



**INSTITUTO FEDERAL**  
Espírito Santo

# **Trabalho de Banco de Dados :**

## **Cardápio QR Code**

Grupo: Renzo Fraga, João Vitor Ferrareis, Perseu Fernandes, Felipe Beccali



# Motivação e Minimundo

**-Restaurantes 4.0, Criatividade, Inovação, Usabilidade e Melhoria de Atendimento.**

## Motivação:

- Inovação e introdução ao mundo 4.0.
- Experiência gastronômica como fusão de criatividade, comodidade e interatividade.
- Cardápio QR Code para explorar opções gastronômicas de forma rápida, fácil e segura.
- Atendimento a pedidos aprimorado e otimização das tarefas na cozinha.

## Minimundo:

- O restaurante possui um nome, código de identificação e um Admin responsável pelo gerenciamento.
- Existem clientes que possuem nome, email e telefone.
- O Cardápio é composto por itens que têm nome, descrição e preço, e categorias que possuem o nome da categoria.
- Cada mesa tem um código de identificação e um QR Code associado.
- O Admin pode criar ou destruir QR Codes, alterar a associação entre QR Codes e mesas, e modificar o cardápio.
- A Cozinha recebe os pedidos dos clientes e os prepara.
- Os clientes acessam o cardápio através do QR Code, fazem pedidos e realizam o pagamento.
- O pedido é enviado para a Cozinha, juntamente com o número da mesa.



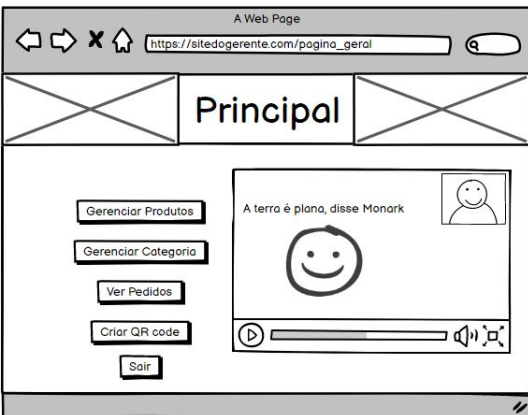
# Prototipação e Tabela de Dados.

Prototipação:

<https://github.com/FeliXp90/CardapioQRCode/blob/master/arquivos/PrototipacaoCardapioQRCode.pdf>

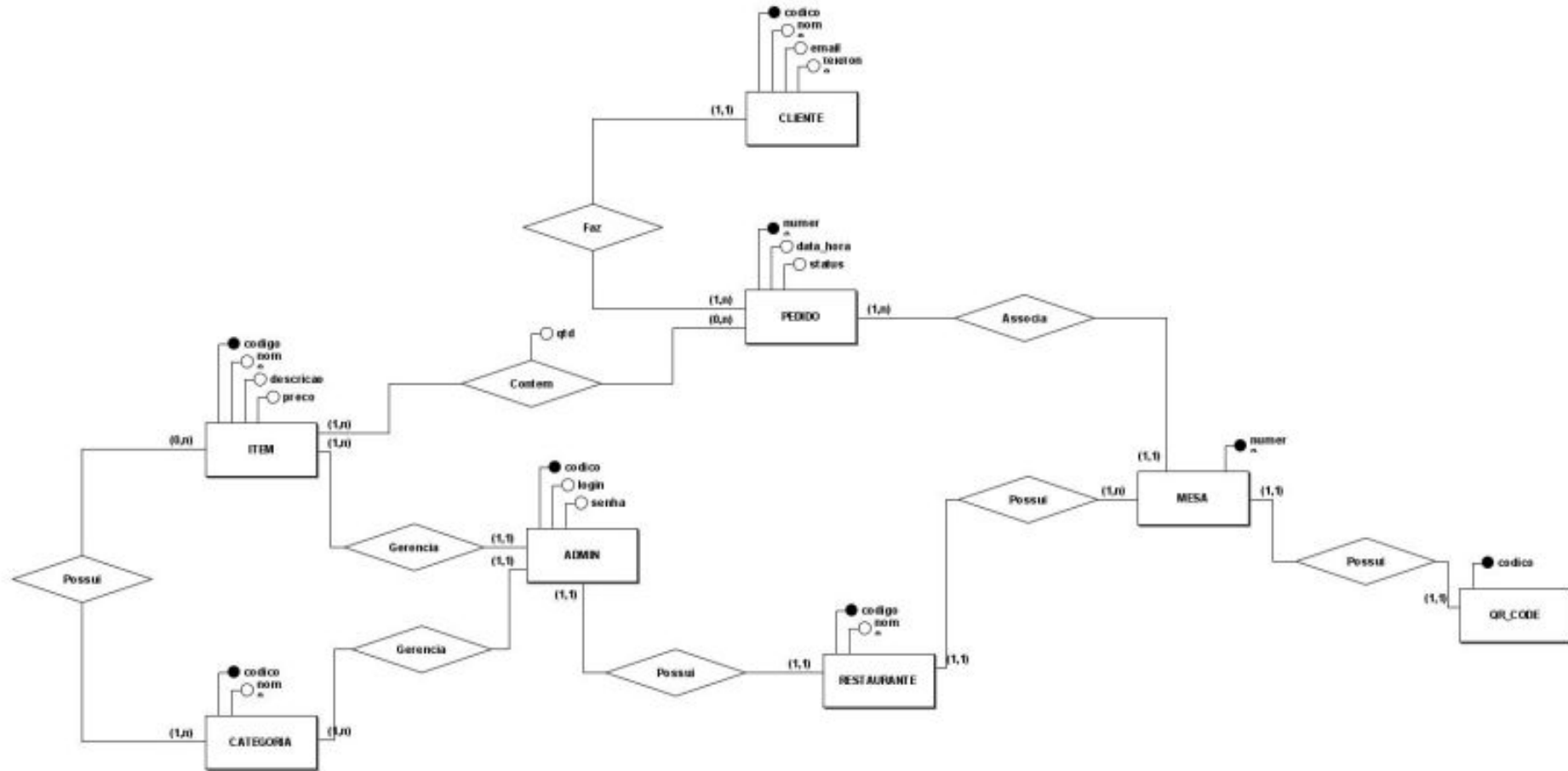
Tabela de dados:

[https://docs.google.com/spreadsheets/d/1n-Mo7cHK3E\\_brFU1ScfYX4t0fWqM9vOJ039ClqLGFXo/edit#gid=0](https://docs.google.com/spreadsheets/d/1n-Mo7cHK3E_brFU1ScfYX4t0fWqM9vOJ039ClqLGFXo/edit#gid=0)

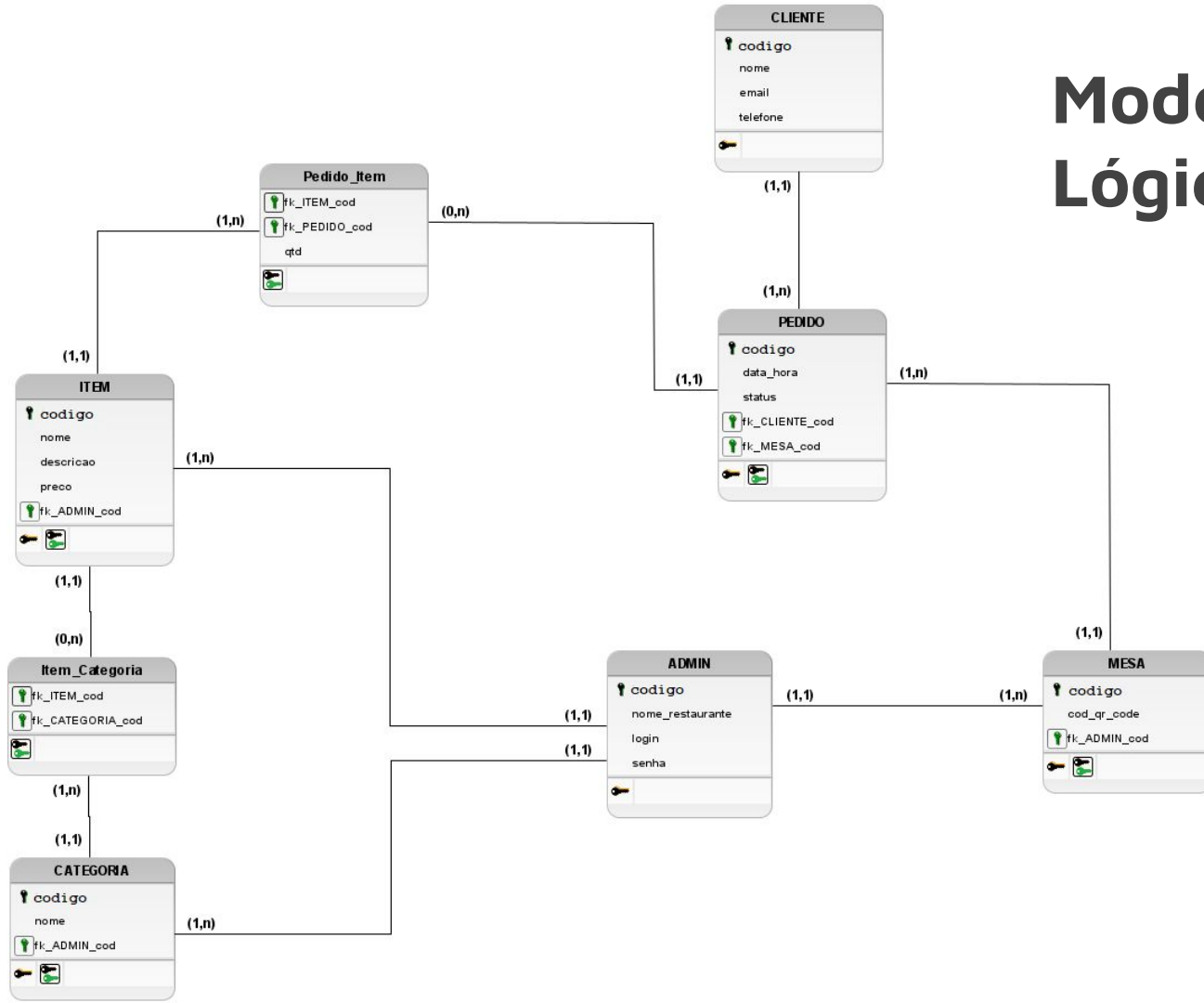


CLIENTE			
Código	Nome	Telefone	Email
1	Wagner Moura	98765-4321	wagnermoura@gmail.com
2	Gabriel Pensado	93456-7800	gabriel@gmail.com
3	Romeu Banhos	91345-6789	romeu@gmail.com
4	Eliseu Faria	92145-6789	eliseu@gmail.com
5	Bartolomeu Brag	92345-6789	bartolomeu@gmail.com
6	Carlos Roberto	94312-6789	carlos@gmail.com
7	João Luiz	93258-1234	joaoluiz@gmail.com
8	Hannah Montann	90028-9220	hannymtn@gmail.com
9	Perseu Fraga	98760-1234	perseufmofraga@gmail.com
9	Felipe Ferrareis	93114-1234	ferrareisfbt@gmail.com

# Modelo Conceitual:



# Modelo Lógico:





## Modelo Físico:

```
CREATE TABLE CLIENTE(  
  CODIGO int,  
  NOME VARCHAR(45),  
  TELEFONE VARCHAR(45),  
  EMAIL VARCHAR(45));
```

```
CREATE TABLE PEDIDO(  
  CODIGO int,  
  DATA_HORA timestamp,  
  STATUS VARCHAR(45),  
  FK_CLIENTE_COD int,  
  FK_MESA_COD int);
```

```
CREATE TABLE PEDIDO_ITEM(  
  FK_PEDIDO_COD int,  
  FK_ITEM_COD int,  
  QUANTIDADE int);
```

```
CREATE TABLE ITEM(  
  CODIGO int,  
  NOME VARCHAR(45),  
  DESCRICAO VARCHAR(45),  
  PRECO real  
);
```

```
CREATE TABLE MESA(  
  CODIGO int,  
  COD_QR_CODE VARCHAR(45)  
);
```

```
CREATE TABLE ADMIN(  
  CODIGO int,  
  NOME_RESTAURANTE VARCHAR(45),  
  LOGIN VARCHAR(45),  
  SENHA VARCHAR(45)  
);
```

```
CREATE TABLE CATEGORIA(  
  CODIGO int,  
  NOME VARCHAR(45)  
);
```

```
CREATE TABLE ITEM_CATEGORIA(  
  FK_ITEM_COD int,  
  FK_CATEGORIA_COD int  
);
```

# Inserts:

```
INSERT INTO CLIENTE(CODIGO,NOME,TELEFONE,EMAIL)  
VALUES
```

```
(1,'Wagner Moura','98765-4321','wagnermoura@gmail.com'),
```

```
(2,'Gabriel Pensador','93456-7800','gabriel@gmail.com'),
```

```
(3,'Romeu Banhos','91345-6789','romeu@gmail.com'),
```

```
(4,'Eliseu Faria','92145-6789','eliseu@gmail.com'),
```

```
(5,'Bartolomeu Bragança','92345-6789','bartolomeu@gmail.com'),
```

```
(6,'Carlos Roberto','94312-6789','carlos@gmail.com'),
```

```
INSERT INTO MESA(CODIGO,COD_QR_CODE)  
VALUES
```

```
(1,'/cardapio/mesa=1'),
```

```
(2,'/cardapio/mesa=2'),
```

```
(3,'/cardapio/mesa=3'),
```

```
INSERT INTO CATEGORIA(CODIGO,NOME) VALUES
```

```
(1,'Carnes'), (2,'Petiscos'), (3,'Bebidas'),
```

```
INSERT INTO
```

```
PEDIDO(CODIGO,DATA_HORA,STATUS,FK_CLIENTE_COD,F  
K_MESA_COD) VALUES
```

```
(1,'11/04/2023 16:03:00 -3:00','Entregue',2,2),
```

```
(2,'11/04/2023 16:10:00 -3:00','Entregue',1,3),
```

```
(3,'11/04/2023 16:11:00 -3:00','Entregue',5,1),
```

```
INSERT INTO PEDIDO_ITEM
```

```
(FK_PEDIDO_COD,FK_ITEM_COD,QUANTIDADE) VALUES
```

```
(1,2,1), (1,6,2), (2,3,1)
```

```
INSERT INTO ITEM(CODIGO,NOME,DESCRICAO,PRECO) VALUES
```

```
(1,'Coraçãozinho','200g de coração de galinha',10.00),
```

```
(2,'Picanha','200g de picanha',12.00),
```

```
(3,'Alcatra','200g de alcatra',11.00),
```

```
INSERT INTO ADMIN
```

```
(CODIGO,NOME_RESTAURANTE,LOGIN,SENHA) VALUES
```

```
(1,'Rei da Picanha','Admin_Luiz','Flamengo123');
```

```
INSERT INTO ITEM_CATEGORIA
```

```
(FK_ITEM_COD,FK_CATEGORIA_COD) VALUES
```

```
(1,1), (2,1), (3,1),
```

```
SELECT cliente.nome,telefone,data_hora,status,descricao

FROM cliente INNER JOIN pedido

ON pedido.fk_cliente_cod = cliente.codigo

INNER JOIN pedido_item

ON pedido.codigo=pedido_item.fk_pedido_cod

INNER JOIN item

ON pedido_item.fk_item_cod=item.codigo

INNER JOIN item_categoria

ON item_categoria.fk_item_cod=item.codigo

INNER JOIN categoria

ON categoria.codigo=fk_categoria_cod

INNER JOIN admin

ON admin.codigo=categoria.codigo

INNER JOIN mesa

ON pedido.fk_mesa_cod=mesa.codigo

ORDER BY cliente.telefone DESC;
```

# Consulta Inner Join 1: Visão da Cozinha do Sistema.

nome	telefone	data_hora		status	descricao
Tatiana Rocha	98765-4322	2023-12-07	17:00:00 +0000	Entregue	200g de picanha
Wagner Moura	98765-4321	2023-11-04	16:10:00 +0000	Entregue	200g de alcatra
Wagner Moura	98765-4321	2023-11-04	16:40:00 +0000	Preparando...	Prato de feijoada completa
Renato Souza	94312-6784	2023-12-08	16:50:00 +0000	Entregue	200g de alcatra
Giovanni Barbosa	93114-1237	2023-12-12	17:20:00 +0000	Preparando...	Porção de tilápia grelhada
Giovanni Barbosa	93114-1237	2023-12-12	17:20:00 +0000	Preparando...	200g de coração de galinha
Carolina Mendes	92345-6719	2023-12-04	17:30:00 +0000	Entregue	Porção de linguiça toscana



```
SELECT cliente.nome,data_hora,status,fk_mesa_cod as
mesa,quantidade,item.nome as item
FROM CLIENTE cliente
JOIN PEDIDO pedido
ON cliente.codigo=pedido.fk_cliente_cod

JOIN MESA mesa
ON pedido.fk_mesa_cod=mesa.codigo

JOIN PEDIDO_ITEM pedido_item
ON pedido_item.fk_pedido_cod=pedido.codigo

JOIN ITEM item
ON item.codigo=pedido_item.fk_item_cod

ORDER BY CASE
  WHEN status = 'Preparando...' THEN 0
  WHEN status = 'Entregue' THEN 1
  ELSE 2
END;
```

## Consulta Inner Join 2:

### Visão melhorada da Cozinha:

nome	data_hora	status	mesa	quantidade	item
Ricardo Pereira	2023-12-06 17:20:00 +0000	Preparando...	4	2	Bife à Parmegiana
Mariana Souza	2023-12-04 16:30:00 +0000	Preparando...	4	2	Pastel de Queijo
Rafaela Almeida	2023-12-04 17:10:00 +0000	Preparando...	4	2	Mousse de Maracujá
Giovanni Barbosa	2023-12-12 17:20:00 +0000	Preparando...	4	2	Tilápia Grelhada

# Funções de Agrupamento:



## Consultas com Group by:

Query 1:

```
SELECT  
FK_ITEM_COD,COUNT(FK_ITEM_COD) AS  
QTD FROM PEDIDO_ITEM GROUP BY  
FK_ITEM_COD ORDER BY QTD DESC;
```

Query 2:

```
SELECT FK_PEDIDO_COD FROM  
PEDIDO_ITEM GROUP BY  
FK_PEDIDO_COD;
```

Query 1:

	cliente_id	qtd
0	1	5
1	4	4
2	10	4
3	5	4
4	6	3
5	2	3
6	29	2
7	40	2
8	32	2

Query 2:

fk_pedido_cod
42
29
4
34

# Funções de Agrupamento:



## Consultas com Group by:

Query 3:

```
SELECT C.CODIGO, C.NOME, SUM(I.PRECO * PI.QUANTIDADE) AS  
TOTALGASTO  
  
FROM CLIENTE C  
  
JOIN PEDIDO P ON C.CODIGO = P.FK_CLIENTE_COD  
  
JOIN PEDIDO_ITEM PI ON p.CODIGO = PI.FK_PEDIDO_COD  
  
JOIN ITEM I ON PI.FK_ITEM_COD = I.CODIGO  
  
GROUP BY C.CODIGO, C.NOME  
  
ORDER BY TOTALGASTO DESC;
```

Query 3:

<b>codigo</b>	<b>nome</b>	<b>totalgasto</b>
40	Giovanni Barbosa	86.0
14	Gustavo Pereira	60.0
29	Leticia Ribeiro	56.0
2	Gabriel Pensador	54.0
4	Eliseu Faria	53.0
8	Hannah Montanna	49.5
3	Romeu Banhos	43.5
9	Perseu Fraga	40.5
15	Carolina Mendes	39.0

# Funções de Agrupamento:



## Consultas com Group by:

Query 5:

```
SELECT PI.FK_ITEM_COD AS CLIENT_ID, I.NOME AS ITEM,  
SUM(PI.QUANTIDADE) AS TOTAL_VENDIDO  
FROM PEDIDO_ITEM PI  
JOIN ITEM I ON PI.FK_ITEM_COD = I.CODIGO  
GROUP BY CLIENT_ID, I.NOME  
ORDER BY TOTAL_VENDIDO DESC;
```

Query 5:

client_id	item	total_vendido
4	Pão de Alho	11
10	Cerveja Skol	8
5	Coca 2L	8
1	Coraçãozinho	8
33	Cachorro-Quente	6
2	Picanha	5
6	Cerveja Brahma	5
7	Dollynho	5
25	Linguiça Toscana	4

```
SELECT cliente.nome as nome_cliente,pedido.status,item.nome
FROM CLIENTE cliente
FULL JOIN PEDIDO pedido
ON cliente.codigo=pedido.fk_cliente_cod
JOIN PEDIDO_ITEM pedido_item
ON pedido.fk_cliente_cod=pedido_item.fk_pedido_cod
JOIN ITEM item
ON pedido_item.fk_ITEM_cod=item.codigo
where cliente.nome='Wagner Moura';
```

Consulta que use Right,  
Self, Left, ou Full join:

nome_cliente	status	nome
Wagner Moura	Preparando...	Picanha
Wagner Moura	Entregue	Picanha
Wagner Moura	Preparando...	Cerveja Brahma
Wagner Moura	Entregue	Cerveja Brahma



## Select com View com subconsulta

```
CREATE VIEW View3 AS
SELECT nome, total_pedidos
FROM CLIENTE
JOIN (
  SELECT FK_CLIENTE_COD, COUNT(*) AS
  total_pedidos
  FROM PEDIDO
  GROUP BY FK_CLIENTE_COD
) SUBC
ON CLIENTE.CODIGO = SUBC.FK_CLIENTE_COD;
```

nome	total_pedidos
Bruno Costa	1
Ana Silva	1
Giovanni Barbosa	1
Perseu Fraga	2
Carolina Mendes	1



# Perguntas a Serem Respondidas:

Relatório 1: Quais são os Pratos mais pedidos?

Relatório 2: Quais são os Pratos que geram mais receita?

Relatório 3: Quais são os pedidos que ainda estão sendo Preparados? E quais já estão Prontos?

Relatório 4: Quais são os Clientes mais fiéis do restaurante? (Os que fizeram mais pedidos)

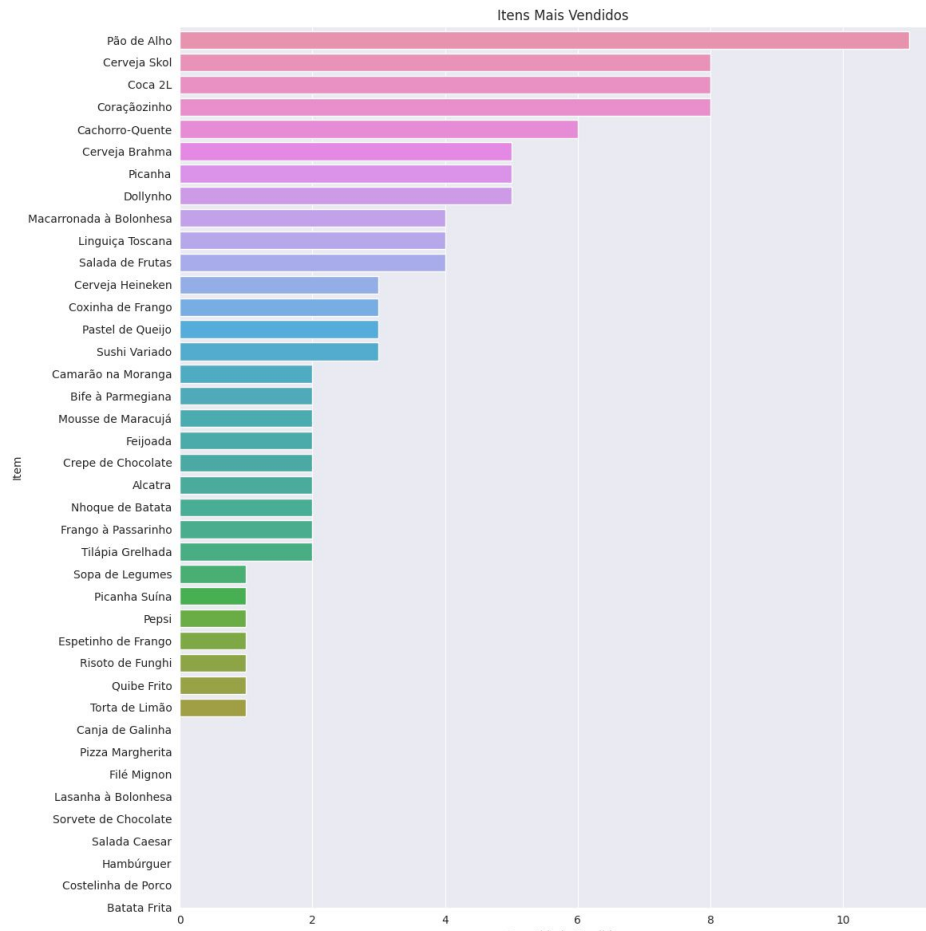
Relatório 5: Quais são os Clientes que já gastaram mais dinheiro no restaurante?

Relatório 6: Quais são as categorias de itens mais pedidos?

# Quais são os Pratos Mais vendidos??

```
res = pd.read_sql_query("""SELECT I.CODIGO, I.NOME AS ITEM,  
COALESCE(SUM(PI.QUANTIDADE), 0) AS TOTAL_VENDIDO  
FROM ITEM I  
LEFT JOIN PEDIDO_ITEM PI ON I.CODIGO = PI.FK_ITEM_COD  
GROUP BY I.CODIGO, I.NOME  
ORDER BY TOTAL_VENDIDO DESC;""", conn)  
res
```

```
sns.set_style('darkgrid')  
plt.figure(figsize=(12, 12))  
res_sorted = res.sort_values(by='total_vendido', ascending=True)  
sns.barplot(x='total_vendido', y='item', data=res_sorted)  
plt.xlabel('Quantidade Vendida')  
plt.ylabel('Item')  
plt.ylim(bottom=0, top=max(res['total_vendido']) * 3.6)  
plt.title('Itens Mais Vendidos')  
plt.tight_layout()  
plt.show()
```

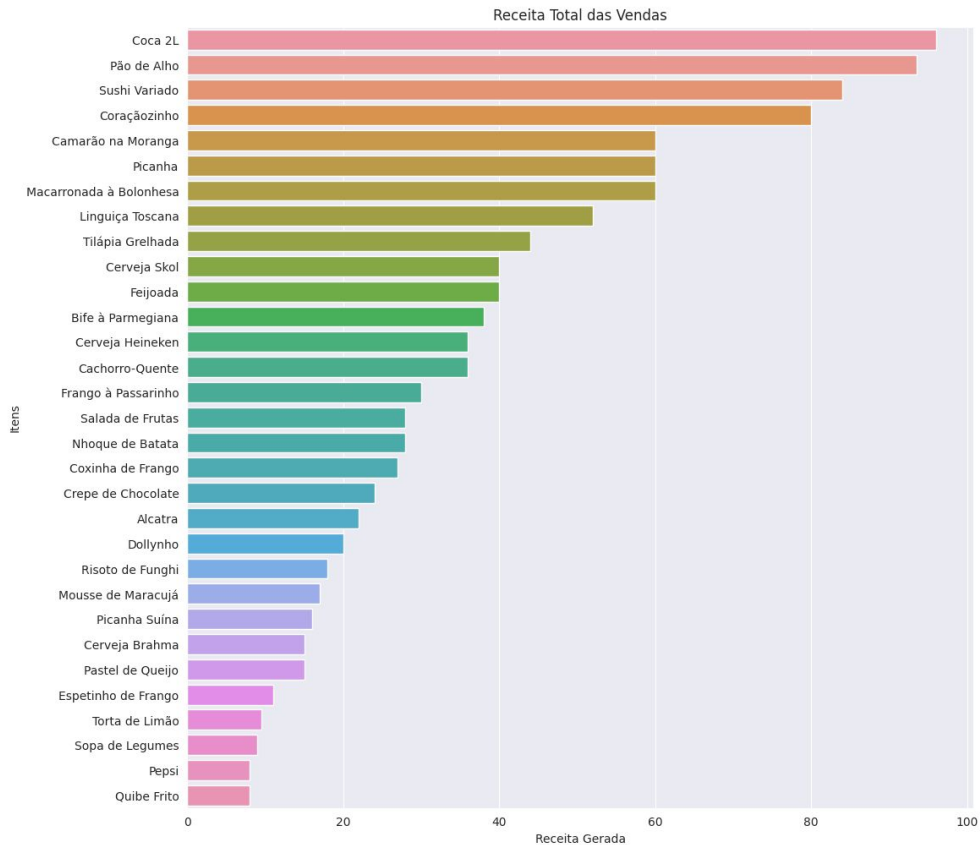




# Quais são os Pratos que geram mais receita?

```
res = pd.read_sql_query("""SELECT I.CODIGO,  
I.NOME AS ITEM, SUM(I.PRECO * PI.QUANTIDADE)  
AS RECEITA_TOTAL  
FROM PEDIDO_ITEM PI  
JOIN ITEM I ON PI.FK_ITEM_COD = I.CODIGO  
GROUP BY I.CODIGO, ITEM  
ORDER BY RECEITA_TOTAL DESC;""", conn)  
res
```

```
sns.set_style('darkgrid')  
res_sorted = res.sort_values(by='receita_total',  
ascending=False)  
plt.figure(figsize=(12, 12))  
sns.barplot(x='receita_total', y='item', data=res_sorted)  
plt.xlabel('Receita Gerada')  
plt.ylabel('Itens')  
plt.title('Receita Total das Vendas')  
plt.show()
```



```

res = pd.read_sql_query("""SELECT status,fk_mesa_cod
as mesa,quantidade,item.nome as item
FROM CLIENTE cliente
JOIN PEDIDO pedido
ON cliente.codigo=pedido.fk_cliente_cod

JOIN MESA mesa
ON pedido.fk_mesa_cod=mesa.codigo

JOIN PEDIDO_ITEM pedido_item
ON pedido_item.fk_pedido_cod=pedido.codigo

JOIN ITEM item
ON item.codigo=pedido_item.fk_item_cod

ORDER BY CASE
  WHEN status = 'Preparando...' THEN 0
  WHEN status = 'Entregue' THEN 1
  ELSE 2
END;""",conn)
res

```

## Quais são os Pedidos que ainda estão em Preparo? E quais já estão prontos?

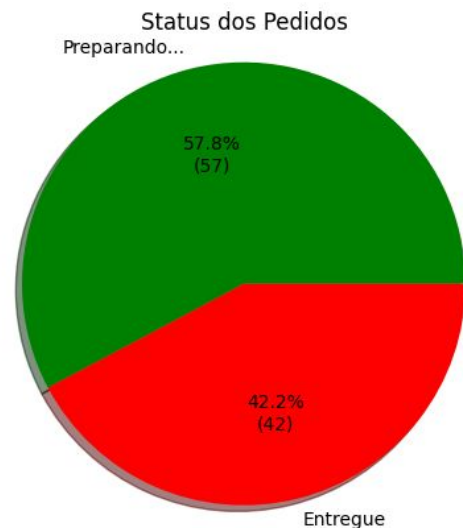
```

status_counts = res['status'].value_counts()
labels = status_counts.index
data = status_counts.values
my_colors = ['lightblue', 'silver']

def label_format(value):
    count = int(value)
    percent = '{:.1f}%'.format(value)
    return '{}\n({})'.format(percent, count)

plt.pie(data, labels=labels,
autopct=label_format, colors=my_colors,
shadow=True)
plt.title('Status dos Pedidos')
plt.axis('equal')
plt.show()

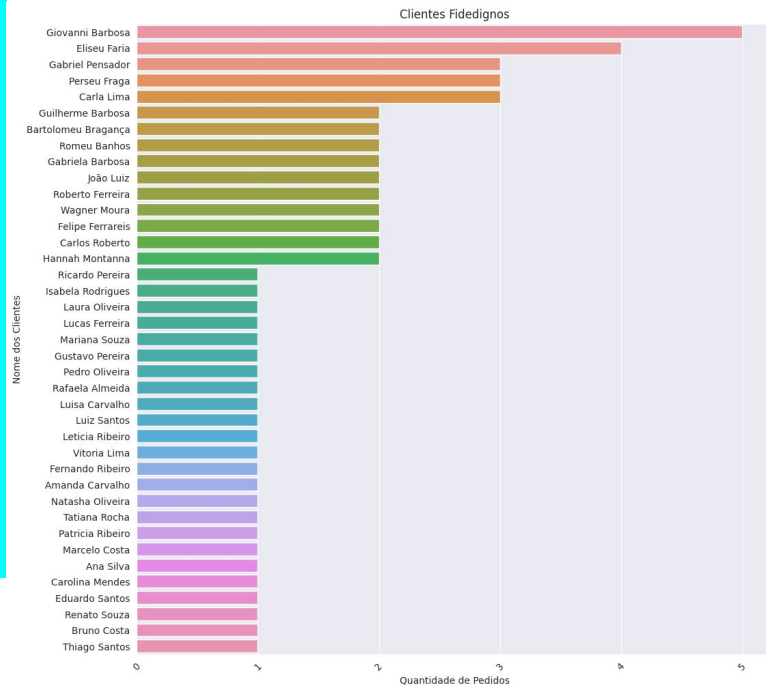
```



# Quais são os clientes mais fiéis?

```
res = pd.read_sql_query("""SELECT cliente.nome AS
Nome_Cliente, COUNT(pedido.codigo) AS
Total_Pedidos
FROM cliente
JOIN pedido ON cliente.codigo =
pedido.fk_cliente_cod
JOIN pedido_item ON pedido.codigo =
pedido_item.fk_pedido_cod
JOIN item ON item.codigo = pedido_item.fk_item_cod
JOIN item_categoria ON item.codigo =
item_categoria.fk_item_cod
JOIN categoria ON item_categoria.fk_categoria_cod =
categoria.codigo
GROUP BY cliente.nome
ORDER BY Total_Pedidos DESC;""",conn)
res
```

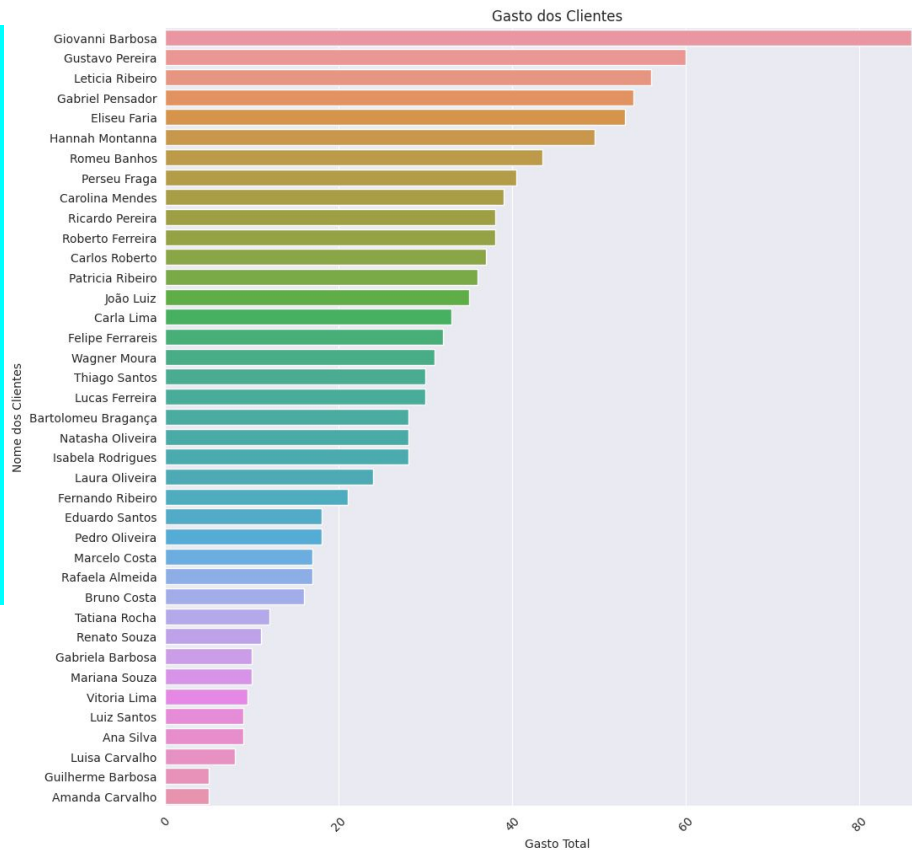
```
sns.set_style('darkgrid')
plt.figure(figsize=(12, 12))
plt.xticks(rotation=45)
sns.barplot(x='total_pedido', y='nome_cliente', data=res,)
plt.xlabel('Quantidade de Pedidos')
plt.ylabel('Nome dos Clientes')
plt.title('Clientes Fidedignos')
plt.show()
```



# Quais clientes já gastaram mais dinheiro no restaurante?

```
res = pd.read_sql_query("""SELECT
cliente.nome AS Nome_Cliente,
SUM(item.preco * pedido_item.quantidade)
AS Total_Gasto
FROM cliente
JOIN pedido ON cliente.codigo =
pedido.fk_cliente_cod
JOIN pedido_item ON pedido.codigo =
pedido_item.fk_pedido_cod
JOIN item ON item.codigo =
pedido_item.fk_item_cod
GROUP BY cliente.nome
ORDER BY Total_Gasto DESC;""",conn)
res
```

```
sns.set_style('darkgrid')
plt.figure(figsize=(12, 12))
plt.xticks(rotation=45)
sns.barplot(y='nome_cliente',x='total_gasto',data=res)
plt.xlabel('Gasto Total')
plt.ylabel('Nome dos Clientes')
plt.title('Gasto dos Clientes')
plt.show()
```



# Quais são as categorias de itens mais vendidas?

```
res = pd.read_sql_query("""SELECT
categoria.nome AS Nome_Categoria,
SUM(pedido_item.quantidade) AS
Quantidade_Itens_Pedidos
FROM pedido
JOIN pedido_item ON pedido.codigo =
pedido_item.fk_pedido_cod
JOIN item ON item.codigo =
pedido_item.fk_item_cod
JOIN item_categoria ON item.codigo =
item_categoria.fk_item_cod
JOIN categoria ON
item_categoria.fk_categoria_cod =
categoria.codigo
GROUP BY categoria.nome;""",conn)
res
```

```
sns.set_style('darkgrid')
plt.figure(figsize=(12, 12))
res_sorted =
res.sort_values(by='quantidade_ite
ns', ascending=False)
plt.xticks(rotation=45)
sns.barplot(y='nome_categoria',x='quantidade
_ite
ns_pedidos',data=res_sorted)
plt.xlabel('Quantidade')
plt.ylabel('Nome da Categoria')
plt.title('Quantidade de Pedidos por
Categoria')
plt.show()
```

