

Atividade Coffee Shops Tia Rosa

O programa inicia no arquivo index com um loop while com a cláusula sempre verdadeira. Isso faz com que ele continue rodando a cada ação ou consulta feita pelo usuário.

É recebido a primeira escolha do usuário que pode ser de 3 formas:

- 1-Clientes
- 2-Pedidos
- 3-Produtos

Cada caso, após ser checado se se enquadra nas possibilidades (1,2 ou 3), é tratado no segundo passo, nos arquivos:

- EscolhaCliente
- EscolhaPedido
- EscolhaProduto

Escolha Cliente

É recebida a segunda escolha, que pode ser:

- Criar
- Listar
- UsarPontos

Os dados são recebidos pelo usuário e ao final é direcionado ao método correspondente dentro do arquivo ClasseCliente.

A classe, além dos métodos de criação, listagem e busca por id, conta também com os métodos aumentarPontos e usarPontos.

A quantidade de pontos ganhos por real é definida no início do arquivo.

usarPontos checa se o cliente tem os pontos suficientes e desconta caso tenha.

Escolha Produto

É recebida a segunda escolha, que pode ser:

- Criar
- Listar
- Criar Promoção
- Encerrar Promoção

Os dados são recebidos pelo usuário e ao final é direcionado ao método correspondente dentro do arquivo ClasseProduto.

A classe conta, além de preço, o atributo preçoSemPromoção.

Para que quando o preço for alterado pela promoção, armazene o valor original.

Esse valor então é usado no método encerrarPromoção.

Escolha Pedido

É recebida a segunda escolha, que pode ser:

- Criar
- Listar

Os dados são recebidos pelo usuário e ao final é direcionado ao método correspondente dentro do arquivo ClassePedido.

Na criação, é checado se o cliente escolhido pelo usuário existe.

É criada uma listaProdutos vazia para os produtos daquele pedido

Para adicionar itens ao pedido, há um loop, ainda no arquivo EscolhaPedido, que chama várias vezes o método adicionarProduto, checando sempre se o produto existe.

É adicionado um dicionário à listaProdutos contendo o objeto do produto e o preço pago (já que com as promoções mudando, o valor atual do produto pode não corresponder ao que o cliente pagou na hora da compra).

Quando o usuário digita 0, chama o método finalizarPedido, que passa pela listaProdutos, somando ao total.

É também adicionado ao cliente a quantidade de pontos ganhos com aquela compra, em função de quanto ele gastou

Ao final é mostrado o resumo do pedido.

Conclusão

Foi exercitado:

- Criação de um sistema complexo de condições e inputs
- Exibição de uma interface de interação no console e utilização de cores
- Entendimento de classes e como suas instâncias se relacionam (Todo pedido possui apenas um cliente de origem e vários produtos)

Qual das opções você deseja criar ou listar?

1-Cliente | 2-Pedido | 3-Produto

2

Criar ou listar?

1-Criar | 2-Listar

2

Pedidos:

ID:1 - Cliente: Ana (id Cliente:2)

Total: R\$ 0.00

=====

ID:2 - Cliente: Pedro (id Cliente:1)

R\$ 6.50 - Café Artesanal (Promoção)

R\$ 12.40 - Bolo

Total: R\$ 18.90

=====

ID:3 - Cliente: Ana (id Cliente:2)

Total: R\$ 0.00

=====

ID:4 - Cliente: Ana (id Cliente:2)

R\$ 7.90 - Café Artesanal

R\$ 7.90 - Café Artesanal

R\$ 12.40 - Bolo

Total: R\$ 28.20

=====

Qual das opções você deseja criar ou listar?

1-Cliente | 2-Pedido | 3-Produto

1

Criar ou listar?

1-Criar | 2-Listar | 3-Usar Pontos

2

Clientes:

ID:1 - Pedro - 189.0 pontos

ID:2 - Ana - 282.0 pontos

Qual das opções você deseja criar ou listar?

1-Cliente | 2-Pedido | 3-Produto

1

Criar ou listar?

1-Criar | 2-Listar | 3-Usar Pontos

3

Digite o id do Cliente:

1

Digite a quantidade de pontos a serem usados:

200

Saldo insuficiente de pontos

Qual das opções você deseja criar ou listar?

1-Cliente | 2-Pedido | 3-Produto

1

Criar ou listar?

1-Criar | 2-Listar | 3-Usar Pontos

3

Digite o id do Cliente:

1

Digite a quantidade de pontos a serem usados:

150

150 pontos usados por Pedro!

```
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
3
Criar ou listar?
1-Criar | 2-Listar | 3-Criar Promoção | 4-Encerrar Promoção
4
Digite o id do Produto:
1
Promoção encerrada com sucesso!
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
2
Criar ou listar?
1-Criar | 2-Listar
1
Digite o id do Cliente:
2
Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido
1
Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido
1
Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido
2
Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido
0
Pedido criado com sucesso!
282 pontos ganhos
ID:4 - Cliente: Ana (id Cliente:2)
R$ 7.90 - Café Artesanal
R$ 7.90 - Café Artesanal
R$ 12.40 - Bolo
Total: R$ 28.20
=====
```

```
Digite o id do Produto:
1
Digite o preço da promoção no formato 0:00:
6.50
Promoção feita com sucesso!
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
3
Criar ou listar?
1-Criar | 2-Listar | 3-Criar Promoção | 4-Encerrar Promoção
2
Produtos:
ID:1 - Café Artesanal - R$ 6.50 (Promoção)
Ingredientes: Grão de café, Canela
ID:2 - Bolo - R$ 12.40
Ingredientes: Farinha, Ovo, Açúcar
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
2
Criar ou listar?
1-Criar | 2-Listar
1
Digite o id do Cliente:
1
Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido
1
Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido
2
Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido
0
Pedido criado com sucesso!
189 pontos ganhos
ID:2 - Cliente: Pedro (id Cliente:1)
R$ 6.50 - Café Artesanal (Promoção)
R$ 12.40 - Bolo
Total: R$ 18.90
```

```
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
1
Criar ou listar?
1-Criar | 2-Listar | 3-Usar Pontos
4
Número inválido
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
4
Número inválido
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
2
Criar ou listar?
1-Criar | 2-Listar
1
Digite o id do Cliente:
4
Não existe Cliente com este id
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
2
Criar ou listar?
1-Criar | 2-Listar
1
Digite o id do Cliente:
2
Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido
4
Não existe Produto com este id
```

```
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
3
Criar ou listar?
1-Criar | 2-Listar | 3-Criar Promoção | 4-Encerrar Promoção
1
Digite o nome:
Café Artesanal
Digite os ingredientes:
Grão de café, Canela
Digite o preço no formato 0:00:
7.90
Produto criado com sucesso!
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
3
Criar ou listar?
1-Criar | 2-Listar | 3-Criar Promoção | 4-Encerrar Promoção
1
Digite o nome:
Bolo
Digite os ingredientes:
Farinha, Ovo, Açúcar
Digite o preço no formato 0:00:
12.40
Produto criado com sucesso!
```

```
Meu_Terminal$ python3 index.py
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
1
Criar ou listar?
1-Criar | 2-Listar | 3-Usar Pontos
1
Digite o nome:
Pedro
Cliente criado com sucesso!
Qual das opções você deseja criar ou listar?
1-Cliente | 2-Pedido | 3-Produto
1
Criar ou listar?
1-Criar | 2-Listar | 3-Usar Pontos
1
Digite o nome:
Ana
Cliente criado com sucesso!
```

ClasseCliente.py

```
from textoColorido import textoColorido
id=1
RELACAO_PONTO_PRECO=10 # Define quantos pontos o Cliente ganha para cada real gasto
class Cliente:
    clientes=[]

    def __init__(self,nome): # Método de criação de um objeto da classe Cliente
        global id
        self.id=id
        id=id+1
        self.nome=nome
        self.pontos=0
        Cliente.clientes.append(self) # Adiciona o novo Cliente à lista
        textoColorido(f'Cliente criado com sucesso!','verde')
    def listar():
        textoColorido('Clientes:','azul')
        for c in Cliente.clientes:
            print(f'ID:{c.id} - {c.nome} - {c.pontos} pontos')
    def encontrarClientePorId(idCliente):
        for c in Cliente.clientes:
            if(c.id==idCliente):
                return c
    def aumentarPontos(self,valorGasto):
        # Método que adiciona pontos ao Cliente, baseado na relação definida no início deste arquivo
        pontosGanhos=valorGasto*RELACAO_PONTO_PRECO
        self.pontos=self.pontos+pontosGanhos
        textoColorido(f'{pontosGanhos:.0f} pontos ganhos','azul')
    def usarPontos(cliente,pontosUsados):
        # Checa se o Cliente tem pontos o suficiente para o pedido de uso
        if(cliente.pontos>=pontosUsados):
            cliente.pontos=cliente.pontos-pontosUsados # Desconta os pontos usados
            textoColorido(f'{pontosUsados} pontos usados por {cliente.nome}!','verde')
        else:
            textoColorido(f'Saldo insuficiente de pontos','vermelho')
```

ClassePedido.py

```
from ClasseProduto import Produto
from ClasseCliente import Cliente
from textoColorido import textoColorido
id=1
class Pedido:
    pedidos=[]

    def __init__(self,cliente): # Método de criação de um objeto da classe Pedido
        global id
        self.id=id
        id=id+1
        self.cliente=cliente
        self.listaProdutos=[] # Cria a lista onde serão adicionados os itens do Pedido
        self.total=0
        Pedido.pedidos.append(self) # Adiciona o novo Pedido à lista

    def adicionarProduto(self,produto):
        # Adiciona um dicionário com o Produto e o Preço pago (já que ele pode ser feito em quanto t
        self.listaProdutos.append({'infos':produto,'precoComprado':produto.preco})
    def finalizarPedido(self):
        t=0
        for dicionarioProduto in self.listaProdutos:
            # Passa por cada Produto e adiciona o preço atual ao total
            t=t+dicionarioProduto['precoComprado']
        self.total=t
        cliente=self.cliente
        textoColorido(f'Pedido criado com sucesso!','verde')
        Cliente.aumentarPontos(cliente,t) # Adiciona pontos ao Cliente, baseado no preço (a relação p
        Pedido.imprimirPedido(self)
    def listar():
        # Checagem de Pedidos
        textoColorido('Pedidos:','azul')
        for p in Pedido.pedidos:
            Pedido.imprimirPedido(p)
```

```

def imprimirPedido(p):
    # Método que imprime as informações do Pedido como os itens e o total a pagar
    textoColorido(f'ID:{p.id} - Cliente: {p.cliente.nome} (id Cliente:{p.cliente.id})','amarelo')
    for dicionarioProduto in p.listaProdutos:
        produto=dicionarioProduto['infos']
        precoComprado=dicionarioProduto['precoComprado']
        if(precoComprado!=produto.precoSemPromocao):
            print(f'R$ {precoComprado:.2f} - {produto.nome} (Promoção)')
        else:
            print(f'R$ {precoComprado:.2f} - {produto.nome}')
    textoColorido(f'Total: R$ {p.total:.2f}','verde')
    print('=====')

```

ClasseProduto.py

```

from textoColorido import textoColorido
id=1
class Produto:
    produtos=[]

    def __init__(self,nome,ingredientes,preco): # Método de criação de um objeto da classe Produto
        global id
        self.id=id
        id=id+1
        self.nome=nome
        self.ingredientes=ingredientes
        self.preco=preco
        self.precoSemPromocao=preco # Relembra o preço inicial, para quando a promoção for encerrar
        Produto.produtos.append(self) # Adiciona o novo Produto à lista
        textoColorido(f'Produto criado com sucesso!','verde')
    def alterarPreco(self,novoPreco):
        self.preco=novoPreco
    def listar():
        # Checagem de Produtos com suas descrições e preços atuais
        textoColorido('Produtos:','azul')
        for p in Produto.produtos:
            if(p.preco!=p.precoSemPromocao):
                textoColorido(f'ID:{p.id} - {p.nome} - R$ {p.preco:.2f} (Promoção)','amarelo')
            else:
                textoColorido(f'ID:{p.id} - {p.nome} - R$ {p.preco:.2f}','amarelo')
            print(f'Ingredientes: {p.ingredientes}')
    def criarPromocao(produto,novoPreco):
        produto.alterarPreco(novoPreco) # Define o novo preço da promoção
        textoColorido(f'Promoção feita com sucesso!','verde')
    def encerrarPromocao(produto):
        if(produto.preco==produto.precoSemPromocao): # Checa se o preço atual é o mesmo do original
            textoColorido(f'Este Produto não está na promoção','vermelho')
        else:
            produto.alterarPreco(produto.precoSemPromocao) # Volta o preço original do Produto
            textoColorido(f'Promoção encerrada com sucesso!','verde')
    def encontrarProdutoPorId(idProduto):
        for p in Produto.produtos:
            if(p.id==idProduto):
                return p

```

EscolhaCliente.py

```

from ClasseCliente import Cliente
from textoColorido import textoColorido
def escolhaCliente():
    textoColorido('Criar ou listar?','azul')
    textoColorido('1-Criar | 2-Listar | 3-Usar Pontos','azul')
    texto=input('')
    if(not texto.isdigit()):
        textoColorido('Número inválido','vermelho')
        return
    decisao=int(texto)
    if(decisao==1): # Criação de Cliente
        textoColorido('Digite o nome:','azul')
        nome=input('')
        Cliente(nome)

```



```

elif(decisao==2): # Checagem dos Clientes
    Cliente.listar()
elif(decisao==3): # Desconto de pontos do Cliente, caso tenha saldo suficiente
    textoColorido('Digite o id do Cliente:', 'azul')
    texto=input('')
    if(not texto.isdigit()):
        textoColorido('Número inválido', 'vermelho')
        return
    idCliente=int(texto)
    cliente=Cliente.encontrarClientePorId(idCliente) # Procura a instância de Cliente com o id p
    if(not cliente): # Checa se o valor retornado é nulo
        textoColorido('Não existe Cliente com este id', 'vermelho')
        return
    textoColorido('Digite a quantidade de pontos a serem usados:', 'azul')
    texto=input('')
    if(not texto.isdigit()):
        textoColorido('Número inválido', 'vermelho')
        return
    pontos=int(texto)
    Cliente.usarPontos(cliente, pontos) # Chama o método usarPontos, com a quantidade passada pelo
else:
    textoColorido('Número inválido', 'vermelho')

```

EscolhaPedido.py

```

from ClassePedido import Pedido
from ClasseProduto import Produto
from ClasseCliente import Cliente
from textoColorido import textoColorido
def escolhaPedido():
    textoColorido('Criar ou listar?', 'azul')
    textoColorido('1-Criar | 2-Listar', 'azul')
    texto=input('')
    if(not texto.isdigit()):
        textoColorido('Número inválido', 'vermelho')
        return
    decisao=int(texto)
    if(decisao==1): # Criação de um pedido
        textoColorido('Digite o id do Cliente:', 'azul')
        texto=input('')
        if(not texto.isdigit()):
            textoColorido('Número inválido', 'vermelho')
            return
        idCliente=int(texto)
        cliente=Cliente.encontrarClientePorId(idCliente)
        if(not cliente): # Checa existência do Cliente
            textoColorido('Não existe Cliente com este id', 'vermelho')
            return
        novoPedido=Pedido(cliente)
        novoItem=None
        while(novoItem!=0): # Inicia um loop para adicionar itens ao pedido, até ser encerrado com o
            textoColorido('Digite o id do Produto que quer adicionar, ou 0 para finalizar o Pedido',
            texto=input('')
            if(not texto.isdigit()):
                textoColorido('Número inválido', 'vermelho')
                return
            novoItem=int(texto)
            if(novoItem!=0):
                produto=Produto.encontrarProdutoPorId(novoItem)
                if(not produto): # Checa existência do Produto
                    textoColorido('Não existe Produto com este id', 'vermelho')
                    return
                novoPedido.adicionarProduto(produto)
            novoPedido.finalizarPedido()
    elif(decisao==2): # Checagem de Pedidos
        Pedido.listar()
    else:
        textoColorido('Número inválido', 'vermelho')

```

EscolhaProduto.py

```

from ClasseProduto import Produto
from textoColorido import textoColorido
def escolhaProduto():
    textoColorido('Criar ou listar?','azul')
    textoColorido('1-Criar | 2-Listar | 3-Criar Promoção | 4-Encerrar Promoção','azul')
    texto=input('')
    if(not texto.isdigit()):
        textoColorido('Número inválido','vermelho')
        return
    decisao=int(texto)
    if(decisao==1): # Criação de Produto
        textoColorido('Digite o nome:','azul')
        nome=input('')
        textoColorido('Digite os ingredientes:','azul')
        ingredientes=input('')
        textoColorido('Digite o preço no formato 0:00:','azul')
        texto=input('')
        if(not texto.replace('.', '').isdigit()):
            textoColorido('Número inválido','vermelho')
            return
        preco=float(texto)
        Produto(nome,ingredientes,preco)
    elif(decisao==2): # Checagem de Produtos
        Produto.listar()
    elif(decisao==3): # Fazer promoção de um produto
        textoColorido('Digite o id do Produto:','azul')
        texto=input('')
        if (not texto.isdigit()):
            textoColorido('Número inválido','vermelho')
            return
        idProduto=int(texto)
        produto=Produto.encontrarProdutoPorId(idProduto)
        if(not produto):
            textoColorido('Não existe Produto com este id','vermelho')
            return
        textoColorido('Digite o preço da promoção no formato 0:00:','azul')
        texto=input('')
        if (not texto.replace('.', '').isdigit()):
            textoColorido('Número inválido','vermelho')
            return
        preco=float(texto)
        Produto.criarPromocao(produto,preco)
    elif(decisao==4): # Encerrar uma promoção já criada
        textoColorido('Digite o id do Produto:','azul')
        texto=input('')
        if (not texto.isdigit()):
            textoColorido('Número inválido','vermelho')
            return
        idProduto=int(texto)
        produto=Produto.encontrarProdutoPorId(idProduto)
        if(not produto):
            textoColorido('Não existe Produto com este id','vermelho')
            return
        Produto.encerrarPromocao(produto)
    else:
        textoColorido('Número inválido','vermelho')

```

index.py

```

from EscolhaCliente import escolhaCliente
from EscolhaPedido import escolhaPedido
from EscolhaProduto import escolhaProduto
from textoColorido import textoColorido
# O loop while continuará sendo repetido, mantendo o programa rodando
while(True):
    textoColorido('Qual das opções você deseja criar ou listar?','azul')
    textoColorido('1-Cliente | 2-Pedido | 3-Produto','azul')
    texto=input('')
    if(not texto.isdigit()): # O texto digitado pelo usuário é verificado se é numerico, para não gerar erro
        textoColorido('Número inválido','vermelho')
        continue
    decisao=int(texto) # A escolha do usuário é convertida para número e entra na condição correspondente
    if(decisao==1):
        escolhaCliente() # Instância Cliente foi escolhida, continua no arquivo EscolhaCliente.py

```

```
elif(decisao==2):
    escolhaPedido() # Instância Pedido foi escolhida, continua no arquivo EscolhaPedido.py
elif(decisao==3):
    escolhaProduto() # Instância Produto foi escolhida, continua no arquivo EscolhaProduto.py
else:
    textoColorido('Número inválido','vermelho')
    # A resposta não é válida, o programa exibe a mensagem de erro e volta ao início
```

textoColorido.py

```
def textoColorido(texto,cor):
    if(cor=='vermelho'):
        codigo="\033[31m"
    elif(cor=='verde'):
        codigo="\033[32m"
    elif(cor=='azul'):
        codigo="\033[34m"
    elif(cor=='amarelo'):
        codigo="\033[33m"
    finalCodigo="\033[0m"
    # Essa função recebe o texto e a cor escolhida
    # E Junta o código da cor com o texto
    print(codigo+texto+finalCodigo)
```