Serviço Público Federal
Ministério da Educação
Fundação Universidade Federal de Mato Grosso do Sul
Câmpus de Três Lagoas
Curso de Bacharel em Sistemas de Informação

Murilo José Silva

Kaique Josias Brito

Sistema de Gerenciamento de Restaurante

Trabalho apresentado à Disciplina de Laboratório de Banco de Dados,

Professora Ivone Penque Matsuno Yugoshi

1. Introdução

A proposta de desenvolvimento deste banco de dados relacional para o "Sabor do Rio", um restaurante situado em Três Lagoas, surge da necessidade de estabelecer uma base estruturada para gerenciar todas as operações essenciais ao funcionamento do estabelecimento. Ao criar uma infraestrutura robusta, a intenção é não apenas armazenar dados, mas também proporcionar uma gestão eficaz e integrada das informações relacionadas aos aspectos cruciais do restaurante.

O esquema conceitual delineado visa mapear, de forma abrangente, as diversas entidades envolvidas nas operações diárias do restaurante. A inclusão de tabelas específicas para representar elementos como restaurante, funcionário, cliente, pedido, prato e métodos de pagamento demonstra uma abordagem holística na captura e organização das informações relevantes. Cada tabela é cuidadosamente projetada para garantir a integridade dos dados, evitando inconsistências e redundâncias.

A representação do esquema lógico complementa essa visão, refletindo de maneira precisa a estrutura necessária para as operações diárias. A interconexão entre as tabelas é estabelecida por meio de chaves estrangeiras, facilitando a recuperação de dados e permitindo uma visão abrangente das relações entre as diferentes entidades.

Este banco de dados não se limita apenas ao armazenamento de informações transacionais. Ele é concebido para ser uma ferramenta dinâmica que se adapta às complexidades do ambiente do "Sabor do Rio".

Essa estrutura é baseada na rede de restaurantes Burger King (BK).

2. Descrição do cenário

Às margens do tranquilo Rio Paraná, ergue-se o "Sabor do Rio", um recém-inaugurado ponto gastronômico em Três Lagoas. Este restaurante nasceu da paixão pela culinária regional e da inspiração nas riquezas naturais do Brasil.

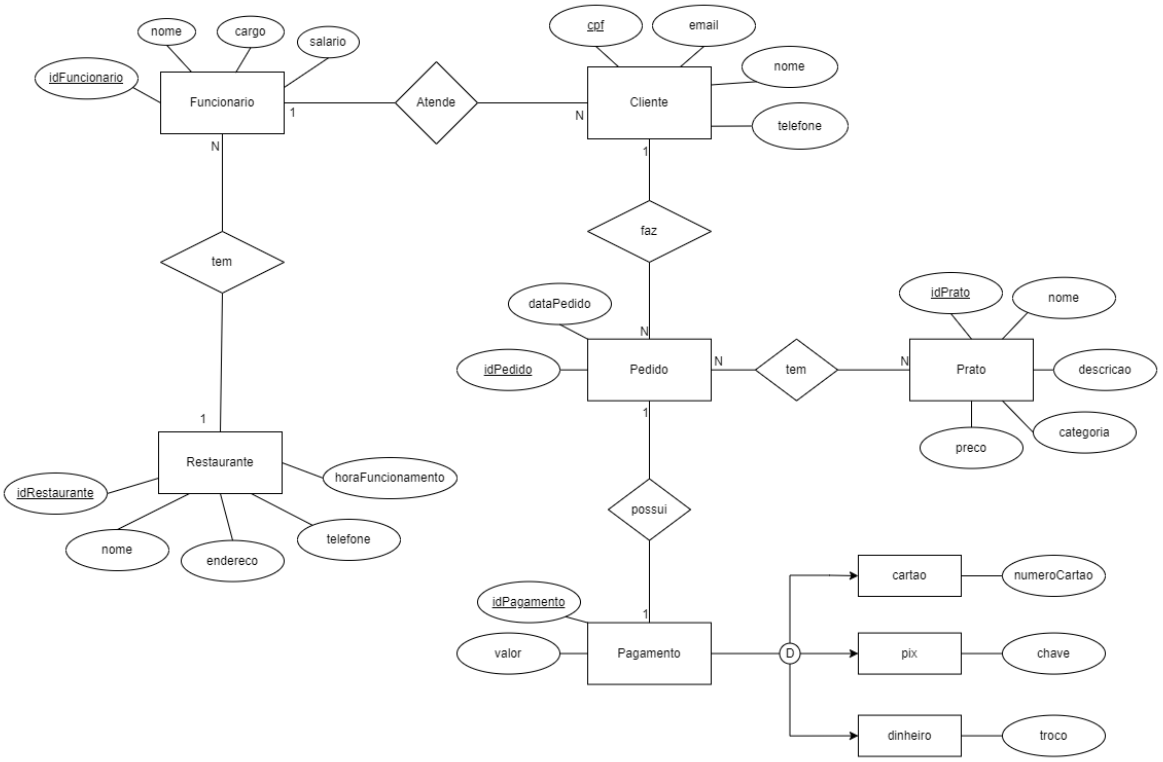
Com uma vista deslumbrante do rio, o ambiente do Sabor do Rio é um convite à descontração. A decoração moderna destaca tons que refletem as águas serenas do Paraná, proporcionando uma experiência visual única aos visitantes. As mesas elegantes, adornadas com arranjos de flores locais, completam o cenário acolhedor.

O cardápio é um verdadeiro mergulho nos sabores autênticos da região. Pratos como Peixe na Telha, Moqueca Pantaneira e Frango com Guavira destacam a diversidade da culinária brasileira. Além disso, opções vegetarianas e veganas ressaltam a riqueza dos produtos cultivados localmente.

O Sabor do Rio não é apenas um restaurante, mas um espaço para a comunidade se reunir. Noites de música ao vivo, exposições de artistas locais e eventos temáticos sazonais enriquecem a experiência, tornando-o mais do que um local de refeições, mas um ponto de encontro para apreciar a autenticidade da culinária brasileira.

Seja bem-vindo ao Sabor do Rio, onde cada refeição é uma jornada pelos sabores locais e uma celebração da beleza natural que rodeia Três Lagoas.

3. Esquema conceitual



4. Esquema lógico

Restaurante(idRestaurante, nome, endereco, telefone, HoraFuncionamento);

Funcionario(idFuncionario, nome, cargo, salario, **IdRestaurante**)

Cliente(cpf, email, nome, telefone);

Pedido(idPedido, dataPedido, **cpf**);

Prato(idPrato, nome, descricao, categoria, preco);

PedidoPrato(***idPedido, idPrato***);

Pagamento(idPagamento, valor, **idPedido**);

Cartao(***idPagamente***, numeroCartao);

Pix(***idPagamente*,** chave);

Dinheiro(***idPagamente***, troco);

Legendas:

teste : primária.
testes2: estrangeira.
teste3: primária e estrangeira.

Tabela 1 - Restaurante

Atributo	**Tipo de Dados**	**Restrição de Domínio**	**Restrição de Entidade**	**Restrição Referencial**
idRestaurante	INTEGER	Obrigatório	Sim	
nome	VARCHAR(255)	Obrigatório		
endereco	VARCHAR(255)	Obrigatório		
telefone	VARCHAR(20)	Obrigatório		
HoraFuncionamento	VARCHAR(100)			

Tabela 2 - Funcionario

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idFuncionario	INTEGER	Obrigatório	Sim	
nome	VARCHAR(255)	Obrigatório		
cargo	VARCHAR(100)			
salario	NUMERIC(10, 2)			
idRestaurante	INTEGER			Sim

Tabela 3 - Cliente

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
cpf	VARCHAR(11)	Obrigatório	Sim	
email	VARCHAR(255)	Obrigatório		
nome	VARCHAR(100)	Obrigatório		
telefone	VARCHAR(20)	Obrigatório		
idFuncionario	INTEGER			Sim

Tabela 4 - Pedido

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idPedido	INTEGER	Obrigatório	Sim	
dataPedido	DATE			
cpf	VARCHAR(11)			Sim

Tabela 6 - Prato

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idPrato	INTEGER	Obrigatório	Sim	
nome	VARCHAR(255)	Obrigatório		
descricao	VARCHAR(255)			Sim
categoria	VARCHAR(50)			
preco	NUMERIC(10, 2)			

Tabela 7 - PedidoPrato

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idPedido	INTEGER	Obrigatório	Sim	Sim
idPrato	INTEGER	Obrigatório	Sim	Sim

Tabela 8 - Pagamento

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idPagamento	INTEGER	Obrigatório	Sim	
valor	VARCHAR(255)	Obrigatório		
idPedido	INTEGER	Obrigatório		Sim

Tabela 9 - Cartao

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idPagamento	INTEGER	Obrigatório	Sim	Sim
numeroCartao	VARCHAR(16)	Obrigatório		

Tabela 9 - Dinheiro

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idPagamento	INTEGER	Obrigatório	Sim	Sim
troco	NUMERIC(10,2)	Obrigatório		

Tabela 9 - Pix

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idPagamento	INTEGER	Obrigatório	Sim	Sim
chave	VARCHAR(50)	Obrigatório		

Tabela 10 - LogPagamento

Atributo	Tipo de Dados	Restrição de Domínio	Restrição de Entidade	Restrição Referencial
idLog	INTEGER	Obrigatório	Sim	
valor	NUMERIC(10, 2)	Obrigatório		
idPedido	INTEGER	Obrigatório		
data	DATE	Obrigatório		
formaPagamento	VARCHAR(50)	Obrigatório		

5. Scripts para criação do banco de dados

```
Unset
CREATE DATABASE restaurante_lab_trabalho;

-- Criação da tabela Restaurante
CREATE TABLE Restaurante (
  idRestaurante SERIAL PRIMARY KEY,
  nome VARCHAR(255) NOT NULL,
  endereco VARCHAR(255) NOT NULL,
  telefone VARCHAR(20) NOT NULL,
  HoraFuncionamento VARCHAR(100)
);
-- Criação da tabela Funcionario
CREATE TABLE Funcionario (
  idFuncionario SERIAL PRIMARY KEY,
  nome VARCHAR(255) NOT NULL,
  cargo VARCHAR(100),
  salario NUMERIC(10, 2),
  IdRestaurante INT,
  FOREIGN KEY (IdRestaurante) REFERENCES Restaurante(idRestaurante));
```

```
-- Criação da tabela Cliente
CREATE TABLE Cliente (
  cpf VARCHAR(11) PRIMARY KEY,
  email VARCHAR(255) NOT NULL,
  nome VARCHAR(255) NOT NULL,
  telefone VARCHAR(20) NOT NULL,
  idFuncionario INT,
  FOREIGN KEY (idFuncionario) REFERENCES Funcionario(idFuncionario)
);


-- Criação da tabela Pedido
CREATE TABLE Pedido (
  idPedido SERIAL PRIMARY KEY,
  dataPedido DATE,
  cpf VARCHAR(11),
  FOREIGN KEY (cpf) REFERENCES Cliente(cpf)
);

-- Criação da tabela Prato
CREATE TABLE Prato (
  idPrato SERIAL PRIMARY KEY,
  nome VARCHAR(255) NOT NULL,
  descricao TEXT,
  categoria VARCHAR(50),
  preco NUMERIC(10, 2)
);

-- Criação da tabela PedidoPrato (tabela de relacionamento
muitos-para-muitos)
CREATE TABLE PedidoPrato (
  idPedido INT,
  idPrato INT,
  PRIMARY KEY (idPedido, idPrato),
  FOREIGN KEY (idPedido) REFERENCES Pedido(idPedido),
  FOREIGN KEY (idPrato) REFERENCES Prato(idPrato)
);
```

```
-- Criação da tabela Pagamento
CREATE TABLE Pagamento (
  idPagamento SERIAL PRIMARY KEY,
  valor NUMERIC(10, 2),
  idPedido INT,
  FOREIGN KEY (idPedido) REFERENCES Pedido(idPedido)
);

-- Criação da tabela Cartao (tabela de pagamento com cartão)
CREATE TABLE Cartao (
  idPagamento INT,
  numeroCartao VARCHAR(16),
  PRIMARY KEY (idPagamento),
  FOREIGN KEY (idPagamento) REFERENCES Pagamento(idPagamento)
);

-- Criação da tabela Pix (tabela de pagamento com Pix)
CREATE TABLE Pix (
  idPagamento INT,
  chave VARCHAR(50),
  PRIMARY KEY (idPagamento),
  FOREIGN KEY (idPagamento) REFERENCES Pagamento(idPagamento)
);

-- Criação da tabela Dinheiro (tabela de pagamento em dinheiro)
CREATE TABLE Dinheiro (
  idPagamento INT,
  troco NUMERIC(10, 2),
  PRIMARY KEY (idPagamento),
  FOREIGN KEY (idPagamento) REFERENCES Pagamento(idPagamento)
);

-- Criação da tabela LogPagamento
CREATE TABLE LogPagamento (
	idLog SERIAL PRIMARY KEY,
	valor NUMERIC(10, 2) NOT NULL,
	idPedido INT NOT NULL,
	data DATE NOT NULL,
	formaPagamento VARCHAR(50) NOT NULL
);
```

6. Scripts de povoamento

Unset

```
INSERT INTO Restaurante (nome, endereco, telefone, HoraFuncionamento)
VALUES
  ('Restaurante BK', '123 Rua Principal, Cidade Três Lagoas', '(123)
456-7890', 'Seg-Sex: 11:00-22:00, Sáb-Dom: 12:00-23:00'),
  ('Restaurante BK', '456 Avenida Secundária, Cidade Andradina', '(456)
789-1230', 'Seg-Sex: 10:30-21:30, Sáb-Dom: 12:00-22:00');

INSERT INTO Funcionario (nome, cargo, salario, IdRestaurante)
VALUES
  ('João Silva', 'Garçom', 2000.00, 1),
  ('Maria Pereira', 'Cozinheira', 2200.00, 1),
  ('Pedro Oliveira', 'Gerente', 3000.00, 1),
  ('Ana Santos', 'Garçom', 2000.00, 2),
  ('Luiz Fernandes', 'Cozinheiro', 2200.00, 2),
  ('Sofia Almeida', 'Garçom', 2000.00, 1),
  ('Lucas Costa', 'Cozinheiro', 2200.00, 2),
  ('Mariana Rodrigues', 'Garçom', 2000.00, 2),
  ('Gustavo Lima', 'Cozinheiro', 2200.00, 1),
  ('Clara Fernandes', 'Garçom', 2000.00, 1);
```

```
INSERT INTO Cliente (cpf, email, nome, telefone, idFuncionario)
VALUES
  ('12345678901', 'cliente1@email.com', 'Maria Silva', '(123) 456-7890',
1),
  ('23456789012', 'cliente2@email.com', 'João Santos', '(234) 567-8901',
2),
  ('34567890123', 'cliente3@email.com', 'Pedro Oliveira', '(345)
678-9012', 3),
  ('45678901234', 'cliente4@email.com', 'Ana Pereira', '(456) 789-0123',
4),
  ('56789012345', 'cliente5@email.com', 'Luiza Rodrigues', '(567)
890-1234', 5),
  ('67890123456', 'cliente6@email.com', 'Miguel Costa', '(678)
901-2345', 6),
  ('78901234567', 'cliente7@email.com', 'Isabella Martins', '(789)
012-3456', 7),
  ('89012345678', 'cliente8@email.com', 'Lucas Fernandes', '(890)
123-4567', 8),
  ('90123456789', 'cliente9@email.com', 'Sophia Gonçalves', '(901)
234-5678', 9),
  ('01234567890', 'cliente10@email.com', 'Guilherme Ribeiro', '(012)
345-6789', 10);
```

```
INSERT INTO Pedido (dataPedido, cpf)
VALUES
  ('2023-11-02', '12345678901'),
  ('2023-11-02', '23456789012'),
  ('2023-11-02', '34567890123'),
  ('2023-11-05', '45678901234'),
  ('2023-11-06', '56789012345'),
  ('2023-11-07', '67890123456'),
  ('2023-11-07', '78901234567'),
  ('2023-11-07', '89012345678'),
  ('2023-11-17', '90123456789'),
  ('2023-11-11', '01234567890'),
  ('2023-11-12', '12345678901'),
  ('2023-11-12', '23456789012'),
  ('2023-11-14', '34567890123'),
  ('2023-11-15', '45678901234'),
  ('2023-11-15', '56789012345'),
  ('2023-11-17', '67890123456'),
  ('2023-11-18', '78901234567'),
  ('2023-11-19', '89012345678'),
  ('2023-11-19', '90123456789'),
  ('2023-11-21', '01234567890');
```

```
INSERT INTO Prato (nome, descricao, categoria, preco)
VALUES
  ('Pizza Margherita', 'Pizza com molho de tomate, queijo e manjericão
fresco.', 'Pizza', 15.99),
  ('Salmão Grelhado', 'Salmão grelhado com legumes e arroz.', 'Peixe',
21.50),
  ('Frango ao Curry', 'Frango cozido com molho de curry e arroz basmati.',
'Frango', 18.99),
  ('Spaghetti Carbonara', 'Massa com ovos, queijo parmesão e bacon.',
'Massa', 12.99),
  ('Salada Caesar', 'Salada com alface, croutons, queijo parmesão e molho
Caesar.', 'Salada', 10.50),
  ('Bife à Parmegiana', 'Bife empanado com queijo e molho de tomate,
acompanhado de espaguete.', 'Carne', 19.50),
  ('Sushi Misto', 'Sushi variado com salmão, atum, camarão e legumes.',
'Sushi', 22.99),
  ('Taco de Carne', 'Taco com carne grelhada, guacamole e molho picante.',
'Mexicano', 9.99),
  ('Molho de Alcachofra', 'Alcachofras, espinafre e queijo servidos com
pão torrado.', 'Aperitivo', 14.99),
  ('Tiramisu', 'Sobremesa de mascarpone, café e cacau.', 'Sobremesa',
7.99),
  ('Hambúrguer Clássico', 'Hambúrguer de carne com queijo, alface e
tomate.', 'Hambúrguer', 11.99),
  ('Creme de Abóbora', 'Sopa de abóbora com creme e temperos.', 'Sopa',
8.99),
  ('Sorvete de Chocolate', 'Sorvete de chocolate com calda de morango.',
'Sobremesa', 6.50),
  ('Caipirinha de Morango', 'Coquetel de cachaça, morango e limão.',
'Bebida', 8.50),
  ('Salada de Frutas', 'Frutas frescas cortadas em pedaços.', 'Salada',
5.99),
  ('Sanduíche de Frango Grelhado', 'Sanduíche com frango grelhado e
vegetais.', 'Sanduíche', 10.99),
  ('Sopa de Lentilha', 'Sopa de lentilha com legumes e temperos.', 'Sopa',
8.99),
  ('Sorvete de Baunilha', 'Sorvete de baunilha com cobertura de
caramelo.', 'Sobremesa', 6.50),
  ('Caipirinha de Limão', 'Coquetel de cachaça, limão e açúcar.',
'Bebida', 8.50),
  ('Torta de Maçã', 'Torta de maçã com uma pitada de canela.', 'Sobremesa',
7.99);
```

```
INSERT INTO PedidoPrato (idPedido, idPrato)
VALUES
  (1, 1),
  (1, 2),
  (2, 3),
  (2, 4),
  (3, 5),
  (3, 6),
  (4, 7),
  (4, 8),
  (5, 9),
  (5, 10);

INSERT INTO Pagamento (valor, idPedido)
VALUES
  (25.00, 1),
  (32.50, 2),
  (19.99, 3),
  (28.75, 4),
  (15.50, 5);

INSERT INTO Cartao (idPagamento, numeroCartao)
VALUES
  (1, '1234567890123456'),
  (2, '2345678901234567');

INSERT INTO Pix (idPagamento, chave)
VALUES
  (3, '4567890ABCDEF123'),
  (4, 'XYZ7890LMNOPQRS');

INSERT INTO Dinheiro (idPagamento, troco)
VALUES
  (5, 2.50);
```

7. Scripts de inserção, remoção e atualização

```
Unset
```

```
INSERT INTO Pagamento (valor, idPedido)
VALUES (25.00, 7);

INSERT INTO Dinheiro (idPagamento, troco)
VALUES (11, 2.50);


INSERT INTO Pedido (cpf)
VALUES ('12345678901');

INSERT INTO Pedido (cpf)
VALUES ('45678901234');

INSERT INTO Restaurante (nome, endereco, telefone)
VALUES ('Restaurante BK', '486 Avenida Do Horizonte, Cidade
Bataguassu', '(456) 789-1230');
```

8. Scripts das Consultas

8.1. Consulta 1

8.1.1. SQL

```
Unset
-- Seleção 1: Listar todos os restaurantes
SELECT *
FROM Restaurante;
```

8.1.2. Resultado

	idrestaurante [PK] integer	nome character varying (255)	endereco character varying (255)	telefone character varying (20)	horafuncionamento character varying (100)
1	1	Restaurante BK	123 Rua Principal, Cidade Três Lagoas	(123) 456-7890	Seg-Sex: 11:00-22:00, Sáb-Dom: 12:00-23:00
2	2	Restaurante BK	456 Avenida Secundária, Cidade Andradina	(456) 789-1230	Seg-Sex: 10:30-21:30, Sáb-Dom: 12:00-22:00

8.2. Consulta 2

8.2.1. SQL

```
Unset
```

```
-- Seleção 2: Listar todos os pedidos e os pratos associados
SELECT Pedido.idPedido, Pedido.dataPedido, Cliente.nome AS nomeCliente,
Prato.nome AS nomePrato
FROM Pedido
	JOIN Cliente ON Pedido.cpf = Cliente.cpf
	JOIN PedidoPrato ON Pedido.idPedido = PedidoPrato.idPedido
	JOIN Prato ON PedidoPrato.idPrato = Prato.idPrato;
```

8.2.2. Resultado

	idpedido integer	datapedido date	nomecliente character varying (255)	nomeprato character varying (255)
1	1	2023-11-02	Maria Silva	Pizza Margherita
2	1	2023-11-02	Maria Silva	Salmão Grelhado
3	2	2023-11-02	João Santos	Frango ao Curry
4	2	2023-11-02	João Santos	Spaghetti Carbonara
5	3	2023-11-02	Pedro Oliveira	Salada Caesar
6	3	2023-11-02	Pedro Oliveira	Bife à Parmegiana
7	4	2023-11-05	Ana Pereira	Sushi Misto
8	4	2023-11-05	Ana Pereira	Taco de Carne
9	5	2023-11-06	Luiza Rodrigues	Molho de Alcachofra
10	5	2023-11-06	Luiza Rodrigues	Tiramisu

8.3. Consulta 3

8.3.1. SQL

Unset

```
-- Seleção 3: Listar os pagamentos com método de pagamento
SELECT Pagamento.idPagamento, Pagamento.valor, Cartao.numeroCartao,
Pix.chave, Dinheiro.troco
	FROM Pagamento
	LEFT JOIN Cartao ON Pagamento.idPagamento = Cartao.idPagamento
	LEFT JOIN Pix ON Pagamento.idPagamento = Pix.idPagamento
	LEFT JOIN Dinheiro ON Pagamento.idPagamento =
	Dinheiro.idPagamento;
```

8.3.2. Resultado

	idpagamento integer	valor numeric (10,2)	numerocartao character varying (16)	chave character varying (50)	troco numeric (10,2)
1	5	15.50	[null]	[null]	2.50
2	2	32.50	2345678901234567	[null]	[null]
3	4	28.75	[null]	XYZ7890LMNOPQRS	[null]
4	1	25.00	1234567890123456	[null]	[null]
5	3	19.99	[null]	4567890ABCDEF123	[null]

8.4. Consulta 4

8.4.1. SQL

Unset

```
-- Seleção 4: Total de vendas por categoria de prato
SELECT Prato.categoria, SUM(Pagamento.valor) AS totalVendas
FROM PedidoPrato
	JOIN Prato ON PedidoPrato.idPrato = Prato.idPrato
	JOIN Pagamento ON PedidoPrato.idPedido = Pagamento.idPedido
GROUP BY Prato.categoria;
```

8.4.2. Resultado

	categoria character varying (50)	totalvendas numeric
1	Mexicano	28.75
2	Sobremesa	15.50
3	Sushi	28.75
4	Frango	32.50
5	Pizza	25.00
6	Aperitivo	15.50
7	Peixe	25.00
8	Massa	32.50
9	Carne	19.99
10	Salada	19.99

8.5. Consulta 5

8.5.1. SQL

```
Unset
-- Seleção 5: Listar clientes e funcionários associados
SELECT Cliente.nome AS nomeCliente, Funcionario.nome AS nomeFuncionario
FROM Cliente
	JOIN Funcionario ON Cliente.idFuncionario =
	Funcionario.idFuncionario;
```

8.5.2. Resultado

	nomecliente character varying (255)	nomefuncionario character varying (255)
1	Maria Silva	João Silva
2	João Santos	Maria Pereira
3	Pedro Oliveira	Pedro Oliveira
4	Ana Pereira	Ana Santos
5	Luiza Rodrigues	Luiz Fernandes
6	Miguel Costa	Sofia Almeida
7	Isabella Martins	Lucas Costa
8	Lucas Fernandes	Mariana Rodrigues
9	Sophia Gonçalves	Gustavo Lima
10	Guilherme Ribeiro	Clara Fernandes

8.6. Consulta 6

8.6.1. SQL

Unset

```
-- Seleção 6: Valor total recebido em cada forma de pagamento
SELECT 'Cartão' AS formaPagamento, SUM(Pagamento.valor) AS
totalPagamento
FROM Cartao
      JOIN Pagamento ON Cartao.idPagamento = Pagamento.idPagamento
UNION
SELECT 'Pix' AS formaPagamento, SUM(Pagamento.valor) AS totalPagamento
FROM Pix
      JOIN Pagamento ON Pix.idPagamento = Pagamento.idPagamento
UNION
SELECT 'Dinheiro' AS formaPagamento, SUM(Pagamento.valor) AS
totalPagamento FROM Dinheiro
      JOIN Pagamento ON Dinheiro.idPagamento = Pagamento.idPagamento;
```

8.6.2. Resultado

	formapagamento text	totalpagamento numeric
1	Cartão	57.50
2	Pix	48.74
3	Dinheiro	15.50

8.7. Consulta 7

8.7.1. SQL

Unset

```
-- Seleção 7: Listar os pratos e suas categorias
SELECT nome, categoria
FROM Prato;
```

8.7.2. Resultado

	nome character varying (255)	categoria character varying (50)
1	Pizza Margherita	Pizza
2	Salmão Grelhado	Peixe
3	Frango ao Curry	Frango
4	Spaghetti Carbonara	Massa
5	Salada Caesar	Salada
6	Bife à Parmegiana	Carne
7	Sushi Misto	Sushi
8	Taco de Carne	Mexicano
9	Molho de Alcachofra	Aperitivo
10	Tiramisu	Sobremesa
11	Hambúrguer Clássico	Hambúrguer
12	Creme de Abóbora	Sopa
13	Sorvete de Chocolate	Sobremesa
14	Caipirinha de Morango	Bebida
15	Salada de Frutas	Salada
16	Sanduíche de Frango Grelhado	Sanduíche
17	Sopa de Lentilha	Sopa
18	Sorvete de Baunilha	Sobremesa
19	Caipirinha de Limão	Bebida
20	Torta de Maçã	Sobremesa

8.8. Consulta 8

8.8.1. SQL

Unset

```
-- Seleção 8: Listar todos os pedidos do mês atual (novembro)
SELECT *
FROM Pedido
WHERE EXTRACT(MONTH FROM datapedido) = EXTRACT(MONTH FROM NOW());
```

8.8.2. Resultado

	idpedido [PK] integer	datapedido date	cpf character varying (11)
1	1	2023-11-02	12345678901
2	2	2023-11-02	23456789012
3	3	2023-11-02	34567890123
4	4	2023-11-05	45678901234
5	5	2023-11-06	56789012345
6	6	2023-11-07	67890123456
7	7	2023-11-07	78901234567
8	8	2023-11-07	89012345678
9	9	2023-11-17	90123456789
10	10	2023-11-11	01234567890
11	11	2023-11-12	12345678901
12	12	2023-11-12	23456789012
13	13	2023-11-14	34567890123
14	14	2023-11-15	45678901234
15	15	2023-11-15	56789012345
16	16	2023-11-17	67890123456
17	17	2023-11-18	78901234567
18	18	2023-11-19	89012345678
19	19	2023-11-19	90123456789
20	20	2023-11-21	01234567890

8.9. Consulta 9

8.9.1. SQL

```
Unset
-- Seleção 9: Valor Médio dos Pratos por Categoria
SELECT categoria, AVG(Prato.preco) AS valorMedio
FROM Prato
GROUP BY categoria;
```

8.9.2. Resultado

	categoria character varying (50)	valormedio numeric
1	Hambúrguer	11.9900000000000000
2	Mexicano	9.9900000000000000
3	Bebida	8.5000000000000000
4	Sobremesa	7.2450000000000000
5	Sushi	22.9900000000000000
6	Frango	18.9900000000000000
7	Pizza	15.9900000000000000
8	Sanduíche	10.9900000000000000
9	Aperitivo	14.9900000000000000
10	Peixe	21.5000000000000000
11	Massa	12.9900000000000000
12	Sopa	8.9900000000000000
13	Carne	19.5000000000000000
14	Salada	8.2450000000000000

8.10. Consulta 10

8.10.1. SQL

```
Unset
-- Seleção 10: Valor Total Recebido por Mês:
SELECT EXTRACT(MONTH FROM dataPedido) AS mes, SUM(valor) AS
totalRecebido
FROM Pagamento
      JOIN Pedido ON Pagamento.idPedido = Pedido.idPedido
GROUP BY mes;
```

8.10.2. Resultado

	mes numeric	totalrecebido numeric
1	11	121.74

8.11. Consulta 11

8.11.1. SQL

```
Unset
-- Seleção 11: Listar o total de clientes que foram a cada
estabelecimento
SELECT r.nome, r.endereco, count(r.endereco) As totalCliente
FROM Restaurante As r
      JOIN Funcionario As f ON r.idRestaurante = f.idRestaurante
      JOIN Cliente As c ON f.idFuncionario = c.idFuncionario
GROUP BY r.nome, r.endereco;
```

8.11.2. Resultado

	nome character varying (255)	endereco character varying (255)	totalcliente bigint
1	Restaurante BK	123 Rua Principal, Cidade Três Lagoas	6
2	Restaurante BK	456 Avenida Secundária, Cidade Andradina	4

9. Scripts das Funções, Stored Procedures e Triggers.

Unset

```
-- Função 1: Calcular o total de vendas para um determinado cliente
CREATE OR REPLACE FUNCTION calcularTotalVendasCliente(cpf_cliente
VARCHAR)
RETURNS NUMERIC(10, 2) AS $$
DECLARE
    total_vendas NUMERIC(10, 2);
BEGIN
    SELECT COALESCE(SUM(valor), 0) INTO total_vendas
    FROM Pagamento
    WHERE idPedido IN (SELECT idPedido FROM Pedido WHERE cpf =
cpf_cliente);

    RETURN total_vendas;
END;
$$ LANGUAGE PLPGSQL;

SELECT calcularTotalVendasCliente('12345678901') AS
totalVendasCliente;

-- Função 2: Listar os pratos mais vendidos
CREATE OR REPLACE FUNCTION pratosMaisVendidos()
RETURNS TABLE (nome_prato VARCHAR(255)) AS $$
BEGIN
    FOR nome_prato IN
        SELECT Prato.nome
        FROM PedidoPrato
        JOIN Prato ON PedidoPrato.idPrato = Prato.idPrato
        GROUP BY Prato.nome
        ORDER BY COUNT(*) DESC
        LIMIT 5
    LOOP
        RETURN NEXT;
    END LOOP;

    RETURN;
END;
$$
LANGUAGE 'plpgsql';
```

```
SELECT * FROM pratosMaisVendidos();

-- Função 3: Calcular a média salarial dos funcionários
CREATE OR REPLACE FUNCTION calcularMediaSalarialFuncionarios()
RETURNS NUMERIC(10, 2) AS $$
DECLARE
    media_salarial NUMERIC(10, 2);
BEGIN
    SELECT COALESCE(AVG(salario), 0) INTO media_salarial
    FROM Funcionario;

    RETURN media_salarial;
END;
$$ LANGUAGE PLPGSQL;

SELECT calcularMediaSalarialFuncionarios() AS
mediaSalarialFuncionarios;

-- Procedure 1: Atualizar o salário de um funcionário
CREATE OR REPLACE PROCEDURE
atualizarSalarioFuncionario(id_funcionario INT,
novo_salarioNUMERIC(10, 2)) AS $$
BEGIN
    UPDATE Funcionario
    SET salario = novo_salario
    WHERE idFuncionario = id_funcionario;
END;
$$ LANGUAGE PLPGSQL;

CALL atualizarSalarioFuncionario(1, 2500.00);
```

```
-- Procedure 2: Remover um cliente
CREATE OR REPLACE PROCEDURE removerCliente(cpf_cliente VARCHAR(11))
AS $$
DECLARE
    id_pedido_cliente INT;
    id_pagamento_cliente INT;
BEGIN
    SELECT idPedido INTO id_pedido_cliente FROM Pedido WHERE cpf =
cpf_cliente;

    SELECT idPagamento INTO id_pagamento_cliente FROM Pagamento WHERE
idPedido = id_pedido_cliente;

    IF EXISTS (SELECT 1 FROM Cartao WHERE idPagamento =
id_pagamento_cliente) THEN
        DELETE FROM cartao WHERE idpagamento = id_pagamento_cliente;
    END IF;

    IF EXISTS (SELECT 1 FROM Pix WHERE idPagamento =
id_pagamento_cliente) THEN
        DELETE FROM pix WHERE idpagamento = id_pagamento_cliente;
    END IF;

    IF EXISTS (SELECT 1 FROM Dinheiro WHERE idPagamento =
id_pagamento_cliente) THEN
        DELETE FROM dinheiro WHERE idpagamento =
id_pagamento_cliente;
    END IF;

    DELETE FROM PedidoPrato WHERE idPedido = id_pedido_cliente;

    DELETE FROM Pagamento WHERE idPedido = id_pedido_cliente;

    DELETE FROM pedido where cpf = cpf_cliente;

    DELETE FROM Cliente WHERE cpf = cpf_cliente;
END;
$$
LANGUAGE 'plpgsql';

CALL removerCliente('12345678901');
```

```
-- Procedure 3: Adicionar um novo prato
CREATE OR REPLACE PROCEDURE adicionarPrato(
    nome_prato VARCHAR(255),
    descricao_prato TEXT,
    categoria_prato VARCHAR(255),
    preco_prato DECIMAL(10, 2)
)
AS $$
BEGIN
    INSERT INTO Prato (nome, descricao, categoria, preco)
    VALUES (nome_prato, descricao_prato, categoria_prato,
preco_prato);
END;
$$
LANGUAGE 'plpgsql';

CALL adicionarPrato('Macarrão à Bolonhesa', 'Macarrão com molho à
bolonhesa', 'Massa', 14.99);


-- Trigger 1: Registrar log ao inserir um pagamento
CREATE OR REPLACE FUNCTION registrarLogPagamento()
RETURNS TRIGGER AS $$
BEGIN

      INSERT INTO LogPagamento (valor, idPedido, data)
      VALUES (NEW.valor, NEW.idPedido, CURRENT_DATE);
      RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER pagamento_after_insert
AFTER INSERT ON Pagamento
FOR EACH ROW
EXECUTE FUNCTION registrarLogPagamento();
```

```
-- Trigger 2: Atualizar data de pedido ao inserir um prato em um
pedido
CREATE OR REPLACE FUNCTION atualizarDataPedido()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE Pedido
    SET dataPedido = CURRENT_DATE
    WHERE idPedido = NEW.idPedido;
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER Pedido_after_insert
AFTER INSERT ON Pedido
FOR EACH ROW
EXECUTE FUNCTION atualizarDataPedido();

-- Trigger 3: Atualizar hora de funcionamento ao inserir um novo
restaurante
CREATE OR REPLACE FUNCTION atualizarHoraFuncionamentoRestaurante()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE Restaurante
    SET HoraFuncionamento = 'Seg-Sex: 10:30-21:30, Sáb-Dom:
12:00-22:00'
    WHERE idRestaurante = NEW.idRestaurante;
    RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER restaurante_after_insert
AFTER INSERT ON Restaurante
FOR EACH ROW
EXECUTE FUNCTION atualizarHoraFuncionamentoRestaurante();
```

10. Conclusão

Em resumo, o desenvolvimento do banco de dados para o "Sabor do Rio" é um passo sólido em direção à gestão eficiente e integrada do restaurante. Inspirado no modelo da rede Burger King, cuidamos para elaborar um esquema conceitual e lógico robusto, garantindo que os dados estejam sempre íntegros. A flexibilidade do design permite que o banco de dados evolua dinamicamente para lidar com as complexidades do ambiente do "Sabor do Rio". Este não é apenas um repositório de informações transacionais, mas uma ferramenta essencial para análises estratégicas e tomada de decisões embasadas. Assim, contribuímos para o aprimoramento contínuo das operações do restaurante, proporcionando uma base sólida para o sucesso duradouro do "Sabor do Rio".