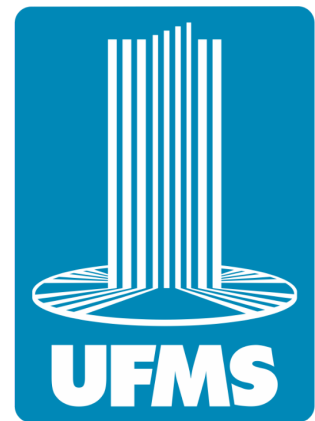


**UNIVERSIDADE FEDERAL  
DE MATO GROSSO DO SUL**



Laboratório de Banco de Dados

## SISTEMA DE GERENCIAMENTO PARA SALÃO DE BELEZA

Eduardo Alves de Oliveira Freitas

## 1. Descrição do cenário do problema

Em um bairro movimentado da cidade, encontra-se o "Beleza Radiante", um salão de beleza próspero e bem conceituado, conhecido por seu ambiente acolhedor e serviços de alta qualidade. Com uma clientela diversificada e em expansão, o salão enfrenta desafios típicos de um negócio em crescimento, como a necessidade de gerenciar eficientemente os agendamentos, manter um registro preciso dos serviços prestados e gerir o estoque de produtos de beleza.

O sistema de gerenciamento do salão mantém um registro detalhado de cada cliente, incluindo informações pessoais e preferências específicas. Isso permite que a equipe ofereça um serviço personalizado e atencioso, essencial para manter a satisfação e a fidelidade dos clientes. As informações sobre alergias são particularmente importantes, garantindo a segurança e o bem-estar dos clientes durante os tratamentos.

O salão emprega uma variedade de profissionais, incluindo cabeleireiros, manicures, esteticistas e terapeutas de beleza. O sistema armazena informações completas sobre cada funcionário, incluindo qualificações, experiência e horários de trabalho. Isso facilita a organização das escalas de trabalho, a gestão de folgas e férias e a alocação eficiente de recursos humanos.

O sucesso do salão depende não apenas da habilidade de seus profissionais, mas também da qualidade dos produtos utilizados. O sistema de gerenciamento monitora de perto o estoque de produtos de beleza, registrando cada item utilizado. Isso não só ajuda a prevenir a escassez de produtos essenciais, mas também auxilia na previsão e no planejamento de compras futuras.

Um dos aspectos mais críticos do funcionamento do salão é o agendamento de serviços. Os clientes podem agendar horários por telefone ou pessoalmente, e a secretária, com a ajuda do sistema, organiza a agenda de forma eficaz. O sistema permite ajustar os horários conforme a disponibilidade dos funcionários e as preferências dos clientes, minimizando conflitos de agendamento e tempo de inatividade.

Cada atendimento é cuidadosamente registrado no sistema, formando um histórico detalhado para cada cliente. Isso inclui o tipo de serviço prestado, os produtos utilizados e notas sobre as preferências ou comentários do cliente. Esse histórico é inestimável para fornecer serviços personalizados e para entender as tendências e as necessidades dos clientes ao longo do tempo.

O "Beleza Radiante" oferece uma vasta gama de serviços, cada um com sua própria estrutura de preços. O sistema ajuda a gerenciar essa complexidade, assegurando que os preços sejam atualizados e que os custos dos serviços sejam claramente comunicados aos clientes. Isso inclui serviços como cortes de cabelo, tratamentos de coloração, manicure e pedicure, design de sobrancelhas, maquiagem, massagens e outros tratamentos de beleza.

Ao concluir os serviços, os clientes são apresentados com várias opções de pagamento, facilitadas pelo sistema. Ele calcula automaticamente o total a ser pago, integrando os custos de todos os serviços e produtos utilizados. Além disso, o sistema gera relatórios financeiros detalhados, permitindo que a gerência do salão monitore o desempenho do negócio, identifique tendências de vendas e faça ajustes operacionais conforme necessário.

## 2. Esquema Conceitual (Entidade-Relacionamento)

Apresentamos o Diagrama de Entidade-Relacionamento (DER), uma representação gráfica que detalha as estruturas de dados e as relações entre as entidades no sistema de gerenciamento do salão de beleza, fornecendo uma visão clara da organização e do fluxo de informações dentro do banco de dados.

### Legenda do DER:



Chave Primária



Chave Estrangeira



- Campo Obrigatório.



- Campo Não Obrigatório.

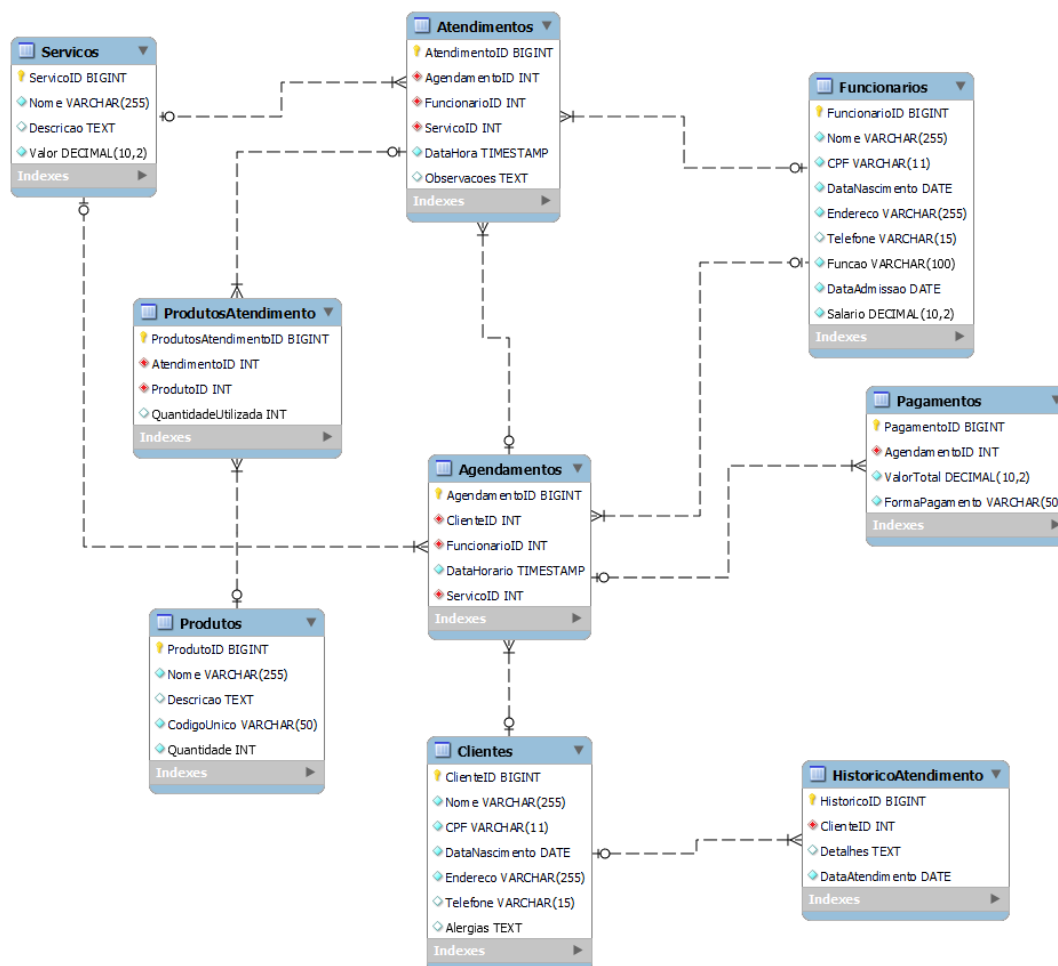


imagem 1: Diagrama entidade relacionamento(acesse online com melhor qualidade em: <https://ibb.co/NngSpBs>)

## 2.1.Relacionamentos

Agendamentos ↔ Clientes

- Um Cliente pode ter vários Agendamentos.
- Um Agendamento está associado a um único Cliente.

Agendamentos ↔ Funcionários

- Um Funcionário pode estar associado a vários Agendamentos.
- Um Agendamento é atendido por um único Funcionário.

Agendamentos ↔ Serviços

- Um Serviço pode ser oferecido em vários Agendamentos.
- Um Agendamento envolve um único Serviço.

Agendamentos ↔ Atendimentos

- Um agendamento pode levar a um atendimento.
- Um atendimento está associado a um agendamento.

## 3. Esquema Lógico (Modelo Relacional)

Clientes

- ClienteID (Chave Primária, serial)
- Nome (VARCHAR(255))
- CPF (VARCHAR(11), Único)
- DataNascimento (DATE)
- Endereco (VARCHAR(255))
- Telefone (VARCHAR(15))
- Alergias (TEXT)

Funcionarios

- FuncionarioID (Chave Primária, serial)
- Nome (VARCHAR(255))
- CPF (VARCHAR(11), Único)
- DataNascimento (DATE)
- Endereco (VARCHAR(255))
- Telefone (VARCHAR(15))
- Funcao (VARCHAR(100))
- DataAdmissao (DATE)
- Salario (DECIMAL(10, 2))

Produtos

- ProdutoID (Chave Primária, serial)
- Nome (VARCHAR(255))
- Descricao (TEXT)

- CodigoUnico (VARCHAR(50), Único)
- Quantidade (INT)
- PrecoUnitario (DECIMAL(10, 2))

#### Servicos

- ServicoID (Chave Primária, serial)
- Nome (VARCHAR(255))
- Descricao (TEXT)
- Valor (DECIMAL(10, 2))

#### Agendamentos

- AgendamentoID (Chave Primária, serial)
- ClienteID (INT, Chave Estrangeira para Clientes)
- FuncionarioID (INT, Chave Estrangeira para Funcionarios)
- DataHorario (TIMESTAMP)
- ServicoID (INT, Chave Estrangeira para Servicos)
- AtendimentoID (INT, Chave Estrangeira para Atendimentos)

#### HistoricoAtendimento

- HistoricoID (Chave Primária, serial)
- ClienteID (INT, Chave Estrangeira para Clientes)
- Detalhes (TEXT)
- DataAtendimento (DATE)

#### Pagamentos

- PagamentoID (Chave Primária, serial)
- AgendamentoID (INT, Chave Estrangeira para Agendamentos)
- ValorTotal (DECIMAL(10, 2))
- FormaPagamento (VARCHAR(50))

#### Atendimentos

- AtendimentoID (Chave Primária, serial)
- AgendamentoID (INT, Chave Estrangeira para Agendamentos)
- FuncionarioID (INT, Chave Estrangeira para Funcionarios)
- ServicoID (INT, Chave Estrangeira para Servicos)
- DataHora (TIMESTAMP)
- Observacoes (TEXT)
- idPagamento (INT, Chave Estrangeira para Pagamentos)

#### ProdutosAtendimento

- ProdutosAtendimentoID (Chave Primária, serial)
- AtendimentoID (INT, Chave Estrangeira para Atendimentos)
- ProdutoID (INT, Chave Estrangeira para Produtos)
- QuantidadeUtilizada (INT)

## 4. Domínio dos Atributos

Cliente:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
ClienteID	Serial	Requerido	Sim	-
Nome	VARCHAR(255)	Requerido	-	-
CPF	VARCHAR(11)	Requerido	-	-
DataNascimento	DATE	Requerido	-	-
Endereco	VARCHAR(255)	Requerido	-	-
Telefone	VARCHAR(15)	-	-	-

Funcionário:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
FuncionarioID	Serial	Requerido	Sim	-
Nome	VARCHAR(255)	Requerido	-	-
CPF	VARCHAR(11)	Requerido	-	-
DataNascimento	DATE	Requerido	-	-
Endereco	VARCHAR(255)	Requerido	-	-
Telefone	VARCHAR(15)	-	-	-
Funcao	VARCHAR(100)	Requerido	-	-
DataAdmissao	DATE	Requerido	-	-
Salario	DECIMAL	Requerido	-	-

Produtos:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
ProdutoID	Serial	Requerido	Sim	-
Nome	VARCHAR(255)	Requerido	-	-
Descricao	TEXT	-	-	-
CodigoUnico	VARCHAR(50)	Requerido	-	-
Quantidade	INT	Requerido	-	-
PrecoUnitario	DECIMAL(10,2)	Requerido	-	-

Serviços:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
ServicoID	Serial	Requerido	Sim	-
Nome	VARCHAR(255)	Requerido	-	-
Descricao	TEXT	-	-	-
Valor	DECIMAL(10,2)	Requerido	-	-

Agendamentos:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
AgendamentoID	Serial	Requerido	Sim	-
ClienteID	INT	FOREIGN KEY	-	Sim
FuncionarioID	INT	FOREIGN KEY	-	Sim
DataHorario	TIME	Requerido	-	-
ServicoID	INT	FOREIGN KEY	-	Sim
AtendimentoID	INT	FOREIGN KEY	-	Sim



Historico Atendimento:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
HistoricoID	Serial	Requerido	Sim	-
ClienteID	INT	FOREIGN KEY	-	Sim
Detalhes	TEXT	-	-	-
DataAtendimento	TIME	Requerido	-	-

Pagamentos:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
AgendamentoID	INT	FOREIGN KEY	-	SIM
PagamentoID	INT	Requerido	Sim	-
ValorTotal	DECIMAL(10,2)	Required	-	-
FormaPagamento	VARCHAR(50)	Requerido	-	-

Atendimentos:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
AtendimentoID	Serial	Required	Sim	-
AgendamentoID	INT	FOREIGN KEY	-	Sim
FuncionarioID	INT	FOREIGN KEY	-	Sim
ServicoID	INT	FOREIGN KEY	-	Sim
DataHora	TIME	REQUIRED	-	-
Observacao	Text	-	-	-
Idpagamento	INT	FOREIGN KEY	-	Sim

Produtos Atendimento:

ATRIBUTO	TIPO DE DADO	RESTRIÇÃO CAMPOS	CHAVE ENTIDADE	CHAVE REFERENCIAL
ProdutosAtendimentoID	Serial	Required	Sim	-
AtendimentoID	INT	FOREIGN KEY	-	Sim
ProdutoID	INT	FOREIGN KEY	-	Sim
QuantidadeUtilizada	INT	Requerido	-	-

## 5. Script de Criação do Banco de Dados - DDL

```
CREATE TABLE Clientes (  
    ClienteID serial PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    CPF VARCHAR(11) UNIQUE NOT NULL,  
    DataNascimento DATE NOT NULL,  
    Endereco VARCHAR(255) NOT NULL,  
    Telefone VARCHAR(15),  
    Alergias TEXT  
);
```

```
CREATE TABLE Funcionarios (  
    FuncionarioID serial PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    CPF VARCHAR(11) UNIQUE NOT NULL,  
    DataNascimento DATE NOT NULL,  
    Endereco VARCHAR(255) NOT NULL,  
    Telefone VARCHAR(15),  
    Funcao VARCHAR(100) NOT NULL,  
    DataAdmissao DATE NOT NULL,  
    Salario DECIMAL(10, 2) NOT NULL  
);
```

```
CREATE TABLE Produtos (  
    ProdutoID serial PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    Descricao TEXT,  
    CodigoUnico VARCHAR(50) UNIQUE NOT NULL,
```

```
Quantidade INT NOT NULL  
);
```

```
ALTER TABLE Produtos  
ADD COLUMN PrecoUnitario DECIMAL(10, 2);
```

```
CREATE TABLE Servicos (  
    ServicoID serial PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    Descricao TEXT,  
    Valor DECIMAL(10, 2) NOT NULL  
);
```

```
CREATE TABLE Agendamentos (  
    AgendamentoID serial PRIMARY KEY,  
    ClienteID INT REFERENCES Clientes(ClienteID),  
    FuncionarioID INT REFERENCES Funcionarios(FuncionarioID),  
    DataHorario TIMESTAMP NOT NULL,  
    ServicoID INT REFERENCES Servicos(ServicoID)  
);  
ALTER TABLE Agendamentos  
ADD COLUMN AtendimentoID INT REFERENCES Atendimentos(AtendimentoID);
```

```
CREATE TABLE HistoricoAtendimento (  
    HistoricoID serial PRIMARY KEY,  
    ClienteID INT REFERENCES Clientes(ClienteID),  
    Detalhes TEXT,  
    DataAtendimento DATE NOT NULL  
);
```

```
CREATE TABLE Pagamentos (  
    PagamentoID serial PRIMARY KEY,  
    AgendamentoID INT REFERENCES Agendamentos(AgendamentoID),  
    ValorTotal DECIMAL(10, 2) NOT NULL,  
    FormaPagamento VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Atendimentos (  
    AtendimentoID serial PRIMARY KEY,  
    AgendamentoID INT REFERENCES Agendamentos(AgendamentoID),
```

```

FuncionarioID INT REFERENCES Funcionarios(FuncionarioID),
ServicoID INT REFERENCES Servicos(ServicoID),
DataHora TIMESTAMP NOT NULL,
Observacoes TEXT
);

ALTER TABLE Atendimentos
ADD COLUMN idPagamento INT REFERENCES Pagamentos(PagamentoID);

CREATE TABLE ProdutosAtendimento (
    ProdutosAtendimentoID serial PRIMARY KEY,
    AtendimentoID INT REFERENCES Atendimentos(AtendimentoID),
    ProdutoID INT REFERENCES Produtos(ProdutoID),
    QuantidadeUtilizada INT
);

```

## 6. Script de Cadastros do Banco de Dados

### –Clientes:

```

INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Natália Duarte', '36174925099', '2001-03-04', 'Recanto de Rodrigues, 8,
Madri, 47244525 Barbosa de Aragão / PE', '09004899277', 'quidem');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Valentina Lima', '07532496180', '1955-07-14', 'Lagoa Joaquim Castro, 19,
Conjunto Jatoba, 12990578 Silva da Mata / PI', '2166767863', 'praesentium');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Emanuel Teixeira', '82079316559', '1985-03-26', 'Alameda Rezende, 87,
Vila Antena Montanhês, 62680-156 da Luz de Cunha / PR', '8426197710', 'vel');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Vitor Hugo Porto', '93218760496', '1976-10-27', 'Praia Bernardo Costela,
7, Ouro Minas, 39223048 Viana da Praia / TO', '552180357991', 'id');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Brenda Caldeira', '79568412085', '1968-03-30', 'Residencial Giovanna
Rezende, 55, Flavio De Oliveira, 50734-539 Nunes / CE', '4185327680', 'ipsam');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Igor da Mata', '42905761849', '1977-09-16', 'Campo Mendes, 74, Madri,
54273596 Azevedo / RS', '5006133929', 'in');

```

```

INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Kamilly Sales', '89510734241', '1993-08-02', 'Parque Gabriel Rodrigues,
93, Maria Tereza, 44787694 Santos do Galho / AC', '554184603234', 'ipsam');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Ana Júlia da Rosa', '95817032414', '2002-07-24', 'Rua Kaique Aragão,
405, Vila Nova Gameleira 3ª Seção, 58350285 Ribeiro dos Dourados / RN',
'5508421287792', 'repellendus');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Eduardo Barros', '15629487011', '1992-11-15', 'Vila de Castro, 65,
Palmares, 73770301 Ribeiro Grande / MT', '11946792830', 'lactose');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Laura Nogueira', '47921586032', '1987-06-22', 'Travessa Farias, 403,
Jardim Laguna, 52073291 Cavalcanti / AL', '21938294718', 'glúten');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Fábio da Luz', '39847102549', '1979-08-30', 'Alameda Souza, 12, Jardim
Leblon, 79925381 Nunes / RN', '84926483107', 'poeira');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Alice Castro', '69024817539', '2000-05-19', 'Loteamento Figueiredo, 77,
Jaqueline, 20274468 Carvalho / SE', '79938217460', 'pólen');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Lucas Silva', '31294705860', '1974-01-28', 'Vila de Almeida, 94, Jardim
Felicidade, 50710485 Teixeira / PE', '81947382016', 'amendoim');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Pedro Lucas Pires', '43275680900', '1946-03-11', 'Recanto Monteiro, 48,
Vila Boa Vista, 38607-302 Teixeira / RS', '08400400260', 'maiores');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Bianca Moura', '52734896109', '1996-01-03', 'Conjunto Freitas, 32,
Buraco Quente, 67949-195 Costela / ES', '04165917342', 'quae');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Maitê Rezende', '59476302810', '1979-06-29', 'Vila de Araújo, 7, Vila
Copasa, 09902238 Barros de Almeida / AM', '2140338728', 'voluptatibus');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Sr. Erick Rezende', '81974352005', '1962-02-12', 'Esplanada de da Rosa,
40, Santa Tereza, 34428-130 Souza de Aragão / PB', '05002464916', 'neque');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Giovanna Souza', '61948753057', '1991-05-08', 'Alameda Alana Nogueira,
46, São Luiz, 64074-128 Cunha / SC', '01111655588', 'maxime');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Sr. Cauê Moura', '31490625780', '1968-08-13', 'Passarela Ana Beatriz da
Cunha, 8, Vila Cloris, 00928-860 Barros / MG', '02126583110', 'eum');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('João Pedro Monteiro', '28179540685', '1990-10-11', 'Esplanada de
Moreira, 227, Glória, 84147-639 Souza / PR', '6173169022', 'ipsum');

```

```

INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Eduardo Moraes', '70568913493', '1948-09-10', 'Travessa Dias, Garças,
90127553 Peixoto dos Dourados / AC', '03189257093', 'odit');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Kamilly Teixeira', '84597130241', '1947-07-08', 'Pátio Teixeira, 891, Vila
Engenho Nogueira, 16770396 Moreira / AM', '02107207479', 'nam');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Sabrina Moraes', '81473506948', '1979-11-17', 'Ladeira Melissa Duarte,
546, Mala E Cuia, 55027-901 Pereira da Praia / RO', '8474965060', 'accusantium');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Henrique Ribeiro', '36501748208', '1961-01-03', 'Alameda de Monteiro,
Gutierrez, 94308-581 Caldeira / MG', '551141097933', 'repellat');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Srta. Bruna Correia', '42095137841', '1973-08-08', 'Conjunto de
Gonçalves, 91, Lorena, 90253-721 da Luz do Norte / PA', '6167334735', 'dolore');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Igor Freitas', '38274169069', '1967-01-15', 'Viaduto Emanuely Araújo, 93,
Bonfim, 68543-306 Ribeiro de da Rosa / RO', '8000544370', 'qui');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Joana Rodrigues', '12849306533', '1944-07-10', 'Lago Farias, 4, Vila
Betânia, 88747634 Duarte de Minas / RN', '58112031672', 'necessitatibus');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Sra. Evelyn Rodrigues', '73209846103', '2003-12-18', 'Feira de Cavalcanti,
697, Vila Nova Gameleira 3ª Seção, 88324-961 das Neves / MT', '8001037839',
'pariatur');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Carlos Eduardo Carvalho', '86901324570', '1991-09-13', 'Vila Peixoto, 61,
Outro, 48562-598 Teixeira de Rocha / PE', '01176301157', 'tenetur');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('João Felipe da Costa', '10732859441', '1972-04-16', 'Praça da Paz, 942,
Vila Nova Gameleira 1ª Seção, 26809902 Almeida / MT', '1103686894', 'hic');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Srta. Isis Campos', '58039627168', '1953-03-04', 'Esplanada de Silva, 8,
Vila De Sá, 04501-308 Novaes das Pedras / DF', '05009650222', 'tempore');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Thales Cunha', '31498275664', '2002-04-22', 'Rodovia Cavalcanti, Serra
Verde, 43223702 Cardoso de Minas / AC', '7187820244', 'amet');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Luiz Henrique Viana', '43167502835', '1943-09-09', 'Rua Silva, 59, Cdi
Jatoba, 52783-227 da Rocha da Praia / SP', '8496588506', 'veritatis');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Ana Sophia Souza', '45936028747', '1996-07-02', 'Parque Raul Oliveira,
840, Vista Do Sol, 55136-131 Barbosa Alegre / PI', '3198906322', 'tempora');

```

```

INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Anthony Cavalcanti', '03968247574', '1963-08-20', 'Sítio Alana Nogueira,
48, São Damião, 61109-951 Costela de Araújo / RS', '05003970946', 'quas');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Sra. Helena Costa', '45812376026', '1997-04-27', 'Lagoa Joaquim da
Mata, 86, Horto Florestal, 10096-939 Lima de Gonçalves / PB', '4162720527',
'repudiandae');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Enzo Oliveira', '40672859149', '1989-02-01', 'Lago Farias, Santa Lúcia,
91441-798 Carvalho das Flores / AP', '7108632075', 'sequi');
INSERT INTO Clientes (Nome, CPF, DataNascimento, Endereco, Telefone, Alergias)
VALUES ('Natália da Cunha', '02591846324', '1978-04-03', 'Setor Azevedo, 17,
Jardim Leblon, 78176-889 Mendes / AM', '3121317715', 'amet');

```

### **Funcionários:**

```

INSERT INTO Funcionarios (Nome, CPF, DataNascimento, Endereco, Telefone,
Funcao, DataAdmissao, Salario) VALUES ('Eduardo', '76821594058', '2002-11-22',
'Praça Isabel Ferreira, 96, São João, 83890-595 da Cunha / RJ', '+5507143259897',
'Técnico em aviação', '2017-07-28', 4505.99);
INSERT INTO Funcionarios (Nome, CPF, DataNascimento, Endereco, Telefone,
Funcao, DataAdmissao, Salario) VALUES ('Igor', '71369452080', '1980-09-19',
'Loteamento Levi Mendes, 1, Vila Mantiqueira, 30830449 Nunes de Minas / SP',
'+5507178258038', 'Museólogo', '2016-11-24', 9628.54);
INSERT INTO Funcionarios (Nome, CPF, DataNascimento, Endereco, Telefone,
Funcao, DataAdmissao, Salario) VALUES ('Ivone', '40728361507', '1987-01-28',
'Aeroporto Maria Eduarda Ferreira, São João Batista, 68454610 Moraes da Serra /
RR', '+557197609762', 'Produtor editorial', '2018-09-24', 8189.4);
INSERT INTO Funcionarios (Nome, CPF, DataNascimento, Endereco, Telefone,
Funcao, DataAdmissao, Salario) VALUES ('Juliano', '67935248010', '1968-08-30',
'Passarela Lorena Cardoso, 7, Conjunto Lagoa, 75438751 Nascimento / SC',
'04148891028', 'Terapeuta Holístico', '2021-11-05', 4906.36);
INSERT INTO Funcionarios (Nome, CPF, DataNascimento, Endereco, Telefone,
Funcao, DataAdmissao, Salario) VALUES ('Belo', '50821947630', '1992-11-10',
'Esplanada Sofia das Neves, 60, Vila Madre Gertrudes 4ª Seção, 28118817 Barros
do Campo / MG', '03005416876', 'Cerimonialista', '2023-06-30', 8817.89);

```

### **Produtos:**

```

INSERT INTO Produtos (Nome, Descricao,CodigoUnico, Quantidade) VALUES
('Shampoo Hidratante', 'Shampoo para cabelos secos, 500ml', 'SH001', 50);

```

```

INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Condicionador Nutritivo', 'Condicionador para cabelos danificados, 500ml', 'CN002',
40);
INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Máscara Capilar', 'Máscara de tratamento intensivo, 300ml', 'MC003', 30);
INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Óleo de Argan', 'Óleo nutritivo para cabelos, 100ml', 'OA004', 25);
INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Creme de Pentear', 'Creme para desembaraçar e modelar cabelos, 250ml',
'CP005', 45);
INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Tintura Permanente', 'Tinta para cabelo permanente, diversas cores', 'TP006', 60);
INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Esmalte', 'Esmalte de longa duração, diversas cores', 'ES007', 100);
INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Removedor de Esmalte', 'Removedor de esmalte com aloe vera, 200ml', 'RE008',
20);
INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Cera Depilatória', 'Cera quente para depilação, 500g', 'CD009', 15);
INSERT INTO Produtos (Nome, Descricao, CodigoUnico, Quantidade) VALUES
('Gel Fixador', 'Gel fixador extra forte, 150ml', 'GF010', 35);

```

### **Serviços:**

```

INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Corte de Cabelo', 'Corte
de cabelo feminino ou masculino', 50.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Tintura', 'Tintura
completa para cabelo', 150.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Manicure', 'Manicure
completa', 30.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Pedicure', 'Pedicure
completa', 35.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Design de
Sobrancelhas', 'Design e modelagem de sobrancelhas', 40.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Maquiagem',
'Maquiagem completa para eventos', 120.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Depilação', 'Depilação
com cera em áreas específicas', 60.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Hidratação Capilar',
'Tratamento de hidratação profunda para cabelos', 70.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Escova Progressiva',
'Escova progressiva para alisamento', 200.0);
INSERT INTO Servicos (Nome, Descricao, Valor) VALUES ('Massagem Relaxante',
'Massagem corporal relaxante', 90.0);

```



## 7. Exemplos Updates e Deletes do Banco de Dados

### 7.1.Updates

#### **Atualizar o Endereço de um Cliente:**

*UPDATE Clientes*

*SET Endereco = 'Rua senhor Mouse e Teclado da Silva, numero 321'*

*WHERE ClienteID = 1;*

#### **Atualizar o Salário de um Funcionário:**

*UPDATE Funcionarios*

*SET Salario = Salario + 500*

*WHERE FuncionarioID = 2;*

#### **Atualizar a Quantidade de um Produto em Estoque:**

*UPDATE Produtos*

*SET Quantidade = Quantidade - 1*

*WHERE ProdutoID = 3;*

#### **Atualizar a Data de Agendamento:**

*UPDATE Agendamentos*

*SET DataHorario = '2023-11-20 14:00:00'*

*WHERE AgendamentoID = 4;*

### 7.2.Deletes

#### **Remover um Cliente:**

*DELETE FROM Clientes*

*WHERE ClienteID = 1;*

**Remover um Funcionário:**

```
DELETE FROM Funcionarios  
WHERE FuncionarioID = 2;
```

**Remover um Produto:**

```
DELETE FROM Produtos  
WHERE ProdutoID = 3;
```

**Cancelar um Agendamento:**

```
DELETE FROM Agendamentos  
WHERE AgendamentoID = 4;
```

## 8. Selects do Banco de Dados

**Listar Todos os Agendamentos para uma Data Específica**

```
SELECT *  
FROM Agendamentos  
WHERE DataHorario::date = '2023-11-15'  
order by datahorario
```

Resultado:

	agendamentoid [PK] integer	clienteid integer	funcionarioid integer	datahorario timestamp without time zone	servicoid integer	atendimentoid integer
1	3	1	2	2023-11-15 11:00:00	3	3
2	1	1	2	2023-11-15 15:00:00	3	5
3	2	1	2	2023-11-15 19:00:00	3	2

**Verificar a Agenda de um Funcionário em Particular para um Dia Específico:**

```
SELECT A.AgendamentoID, A.DataHorario, C.Nome AS Cliente, S.Nome AS  
Servico  
FROM Agendamentos A  
JOIN Clientes C ON A.ClienteID = C.ClienteID  
JOIN Servicos S ON A.ServicoID = S.ServicoID  
WHERE A.FuncionarioID = 2 AND A.DataHorario::date = '2023-11-15'  
order by datahorario
```

Resultado:

	agendamentoid integer	datahorario timestamp without time zone	cliente character varying (255)	servico character varying (255)
1	3	2023-11-15 11:00:00	Davi Gonçalves	Manicure
2	1	2023-11-15 15:00:00	Davi Gonçalves	Manicure
3	2	2023-11-15 19:00:00	Davi Gonçalves	Manicure

### Listar Atendimentos Realizados por um Funcionário Específico:

```
SELECT At.AtendimentoID, At.DataHora, C.Nome AS Cliente, S.Nome AS Servico,  
At.Observacoes  
FROM Atendimentos At  
JOIN Clientes C ON At.AgendamentoID = C.ClienteID  
JOIN Servicos S ON At.ServicoID = S.ServicoID  
WHERE At.FuncionarioID = 2;
```

Resultado:

	atendimentoid integer	datahora timestamp without time zone	cliente character varying (255)	servico character varying (255)	observacoes text
1	5	2023-11-15 15:00:00	Davi Gonçalves	Manicure	[null]
2	4	2023-11-15 15:00:00	Davi Gonçalves	Manicure	[null]
3	1	2023-11-15 15:00:00	Davi Gonçalves	Manicure	[null]
4	3	2023-11-15 11:00:00	Natália Duarte	Manicure	[null]

### Verificar Atendimentos Que Ainda Não Foram Pagos:

```
SELECT At.AtendimentoID, At.DataHora, C.Nome AS Cliente, S.Nome AS Servico  
FROM Atendimentos At  
JOIN Clientes C ON At.AgendamentoID = C.ClienteID  
JOIN Servicos S ON At.ServicoID = S.ServicoID  
WHERE At.idPagamento IS NULL;
```

Resultado:

	atendimentoid integer	datahora timestamp without time zone	cliente character varying (255)	servico character varying (255)
1	5	2023-11-15 15:00:00	Davi Gonçalves	Manicure
2	4	2023-11-15 15:00:00	Davi Gonçalves	Manicure
3	3	2023-11-15 11:00:00	Natália Duarte	Manicure

### Detalhar Produtos Utilizados em um Atendimento Específico:

```
SELECT P.Nome, PA.QuantidadeUtilizada
FROM ProdutosAtendimento PA
JOIN Produtos P ON PA.ProdutoID = P.ProdutoID
WHERE PA.AtendimentoID = 1;
```

Resultado:

	nome character varying (255)	quantidadeutilizada integer
1	Óleo de Argan	1

### Histórico de Atendimentos de um Cliente:

```
SELECT HA.HistoricoID, HA.DataAtendimento, HA.Detalhes
FROM HistoricoAtendimento HA
WHERE HA.ClienteID = 1;
```

Resultado:

	historicoId [PK] integer	dataatendimento date	detalhes text
1	1	2023-11-15	4 - Detalhes do atendimento
2	2	2023-11-15	5 - Detalhes do atendimento

### Listar Todos os Pagamentos Realizados em um Dia Específico:

```
SELECT
  P.PagamentoID,
  MIN(A.DataHora) AS PrimeiroAtendimento,
  MAX(A.DataHora) AS UltimoAtendimento,
  P.ValorTotal,
  P.FormaPagamento,
  STRING_AGG(C.Nome, ' ') AS Clientes
FROM Pagamentos P
JOIN Atendimentos A ON P.AgendamentoID = A.AgendamentoID
JOIN Agendamentos AG ON A.AgendamentoID = AG.AgendamentoID
JOIN Clientes C ON AG.ClienteID = C.ClienteID
GROUP BY P.PagamentoID, P.ValorTotal, P.FormaPagamento
```

HAVING MIN(A.DataHora)::date = '2023-11-15';

Resultado:

	pagamentoid [PK] integer	primeiroatendimento timestamp without time zone	ultimoatendimento timestamp without time zone	valortotal numeric (10,2)	formapagamento character varying (50)	clientes text
1	1	2023-11-15 15:00:00	2023-11-15 15:00:00	44.50	Dinheiro	Davi Gon...

## 9. Procedures e Functions do Banco de Dados

**Procedure para Realizar Atendimentos:**

```
CREATE OR REPLACE PROCEDURE realizar_atendimento(  
    _clienteID INT,  
    _funcionarioID INT,  
    _servicoID INT,  
    _produtoID INT,  
    _quantidadeProduto INT,  
    _dataHorario TIMESTAMP,  
    _detalhesBase TEXT  
)  
LANGUAGE plpgsql  
AS $$  
DECLARE  
    _agendamentoID INT;  
    _atendimentoID INT;  
    _detalhes TEXT;  
BEGIN  
  
    SELECT AgendamentoID INTO _agendamentoID FROM Agendamentos  
    WHERE ClienteID = _clienteID AND FuncionarioID = _funcionarioID AND  
    DataHorario = _dataHorario;  
  
    IF _agendamentoID IS NULL THEN  
  
        INSERT INTO Agendamentos (ClienteID, FuncionarioID, ServicoID,  
DataHorario)  
        VALUES (_clienteID, _funcionarioID, _servicoID, _dataHorario)  
        RETURNING AgendamentoID INTO _agendamentoID;
```

*END IF;*

*INSERT INTO Atendimentos (AgendamentoID, FuncionarioID, ServicoID, DataHora)*  
*VALUES (\_agendamentoID, \_funcionarioID, \_servicoID, \_dataHorario)*  
*RETURNING AtendimentoID INTO \_atendimentoID;*

*INSERT INTO ProdutosAtendimento (AtendimentoID, ProdutoID, QuantidadeUtilizada)*  
*VALUES (\_atendimentoID, \_produtoID, \_quantidadeProduto);*

*UPDATE Agendamentos*  
*SET AtendimentoID = \_atendimentoID*  
*WHERE AgendamentoID = \_agendamentoID;*

*\_detalhes := \_atendimentoID || ' - ' || \_detalhesBase;*

*INSERT INTO HistoricoAtendimento (ClienteID, Detalhes, DataAtendimento)*  
*VALUES (\_clienteID, \_detalhes, \_dataHorario::date);*

*RAISE NOTICE 'Atendimento realizado com sucesso.';*  
*END;*  
*\$\$;*

### **Procedure para Faturar um Atendimento:**

*CREATE OR REPLACE PROCEDURE registrar\_pagamento(\_agendamentoID INT,*  
*\_valorTotal DECIMAL(10, 2), \_formaPagamento VARCHAR(50))*  
*LANGUAGE plpgsql*  
*AS \$\$*  
*DECLARE*  
*\_pagamentoID INT;*  
*\_atendimentoID INT;*  
*\_totalAtendimento DECIMAL(10, 2) := 0;*  
*\_totalProdutos DECIMAL(10, 2) := 0;*  
*BEGIN*  
  
*SELECT AtendimentoID INTO \_atendimentoID FROM Atendimentos*

*WHERE AgendamentoID = \_agendamentoID;*

*IF \_atendimentoID IS NOT NULL THEN*

*SELECT COALESCE(SUM(Servicos.Valor), 0) INTO \_totalAtendimento  
FROM Atendimentos  
JOIN Servicos ON Atendimentos.ServicoID = Servicos.ServicoID  
WHERE Atendimentos.AtendimentoID = \_atendimentoID;*

*SELECT COALESCE(SUM(Produtos.PrecoUnitario \*  
ProdutosAtendimento.QuantidadeUtilizada), 0) INTO \_totalProdutos  
FROM ProdutosAtendimento  
JOIN Produtos ON ProdutosAtendimento.ProdutoID = Produtos.ProdutoID  
WHERE ProdutosAtendimento.AtendimentoID = \_atendimentoID;*

*\_totalAtendimento := \_totalAtendimento + \_totalProdutos;*

*IF \_valorTotal = \_totalAtendimento THEN*

*INSERT INTO Pagamentos (AgendamentoID, ValorTotal, FormaPagamento)  
VALUES (\_agendamentoID, \_valorTotal, \_formaPagamento)  
RETURNING PagamentoID INTO \_pagamentoID;*

*UPDATE Atendimentos  
SET idPagamento = \_pagamentoID  
WHERE AtendimentoID = \_atendimentoID;*

*RAISE NOTICE 'Pagamento registrado e atendimento atualizado com  
sucesso.';*

*ELSE*

*RAISE EXCEPTION 'Falha no registro de pagamento: o valor do pagamento  
não corresponde ao total do atendimento.';*

*END IF;*

*ELSE*

*RAISE EXCEPTION 'Falha no registro de pagamento: nenhum atendimento  
encontrado para o agendamento fornecido.';*

*END IF;*

*END;*

*\$\$;*

### **Função para Agendar Atendimento:**

```
CREATE OR REPLACE FUNCTION agendar_atendimento(_clienteID INT,
_funcionarioID INT, _servicoID INT, _dataHorario TIMESTAMP)
RETURNS VARCHAR AS $$
DECLARE
    conflito INT;
BEGIN
    -- Verifica se há conflito de horário
    SELECT COUNT(*) INTO conflito FROM Agendamentos
    WHERE FuncionarioID = _funcionarioID AND DataHorario = _dataHorario;

    IF conflito = 0 THEN
        -- Insere o novo agendamento
        INSERT INTO Agendamentos (ClienteID, FuncionarioID, ServicoID,
DataHorario)
VALUES (_clienteID, _funcionarioID, _servicoID, _dataHorario);

        RETURN 'Agendamento realizado com sucesso.';
    ELSE
        -- Conflito de horário
        RETURN 'Falha no agendamento: o funcionário já tem um compromisso neste
horário.';
    END IF;
END;
$$ LANGUAGE plpgsql;
```

### **Função para atualizar o estoque sempre que um produto é usado no atendimento:**

```
CREATE OR REPLACE FUNCTION update_estoque()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE Produtos
    SET Quantidade = Quantidade - NEW.QuantidadeUtilizada
    WHERE ProdutoID = NEW.ProdutoID;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

### **Função para checar se um funcionário tem disponibilidade naquele horário específico:**



```

CREATE OR REPLACE FUNCTION check_funcionario_disponibilidade()
RETURNS TRIGGER AS $$
BEGIN
    -- Verifica se existe algum agendamento no mesmo horário para o funcionário
    IF EXISTS (
        SELECT 1 FROM Agendamentos
        WHERE FuncionarioID = NEW.FuncionarioID
        AND DataHorario = NEW.DataHorario
    ) THEN
        -- Se existir, lança um erro
        RAISE EXCEPTION 'Funcionário já possui um agendamento para este horário.';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

## 10. Triggers do Banco de Dados

**Trigger para chamar a função de atualização de estoque após atendimento:**

```

CREATE TRIGGER update_estoque_after_atendimento
AFTER INSERT ON ProdutosAtendimento
FOR EACH ROW
EXECUTE FUNCTION update_estoque();

```

**Trigger para chamar a função de disponibilidade de funcionario ao realizar um agendamento:**

```

CREATE TRIGGER trg_check_disponibilidade
BEFORE INSERT ON Agendamentos
FOR EACH ROW
EXECUTE FUNCTION check_funcionario_disponibilidade();

```

## 11. Considerações Finais

Eduardo:

O desenvolvimento deste projeto sob minha responsabilidade revelou-se extremamente gratificante. Lamentavelmente, meu colega Igor optou por descontinuar sua participação na disciplina LAB-BD, resultando na necessidade de eu assumir integralmente a execução do projeto. Isso incluiu a criação do banco de dados, o processo de povoamento, a elaboração do relatório e tarefas correlatas. Embora tenha representado um desafio significativo, expressei minha profunda satisfação com os resultados alcançados. Esta experiência fortaleceu minha confiança e competência na área de bancos de dados, a ponto de considerar a possibilidade de concorrer a oportunidades profissionais no setor. No ambiente prático, onde é possível realizar consultas à internet conforme necessário, senti-me bastante à vontade. Expresso meus sinceros agradecimentos a todos que dedicaram tempo para ler este relato até o final.