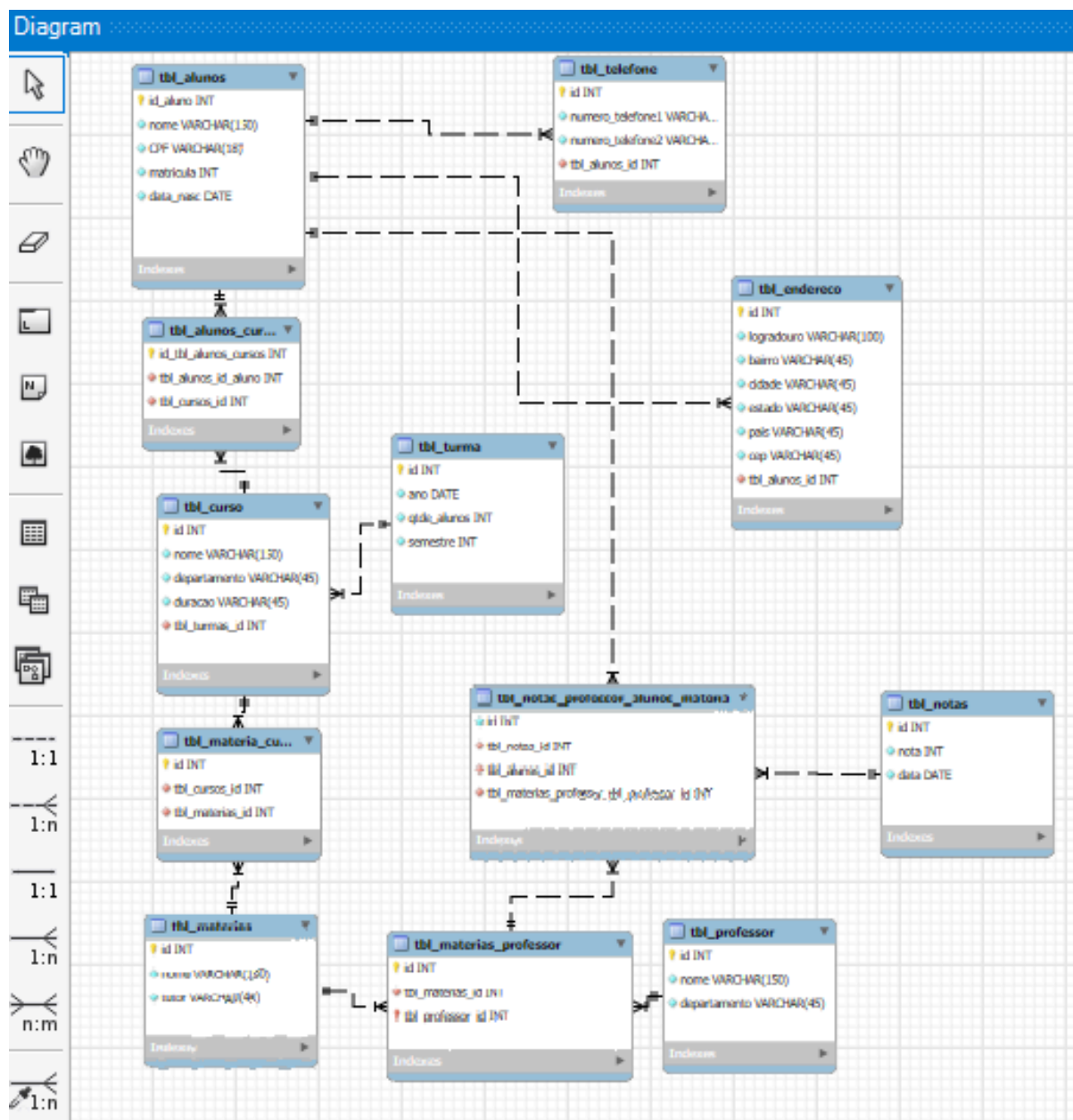
Com base no que foi apresentado acima, o aluno deve criar um banco de dados que ofereça suporte para que um sistema possa armazenar informações que atendam a necessidade do cliente.

2 MODELOS

2.1 modelagem conceitual (MER)



2.2 Modelagem lógica (DER)



2.3 Modelo físico

```
'create database db_portfolio;  
use db_portfolio;
```

```
create table tbl_alunos(  
  
id_alunos int not null primary key auto_increment,  
nome varchar(150) not null,  
cpf VARCHAR(18) not null,  
matricula int not null,  
data_nasc date not null,  
unique index (id_alunos)  
);
```

```
create table tbl_turma(  
  
id_turma int not null primary key auto_increment,  
    nome VARCHAR(10),  
    ano date not null,  
    qtde_alunos int not null,  
    semestre int not null,  
    unique index (id_turma)  
);
```

```
create table tbl_materias(  
id_materias int not null primary key auto_increment,  
    nome VARCHAR(150) not null,  
    tutor VARCHAR(150) not null,  
    unique index (id_materias)  
);
```

```
create table tbl_professor(  
id_professor int not null primary key auto_increment,  
    nome VARCHAR(150) not null,  
    departamento VARCHAR(150) not null,  
    unique index (id_professor)  
);
```

```
create table tbl_notas(  
id_notas int not null primary key auto_increment,  
    nota decimal(10,2),  
    data_prova date,  
    unique index (id_notas)  
);
```

```
create table tbl_telefone(  
id_telefone int not null primary key auto_increment,  
    numero_telefone1 VARCHAR(15),  
    numero_telefone2 VARCHAR(15),  
    id_alunos int not null,  
constraint FK_ALUNO_TELEFONE  
foreign key (id_alunos)  
references tbl_alunos (id_alunos),  
    unique index (id_telefone)  
);
```

```
create table tbl_endereco(  
id_endereco int not null primary key auto_increment,  
    logradouro VARCHAR(150),  
    bairro VARCHAR (45),  
    cidade VARCHAR (45),  
    estado VARCHAR (45),
```

TABOÃO DA SERRA, SP
2024


```
pais VARCHAR (45),
id_alunos int not null,
    constraint FK_ALUNO_ENDERECO
    foreign key (id_alunos)
    references tbl_alunos (id_alunos),
    unique index (id_endereco)
);
```

```
create table tbl_curso(
id_curso int not null primary key auto_increment,
    nome VARCHAR (150) not null,
    departamento VARCHAR (150) not null,
    duracao VARCHAR (45) not null,
    id_turma int not null,
constraint FK_CURSO_TURMAS
foreign key (id_turma)
references tbl_turma (id_turma),
unique index (id_curso)

);
```

```
create table tbl_alunos_curso(
id_alunos_curso int not null primary key auto_increment,
    id_alunos int not null,
    id_curso int not null,
constraint FK_ALUNOS_ALUNOS_CURSO
    foreign key (id_alunos)
    references tbl_alunos (id_alunos),
```

```
constraint FK_CURSO_ALUNOS_CURSO
    foreign key (id_curso)
    references tbl_curso (id_curso),
unique index (id_alunos_curso)
);
```

```
create table tbl_materias_curso(
id_materias_curso int not null primary key auto_increment,
id_curso int not null,
    id_materias int not null,
constraint FK_CURSOS_MATERIAS_CURSO
    foreign key (id_curso)
    references tbl_curso (id_curso),
```

```
    constraint FK_MATERIAS_MATERIAS_CURSO
    foreign key (id_materias)
    references tbl_materias (id_materias),
unique index (id_materias_curso)
);
```

```
create table tbl_materias_professor(
id_materias_professor int not null primary key auto_increment,
    id_materias int not null,
    id_professor int not null,
constraint FK_MATERIAS_MATERIAS_PROFESSOR
    foreign key (id_materias)
    references tbl_materias (id_materias),
```

```
    constraint FK_PROFESSOR_MATERIAS_PROFESSOR
    foreign key (id_professor)
    references tbl_professor (id_professor),
unique index (id_materias_professor)
);
```

```

create table tbl_notas_professor_alunos_materias(
id_notas_professor_alunos_materias int not null primary key auto_increment
    id_notas int not null,
    id_materias_professor int not null,
    id_alunos int not null,

constraint FK_NOTAS_NOTAS_PROFESSOR_ALUNO_MATERIAS
    foreign key (id_notas)
    references tbl_notas (id_notas),

constraint
FK_PROFESSOR_MATERIAS_NOTAS_PROFESSOR_ALUNOS_MATERIAS
    foreign key (id_materias_professor)
    references tbl_materias_professor (id_materias_professor),

constraint FK_ALUNOS_NOTAS_PROFESSOR_ALUNOS_MATERIAS
    foreign key (id_alunos)
    references tbl_alunos (id_alunos),

unique index (id_notas_professor_alunos_materias)
);

```

[Link do GITHUB](#)

4 CONCLUSÃO

O desenvolvimento de um banco de dados eficiente é essencial para atender às necessidades de uma instituição acadêmica, permitindo a organização e o controle das informações de forma estruturada e acessível. Ao implementar um sistema que armazena dados sobre alunos, cursos, matérias e professores, possibilitamos que a faculdade realize a gestão de turmas e notas com maior precisão e segurança. Este projeto demonstra a importância de uma base de dados robusta no apoio às operações diárias da instituição, proporcionando melhorias no processo de tomada de decisões e na qualidade do ensino.