

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

**Nolan Lemes Marcondes
CJ3030024**

SISTEMA PARA GESTÃO DE BARBEARIA

**CAMPOS DO JORDÃO
2025**

RESUMO

O objetivo de um sistema de banco de dados para uma barbearia é organizar, armazenar e gerenciar as informações de maneira eficiente, permitindo otimizar os processos de administração e melhorar a experiência tanto para os clientes quanto para os empregados. Os principais objetivos incluem: Gestão de Clientes; Controle de Agendamentos; Gestão de Funcionários; Controle de Serviços; Relatórios e Análises; Promoções e Fidelização de Clientes.

Uma metodologia eficiente para o desenvolvimento desse sistema pode seguir o ciclo de vida de desenvolvimento de software (SDLC - Software Development Life Cycle), que inclui as fases de levantamento de requisitos, análise, design, implementação, testes e manutenção. É importante envolver os usuários finais (funcionários da barbearia) no processo para garantir que o sistema atenda às necessidades reais do negócio. Além disso, a escolha de uma boa arquitetura e tecnologias apropriadas contribui para o sucesso do sistema a longo prazo.

Palavras-Chave: Barbearia; Sistema; Negócio.

ABSTRACT

The purpose of a database system for a barbershop is to organize, store, and manage information efficiently, allowing you to optimize administration processes and improve the experience for both customers and owners and employees. The main objectives include: Customer Management; Scheduling Control; Employee Management; Service Control; Reports and Analysis; Promotions and Customer Loyalty.

An efficient methodology for developing this system can follow the Software Development Life Cycle (SDLC), which includes the phases of requirements gathering, analysis, design, implementation, testing, and maintenance. It is important to involve the end users (barbershop employees) in the process to ensure that the system meets the actual needs of the business. In addition, the choice of a good architecture and appropriate technologies contributes to the long-term success of the system.

Keywords: Barbershop; System; Business.

SUMÁRIO

1	INTRODUÇÃO _____	12
1.1	Metodologia _____	12
2	RESULTADOS OBTIDOS _____	14
3	CONCLUSÃO _____	15
4	REFERÊNCIAS _____	20

1 INTRODUÇÃO

O banco de dados "**BarbeariaDB**" foi projetado de forma eficiente para atender às principais necessidades operacionais de uma barbearia. Ele possui uma estrutura relacional bem definida, composta por cinco tabelas principais que se interligam para garantir o controle e a organização dos processos.

- A tabela "**Clientes**" permite o cadastro dos clientes, armazenando informações como nome, telefone, e-mail e a data de cadastro.
- A tabela "**Funcionarios**" registra os dados dos profissionais, incluindo suas especialidades, contatos, data de contratação e salário.
- A tabela "**Servicos**" lista os serviços oferecidos pela barbearia, com valores, nomes e duração de cada atendimento.
- A tabela "**Agendamentos**" conecta clientes, funcionários e serviços, organizando os horários dos atendimentos e seu status (agendado, concluído ou cancelado).

- A tabela "**Pagamentos**" está diretamente vinculada aos agendamentos e permite o controle dos recebimentos, registrando valores, formas de pagamento e datas.

Além disso, o banco de dados foi desenvolvido com o uso adequado de chaves primárias e estrangeiras, garantindo a integridade referencial entre as tabelas. Os tipos de dados escolhidos são apropriados para as informações que cada tabela armazena, e o uso de campos ENUM nos status e formas de pagamento padroniza as opções, facilitando a consistência dos dados.

1.1 metodologia

A metodologia utilizada na construção desse banco de dados segue os princípios da **Modelagem de Dados Relacional**, baseada no modelo relacional proposto por **Edgar F. Codd**. Este modelo organiza os dados em tabelas (ou relações) que se conectam através de chaves primárias e estrangeiras.

1. Levantamento de Requisitos

- a. Identificação das principais entidades do negócio (Clientes, Funcionários, Serviços, Agendamentos, Pagamentos).
- b. Entendimento dos processos da barbearia, como cadastros, agendamentos e controle de pagamentos.

2. Modelagem Conceitual

- a. Definição das entidades e dos atributos.
- b. Relacionamentos identificados entre elas.
- c. Normalização para evitar redundância de dados.

3. Modelagem Lógica

- a. Transformação do modelo conceitual em um modelo lógico relacional.
- b. Definição de tabelas, campos, tipos de dados, chaves primárias e estrangeiras.

4. Modelagem Física

- a. Implementação do banco de dados em um SGBD (Sistema Gerenciador de Banco de Dados), utilizando comandos SQL.
- b. Aplicação de restrições como AUTO_INCREMENT, PRIMARY KEY, FOREIGN KEY e ENUM.

2 RESULTADOS OBTIDOS

A implementação desse banco de dados trouxe diversos resultados práticos e operacionais para a gestão da barbearia. O banco de dados "**BarbeariaDB**" proporcionou:

- **Eficiência operacional,**
- **Melhor organização,**
- **Gestão financeira robusta,**
- e **base de dados sólida** para tomadas de decisão e crescimento sustentável da barbearia.

3 CONCLUSÃO

O desenvolvimento do banco de dados "BarbeariaDB" representa uma solução tecnológica fundamental para atender às necessidades operacionais, administrativas e financeiras de uma barbearia moderna. Sua estrutura foi cuidadosamente planejada para garantir eficiência, organização e controle de todas as atividades envolvidas no funcionamento do negócio.

A criação de tabelas específicas para clientes, funcionários, serviços, agendamentos e pagamentos permite que as informações sejam devidamente categorizadas e relacionadas, facilitando consultas rápidas, geração de relatórios e acompanhamento em tempo real dos processos.

Do ponto de vista da operação diária, o banco oferece uma grande vantagem ao permitir um controle preciso dos agendamentos, possibilitando visualizar facilmente quais serviços estão marcados, para qual funcionário, qual cliente e em qual horário. Isso reduz drasticamente erros operacionais, conflitos de horários e esquecimentos, além de garantir uma melhor experiência tanto para o cliente quanto para o funcionário.

No aspecto financeiro, o controle dos pagamentos, com registros detalhados de valores, formas de pagamento e datas, permite à administração da barbearia ter total clareza sobre sua saúde financeira, facilitando análises de receitas, identificação de períodos de maior ou menor movimento e planejamento estratégico.

Além disso, a utilização de chaves primárias e estrangeiras, juntamente com tipos de dados adequados e restrições como ENUM, promove a integridade dos dados, evitando inconsistências e garantindo que todas as informações estejam corretas e atualizadas.

O banco também foi projetado com foco na escalabilidade, permitindo que, à medida que a barbearia cresça — seja com mais clientes, novos serviços, contratação de mais funcionários ou abertura de novas unidades —, o sistema continue sendo funcional e adaptável.

Em suma, o banco de dados "BarbeariaDB" não só automatiza processos, mas também agrega valor ao negócio, proporcionando uma gestão mais eficiente, maior controle sobre as operações e dados confiáveis para a tomada de decisões. Ele é

uma ferramenta essencial para qualquer barbearia que busca se profissionalizar, crescer e oferecer um serviço de qualidade e excelência aos seus clientes.

Como já foi apresentado um pouco do projeto no final da fundamentação teórica, aqui já será falado tudo sobre o projeto, como foi desenvolvido (crie subseções adequadas ao seu trabalho). Exemplo:

Nesta seção serão apresentadas detalhadamente a metodologia utilizada neste trabalho, porque esta foi a escolhida (porque esta é mais adequada que outras) e suas etapas, os documentos referentes ao sistema proposto, como eles foram elaborados e demais artefatos referentes a este projeto.

1 REFERÊNCIAS BIBLIOGRÁFICAS

1. **Date, C. J.**
Introdução a Sistemas de Banco de Dados. 8ª edição. Rio de Janeiro: Campus, 2004.
2. **Elmasri, R.; Navathe, S. B.**
Sistemas de Banco de Dados. 7ª edição. São Paulo: Pearson, 2019.
3. **Coronel, C.; Morris, S.; Rob, P.**
Database Systems: Design, Implementation, & Management. 13th Edition. Cengage Learning, 2019.
4. **Silberschatz, A.; Korth, H.; Sudarshan, S.**
Sistemas de Banco de Dados. 6ª edição. São Paulo: Pearson, 2013.

---- CÓDIGO DO BANCO DE DADOS ----

```
-- Criar o banco de dados
CREATE DATABASE BarbeariaDB;
USE BarbeariaDB;

-- Tabela de Clientes
CREATE TABLE Clientes (
    id_cliente INT AUTO_INCREMENT PRIMARY KEY,
```

```

    nome VARCHAR(100),
    telefone VARCHAR(15),
    email VARCHAR(100),
    data_cadastro DATE
);

-- Tabela de Funcionários
CREATE TABLE Funcionarios (
    id_funcionario INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100),
    especialidade VARCHAR(50),
    telefone VARCHAR(15),
    email VARCHAR(100),
    data_contratacao DATE,
    salario DECIMAL(10,2)
);

-- Tabela de Serviços
CREATE TABLE Servicos (
    id_servico INT AUTO_INCREMENT PRIMARY KEY,
    nome_servico VARCHAR(100),
    preco DECIMAL(10,2),
    duracao_minutos INT
);

-- Tabela de Agendamentos
CREATE TABLE Agendamentos (
    id_agendamento INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT,
    id_funcionario INT,
    id_servico INT,
    data_hora DATETIME,
    status ENUM('Agendado', 'Concluído', 'Cancelado'),
    FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente),
    FOREIGN KEY (id_funcionario) REFERENCES Funcionarios(id_funcionario),
    FOREIGN KEY (id_servico) REFERENCES Servicos(id_servico)
);

-- Tabela de Pagamentos
CREATE TABLE Pagamentos (
    id_pagamento INT AUTO_INCREMENT PRIMARY KEY,
    id_agendamento INT,
    valor_pago DECIMAL(10,2),
    forma_pagamento ENUM('Dinheiro', 'Cartão', 'Pix'),
    data_pagamento DATE,
    FOREIGN KEY (id_agendamento) REFERENCES Agendamentos(id_agenda-
mento)
);

-- INSERÇÃO DE DADOS

-- Clientes

```



```
INSERT INTO Clientes (nome, telefone, email, data_cadastro)
VALUES
('João Silva', '11999998888', 'joao@email.com', '2024-12-01'),
('Carlos Lima', '11888887777', 'carlos@email.com', '2025-01-15'),
('Marcos Souza', '11777776666', 'marcos@email.com', '2025-03-10');
```

-- Funcionários

```
INSERT INTO Funcionarios (nome, especialidade, telefone, email, data_contratacao,
salario)
VALUES
('Pedro Oliveira', 'Corte Masculino', '11666665555', 'pedro@email.com', '2023-11-01',
2500.00),
('Lucas Mendes', 'Barba e Sobancelha', '11555554444', 'lucas@email.com', '2024-
01-10', 2600.00);
```

-- Serviços

```
INSERT INTO Servicos (nome_servico, preco, duracao_minutos)
VALUES
('Corte de Cabelo', 40.00, 30),
('Barba', 25.00, 20),
('Corte + Barba', 60.00, 50),
('Sobancelha', 15.00, 10);
```

-- Agendamentos

```
INSERT INTO Agendamentos (id_cliente, id_funcionario, id_servico, data_hora, sta-
tus)
VALUES
(1, 1, 1, '2025-05-20 14:00:00', 'Concluído'),
(2, 2, 2, '2025-05-21 16:00:00', 'Agendado'),
(3, 1, 3, '2025-05-18 13:30:00', 'Cancelado'),
(1, 2, 4, '2025-05-17 10:00:00', 'Concluído');
```

-- Pagamentos

```
INSERT INTO Pagamentos (id_agendamento, valor_pago, forma_pagamento,
data_pagamento)
VALUES
(1, 40.00, 'Pix', '2025-05-20'),
(4, 15.00, 'Cartão', '2025-05-17');
```

-- 20 CONSULTAS

-- Clientes que realizaram o pagamento

```
SELECT DISTINCT c.nome
FROM Clientes c
JOIN Agendamentos a ON c.id_cliente = a.id_cliente
JOIN Pagamentos p ON a.id_agendamento = p.id_agendamento;
```

-- Funcionários que mais finalizaram atendimentos

```
SELECT f.nome, COUNT(*) AS total_atendimentos
FROM Agendamentos a
JOIN Funcionarios f ON a.id_funcionario = f.id_funcionario
WHERE a.status = 'Concluído'
GROUP BY f.id_funcionario
ORDER BY total_atendimentos DESC;
```

-- Serviço mais comprado (Baseado em pagamentos)

```
SELECT s.nome_servico, SUM(p.valor_pago) AS total_receita
FROM Pagamentos p
JOIN Agendamentos a ON p.id_agendamento = a.id_agendamento
JOIN Servicos s ON a.id_servico = s.id_servico
GROUP BY s.id_servico
ORDER BY total_receita DESC
LIMIT 1;
```

-- Receita Mensal da barbearia

```
SELECT DATE_FORMAT(data_pagamento, '%Y-%m') AS mes, SUM(valor_pago)
AS receita
FROM Pagamentos
GROUP BY mes
ORDER BY mes DESC;
```

-- Agendamentos Cancelados nos últimos 30 dias

```
SELECT * FROM Agendamentos
WHERE status = 'Cancelado'
AND data_hora >= NOW() - INTERVAL 30 DAY;
```

-- Clientes com mais de um agendamento

```
SELECT c.nome, COUNT(a.id_agendamento) AS total
FROM Clientes c
JOIN Agendamentos a ON c.id_cliente = a.id_cliente
GROUP BY c.id_cliente
HAVING total > 1;
```

-- Funcionários que nunca receberam pagamento

```
SELECT f.nome
FROM Funcionarios f
LEFT JOIN Agendamentos a ON f.id_funcionario = a.id_funcionario
LEFT JOIN Pagamentos p ON a.id_agendamento = p.id_agendamento
WHERE p.id_pagamento IS NULL;
```

-- Média de tempo por serviço

```
SELECT nome_servico, AVG(duracao_minutos) AS duracao_media
```

```
FROM Servicos
GROUP BY nome_servico;
```

-- Valor total gasto por cliente

```
SELECT c.nome, SUM(p.valor_pago) AS total_gasto
FROM Clientes c
JOIN Agendamentos a ON c.id_cliente = a.id_cliente
JOIN Pagamentos p ON a.id_agendamento = p.id_agendamento
GROUP BY c.id_cliente;
```

-- Serviços mais agendados

```
SELECT s.nome_servico, COUNT(*) AS total
FROM Agendamentos a
JOIN Servicos s ON a.id_servico = s.id_servico
GROUP BY s.id_servico
ORDER BY total DESC;
```

-- Funcionário com maior receita gerada

```
SELECT f.nome, SUM(p.valor_pago) AS receita_total
FROM Funcionarios f
JOIN Agendamentos a ON f.id_funcionario = a.id_funcionario
JOIN Pagamentos p ON a.id_agendamento = p.id_agendamento
GROUP BY f.id_funcionario
ORDER BY receita_total DESC
LIMIT 1;
```

-- Horários com mais agendamentos (por hora do dia)

```
SELECT HOUR(data_hora) AS hora, COUNT(*) AS qtd
FROM Agendamentos
GROUP BY hora
ORDER BY qtd DESC;
```

-- Clientes que usaram Pix para pagamento

```
SELECT DISTINCT c.nome
FROM Clientes c
JOIN Agendamentos a ON c.id_cliente = a.id_cliente
JOIN Pagamentos p ON a.id_agendamento = p.id_agendamento
WHERE p.forma_pagamento = 'Pix';
```

-- Últimos 5 agendamentos realizados

```
SELECT * FROM Agendamentos
ORDER BY data_hora DESC
LIMIT 5;
```

-- Funcionários que atenderam um cliente específico

```
SELECT DISTINCT f.nome
FROM Funcionarios f
JOIN Agendamentos a ON f.id_funcionario = a.id_funcionario
WHERE a.id_cliente = 1;
```

-- Clientes com agendamentos futuros

```
SELECT DISTINCT c.nome
FROM Clientes c
JOIN Agendamentos a ON c.id_cliente = a.id_cliente
WHERE a.data_hora > NOW();
```

-- Porcentagem de agendamentos concluídos por funcionário

```
SELECT f.nome,
       ROUND(SUM(CASE WHEN a.status = 'Concluído' THEN 1 ELSE 0 END) /
COUNT(*) * 100, 2) AS porcentagem_concluidos
FROM Funcionarios f
JOIN Agendamentos a ON f.id_funcionario = a.id_funcionario
GROUP BY f.id_funcionario;
```

-- Serviços com preços acima da média

```
SELECT nome_servico, preco
FROM Servicos
WHERE preco > (SELECT AVG(preco) FROM Servicos);
```

-- Dias com mais agendamentos

```
SELECT DATE(data_hora) AS dia, COUNT(*) AS total_agendamentos
FROM Agendamentos
GROUP BY dia
ORDER BY total_agendamentos DESC;
```

-- Faturamentos por formas de pagamento

```
SELECT forma_pagamento, SUM(valor_pago) AS total
FROM Pagamentos
GROUP BY forma_pagamento;
```