CESAR SCHOOL

CIÊNCIA DA COMPUTAÇÃO

DISCIPLINA PROBABILIDADE E ESTATÍSTICA

TURMA 4º B, SEMESTRE 2024.2

PROJETO DE MODELAGEM DE BANCO DE DADOS

DUVIDA ZERO

Yara Rodrigues Inácio

Recife,

Maio/2024

CESAR SCHOOL

ANÁLISE ESTATÍSTICA DA RELAÇÃO ENTRE O ORÇAMENTO E A CLASSIFICAÇÃO DE FILMES NA IMDb

Este relatório apresenta uma aplicação web feita em java utilizando o framework Quarkus. Essa aplicação visa realizar consultas no banco de dados MySQL criado e populado no DBeaver e modelado no brModelo.

Orientação: Professora Gabrielle Karine Canalle

Recife,

Maio/2024

1. MINI-MUNDO

Um cursinho particular deseja criar um banco de dados para gerenciar as informações de seus professores, alunos e aulas.

Cada professor tem um CPF, nome, telefone e email, e reside em endereços que incluem cidade, bairro, rua, número e apartamento. Os professores ensinam aulas para turmas compostas por múltiplos alunos, sendo cada turma identificada por um código.

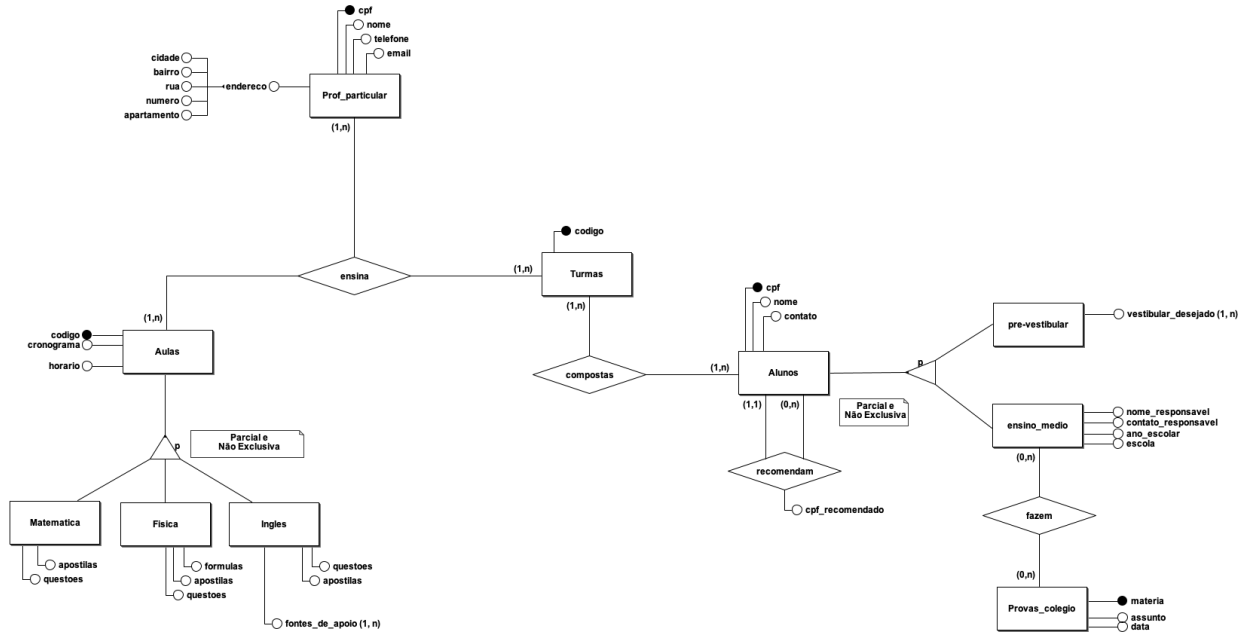
As aulas são identificadas por um código e cronograma, e têm um horário específico. Essas aulas podem ser de Matemática, Física ou Inglês. As aulas de Matemática têm apostilas e questões, as de Física têm apostilas, questões e fórmulas, e as de Inglês têm apostilas, questões e múltiplas fontes de apoio.

Cada aluno tem um CPF, nome e contato. Alunos podem participar de turmas, recomendar outros alunos e podem ser tanto de ensino médio quanto pré-vestibular ou ambos.

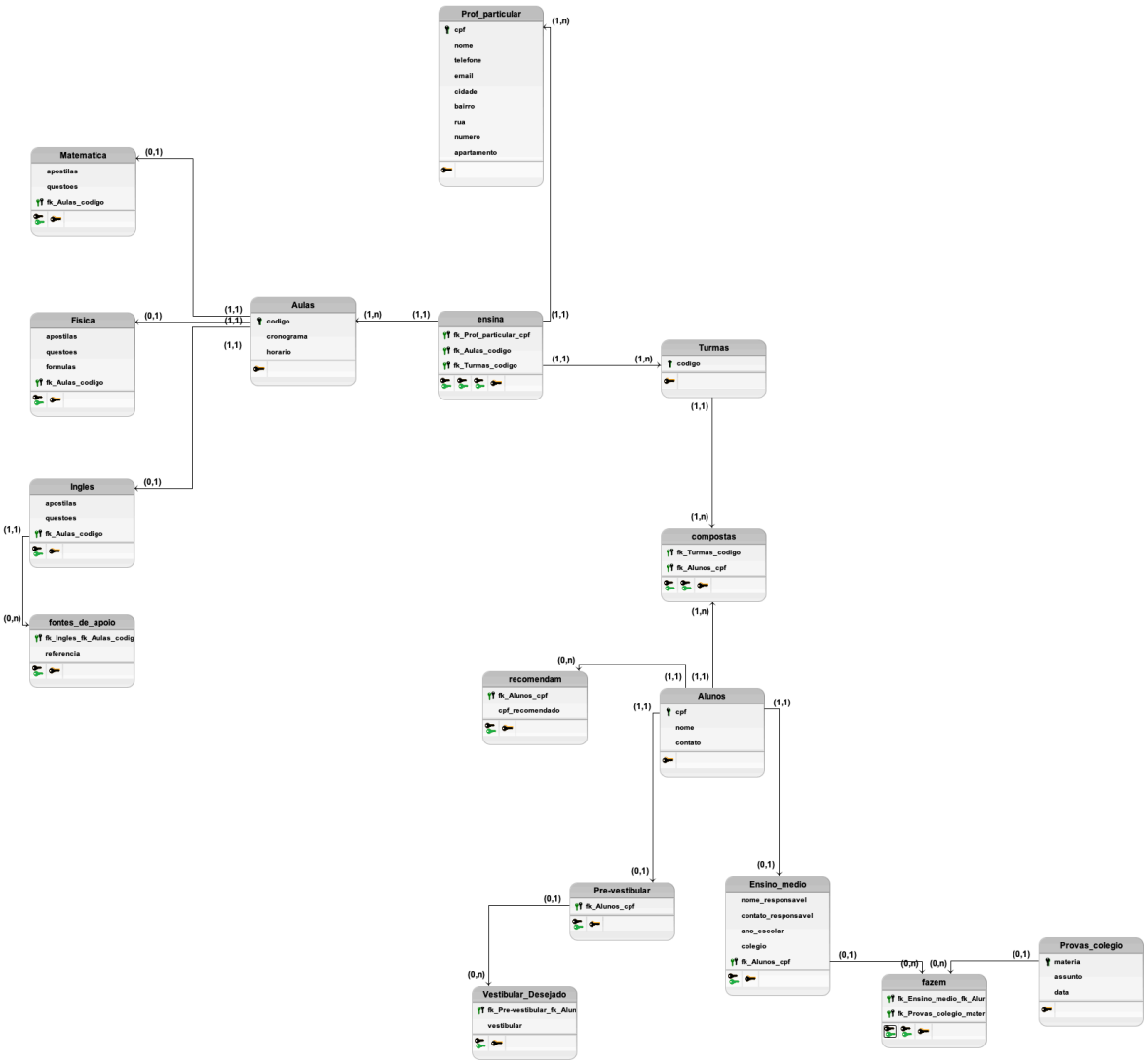
Alunos de ensino médio possuem informações adicionais como nome e contato do responsável, ano escolar e escola. Eles fazem provas no colégio, cujas matérias, assuntos e datas precisam ser registradas. Já os alunos pré-vestibular indicam quais são os vestibulares que desejam fazer.

Este sistema permitirá ao cursinho gerenciar de forma eficiente suas operações, facilitando a administração das aulas, professores e alunos, além de fornecer um suporte completo para as atividades educacionais.

2. MODELO CONCEITUAL



3. MODELO LÓGICO



4.SCRIPT DE CRIAÇÃO

É possível ver que, na criação do banco de dados a seguir, há aplicação de SQL avançado no uso da função Trigger.

```
CREATE DATABASE DuvidaZero;
CREATE TABLE Prof_particular (
  cpf VARCHAR(14) UNIQUE PRIMARY KEY,
  nome VARCHAR(50) NOT NULL,
  telefone VARCHAR(20),
  email VARCHAR(50),
  cidade VARCHAR(20),
  bairro VARCHAR(20),
  rua VARCHAR(20),
  numero int,
  apartamento VARCHAR(20)
);
CREATE TABLE Aulas (
  codigoA UNIQUE VARCHAR(20) UNIQUE,
  cronograma TEXT,
  horario VARCHAR(20)
);
CREATE TABLE Matematica (
  codigoA VARCHAR(20),
  apostilas TEXT,
  questoes TEXT,
     CONSTRAINT fk_Aulas_codigo_M FOREIGN KEY (codigoA) REFERENCES
Aulas(codigoA),
  PRIMARY KEY(codigoA)
);
CREATE TABLE Fisica (
```

```
  codigoA VARCHAR(20),
  formulas TEXT,
  apostilas TEXT,
  questoes TEXT,
     CONSTRAINT fk_Aulas_codigo_F FOREIGN KEY (codigoA) REFERENCES
Aulas(codigoA),
  PRIMARY KEY(codigoA)
);
CREATE TABLE Ingles (
  codigoA VARCHAR(20),
  apostilas TEXT,
  questoes TEXT,
     CONSTRAINT fk_Aulas_codigo_I FOREIGN KEY (codigoA) REFERENCES
Aulas(codigoA),
  PRIMARY KEY(codigoA)
);
CREATE TABLE fontes_de_apoio (
  fk_Ingles_fk_Aulas_codigo VARCHAR(20),
  referencia varchar(100),
                CONSTRAINT fk_Ingles_fk_Aulas_codigo FOREIGN KEY
(fk_Ingles_fk_Aulas_codigo) REFERENCES Ingles(codigoA),
  PRIMARY KEY(fk_Ingles_fk_Aulas_codigo)
);
CREATE TABLE Turmas (
  codigoT VARCHAR(20) UNIQUE
);
CREATE TABLE ensina (
  cpf_prof_particular VARCHAR(14),
  codigo_turma VARCHAR(20),
  codigo_aula VARCHAR(20),
```

```
        CONSTRAINT fk_prof_particular FOREIGN KEY (cpf_prof_particular)
REFERENCES Prof_particular(cpf),
        CONSTRAINT fk_turma FOREIGN KEY (codigo_turma) REFERENCES
Turmas(codigoT),
        CONSTRAINT fk_aula FOREIGN KEY (codigo_aula) REFERENCES
Aulas(codigoA)
);

DELIMITER //
CREATE TRIGGER backup_ensina
BEFORE DELETE ON ensina
FOR EACH ROW
BEGIN
  INSERT INTO ensina_backup (cpf_prof_particular, codigo_turma, codigo_aula)
  VALUES (OLD.cpf_prof_particular, OLD.codigo_turma, OLD.codigo_aula);
END;
//
DELIMITER ;

DROP trigger backup_prof_particular;
DROP trigger backup_ensina;
CREATE TABLE alunos (
  cpf VARCHAR(14) UNIQUE,
  nome VARCHAR(50) NOT NULL,
  contato VARCHAR(50)
);
CREATE TABLE composta (
  codigo_turma VARCHAR(20),
  cpf VARCHAR(14),
        CONSTRAINT fk_turma2 FOREIGN KEY (codigo_turma) REFERENCES
Turmas(codigoT),
```

```
  CONSTRAINT fk_aluno FOREIGN KEY (cpf) REFERENCES alunos(cpf)
);
CREATE TABLE recomendam (
  cpf VARCHAR(14),
  cpf_recomendado VARCHAR(14), -- Adicionando a definição da coluna
   CONSTRAINT Aluno_recomenda PRIMARY KEY(cpf), -- Movendo a definição da chave primária para cá
      CONSTRAINT FK_Aluno_recomenda FOREIGN KEY (cpf) REFERENCES alunos(cpf)
);
CREATE TABLE PreVestibular (
  id INT PRIMARY KEY AUTO_INCREMENT,
  cpf VARCHAR(14),
  CONSTRAINT fk_alunoPV FOREIGN KEY (cpf) REFERENCES alunos(cpf)
);
CREATE TABLE Vestibular_Desejado (
  fk_prevestibular INT,
   CONSTRAINT fk_prevestibular FOREIGN KEY (fk_prevestibular) REFERENCES PreVestibular(id),
  vestibular varchar(10)
);
CREATE TABLE Ensino_Medio(
  nome_responsavel VARCHAR(50) NOT NULL,
  contato_responsavel VARCHAR(50) NOT NULL,
  ano_escolar VARCHAR(10),
  colegio VARCHAR(50),
  cpf VARCHAR(14),
  CONSTRAINT pk_Ensino_Medio PRIMARY KEY (cpf),
  CONSTRAINT fk_alunoEM FOREIGN KEY (cpf) REFERENCES alunos(cpf)
);
```

5.SCRIPT DE POVOAMENTO

```
INSERT INTO Prof_particular (cpf, nome, telefone, email, cidade, bairro, rua, numero, apartamento)
VALUES
('059.843.524-78', 'Yara Faran', '(81) 995594885', 'yri@cesar.school', 'Recife', 'Boa Viagem', 'rua Setúbal', 1314, 'Apto 101'),
('123.456.789-01', 'Pedro Alves', '(81) 987654321', 'pedro.alves@email.com', 'Recife', 'Pina', 'Rua da Praia', 123, 'Apto 202'),
('987.654.321-09', 'Carla Lima', '(81) 999998888', 'carla.lima@email.com', 'Olinda', 'Bairro Novo', 'Rua dos Coqueiros', 456, 'Apto 303'),
('222.333.444-33', 'Marcos Santos', '(81) 987654321', 'marcos.santos@email.com', 'Recife', 'Boa Viagem', 'Rua dos Girassóis', 789, 'Apto 404'),
('333.444.555-44', 'Larissa Oliveira', '(81) 999999999', 'larissa.oliveira@email.com', 'Olinda', 'Cidade Tabajara', 'Avenida Central', 101, 'Apto 202');
INSERT INTO Aulas (codigoA, cronograma, horario)
VALUES
('MAT001', 'Semana 1 ...; Semana 2...; Semana 3...', 'Segunda 15h-16h'),
('MAT002', 'Semana 1 ...; Semana 2...; Semana 3...', 'Quarta, 16h-17h'),
('ING001', 'Semana 1 ...; Semana 2...; Semana 3...', 'Sábado, 10h-12h'),
('FIS001', 'Semana 1 ...; Semana 2...; Semana 3...', 'Quinta, 14h-15h'),
('MAT003', 'Semana 1 ...; Semana 2...; Semana 3...', 'Terça, 14h-16h'),
('ING002', 'Semana 1 ...; Semana 2...; Semana 3...', 'Sábado, 13h-15h'),
('FIS002', 'Semana 1 ...; Semana 2...; Semana 3...', 'Segunda, 18h-20h'),
('ING003', 'Semana 1 ...; Semana 2...; Semana 3...', 'Quinta, 8h-10h'),
('MAT004', 'Semana 1 ...; Semana 2...; Semana 3...', 'Quarta, 14h-16h'),
('FIS003', 'Semana 1 ...; Semana 2...; Semana 3...', 'Sexta, 9h-11h');
INSERT INTO Matematica (codigoA, apostilas, questoes)
VALUES
('MAT001', 'Apostilas de Matemática 1', 'Questões...'),
('MAT002', 'Apostilas de Matemática 2', 'Questões ...'),
```

```
('MAT003', 'Apostilas de Matemática 3', 'Questões ...'),
('MAT004', 'Apostilas de Matemática 4', 'Questões ...');
INSERT INTO Fisica (codigoA, formulas, apostilas, questoes)
VALUES
('FIS001', 'Fórmulas de Física...', 'Apostilas de Física...', 'Questões...'),
('FIS002', 'Fórmulas de Física...', 'Apostilas de Física...', 'Questões...'),
('FIS003', 'Fórmulas de Física...', 'Apostilas de Física...', 'Questões...');
INSERT INTO Ingles (codigoA, apostilas, questoes)
VALUES
('ING001', 'Apostilas de Inglês...', 'Questões...'),
('ING002', 'Apostilas de Inglês...', 'Questões...'),
('ING003', 'Apostilas de Inglês...', 'Questões...');
INSERT INTO fontes_de_apoio (fk_Ingles_fk_Aulas_codigo, referencia)
VALUES
('ING001', 'pt.duolingo.com'),
('ING002', 'www.todamateria.com.br/verbos-regulares-e-irregulares-no-ingles/'),
('ING003', 'dictionary.cambridge.org/pt/dicionario/ingles-portugues/');
INSERT INTO Turmas (codigoT)
VALUES ('TURMA001'), ('TURMA002'), ('TURMA003'), ('TURMA004'), ('TURMA005'),
('TURMA006');
INSERT INTO ensina (cpf_prof_particular, codigo_turma, codigo_aula)
VALUES
('059.843.524-78', 'TURMA001', 'MAT001'),
('059.843.524-78', 'TURMA002', 'FIS001'),
('123.456.789-01', 'TURMA003', 'ING001'),
('987.654.321-09', 'TURMA004', 'MAT002'),
('222.333.444-33', 'TURMA005', 'FIS002'),
('333.444.555-44', 'TURMA006', 'ING002');
INSERT INTO alunos (cpf, nome, contato)
VALUES
('111.222.333-00', 'Rebeka Albuquerque', '(81) 99842-0957'),
```

```
('444.555.666-00', 'João Silva', '(81) 99999-9999'),
('222.333.444-11', 'Lucas Xavier', '(81) 98765-4321'),
('555.666.777-22', 'Ana Costa', '(81) 98888-7777'),
('666.777.888-99', 'Paulo Pereira', '(81) 987654321'),
('777.888.999-00', 'Amanda Souza', '(81) 999999999');
INSERT INTO composta (codigo_turma, cpf)
VALUES
('TURMA001', '111.222.333-00'),
('TURMA002', '444.555.666-00'),
('TURMA003', '222.333.444-11'),
('TURMA004', '555.666.777-22'),
('TURMA005', '666.777.888-99'),
('TURMA006', '777.888.999-00');
INSERT INTO recomendam (cpf, cpf_recomendado)
VALUES
('111.222.333-00', '444.555.666-00'),
('222.333.444-11', '555.666.777-22'),
('666.777.888-99', '777.888.999-00');
INSERT INTO PreVestibular (cpf)
VALUES
('111.222.333-00'), ('444.555.666-00'),
('222.333.444-11'), ('555.666.777-22'),
('666.777.888-99'), ('777.888.999-00');
INSERT INTO Vestibular_Desejado (fk_prevestibular, vestibular)
VALUES
(1, 'ENEM'),
(2, 'AFA'),
(3, 'USP'),
(4, 'EEAR'),
(5, 'ENEM'),
(6, 'ENEM');
```

```
INSERT INTO Ensino_Medio (nome_responsavel, contato_responsavel, ano_escolar, colegio, cpf)
VALUES
('Maria Oliveira', '(81) 5555-5555', '2º ano', 'Colégio Santa Maria', '111.222.333-00'),
('José Santos', '(81) 4444-4444', '3º ano', 'Colégio Boa Viagem', '444.555.666-00'),
('Lucas Xavier', '(81) 98765-4321', '2º ano', 'Cognitivo', '222.333.444-11'),
('Ana Costa', '(81) 98888-7777', '3º ano', 'Damas', '555.666.777-22'),
('Fernanda Silva', '(81) 5555-5555', '2º ano', 'Motivo', '666.777.888-99'),
('Ricardo Almeida', '(81) 4444-4444', '3º ano', 'Colégio Santos Dummont', '777.888.999-00');
```

6.CONSULTAS SQL

As consultas a seguir atendem os requisitos de inserção, deleção, alteração e geração de relatórios, onde abordam conceitos como subconsulta, join, left join, group by, count.

```
SELECT Turmas.codigoT, Aulas.codigoA, Prof_particular.cpf,
Prof_particular.nome, COUNT(composta.codigo_turma) AS alunosporturma
FROM Turmas
JOIN ensina ON Turmas.codigoT = ensina.codigo_turma
JOIN Aulas ON ensina.codigo_aula = Aulas.codigoA
JOIN composta ON Turmas.codigoT = composta.codigo_turma
JOIN Prof_particular ON ensina.cpf_prof_particular = Prof_particular.cpf
WHERE Turmas.codigoT = ?
                OR Aulas.codigoA IN (SELECT codigoA FROM Aulas WHERE
Aulas.codigoA = ?)
GROUP BY Turmas.codigoT, Aulas.codigoA, Prof_particular.cpf,
Prof_particular.nome;
```

```
SELECT Prof_particular.*, COUNT(ensina.cpf_prof_particular) AS turmasporprofessor
FROM Prof_particular
LEFT JOIN ensina ON Prof_particular.cpf = ensina.cpf_prof_particular
WHERE Prof_particular.cpf = ?
GROUP BY Prof_particular.cpf;

INSERT INTO Prof_particular (cpf, nome, telefone, email, cidade, bairro, rua, numero,
apartamento) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);

UPDATE Prof_particular SET nome = ?, telefone = ? WHERE cpf = ?;

DELETE FROM ensina WHERE cpf_prof_particular = ?;
DELETE FROM Prof_particular WHERE cpf = ?;
```

```
SELECT *
FROM Aulas
WHERE codigoA = ?;

INSERT INTO Aulas (codigoA, cronograma, horario) VALUES (?, ?, ?);

UPDATE Aulas SET cronograma = ?, horario = ? WHERE codigoA = ?;

DELETE FROM Aulas WHERE codigoA=?;
```

```sql
SELECT alunos.*,
                Ensino_Medio.nome_responsavel, Ensino_Medio.contato_responsavel,
Ensino_Medio.ano_escolar, Ensino_Medio.colegio,
                VESTIBULAR_DESEJADO.vestibular, composta.codigo_turma
FROM alunos
LEFT JOIN PreVestibular ON alunos.cpf = PreVestibular.cpf
LEFT JOIN Vestibular_Desejado ON PreVestibular.id =
Vestibular_Desejado.fk_prevestibular
LEFT JOIN Ensino_Medio ON alunos.cpf = Ensino_Medio.cpf
LEFT JOIN composta ON composta.cpf = alunos.cpf
WHERE alunos.cpf =?;

INSERT INTO alunos (cpf, nome, contato) VALUES (?, ?, ?);

UPDATE alunos SET nome = ?, contato = ? WHERE cpf = ?;

DELETE FROM Ensino_medio WHERE cpf=?;
DELETE FROM PreVestibular WHERE cpf=?;
DELETE FROM alunos WHERE cpf=?
```

7.REQUISITOS

Ter no mínimo 8 entidades:

1. Prof_particular
2. Aulas
3. Matematica
4. Fisica
5. Ingles
6. Turmas
7. Alunos
8. pre-vestibular
9. ensino_medio
10. Provas_colegio

atributo composto:

endereco (Prof_particular)

atributo multivalorado:

fontes_de_apoio(1,n)

vestibular_desejado(1,n)

cardinalidades diversas:

(1, n); (1, 1); (0, n); (0, 1)

atributo de relacionamento:

ensina

compostas

recomendam

fazem

auto-relacionamento:

recomendam

agregação ou ternário:

Prof_particular ensina Aulas para Turmas

entidade fraca:

Provas_colegio

herança:

Matematica, Fisica, Ingles (Aulas)

pre-vestibular, ensino_medio (Alunos)

Conceitos:

```
CREATE TABLE Prof_particular (
  cpf VARCHAR(14) UNIQUE PRIMARY KEY,

CREATE TABLE Aulas (
  codigoA VARCHAR(20) UNIQUE,

  CREATE TABLE Turmas (
  codigoT VARCHAR(20) UNIQUE
);
 CREATE TABLE alunos (
        cpf VARCHAR(14) UNIQUE,
```

8.ENTREVISTA COM O CLIENTE

Yara (Desenvolvedora): Bom dia! Gostaríamos de entender melhor as necessidades do seu cursinho particular para criar um banco de dados eficiente. Pode nos falar um pouco sobre o que vocês precisam gerenciar?

Yara (Professora Particular): Bom dia! Claro, precisamos de um sistema para gerenciar informações de professores, alunos e aulas.

Yara (Desenvolvedora): Perfeito. Poderia nos detalhar as informações necessárias sobre os professores?

Yara (Professora Particular): Cada professor deve ter registrado seu CPF, nome, telefone, email e endereço completo, incluindo cidade, bairro, rua, número e apartamento.

Yara (Desenvolvedora): Entendi. E quanto às aulas?

Yara (Professora Particular): As aulas devem ser identificadas por um código, cronograma e horário. Podem ser de Matemática, Física ou Inglês. Para Matemática, precisamos registrar apostilas e questões. Física requer apostilas, questões e fórmulas. Inglês precisa de apostilas, questões e várias fontes de apoio.

Yara (Desenvolvedora): E sobre as turmas e alunos?

Yara (Professora Particular): Cada turma tem um código e é composta por vários alunos. Os alunos têm CPF, nome e contato. Eles podem participar de turmas e recomendar outros alunos. Precisamos distinguir entre alunos de ensino médio e pré-vestibular. Alunos de ensino médio precisam de informações adicionais como nome e contato do responsável, ano escolar e escola, além de registrarmos as provas realizadas no colégio com matéria, assunto e data. Alunos pré-vestibular indicam os vestibulares desejados.

Yara (Desenvolvedora): Isso ajudará a organizar melhor as informações. Mais alguma necessidade específica?

Yara (Professora Particular): Sim, além do básico, precisamos de funcionalidades específicas para consultas e manipulação dos dados. Por exemplo, queremos ver a lista de turmas, aulas e professores, bem como a contagem de alunos por turma. Também precisamos de relatórios sobre os professores, como a quantidade de turmas que cada professor ensina. Funções para adicionar, atualizar e remover professores, aulas e alunos também são essenciais.

Yara (Desenvolvedora): Excelente. Isso nos ajuda a entender melhor suas necessidades. Vamos trabalhar nesse projeto para garantir que todas essas funcionalidades estejam integradas no banco de dados. Muito obrigado pela sua colaboração!

Yara (Professora Particular): Obrigado! Estou ansioso para ver o resultado final.

9.TRABALHOS FUTUROS

Devido ao prazo limitado e imprevistos durante a implementação não foi possível implementar as seguintes funcionalidades desejadas:

- Consultar e adicionar recomendações de alunos;
- Consultar, adicionar e editar questões, apostilas, fórmulas e fontes de apoio;
- Consultar quantos alunos foram recomendados por outros;
- Consultar quantos alunos de ensino médio estudam no cursinho Dúvida Zero de cada escola;
- Consultar quais os vestibulares mais desejados;

Planejo, futuramente, adicionar essas funcionalidades na aplicação pois prevejo sua utilidade ao cliente.

10. REFERÊNCIAS

Yara-R. DuvidaZero. 2024. Disponível em: https://github.com/Yara-R/DuvidaZero.git.

Acesso em: 25 Abril 2024.