



Aluno: Marcos André Alves da Silva

Trabalho apresentado como requisito parcial de avaliação da disciplina **Database Modeling & SQL** do Curso de Graduação em **Superior de Tecnologia em Analise e Desenvolvimento de Sistemas** do Centro Universitário UniFECAF.

Tutor: ***Fernando Leonid***

SUMÁRIO

Sumário

1.	CENÁRIO.....	1
2.	LEVANTAMENTO DE REQUISITOS.....	2
3.	MODELO CONCEITUAL	3
4.	MODELO LÓGICO.....	4
5.	MODELO FÍSICO	5
6.	LINKS.....	10

1. CENÁRIO

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. LEVANTAMENTO DE REQUISITOS

1. Quais informações precisam ser armazenadas sobre os alunos?

Resposta: Nome, CPF, data de nascimento, endereço, telefone e histórico acadêmico.

2. Quais informações devem ser registradas sobre os cursos?

Resposta: Nome do curso, carga horária, disciplinas associadas e período letivo.

3. Como as matérias devem ser organizadas dentro do sistema?

Resposta: Cada matéria terá um nome, código, professor responsável, curso vinculado e carga horária, garantindo estrutura e controle acadêmico.

4. Que dados são necessários sobre os professores?

Resposta: Nome, CPF, titulação acadêmica, disciplinas lecionadas e contatos.

5. Como será feito o controle das turmas e alunos matriculados?

Resposta: Cada turma terá um código único, um período letivo, alunos matriculados e um professor responsável.

6. O sistema precisa armazenar notas dos alunos? Como isso deve ser estruturado?

Resposta: Sim. Deve ter registros de avaliações por disciplina, com nota, data e tipo da avaliação.

7. Haverá algum tipo de histórico acadêmico?

Resposta: Sim, o sistema deve armazenar todas as disciplinas cursadas pelos alunos e suas respectivas notas.

8. O sistema deve permitir a geração de relatórios acadêmicos?

Resposta: Sim, relatórios de desempenho dos alunos, frequência e histórico acadêmico.

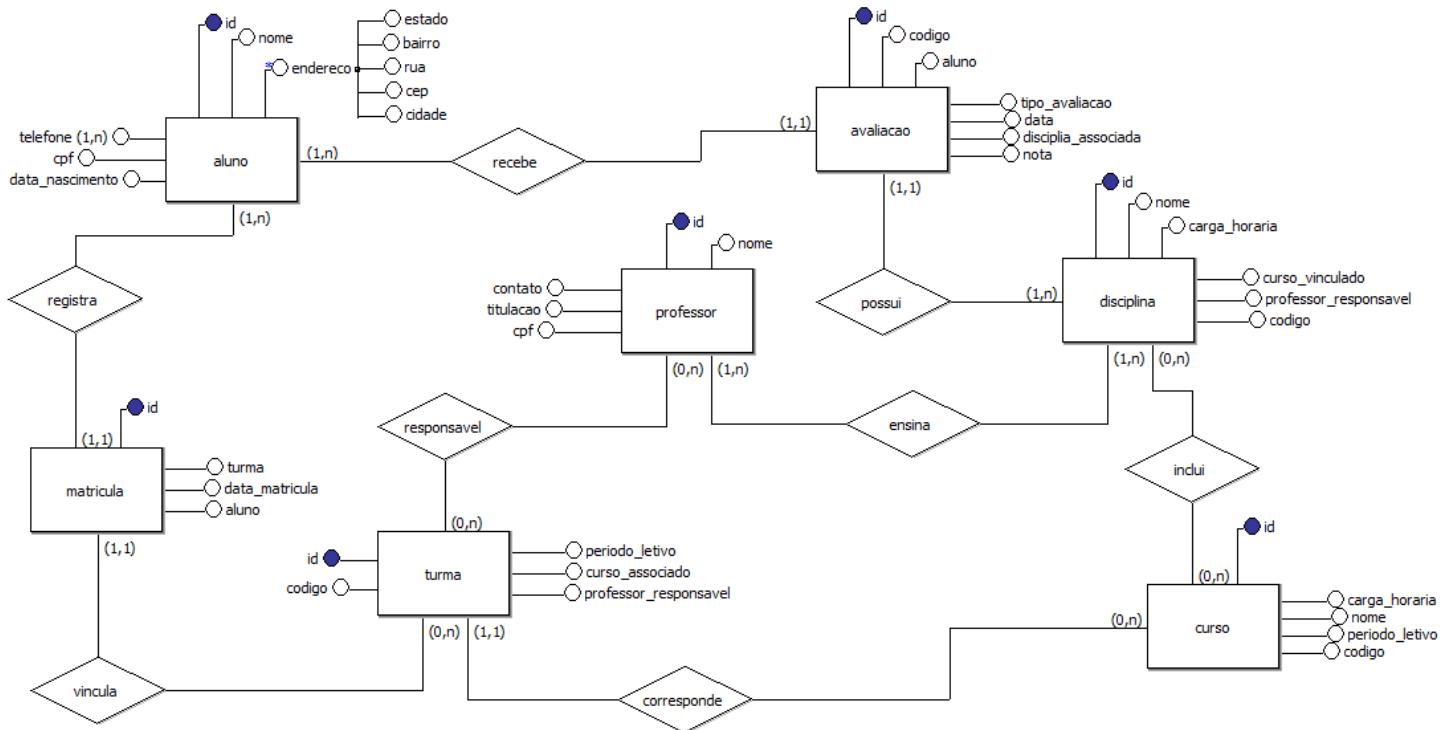
9. Será necessário um controle de usuários e permissões?

Resposta: Sim. Administradores terão acesso total, enquanto professores e alunos terão permissões restritas.

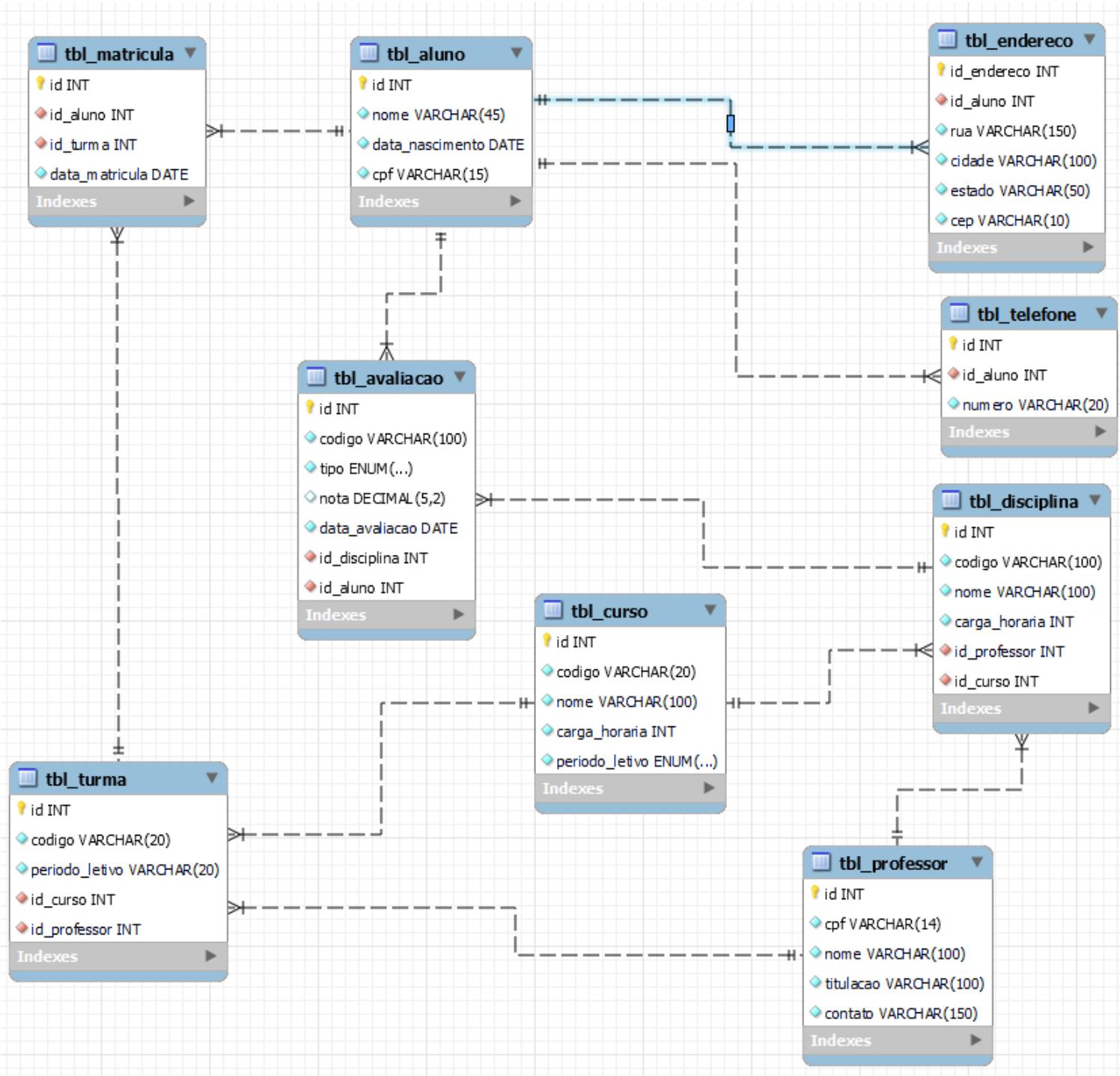
10. Como será feita a integração do sistema com outros serviços?

Resposta: Possível integração com sistemas de pagamento e gestão financeira da faculdade.

3. MODELO CONCEITUAL



4. MODELO LÓGICO



5. MODELO FÍSICO

```
##### Portifolio - Sistema Faculdade #####
```

```
create database db_portifolio_faculdade;
use db_portifolio_faculdade;

create table tbl_aluno (
    id int not null primary key auto_increment,
    nome varchar(45) not null,
    data_nascimento date not null,
    cpf varchar(15) unique not null
);
```

```
alter table endereco rename to tbl_endereco;
create table endereco (
    id_endereco int primary key auto_increment,
    id_aluno int not null,
    rua varchar(150) not null,
    cidade varchar(100) not null,
    estado varchar(50) not null,
    cep varchar(10) not null,
    foreign key (id_aluno) references tbl_aluno(id)
);
```

```
alter table telefone rename to tbl_telefone;
create table telefone (
    id int primary key auto_increment,
    id_aluno int not null,
    numero varchar(20) not null,
    foreign key (id_aluno) references tbl_aluno(id)
);
```

```
create table tbl_curso (
    id int primary key auto_increment,
    codigo varchar(20) unique not null,
    nome varchar(100) not null,
    carga_horaria int not null check (carga_horaria > 0),
    periodo_letivo enum('Semestral', 'Anual', 'Trimestral') not null
);
```

```
create table tbl_professor (
    id int primary key auto_increment,
    cpf varchar(14) unique not null,
    nome varchar(100) not null,
    titulacao varchar(100) not null,
    contato varchar(150) not null
);
```

```
create table tbl_disciplina (
    id int primary key auto_increment,
    codigo varchar(100) not null,
    nome varchar(100) not null,
    carga_horaria int not null check (carga_horaria > 0),
    id_professor int not null,
    id_curso int not null,
    foreign key (id_professor) references tbl_professor(id),
    foreign key (id_curso) references tbl_curso(id)
);
```

```
create table tbl_turma (
    id int primary key auto_increment,
    codigo varchar(20) not null,
    periodo_letivo varchar(20) not null,
    id_curso int not null,
    id_professor int not null,
```

```

foreign key (id_curso) references tbl_curso(id),
foreign key (id_professor) references tbl_professor(id)
);

```

```

create table tbl_matricula (
    id int primary key auto_increment,
    id_aluno int not null,
    id_turma int not null,
    data_matricula date not null,
    foreign key (id_aluno) references tbl_aluno(id),
    foreign key (id_turma) references tbl_turma(id)
);

```

```

create table tbl_avaliacao (
    id int primary key auto_increment,
    codigo varchar(100) not null,
    tipo enum('prova', 'trabalho', 'seminário') not null,
    nota decimal(5,2) check (nota >= 0 and nota <= 10),
    data_avaliacao date not null,
    id_disciplina int not null,
    id_aluno int not null,
    foreign key (id_disciplina) references tbl_disciplina(id),
    foreign key (id_aluno) references tbl_aluno(id)
);

```

Inserindo dados para testes

```

insert into tbl_aluno (nome, data_nascimento, cpf) values
('João Silva', '1990-05-15', '12345678901'),
('Maria Oliveira', '1992-10-22', '98765432100');

```

```

insert into tbl_curso (codigo, nome, carga_horaria, periodo_letivo) values
('MAT101', 'Matemática Básica', 60, 'Semestral'),

```

```
('HIS202', 'História Geral', 80, 'Anual');
```

```
insert into tbl_professor (cpf, nome, titulacao, contato) values
('12345678901', 'Carlos Mendes', 'Doutor em Física', 'carlos@email.com'),
('98765432100', 'Ana Souza', 'Mestre em História', 'ana@email.com');
```

```
select * from tbl_aluno;
select * from tbl_curso;
select * from tbl_professor;
```

```
insert into tbl_disciplina (id, codigo, nome, carga_horaria, id_professor, id_curso) values
(1, 'MAT101', 'Matemática Básica', 60, 1, 1),
(2, 'HIS202', 'História Geral', 80, 2, 2);
```

```
insert into tbl_turma (codigo, periodo_letivo, id_curso, id_professor) values
('TUR001', '2025.1', 1, 1),
('TUR002', '2025.1', 2, 2);
```

```
insert into tbl_matricula (id_aluno, id_turma, data_matricula) values
(1, 1, '2025-02-10'),
(2, 2, '2025-02-12');
```

```
insert into tbl_avaliacao (codigo, tipo, nota, data_avaliacao, id_disciplina, id_aluno)
values
('AV001', 'prova', 8.5, '2025-03-15', 1, 1),
('AV002', 'trabalho', 9.0, '2025-04-10', 2, 2);
```

```
select * from tbl_turma;
select * from tbl_matricula;
select * from tbl_avaliacao;
```

Consultar alunos e suas matrículas

Listar os alunos matriculados e suas respectivas turmas

```
select a.nome, t.codigo as turma, m.data_matricula  
from tbl_matricula m  
join tbl_aluno a on m.id_aluno = a.id  
join tbl_turma t on m.id_turma = t.id;
```

Listar avaliações com nota maior ou igual a 8, incluindo a disciplina e aluno correspondente

```
select a.codigo, a.tipo, a.nota, d.nome as disciplina, al.nome as aluno  
from tbl_avaliacao a  
join tbl_disciplina d on a.id_disciplina = d.id  
join tbl_aluno al on a.id_aluno = al.id  
where a.nota >= 8  
order by a.nota desc;
```

6. LINKS

GitHub	
Vídeo	