



Documentação

O Sistema de Comandas é uma aplicação backend desenvolvida em Python, com foco em facilitar o controle de consumo em estabelecimentos como bares, restaurantes, salões de beleza, eventos e etc.

O sistema permite o **cadastro de itens, a criação de comandas individuais, o registro de produtos consumidos e o gerenciamento de pagamentos,** fornecendo uma visão clara e atualizada do valor total de cada comanda.

Funcionalidades:

- **Cadastro de Clientes:** Cadastro de clientes por nome.
- **Cadastro de produtos:** Cadastro de produtos com o nome e o valor.
- **Comandas por cliente:** Comanda criada por clientes constando o que foi consumido e o valor pago pelo cliente.
- **Cálculo do total da comanda:** Cálculo total de cada comanda.
- **Pagamento parcial ou total :** Pagamento fazendo o controle do valor da comanda e gerando um qr code para pagamento.

Models:

Clientes:

Campos: id=Integer, primary_key, autoincrement
nome = String(80), Not Null
telefone = String(11)

Funções : to_dict para retorno de todos os campos
adicionar_comanda para adicionar uma comanda ao cliente.

Produtos:

Campos: id = Integer, primary_key
nome = String(80), Not Null
preco =Float, Not Null

Funções: to_dict para retorno de todos os campos.

Comandas:

Campos: id = Integer, primary_key
data = String(10)
cliente_id = Integer, ForeignKey('cliente_id')

cliente = relacionamentos com a tabela clientes

Funções: to_dict para retorno de todos os campos

adicionar_produtos para adicionar produtos na comanda onde
eles não estejam com a quantidade menor ou igual a zero

calcula_total para calcular o total da comanda

Comandas Produtos:

Campos: comanda_id = Integer, ForeignKey('comanda.id'), primary_key
produto_id = Integer, ForeignKey('produto.id'), primary_key
quantidade= .Integer, Not Null

Funções: to_dict para retorno de todos os campos

Pagamento:

Campos: id = Integer, Primary_key
value = Float
comanda_id = Integer, ForeignKey('comanda.id'))
paid = Boolean
bank_payment_id = String(200)
qr_code = String(100)
expiration_date = DateTime

Funções: to_dict para retorno de todos os campos

Rotas

1- Clientes

POST /clientes

Responsabilidade: Criar um novo Cliente

payload:

```
{  
  "nome": "string",
```

```
"telefone": "integer"
}
```

retorno:

```
200

{"message": "Cliente cadastrado com sucesso",
 "cliente": {
   "id": novo_cliente.id,
   "nome": novo_cliente.nome,
   "telefone": novo_cliente.telefone }}
```

GET /clientes

Responsabilidade: Listar todos os clientes

retorno:

```
200

[]

{
  "id": cliente.id,
  "nome": cliente.nome,
  "telefone": cliente.telefone
}
```

GET /clientes/{id}

Responsabilidade: Listar apenas um cliente

retorno:

```
200
{
  "id": cliente.id,
```

```
"nome": cliente.nome,  
"telefone": cliente.telefone  
}
```

404

```
{"message": "Cliente não encontrado"}
```

PUT /clientes/{id}

Responsabilidade: Editar informações de um cliente

payload:

```
{  
  "nome": "string",  
  "telefone": "integer"  
}
```

retorno:

200

```
{ "message": "Cliente atualizado com sucesso",  
  "cliente": {  
    "id": cliente.id,  
    "nome": cliente.nome,  
    "telefone": cliente.telefone  
  }  
}
```

404

```
{"message": "Cliente não encontrado"}
```

DELETE /clientes/{id}

Método: DELETE

Responsabilidade: Deletar um cliente

retorno:

```
200
{"message":"Cliente deletado com sucesso"}

404
{"message":"Cliente não encontrado"}
```

2- Produtos

POST /produtos

Responsabilidade: Criar um novo Produto

payload:

```
{
  "nome": "string",
  "preco ": "integer"
}
```

retorno:

```
200

{"message":"Produto cadastrado com sucesso",
"cliente":{
  "id": novo_produto.id,
  "nome": novo_produto.nome,
  "preco ": novo_produto.preco}}
```

GET /produtos

Responsabilidade: Listar todos os produtos

retorno:

```
200
```

```
[]
```

```
{  
  "id": produto.id,  
  "nome": produto.nome,  
  "preco": produto.preco  
}
```

GET /produtos/{id}

Responsabilidade: Listar apenas um produto

retorno:

```
200
```

```
{  
  "id": produto.id,  
  "nome": produto.nome,  
  "preco": produto.preco  
}
```

```
404
```

```
{"message": "Produto não encontrado"}
```

PUT /produtos/{id}

Responsabilidade: Editar informações de um produto

payload:

```
{  
  "nome": "string",  
  "preco": "float"  
}
```

retorno:

```
200
{ "message": "Cliente atualizado com sucesso",
  "produto": {
    "id": produto.id,
    "nome": produto.nome,
    "preco": produto.preco}
}

404
{"message":"Produto não encontrado"}
```

DELETE /produtos/{id}:

Responsabilidade: Deletar um produto

retorno:

```
200
{"message":"Produto deletado com sucesso"}

404
{"message":"Produto não encontrado"}
```

3- Comandas

POST /comandas

Responsabilidade: Criar comandas

payload:

```
{
  "data": "string",
  "cliente_id": "integer"
}
```


retorno:

```
200
nova_comanda.to_dict()

400
{"message": "Cliente e Data são obrigatorios"}
404
{"message": "Cliente não encontrado"}
```

GET /comandas

Responsabilidade: Listando todas as comandas

retorno:

```
200
[]

[
  comanda.to_dict()
]
```

GET /comandas/{id}

Responsabilidade: Listando uma única comand

retorno:

```
200
[
  comanda.to_dict()
]

404
{"message": "Comanda não encontrada"}
```

PUT /comandas/{id}

Responsabilidade: Editando informações da comanda

payload:

```
{
  "data": "string",
  "cliente_id": integer
}
```

retorno:

```
200
{"message": "Informações atualizadas",
 "comanda": comanda.to_dict()}

404
{"message": "Comanda não encontrada"}
```

DELETE /comandas/{id}

Responsabilidade: Deletar uma comanda

retorno:

```
200
{"message": "Comanda deletada com sucesso"}

404
{"message": "Comanda não encontrada"}
```

4- Produtos na Comanda

POST /comandas/{id}/produtos

Responsabilidade: Adicionar produtos a comanda

payload:

```
{
  "produto_id": integer,
  "quantidade": integer
}
```

retorno:

```
200
{"message": "Produto adicionado com sucesso",
 "comanda": comanda.to_dict()}}

400
{"message": "Dados inválidos"}

404
{"message": "Comanda não encontrada"}
{"message": "Produto não encontrado"}
```

PATCH /comandas/{id}/produtos/{produto_id}

Responsabilidade: Editar quantidades dos produtos da comanda

payload:

```
{
  "produto_id": integer,
  "quantidade": integer
}
```

retorno:

```
200
{"message": "Quantidade atualizada com sucesso",
 "comanda": comanda.to_dict()}}

400
{"message": "A quantidade tem que ser maior que zero"}
```

404

```
{"message": "Comanda não encontrada"}  
{"message": "Produto não encontrado na comanda"}
```

DELETE /comandas/{id}/produtos/{produto_id}

Responsabilidade: Deletar quantidades de produtos da uma comanda

payload:

```
{  
  "produto_id": integer,  
  "quantidade": integer  
}
```

retorno:

200

```
{"message": "Produto removido da comanda com sucesso",  
  "comanda": comanda.to_dict()}}
```

400

```
{"message": "Informe uma quantidade válida para remover"}
```

404

```
{"message": "Comanda não encontrada"}  
{"message": "Produto não encontrado na comanda"}
```

5- Pagamento

POST /pagamento

Responsabilidade: Adicionar novo pagamento a comanda

payload:

```
{
  "comanda_id": integer,
  "valor": integer
}
```

retorno:

```
200
{"message": "Pagamento criado com sucesso",
 "pagamento": pagamento.to_dict()}}
```

GET /pagamento/pix/qrcode/<file_name>

Responsabilidade: Busca do QR Code do pagamento

retorno:

```
200
{"message": "Pagamento criado com sucesso",
 "pagamento": pagamento.to_dict()}}
```

POST /pagamento/confirmacao

Responsabilidade: Confirmação de pagamento e calculo do novo saldo da comanda

payload:

```
{
  "comanda_id": integer,
  "valor": integer
}
```

retorno:

200

```
{ "message": "Pagamento confirmado com sucesso",  
  "saldo_restante": numero,  
  "total_comanda": numero,  
  "total_pago": numero  
}
```