



UBEC
UNIVERSIDADE CATÓLICA DE BRASÍLIA
DISCIPLINA LABORATÓRIO DE BANCO DE DADOS

LUCIANA LIEBL DE FREITAS

BANCO DE DADOS DE UNIVERSIDADE

BRASÍLIA – DF
2024

LUCIANA LIEBL DE FREITAS

BANCO DE DADOS PARA UNIVERSIDADES

Protejo de banco de dados apresentado ao curso de Análise e Desenvolvimento de Sistemas, na disciplina Laboratório de Banco de Dados como parte dos requisitos para a conclusão desta.

Professora: Adriana Falcomer Pontes

Disciplina: Laboratório de Banco de Dados

BRASÍLIA – DF

2024



UBEC

UNIVERSIDADE CATÓLICA DE BRASÍLIA
DISCIPLINA LABORATÓRIO DE BANCO DE DADOS

LUCIANA LIEBL DE FREITAS
BANCO DE DADOS PARA UNIVERSIDADES

Trabalho de conclusão de disciplina Banco de Dados, apresentado como requisito parcial para obtenção da aprovação.

Aprovado em: ____de _____ de 2024.

Examinador

(nome, titulação e instituição a que pertence).

AGRADECIMENTOS

A Deus pela força para perseverar.....

A Profª Dr. xxxxx, pela excelente orientação

Aos colegas da turma..

É ótimo celebrar o sucesso, mas mais importante ainda é assimilar as lições trazidas pelos erros que cometemos".

- Bill Gates

RESUMO

Projetaremos um banco de dados para gerenciamento de uma faculdade. Os objetivos deste banco de dados são: Realizar controle centralizado de alunos, professores, cursos, disciplinas, histórico escolar e turmas, desta forma, projetaremos um banco de dados para o uso de uma determinada faculdade, as regras de negócios e requisitos já estão estabelecidas, podendo ser mudado conforme o desejo do cliente e a necessidade de mudanças no projeto. O professor os auxiliará conforme necessário em cada etapa do projeto!

Palavras-chave: Banco de dados. DBA. Entidade, Relacionamento, Atributos.

LISTA DE ABREVIATURAS E SIGLAS

DER	Diagrama de Entidade Relacionamento
ER	Entidade relacionamento (entity-relationship)
RA	Registro do Academico
SENAI	Serviço Nacional de Aprendizagem Industrial
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language

SUMÁRIO

1 INTRODUÇÃO	7
2 REQUISITOS E REGRAS DE NEGÓCIO	7
3 MODELO ER.....	8
4 DIAGRAMA ER	11
5 DICIONARIO DE DADOS	11
7 NORMALIZAÇÃO.....	14
8 IMPLEMENTAÇÃO	15
9 CONCLUSÃO.....	22

1 - INTRODUÇÃO

Para iniciarmos iremos conhecer como nosso projeto funcionara. Seguiremos algumas etapas no decorrer do curso para que possamos avançar aos poucos, até a conclusão deste projeto. As etapas são:

- Levantamento de requisitos
- Identificação de entidades e relacionamentos
- Modelo ER
- Diagrama ER
- Dicionário de dados
- Normalização
- Implementação
- Testes básicos
- Rotinas de manipulação

2 - REQUISITOS E REGRAS DE NEGÓCIO

Tenha em mente que neste documento já constará os requisitos e as regras de negócio pré-estabelecidas e que nos baseando por elas e pelas possíveis novas regras que forem surgindo por parte do cliente construiremos o nosso banco de dados.

Uma faculdade de nome Universidade XXXXX deseja mudar o seu sistema, e para isso ela precisa criar e implementar um novo banco de dados que permita a ela, realizar o controle centralizado de alunos, professores, cursos, disciplinas, histórico escolar e turmas. Após uma longa conversa com o cliente o analista de requisitos chegou-se nas seguintes regras de negócio:

- Um aluno só pode estar matriculado em um curso por vez;
- Alunos possuem um código de identificação (RA);
- Cursos são compostos por várias disciplinas;
- Cada disciplina terá no máximo 30 alunos por turma;
- As disciplinas podem ser obrigatórias ou optativas, dependendo do curso;

- As disciplinas pertencem a departamentos específicos;
- Cada disciplina possui um código de identificação;
- Alunos podem trancar matrícula, não estando então matriculados em nenhuma disciplina no semestre;
- Em cada semestre, cada aluno pode se matricular em no máximo 6 disciplinas;
- O aluno só pode ser reprovado no máximo 3 vezes na mesma disciplina;
- A faculdade terá no máximo 3000 alunos matriculados simultaneamente, em 10 cursos;
- Entram 300 alunos novos por ano;
- Existem 90 disciplinas no total disponíveis;
- Um histórico escolar traz todas as disciplinas cursadas por aluno. Incluindo nota final, frequência e período do curso realizado;
- Professores podem ser cadastrados mesmo sem lecionar disciplinas;
- Existem 40 professores trabalhando na escola;
- Cada professor irá lecionar no máximo 4 disciplinas diferentes;
- Cada professor é vinculado a um departamento;
- Professores são identificados por um código de professor

3- MODELO ER

Liste as entidades encontradas.

- **Departamento**
- **Curso**
- **Aluno**
- **Endereço**
- **Telefone**
- **Disciplina**
- **Curso_Disciplina**
- **Aluno_Disciplina**
- **Historico_Escolar**
- **Turma**
- **Turma_Aluno**
- **Professor**
- **Turma_Disciplina_Professor**

Liste os possíveis atributos de cada entidade encontrada.

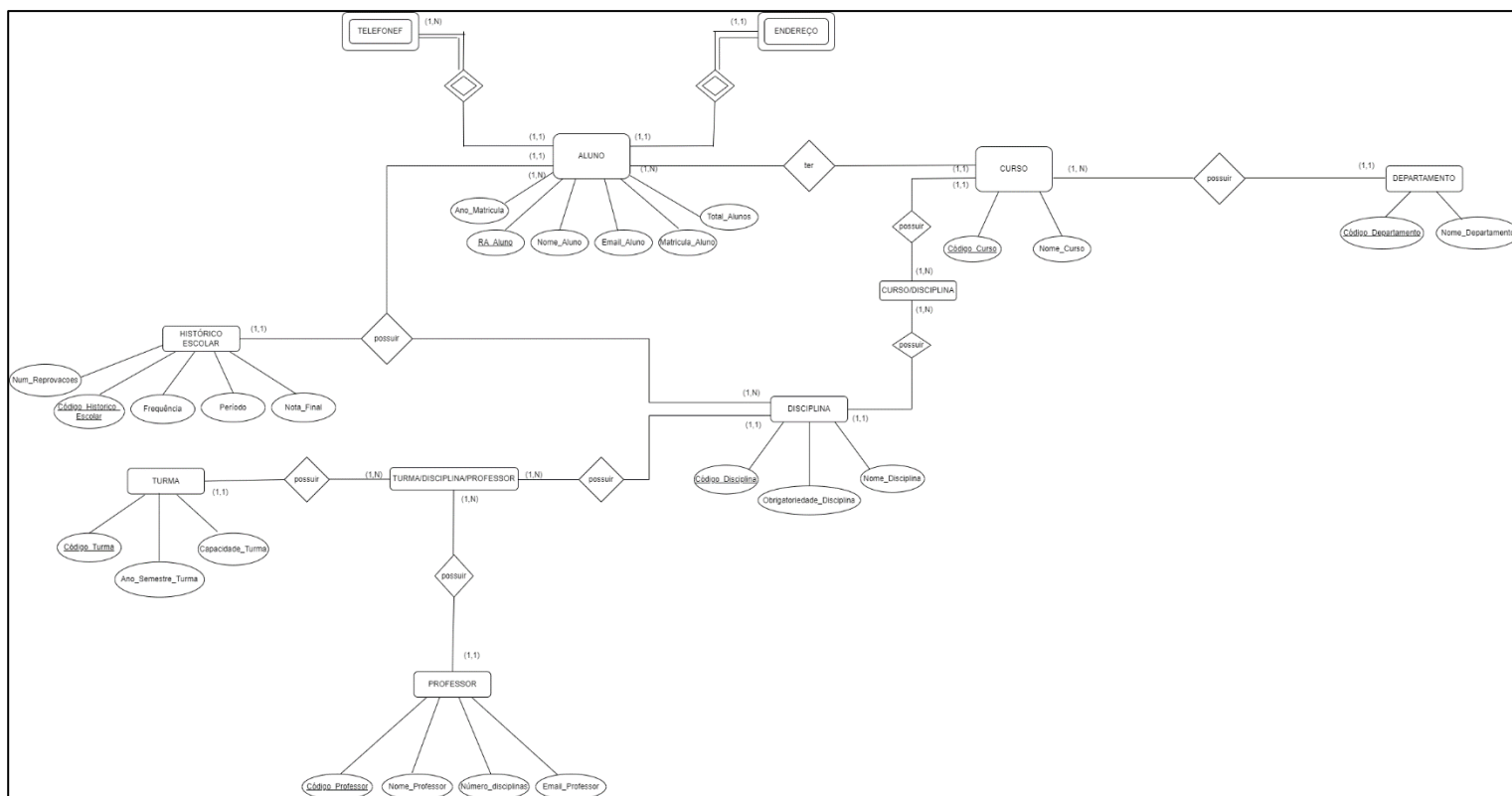
- **Departamento:** Codigo_departamento, Nome_departamento
- **Curso:** Codigo_curso, Nome_curso,
- **Aluno:** RA_aluno, Nome_aluno, Email_aluno, Matricula_aluno, Total_Alunos, Ano_Matricula
- **Endereço:** Endereco_aluno, RA_aluno
- **Telefone:** Telefone_aluno, RA_aluno
- **Disciplina:** Codigo_disciplina, Nome_disciplina, Obrigatoriedade_disciplina
- **Curso_Disciplina:** Codigo_curso, Codigo_disciplina
- **Aluno_Disciplina:** Codigo_disciplina, RA_aluno, Ano_Semestre
- **Historico_Escolar:** Codigo_Historico_Escolar, RA_aluno, Codigo_disciplina, Nota_Final, Frequencia, Periodo, Num_Reprovacoes
- **Turma:** Codigo_Turma, Capacidade_Turma, Ano_Semestre_Turma
- **Turma_Aluno:** Codigo_Turma, RA_aluno, Ano_Semestre_Turma,
- **Professor:** Codigo_Professor, Nome_Professor, Numero_Disciplinas, Email_Professor
- **Turma_Disciplina_Professor:** Codigo_Disciplina, Codigo_Professor, Codigo_Turma

Liste os possíveis relacionamentos encontrados.

- Departamento:
 - Relacionamento 1:N com Curso
- Curso:
 - Relacionamento 1:1 com Departamento
 - Relacionamento 1:N com Aluno
 - Relacionamento 1:N com Curso_Disciplina
- Aluno:
 - Relacionamento 1:N com Telefone
 - Relacionamento 1:1 com Endereço
 - Relacionamento 1:1 com Curso
 - Relacionamento 1:1 com Histórico_Escolar
 - Relacionamento 1:N com Aluno_Disciplina
 - Relacionamento 1:N com Turma_Aluno
- Endereço:
 - Relacionamento 1:1 com Aluno
- Telefone:
 - Relacionamento 1:N com Aluno
- Disciplina:
 - Relacionamento 1:N com Histórico_Escolar
 - Relacionamento 1:N com Turma_Disciplina_Professor

- Relacionamento 1:N com Aluno_Disciplina
- Relacionamento 1:N com Curso_Disciplina
- Curso_Disciplina:
 - Relacionamento 1:N com Curso
 - Relacionamento 1:N Disciplina
- Aluno_Disciplina:
 - Relacionamento 1:N com Aluno
 - Relacionamento 1:N Disciplina
- Historico_Escolar:
 - Relacionamento 1:1 com Aluno
 - Relacionamento 1:1 com Aluno
- Turma:
 - Relacionamento 1:N Turma_Disciplina_Professor
 - Relacionamento 1:N com Tumar_Aluno
- Turma_Aluno:
 - Relacionamento 1:N com Turma
 - Relacionamento 1:N com Aluno
- Professor:
 - Relacionamento 1:N com Turma_Disciplina_Professor
- Turma_Disciplina_Professor:
 - Relacionamento 1:N com Turma
 - Relacionamento 1:N com Disciplina
 - Relacionamento 1:N com Professor

4 - DIAGRAMA ER



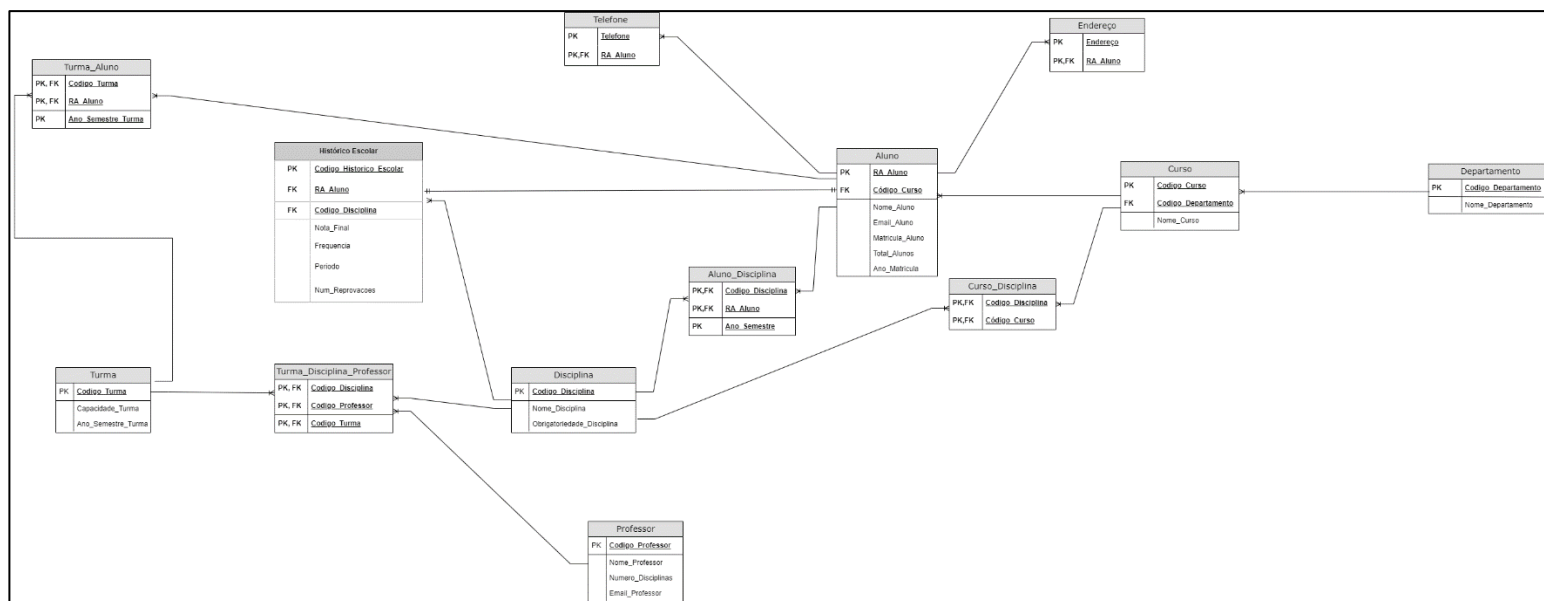
5- DICIONÁRIO DE DADOS

Nome da Tabela	Descrição da Tabela	Relacionamentos da Tabela	Atributos da tabela	Tipos de Atributos
Departamento	Tabela que armazena informações sobre os departamentos da universidade	Relacionamento 1:N com Curso	Codigo_departamento (PK)	int auto_increment
			Nome_departamento	varchar(50) not nul
Curso	Tabela que contém informações sobre os cursos oferecidos pela universidade.	Relacionamento 1:1 com Departamento Relacionamento 1:N com Aluno Relacionamento 1:N com Curso_Disciplina	Codigo_curso	int auto_increment,
			Nome_curso	varchar(100),
			Codigo_departamento (FK)	int
Aluno	Tabela que armazena informações sobre os alunos matriculados na universidade.	Relacionamento 1:N com Telefone Relacionamento 1:1 com Endereço Relacionamento 1:1	RA_aluno (PK)	int auto_increment
			Nome_aluno	varchar(100) not null

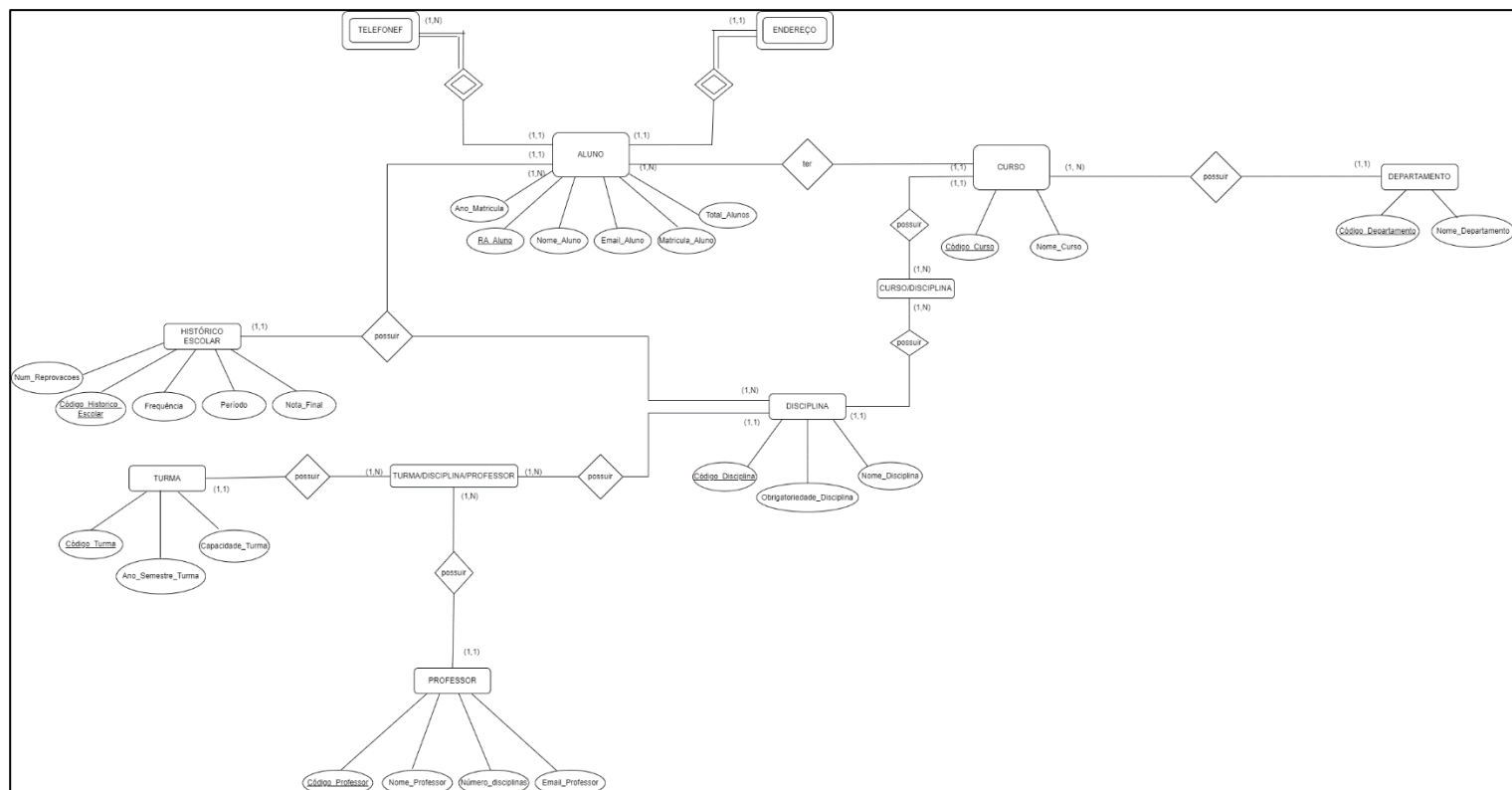
		com Curso Relacionamento 1:1 com Histórico_Escolar Relacionamento 1:N com Aluno_Disciplina Relacionamento 1:N com Turma_Aluno	Email_aluno	varchar(100) not null
			Codigo_curso (FK)	int
			Matricula_aluno	varchar(100) not null
			Total_Alunos	int
			Ano_Matricula	int
Endereço	Tabela que registra os endereços dos alunos.	Relacionamento 1:1 com Aluno	Endereco_aluno (PK)	varchar(50) not null
			RA_aluno (PK, FK)	int
Telefone	Tabela que registra os números de telefone dos alunos.	Relacionamento 1:N com Aluno	Telefone_aluno (PK)	varchar(50) not null
			RA_aluno (PK, FK)	int
Disciplina	Tabela que contém informações sobre as disciplinas oferecidas pela universidade	Relacionamento 1:N com Histórico_Escolar Relacionamento 1:N com Turma_Disciplina_Professor Relacionamento 1:N com Aluno_Disciplina Relacionamento 1:N com Curso_Disciplina	Codigo_disciplina (PK)	int auto_increment
			Nome_disciplina	varchar(100) not null
			Obrigatoriedade_disciplina	enum ('Obrigatoria', 'Optativa') not null
Curso_Disciplina	Tabela que relaciona os cursos às disciplinas que eles oferecem.	Relacionamento 1:N com Curso Relacionamento 1:N Disciplina	Codigo_curso (PK, FK)	int
			Codigo_disciplina (PK, FK)	int
Aluno_Disciplina	Tabela que registra as disciplinas em que os alunos estão matriculados.	Relacionamento 1:N com Aluno Relacionamento 1:N Disciplina	Codigo_disciplina (PK, FK)	int
			RA_aluno (PK, FK)	int
			Ano_Semestre (PK)	varchar(10) not null
Historico_Escolar	Tabela que registra o histórico acadêmico dos alunos em cada disciplina.	Relacionamento 1:1 com Aluno Relacionamento 1:1 com Aluno	Codigo_Historico_Escolar (PK)	int auto_increment
			RA_aluno (FK)	int

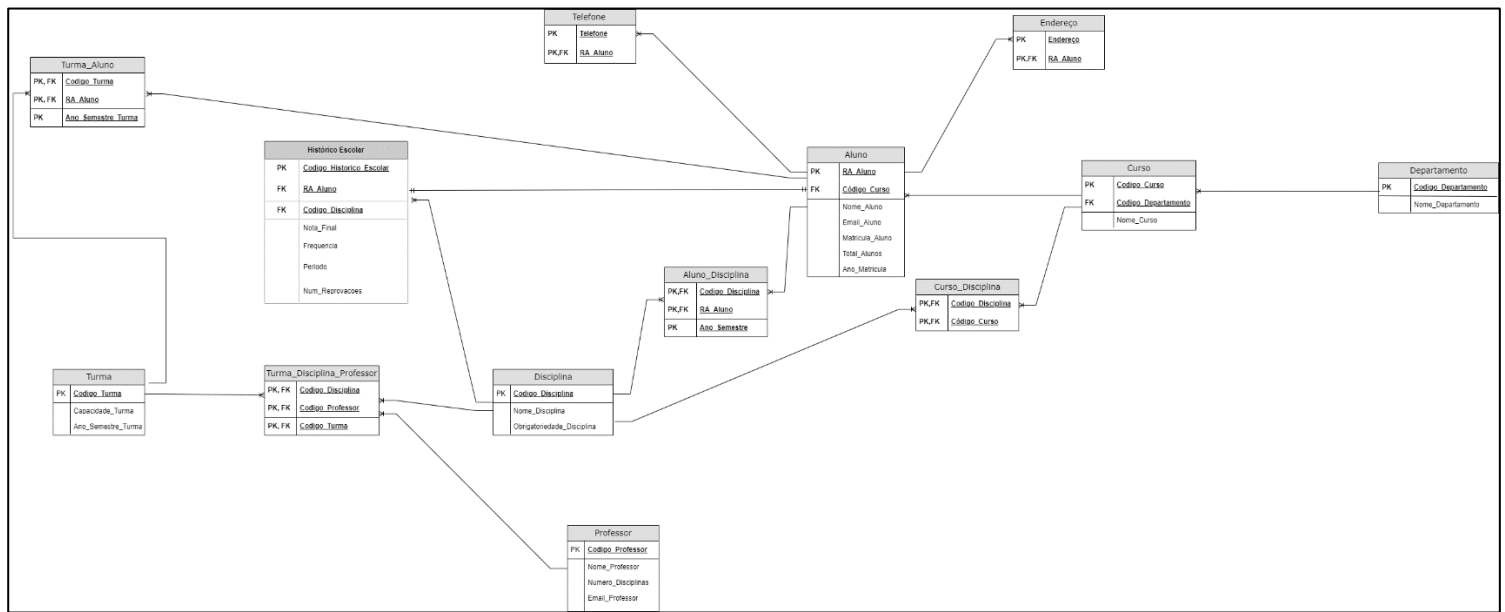
			Codigo_disciplina	int
			Nota_Final	decimal(4, 2)
			Frequencia	int
			Periodo	varchar(10)
			Num_Reprovacoes	int default 0
Turma	Tabela que armazena informações sobre as turmas formadas para as disciplinas.	Relacionamento 1:N Turma_Disciplina_Professor Relacionamento 1:N com Tumar_Aluno	Codigo_Turma (PK)	int auto_increment
			Capacidade_Turma	int not null
			Ano_Semestre_Turma	varchar(10) not null
Turma_Aluno	Tabela que relaciona os alunos matriculados em cada turma.	Relacionamento 1:N com Turma Relacionamento 1:N com Aluno	Codigo_Turma(PK,FK)	int
			RA_aluno (PK,FK)	int
			Ano_Semestre_Turma (PK)	varchar(10)
Professor	Tabela que armazena informações sobre os professores da universidade.	Relacionamento 1:N com Turma_Disciplina_Professor	Codigo_Professor (PK)	int auto_increment
			Nome_Professor	varchar(50) not null
			Numero_Disciplinas	int not null
			Email_Professor	varchar(50) not null
Turma_Disciplina_Professor	Tabela que relaciona as disciplinas, turmas e professores responsáveis por cada turma.	Relacionamento 1:N com Turma Relacionamento 1:N com Disciplina Relacionamento 1:N com Professor	Codigo_Disciplina (PK, FK)	int
			Codigo_Professor (PK, FK)	int
			Codigo_Turma (PK, FK)	int

6- MODELO LÓGICO



7 NORMALIZAÇÃO





8 IMPLEMENTAÇÃO

```
create database universidade
default character set utf8
default collate utf8_general_ci;
```

```
create table Departamento(
    Codigo_departamento int auto_increment,
    Nome_departamento varchar(50) not null,
    PRIMARY KEY (Codigo_departamento)
)default charset = utf8;
```

```
create table Curso (
    Codigo_curso int auto_increment,
    Nome_curso varchar(100),
    Codigo_departamento int,
    PRIMARY KEY (Codigo_curso),
    FOREIGN KEY (Codigo_departamento) references Departamento (Codigo_departamento)
)default charset = utf8;
```

```
create table Aluno(
    RA_aluno int auto_increment,
    Nome_aluno varchar(100) not null,
    Email_aluno varchar(100) not null,
    Matricula_aluno varchar(100) not null,
    Codigo_curso int,
    PRIMARY KEY (RA_aluno),
    FOREIGN KEY (Codigo_curso) references Curso(Codigo_curso)
) default charset = utf8;
```



```
alter table Aluno
add column Total_Alunos int;
```

```
alter table Aluno
add column Ano_Matricula int;
```

```
create table Endereco(
    Endereco_aluno varchar(50) not null,
    RA_aluno int,
    PRIMARY KEY (Endereco_aluno, RA_aluno),
    FOREIGN KEY (RA_aluno) references Aluno(RA_aluno)
) default charset = utf8;
```

```
create table Telefone(
    Telefone_aluno varchar(50) not null,
    RA_aluno int,
    PRIMARY KEY (Telefone_aluno, RA_aluno),
    FOREIGN KEY (RA_aluno) references Aluno(RA_aluno)
) default charset = utf8;
```

```
create table Disciplina(
    Codigo_disciplina int auto_increment,
    Nome_disciplina varchar(100) not null,
    Obrigatoriedade_disciplina enum ('Obrigatoria', 'Optativa') not null,
    PRIMARY KEY (Codigo_disciplina)
) default charset = utf8;
```

```
create table Curso_Disciplina(
    Codigo_curso int,
    Codigo_disciplina int,
    PRIMARY KEY (Codigo_curso, Codigo_disciplina),
    FOREIGN KEY (Codigo_curso) REFERENCES Curso(Codigo_curso),
    FOREIGN KEY (Codigo_disciplina) REFERENCES Disciplina(Codigo_disciplina)
)default charset = utf8;
```

```
create table Aluno_Disciplina(
    Codigo_disciplina int,
    RA_aluno int,
    Ano_Semestre varchar(10) not null,
    PRIMARY KEY (Codigo_disciplina, RA_aluno, Ano_Semestre),
    FOREIGN KEY (Codigo_disciplina) REFERENCES Disciplina(Codigo_disciplina),
    FOREIGN KEY (RA_aluno) REFERENCES Aluno(RA_aluno)
)default charset utf8;
```

```
create table Historico_Escolar(
    Codigo_Historico_Escolar int auto_increment,
    RA_aluno int,
    Codigo_disciplina int,
```

```

        Nota_Final decimal(4, 2),
        Frequencia int,
        Período varchar(10),
        PRIMARY KEY (Codigo_Historico_Escolar),
        FOREIGN KEY (RA_aluno) REFERENCES Aluno (RA_aluno),
        FOREIGN KEY (codigo_disciplina) REFERENCES Disciplina(Codigo_disciplina)
    )default charset utf8;

```

```

alter table historico_escolar
add column Num_Reprovacoes int default 0;

```

```

create table Turma(
    Codigo_Turma int auto_increment,
    Capacidade_Turma int not null,
    Ano_Semestre_Turma varchar(10) not null,
    PRIMARY KEY (Codigo_Turma)
)default charset utf8;

```

```

create table Turma_Aluno(
    Codigo_Turma int,
    RA_aluno int,
    Ano_Semestre_Turma varchar(10),
    PRIMARY KEY (Codigo_Turma, RA_aluno, Ano_Semestre_Turma),
    FOREIGN KEY (Codigo_Turma) REFERENCES Turma(Codigo_Turma),
    FOREIGN KEY (RA_Aluno) REFERENCES Aluno(RA_aluno)
)default charset utf8;

```

```

create table professor(
    Codigo_Professor int auto_increment,
    Nome_Professor varchar(50) not null,
    Numero_Disciplinas int not null,
    Email_Professor varchar(50) not null,
    PRIMARY KEY (Codigo_Professor)
)default charset utf8;

```

```

create table Turma_Disciplina_Professor(
    Codigo_Disciplina int,
    Codigo_Professor int,
    Codigo_Turma int,
    PRIMARY KEY (Codigo_Disciplina, Codigo_Professor, Codigo_Turma),
    FOREIGN KEY (Codigo_Disciplina) REFERENCES Disciplina(Codigo_Disciplina),
    FOREIGN KEY (Codigo_Professor) REFERENCES Professor(Codigo_Professor),
    FOREIGN KEY (Codigo_Turma) REFERENCES Turma(Codigo_Turma)
)default charset utf8;

```

```

-- Cada disciplina terá no máximo 30 alunos por turma;
DELIMITER //
CREATE TRIGGER tr_Limite_Alunos_Turma
BEFORE INSERT ON turma_aluno

```

```

FOR EACH ROW
BEGIN
    DECLARE num_alunos INT;

    SELECT COUNT(*)
    INTO num_alunos
    FROM turma_aluno
    WHERE Codigo_Turma = NEW.Codigo_Turma
        AND Ano_Semestre_Turma = NEW.Ano_Semestre_Turma;

    IF num_alunos >= 30 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'A turma atingiu o limite de 30 alunos.';
    END IF;
END //
DELIMITER ;

-- Garantir que um aluno possa se matricular em no máximo 6 disciplinas
DELIMITER //
CREATE TRIGGER tr_Limite_Disciplinas_Semestre
BEFORE INSERT ON Aluno_Disciplina
FOR EACH ROW
BEGIN
    DECLARE total_disciplinas INT;

    SELECT COUNT(*)
    INTO total_disciplinas
    FROM Aluno_Disciplina
    WHERE RA_aluno = NEW.RA_aluno
        AND Ano_Semestre = NEW.Ano_Semestre;

    IF total_disciplinas >= 6 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'O aluno já está matriculado em 6 disciplinas neste semestre.';
    END IF;
END //
DELIMITER ;

-- Garantir que aluno só pode ser reprovado no máximo 3 vezes na mesma disciplina
DELIMITER //
CREATE TRIGGER tr_Limite_Reprovacoes_Disciplina
BEFORE INSERT ON Historico_Escolar
FOR EACH ROW
BEGIN
    DECLARE num_reprovacoes INT;

    SELECT COUNT(*)
    INTO num_reprovacoes
    FROM Historico_Escolar

```

```

WHERE RA_aluno = NEW.RA_aluno
  AND Codigo_disciplina = NEW.Codigo_disciplina
  AND Nota_Final < 6.0;

IF num_reprovacoes >= 3 THEN
  SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'O aluno já foi reprovado 3 vezes nesta disciplina.';
ELSE
  SET NEW.Num_Reprovacoes = num_reprovacoes + 1;
END IF;
END //
DELIMITER ;

-- Garantir que a faculdade terá no máximo 3000 alunos matriculados simultaneamente, em 10 cursos;
DELIMITER //
CREATE TRIGGER tr_Limite_Alunos_Faculdade
BEFORE INSERT ON aluno
FOR EACH ROW
BEGIN
  DECLARE total_alunos int;
  DECLARE total_alunos_curso int;

  SELECT SUM(Total_Alunos) into total_alunos FROM Aluno;

  SELECT Total_Alunos into total_alunos_curso
  FROM Aluno
  WHERE Codigo_curso = NEW.Codigo_curso;

  IF total_alunos + 1 > 3000 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'A faculdade atingiu o limite de 3.000 alunos matriculados simultaneamente';
  ELSE
    IF total_alunos_curso + 1 > 300 THEN
      SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'O curso atingiu o limite de 300 alunos matriculados simulta-
neamente';
    ELSE
      UPDATE Aluno
      SET Total_Alunos = total_alunos_curso + 1
      WHERE RA_aluno = NEW.RA_aluno;
    END IF;
  END IF;
END //
DELIMITER ;

-- Entram 300 alunos novos por ano;
DELIMITER //

CREATE TRIGGER tr_Limite_Alunos_Novos

```

```

BEFORE INSERT ON Aluno
FOR EACH ROW
BEGIN
    DECLARE total_alunos_ano INT;

    SELECT COUNT(*)
    INTO total_alunos_ano
    FROM Aluno
    WHERE Ano_Matricula = YEAR(NOW());

    IF total_alunos_ano >= 300 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'A faculdade atingiu o limite de 300 alunos novos matriculados neste ano.';
    ELSE
        SET NEW.Ano_Matricula = YEAR(NOW());
    END IF;
END //
DELIMITER ;

-- Cada professor irá lecionar no máximo 4 disciplinas diferentes;
DELIMITER //
CREATE TRIGGER tr_Limite_Disciplinas_Professor
BEFORE INSERT ON Turma_Disciplina_Professor
FOR EACH ROW
BEGIN
    DECLARE num_disciplinas INT;

    SELECT Numero_Disciplinas
    INTO num_disciplinas
    FROM Professor
    WHERE Codigo_Professor = NEW.Codigo_Professor;

    IF num_disciplinas >= 4 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'O professor já está lecionando o máximo de 4 disciplinas.';
    ELSE
        UPDATE Professor
        SET Numero_Disciplinas = Numero_Disciplinas + 1
        WHERE Codigo_Professor = NEW.Codigo_Professor;
    END IF;
END //
DELIMITER ;

-- Existem 40 professores trabalhando na Universidade
INSERT INTO professor (Nome_Professor, Numero_Disciplinas, Email_Professor) VALUES
('Ricardo Santos', 1, 'ricardo.santos@universidade.com'), ('Patrícia Lima', 2, 'patricia.lima@universi-
dade.com'), ('André Fernandes', 2, 'andre.fernandes@universidade.com'), ('Luana Martins', 2, 'luana.mar-
tins@universidade.com'), ('Daniel Almeida', 2, 'daniel.almeida@universidade.com'), ('Carolina Gonçalves',

```

4, 'carolina.goncalves@universidade.com'), ('Thiago Oliveira', 4, 'thiago.oliveira@universidade.com'), ('Fernanda Costa', 2, 'fernanda.costa@universidade.com'), ('Gustavo Silva', 1, 'gustavo.silva@universidade.com'), ('Letícia Mendes', 4, 'leticia.mendes@universidade.com'), ('Isabela Pereira', 1, 'isabela.pereira@universidade.com'), ('Diego Cardoso', 0, 'diego.cardoso@universidade.com'), ('Bianca Mendes', 0, 'bianca.mendes@universidade.com'), ('Paula Costa', 4, 'paula.costa@universidade.com'), ('Gabriel Lopes', 3, 'gabriel.lopes@universidade.com'), ('Amanda Silva', 3, 'amanda.silva@universidade.com'), ('Lucas Nunes', 3, 'lucas.nunes@universidade.com'), ('Eduardo Carvalho', 1, 'eduardo.carvalho@universidade.com'), ('Carlos Silva', 1, 'carlos.silva@universidade.com'), ('Ana Souza', 3, 'ana.souza@universidade.com'), ('Pedro Oliveira', 3, 'pedro.oliveira@universidade.com'), ('Mariana Santos', 3, 'mariana.santos@universidade.com'), ('Bruno Costa', 3, 'bruno.costa@universidade.com'), ('Juliana Rodrigues', 2, 'juliana.rodrigues@universidade.com'), ('Lucas Pereira', 1, 'lucas.pereira@universidade.com'), ('Camila Almeida', 2, 'camila.almeida@universidade.com'), ('Thiago Ferreira', 2, 'thiago.ferreira@universidade.com'), ('Larissa Martins', 3, 'larissa.martins@universidade.com'), ('Rafael Lima', 4, 'rafael.lima@universidade.com'), ('Fernanda Gomes', 4, 'fernanda.gomes@universidade.com'), ('Gustavo Barbosa', 4, 'gustavo.barbosa@universidade.com'), ('Vanessa Oliveira', 2, 'vanessa.oliveira@universidade.com'), ('João Santos', 0, 'joao.santos@universidade.com'), ('Marina Alves', 1, 'marina.alves@universidade.com'), ('Felipe Ribeiro', 2, 'felipe.ribeiro@universidade.com'), ('Renata Castro', 4, 'renata.castro@universidade.com'), ('Leandro Rocha', 4, 'leandro.rocha@universidade.com'), ('Larissa Rodrigues', 3, 'larissa.rodrigues@universidade.com'), ('Mateus Oliveira', 3, 'mateus.oliveira@universidade.com'), ('Mariana Carvalho', 3, 'mariana.carvalho@universidade.com');

select * from disciplina;

-- Existem 90 disciplinas no total disponíveis

INSERT INTO Disciplina (Nome_disciplina, Obrigatoriedade_disciplina)

VALUES

('Introdução à Computação', 'Obrigatoria'), ('Cálculo I', 'Obrigatoria'), ('Álgebra Linear', 'Obrigatoria'), ('Estruturas de Dados', 'Obrigatoria'), ('Física I', 'Obrigatoria'), ('Química Geral', 'Obrigatoria'), ('História da Arte', 'Optativa'), ('Literatura Brasileira', 'Optativa'), ('Filosofia Contemporânea', 'Optativa'), ('Introdução à Economia', 'Optativa'), ('Marketing Digital', 'Optativa'), ('Gestão de Projetos', 'Optativa'), ('Direito Constitucional', 'Optativa'), ('Desenvolvimento Sustentável', 'Optativa'), ('Empreendedorismo', 'Optativa'), ('Administração Financeira', 'Optativa'), ('Psicologia Organizacional', 'Optativa'), ('Ética Profissional', 'Optativa'), ('Design Thinking', 'Optativa'), ('Comunicação Empresarial', 'Optativa'), ('Liderança e Coaching', 'Optativa'), ('Mídias Digitais', 'Optativa'), ('Gestão de Recursos Humanos', 'Optativa'), ('Negociação e Vendas', 'Optativa'), ('Programação Orientada a Objetos', 'Obrigatoria'), ('Banco de Dados', 'Obrigatoria'), ('Engenharia de Software', 'Obrigatoria'), ('Sistemas Operacionais', 'Obrigatoria'), ('Redes de Computadores', 'Obrigatoria'), ('Matemática Discreta', 'Obrigatoria'), ('Direito Empresarial', 'Optativa'), ('Gestão da Qualidade', 'Optativa'), ('Logística Empresarial', 'Optativa'), ('Estratégia de Marketing', 'Optativa'), ('Finanças Corporativas', 'Optativa'), ('Análise de Dados', 'Optativa'), ('Inteligência Artificial', 'Optativa'), ('Gestão de Projetos de TI', 'Optativa'), ('Engenharia Econômica', 'Optativa'), ('Gestão Ambiental', 'Optativa'), ('Comportamento do Consumidor', 'Optativa'), ('Direito Tributário', 'Optativa'), ('Auditoria Interna', 'Optativa'), ('Inovação e Tecnologia', 'Optativa'), ('Comunicação Digital', 'Optativa'), ('Gestão de Conflitos', 'Optativa'), ('Educação Ambiental', 'Optativa'), ('Programação Avançada', 'Obrigatoria'), ('Modelagem de Dados', 'Obrigatoria'), ('Inteligência de Negócios', 'Obrigatoria'), ('Segurança da Informação', 'Obrigatoria'), ('Web Design', 'Obrigatoria'), ('Marketing de Conteúdo', 'Optativa'), ('Gestão Estratégica de Pessoas', 'Optativa'), ('Gestão de Custos', 'Optativa'), ('Direito do Trabalho', 'Optativa'), ('Relações Internacionais', 'Optativa'), ('Finanças Pessoais', 'Optativa'), ('Economia Internacional', 'Optativa'), ('Gestão de Inovação', 'Optativa'), ('Planejamento Tributário', 'Optativa'), ('Gestão de Operações', 'Optativa'), ('Marketing de Serviços', 'Optativa'), ('Tecnologias Emergentes', 'Optativa'), ('Empreendedorismo Social', 'Optativa'), ('Sustentabilidade Corporativa', 'Optativa'), ('Gestão de Startups', 'Optativa'), ('Comunicação

Interpessoal', 'Optativa'), ('Análise de Mercado', 'Optativa'), ('Gestão de Crises', 'Optativa'), ('Direito Ambiental', 'Optativa'), ('Biologia Molecular', 'Obrigatoria'), ('Genética Avançada', 'Obrigatoria'), ('Bioquímica Celular', 'Obrigatoria'), ('História Medieval', 'Optativa'), ('Arqueologia Brasileira', 'Optativa'), ('Geopolítica Contemporânea', 'Optativa'), ('Arte Moderna', 'Optativa'), ('Cinema e Literatura', 'Optativa'), ('Teoria Musical', 'Optativa'), ('Teatro e Sociedade', 'Optativa'), ('Estudos Culturais', 'Optativa'), ('Antropologia Urbana', 'Optativa'), ('Ecologia de Paisagens', 'Optativa'), ('Bioética', 'Optativa'), ('Neurociência Cognitiva', 'Optativa'), ('Psicologia do Desenvolvimento', 'Optativa'), ('Economia Comportamental', 'Optativa');

9 CONCLUSÃO

O presente trabalho buscou implementar um sistema de banco de dados de um Universidade que permita a ela, realizar o controle centralizado de alunos, professores, cursos, disciplinas, histórico escolar e turmas.

Em um primeiro momento, foi desenvolvido o Diagrama Entidade-Relacionamento a fim de identificar a Entidades e seus atributos, bem como os relacionamentos entre a Entidade. Depois, foi criado o Diagrama Lógico, bem como dicionário de dados do Sistema. Por fim, foi implementado o banco de dados em linguagem MySQL.