



PROJETO BANCO DE DADOS CERVEJARIA





Roteiro

01 Integrantes

02 Minimundo

03 Projeto Conceitual

04 Projeto Relacional

05 SQL

06 PL – SQL





01

Integrantes



Integrantes

- **Carlos Eduardo Dias**
(cedfv)
- **Pedro Henrique Medeiros**
(phmfl)
- **João Victor Pereira das Neves**
(jvpn)
- **Bruno Rodrigo dos Santos**
(brsc)
- **Jonathas Vinícius**
(jvras)
- **Vinícius Oliveira**
(vmo2)



02

Minimundo



Cervejaria Du Penguin

A cervejaria Du Penguin, pretende fornecer a seus clientes uma experiência singular através da degustação dos mais diversos tipos de cervejas tradicionais e artesanais, contando com uma enorme variedade de ingredientes nacionais e importados.

Nosso funcionamento acontece de maneira que possamos servir cervejas já prontas, repassadas pelos nossos parceiros e também produzir nossa própria cerveja baseada nos gostos e experiências de nossos colaboradores.



O que buscamos ?



**Produtos de
qualidade**



**Satisfação
dos nossos
clientes**



Cerveja Gelada

Minimundo

- A cervejaria pode fabricar vários tipos de cervejas
- As cervejas podem ser do tipo artesanal ou do tipo industrializada, no qual possuem ingredientes e tipos de produção diferentes.
- A cervejaria também possui vários garçons espalhados pelo salão
- Entre os garçons, pode ou não existir algum garçom que desempenhe o papel de líder, entre os demais..
- Um cliente pode fazer um ou mais pedidos para mais de um garçom e um garçom pode receber pedidos de mais de um cliente.



Minimundo

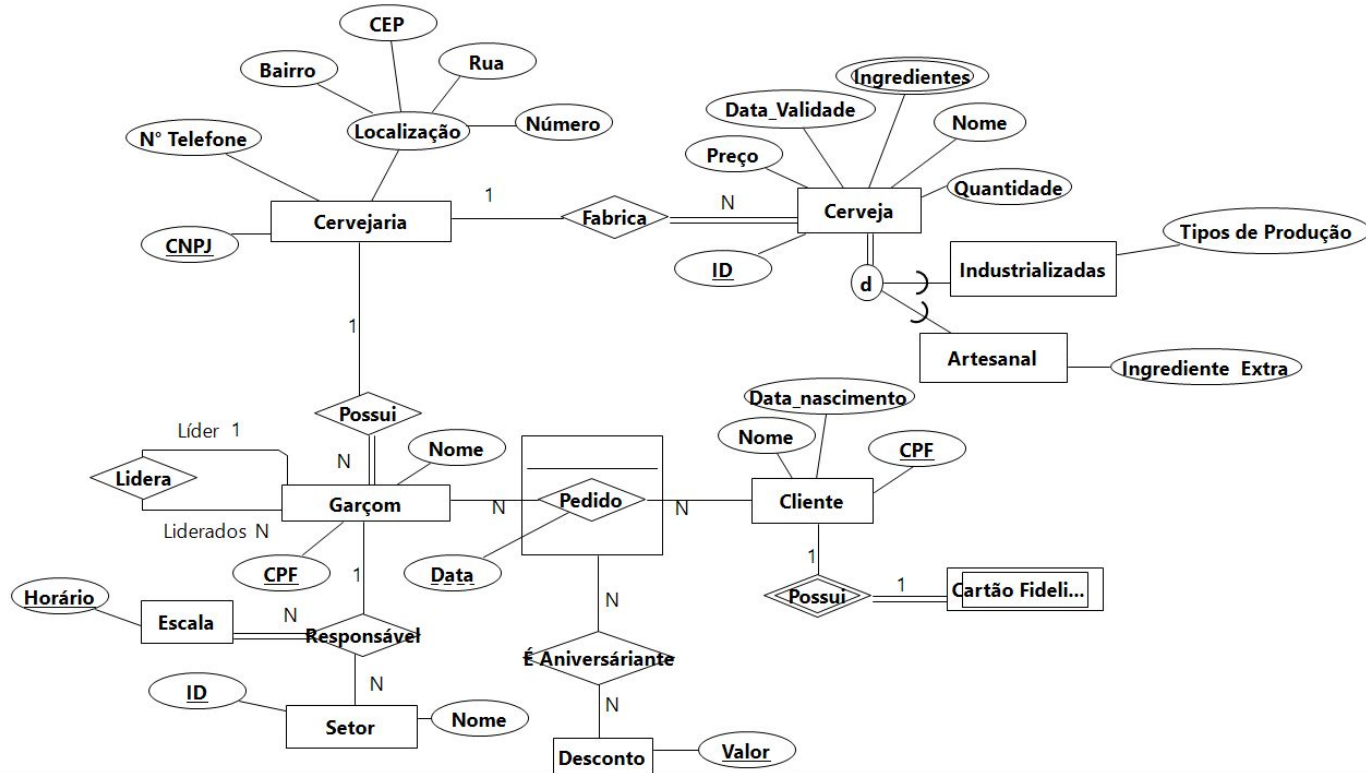
- Caso o cliente esteja na cervejaria em sua data de aniversário, ele receberá um desconto em sua conta final.
- A cada dia que o cliente comparecer e consumir os produtos da cervejaria, ele receberá um cartão fidelidade para futuras bonificações do estabelecimento.
- Um garçom, sendo líder ou não, pode ser responsável por um setor da cervejaria em um determinado turno. Passado o horário, o mesmo funcionário também pode ser responsável por outro setor da cervejaria.





03 Projeto Conceitual

Projeto Conceitual





04

Projeto Relacional

Projeto Relacional

Cervejaria

Cervejaria(CNPJ, n°telefone,
loc_bairro, loc_cep, loc_rua,
loc_numero)



Ingredientes

Ingredientes(ID,ingredientes)
ID → **Cerveja**(ID)



Cerveja

Cerveja(ID, preço, data_validade, nome,
quantidade, CNPJ!)

CNPJ → **Cervejaria**(CNPJ)



Industrializadas

Industrializadas(ID, tipos de produção)
ID → **Cerveja**(ID)



Artesanal

Artesanal (ID, Ingrediente_extra)
ID → **Cerveja**(ID)



Projeto Relacional

Garçom

Garçom (CPF, Nome, LiderCPF, CNPJ!)

LiderCPF → **Garçom**(CPF)

CNPJ → **Cervejaria**(CNPJ)



Responsável

Responsável(CPF!, Horário, ID)

CPF → **Garçom**(CPF)

Horário → **Escala** (Horário)

ID → **Setor** (ID)



Escala

Escala (Horário, CPF_Garçom!)

CPF_Garçom → **Garçom**(CPF)



Setor

Setor(ID, Nome, CPF)

CPF → **Garçom**(CPF)



Cliente

Cliente(CPF, nome, data_nascimento)



Projeto Relacional

Aniversariante

Aniversariante(CPF_cliente,
CPF_garçon,Valor)

Valor → Desconto(Valor)

CPF_cliente → Pedido(CPF_cliente)

CPF_garçon → Pedido(CPF_garçon)



Desconto

Desconto(Valor)



Cartão Fidelidade

Cartão_fidelidade([CPF]!)

CPF → Cliente(CPF)



Pedido

Pedido(CPF_cliente, CPF_garçon, data)

CPF_cliente → Cliente(CPF)

CPF_garçom → Garçom(CPF)





05

SQL – DDL



SQL – DDL

Cervejaria

```
CREATE TABLE Cervejaria (  
  CNPJ VARCHAR(14) PRIMARY KEY,  
  num_telefone VARCHAR(20),  
  loc_bairro VARCHAR(100),  
  loc_cep VARCHAR(15),  
  loc_rua VARCHAR(100),  
  loc_numero VARCHAR(10)  
);
```

Cerveja

```
CREATE TABLE Cerveja (  
  ID INT PRIMARY KEY,  
  preco DECIMAL(10,2),  
  data_validade DATE,  
  nome VARCHAR(100),  
  quantidade INT,  
  CNPJ VARCHAR(14) NOT NULL,  
  FOREIGN KEY (CNPJ) REFERENCES  
  Cervejaria(CNPJ)  
);
```

SQL – DDL

Ingredientes

```
CREATE TABLE Ingredientes (  
  ID INT,  
  ingredientes VARCHAR(100),  
  PRIMARY KEY(ID, ingredientes),  
  FOREIGN KEY (ID) REFERENCES Cerveja(ID)  
);
```

Industrializadas

```
CREATE TABLE Industrializadas (  
  ID INT,  
  tipos_producao VARCHAR(100),  
  PRIMARY KEY(ID, tipos_producao),  
  FOREIGN KEY (ID) REFERENCES Cerveja(ID)  
);
```

Artesanal

```
CREATE TABLE Artesanal (  
  ID INT PRIMARY KEY,  
  ingrediente_extra VARCHAR(100),  
  FOREIGN KEY (ID) REFERENCES Cerveja(ID)  
);
```

SQL – DDL

Garçom

```
CREATE TABLE Garcom (  
  CPF VARCHAR(11) PRIMARY KEY,  
  nome VARCHAR(100),  
  LiderCPF VARCHAR(11),  
  CNPJ VARCHAR(14) NOT NULL,  
  FOREIGN KEY (LiderCPF) REFERENCES Garcom(CPF),  
  FOREIGN KEY (CNPJ) REFERENCES Cervejaria(CNPJ)  
);
```

Escala

```
CREATE TABLE Escala (  
  horario TIMESTAMP,  
  CPF_garcom VARCHAR(11),  
  PRIMARY KEY (horario),  
  FOREIGN KEY (CPF_garcom) REFERENCES Garcom(CPF)  
);
```

SQL – DDL

Responsável

```
CREATE TABLE Responsavel (  
    CPF_garcom VARCHAR(11),  
    horario TIMESTAMP,  
    ID_setor INT,  
    PRIMARY KEY (CPF_garcom, horario, ID_setor),  
    FOREIGN KEY (CPF_garcom) REFERENCES Garcom(CPF),  
    FOREIGN KEY (horario) REFERENCES Escala(horario),  
    FOREIGN KEY (ID_setor) REFERENCES Setor(ID)  
);
```

Setor

```
CREATE TABLE Setor (  
    ID INT PRIMARY KEY,  
    nome VARCHAR(100),  
    CPF_garcom VARCHAR(11),  
    FOREIGN KEY (CPF_garcom) REFERENCES Garcom(CPF)  
);
```

SQL – DDL

Cartão Fidelidade

```
CREATE TABLE Cartao_fidelidade (  
  CPF_cliente VARCHAR(11) PRIMARY KEY,  
  FOREIGN KEY (CPF_cliente) REFERENCES Cliente(CPF)
```

Desconto

```
CREATE TABLE Desconto (  
  valor DECIMAL(10,2) PRIMARY KEY  
);
```

Pedido

```
CREATE TABLE Pedido (  
  CPF_cliente VARCHAR(11),  
  CPF_garcom VARCHAR(11),  
  valor DECIMAL(10,2),  
  data DATE,  
  PRIMARY KEY (CPF_cliente, CPF_garcom, data),  
  FOREIGN KEY (CPF_cliente) REFERENCES Cliente(CPF),  
  FOREIGN KEY (CPF_garcom) REFERENCES Garcom(CPF),  
  FOREIGN KEY (valor) REFERENCES Desconto(valor)  
);
```

Cliente

```
CREATE TABLE Cliente (  
  CPF VARCHAR(11) PRIMARY KEY,  
  nome VARCHAR(100),  
  data_nascimento DATE  
);
```



05 – SQL

CONSULTAS



SQL – DML

Lista de cervejarias com mais de 5 cervejas diferentes (GROUP BY/ HAVING)

```
SELECT c.CNPJ_cervejaria, COUNT(c.ID) as total_cervejas  
FROM Cerveja c  
GROUP BY c.CNPJ_cervejaria  
HAVING COUNT(c.ID) > 5;
```



SQL – DML

Projetar as cervejas que têm pelo menos um ingrediente listado na tabela (SEMI JUNÇÃO)

```
SELECT *  
FROM Cerveja  
WHERE EXISTS (  
    SELECT *  
    FROM Ingredientes  
    WHERE Cerveja.ID = Ingredientes.ID  
);
```



SQL – DML

Lista os clientes com cartão fidelidade (INNER JOIN)

```
SELECT c.CPF, c.nome  
FROM Cliente c  
INNER JOIN Cartao_fidelidade cf ON c.CPF = cf.CPF_cliente;
```



Lista os clientes e o valor total gasto em cada pedido (Incluindo clientes sem pedidos) (OUTER JOIN)

```
SELECT c.CPF, c.nome, p.Valor  
FROM Cliente c  
LEFT OUTER JOIN Pedido p ON c.CPF = p.CPF_cliente;
```

SQL – DML

Projetar as cervejas que não são Artesanais (ANTI JUNCÃO)

```
SELECT *  
FROM Cerveja  
WHERE ID NOT IN (SELECT ID FROM Artesanal);
```



Obter o preço médio das cervejas de cada cervejaria (SUBCONSULTA TIPO ESCALAR)

```
SELECT CNPJ,  
       (SELECT AVG(preco) FROM Cerveja WHERE  
        CNPJ_cervejaria = Cervejaria.CNPJ) AS  
       preco_medio  
FROM Cervejaria;
```



SQL – DML

Lista os nomes dos clientes que possuem um cartão de fidelidade (SUBCONSULTA TIPO LINHA)

```
SELECT cl.nome AS nome_cliente  
FROM Cliente cl  
WHERE EXISTS (  
    SELECT *  
    FROM Cartao_fidelidade cf  
    WHERE cl.CPF = cf.CPF_cliente  
);
```



SQL – DML

Retorna os CPFs e nomes dos clientes que fizeram pedidos na data '18/04/2023'. A SUBCONSULTA DO TIPO TABELA retorna uma tabela com CPFs dos clientes que têm pedidos na data específica

```
SELECT C.CPF, C.nome  
FROM Cliente C  
WHERE C.CPF IN (  
    SELECT CPF_cliente  
    FROM Pedido  
    WHERE dataPedido =  
    TO_DATE('18/04/2023', 'DD/MM/YYYY')  
);
```



SQL – DML

Retorna os nomes dos clientes e garçons para os pedidos realizados na data '18/04/2023' e também para os clientes que são aniversariantes (OPERAÇÃO DE CONJUNTO)

```
SELECT C.nome AS nome_cliente, G.Nome AS nome_garcom
FROM Cliente C, Garcom G
WHERE C.CPF IN (
    SELECT CPF_cliente
    FROM Pedido
    WHERE dataPedido = TO_DATE('18/04/2023', 'DD/MM/YYYY')
)
UNION
SELECT C.nome AS nome_cliente, G.Nome AS nome_garcom
FROM Cliente C, Garcom G
WHERE C.CPF IN (
    SELECT CPF_cliente
    FROM Aniversariante
);
```



PL – SQL

- Função que recebe o nome da cervejaria como parâmetro e utiliza SQL embutido para obter o CNPJ da cervejaria a partir do nome e a quantidade de cervejas distintas produzidas pela cervejaria
- retorna a quantidade de cervejas encontradas

```
CREATE OR REPLACE FUNCTION
quantidade_cervejas_por_cervejaria(cervejaria_nome VARCHAR(100))
RETURNS INTEGER AS $$
DECLARE
    cervejaria_cnpj VARCHAR(14);
    qtd_cervejas INTEGER;
BEGIN
    SELECT CNPJ INTO cervejaria_cnpj FROM Cervejaria WHERE UPPER(loc_bairro)
    = UPPER(cervejaria_nome);
    SELECT COUNT(DISTINCT ID) INTO qtd_cervejas FROM Cerveja WHERE
    CNPJ_cervejaria = cervejaria_cnpj;
    RETURN qtd_cervejas;
END;
$$ LANGUAGE plpgsql;
-- chamando a função
SELECT quantidade_cervejas_por_cervejaria('Pinheiros');
```





OBRIGADO!

