

Trabalho Prático - Comanda Digital

Introdução

A nossa empresa fictícia é um restaurante chamado "Comanda Digital". O objetivo deste sistema é facilitar a divisão de contas entre grupos de clientes de forma igual ou proporcional ao consumo de cada um. Além disso, o sistema permite o gerenciamento de usuários com diferentes níveis de permissão. Os tipos de usuário incluem administrador e cliente, onde o administrador tem acesso completo a todas as funcionalidades, enquanto os clientes possuem permissões limitadas.

Ideia de Funcionamento

Problema a ser resolvido

É comum em um restaurante um grupo de pessoas se confundir quanto a quanto cada um deve pagar ao dividir uma comanda. As dificuldades em pagar lanches em grupos igualitariamente ou igualmente serão resolvidas por nosso sistema, que oferece duas soluções:

1. **Solução de pagamento igual:** O total da comanda dividido pelo número de pessoas do grupo.
2. **Solução igualitária:** Cada um paga em proporção ao seu consumo.

Fluxo de Caixa

Fonte de Renda

1. A primeira fonte de renda vem de anúncios rápidos vistos pelos usuários para usar o programa.
2. A segunda fonte de renda vem do pagamento para remover anúncios pelo usuário.

Reinvestimentos

Os reinvestimentos serão feitos para atualizar, aprimorar e/ou corrigir bugs do sistema.

Doações

10% do lucro será doado para um fundo beneficente na área da saúde e 10% será doado para institutos educacionais de software.

Conclusão:

Foi bem divertido e desafiador fazer este trabalho da comanda digital e ele trouxe muita aprendizagem. Entretanto, faltou fazer a parte de anúncios e pagamento por no adds.

Implementação

Usuários

Estrutura de Dados

Os usuários serão armazenados em um dicionário onde as chaves são os nomes de usuário e os valores são as senhas correspondentes.

python

Copiar código

```
usuarios = {  
    "usuario1": "senha1",  
    "usuario2": "senha2",  
    "admin": "admin123"  
}
```

Arquivo de Registro

O registro dos usuários será feito em um arquivo CSV com duas colunas: `usuario` e `senha`.

Exemplo de conteúdo do arquivo `usuarios.csv`:

Copiar código

```
usuario,senha  
usuario1,senha1  
usuario2,senha2  
admin,admin123
```

Funcionalidades (CRUD)

- **Create:** Adicionar um novo usuário (apenas admin).
- **Read:** Fazer login.
- **Update:** Atualizar senha de um usuário.
- **Delete:** Remover um usuário (apenas admin).

Produtos/Serviços

Estrutura de Dados

Os produtos/serviços serão armazenados em uma lista de dicionários, onde cada dicionário representa um produto/serviço com seus atributos (nome, preço, quantidade).

python

Copiar código

```
produtos = [  
    {"nome": "Café", "preço": 3.50, "quantidade": 100},  
    {"nome": "Sanduíche", "preço": 10.00, "quantidade": 50}  
]
```

Arquivo de Registro

O registro dos produtos será feito em um arquivo CSV com as colunas: **nome**, **preço**, **quantidade**.

Exemplo de conteúdo do arquivo **produtos.csv**:

Copiar código

```
nome,preço,quantidade  
Café,3.50,100  
Sanduíche,10.00,50
```

Funcionalidades (CRUD)

- **Create:** Adicionar um novo produto.
- **Read:** Listar todos os produtos, buscar um registro específico por nome ou código.
- **Update:** Atualizar informações de um produto (preço, quantidade).
- **Delete:** Remover um produto.

Código Fonte

```
usuarios = {
```

```
"usuario1": "senha1",
"usuario2": "senha2",
"admin": "admin123"}

# Função para carregar usuários do arquivo (Read)
def carregar_usuarios():
    global usuarios
    try:
        with open('usuarios.csv', mode='r') as file:
            reader = csv.DictReader(file)
            for row in reader:
                usuarios[row['usuario']] = row['senha']
    except FileNotFoundError:
        print("Arquivo de usuários não encontrado. Iniciando com lista vazia.")

# Função para salvar usuários no arquivo (Create, Update, Delete)
def salvar_usuarios():
    global usuarios
    with open('usuarios.csv', mode='w', newline='') as file:
        writer = csv.DictWriter(file, fieldnames=['usuario', 'senha'])
        writer.writeheader()
        for usuario, senha in usuarios.items():
            writer.writerow({'usuario': usuario, 'senha': senha})

# Função para fazer login (Read)
def fazer_login():
    while True:
        # Entrada de dados
        usuario = input("Digite seu nome de usuário: ")
        senha = input("Digite sua senha: ")

        # Processamento e Saída de dados
        if usuario in usuarios and usuarios[usuario] == senha:
            print(f"Bem-vindo, {usuario}!")
            return usuario
        else:
            print("Nome de usuário ou senha incorretos. Tente novamente.")

# Função para adicionar um novo usuário (Create)
```

```
def adicionar_usuario():
    # Entrada de dados
    novo_usuario = input("Digite o nome do novo usuário: ")
    nova_senha = input("Digite a senha do novo usuário: ")

    # Processamento e Saída de dados
    if novo_usuario in usuarios:
        print(f"Usuário '{novo_usuario}' já existe. Não foi possível
adicionar.")
    else:
        usuarios[novo_usuario] = nova_senha
        salvar_usuarios()
        print(f"Usuário '{novo_usuario}' adicionado com sucesso!")

# Função para carregar produtos do arquivo (Read)
def carregar_produtos():
    produtos = []
    try:
        with open('produtos.csv', mode='r') as file:
            reader = csv.DictReader(file)
            for row in reader:
                produtos.append({
                    "nome": row['nome'],
                    "preço": float(row['preço']),
                    "quantidade": int(row['quantidade'])
                })
    except FileNotFoundError:
        print("Arquivo de produtos não encontrado. Iniciando com lista
vazia.")

    # Adicionando produtos comuns em um restaurante
    produtos = [
        #Comidas:
        {"nome": "Sanduíche", "preço": 15.00, "quantidade": 30},
        {"nome": "Salada", "preço": 12.00, "quantidade": 20},
        {"nome": "Pizza", "preço": 10.00, "quantidade": 15},\
        #Bebidas:
        {"nome": "Suco", "preço": 6.00, "quantidade": 40},
        {"nome": "Café", "preço": 4.00, "quantidade": 50}
    ]
    salvar_produtos(produtos)
```

```
    return produtos

# Função para salvar produtos no arquivo (Create, Update, Delete)
def salvar_produtos(produtos):
    with open('produtos.csv', mode='w', newline='') as file:
        writer = csv.DictWriter(file, fieldnames=['nome', 'preço',
'quantidade'])
        writer.writeheader()
        for produto in produtos:
            writer.writerow(produto)

# Função para adicionar um novo produto (Create)
def adicionar_produto(produtos):
    # Entrada de dados
    nome = input("Digite o nome do produto: ")
    preco = float(input("Digite o preço do produto: "))
    quantidade = int(input("Digite a quantidade do produto: "))

    # Processamento e Saída de dados
    produtos.append({"nome": nome, "preço": preco, "quantidade":
quantidade})
    salvar_produtos(produtos)
    print(f"Produto '{nome}' adicionado com sucesso!")

# Função para listar produtos (Read)
def listar_produtos(produtos):
    # Saída de dados
    print("\nProdutos cadastrados:")
    for produto in produtos:
        print(f"Nome: {produto['nome']}, Preço: R${produto['preço']:.2f},
Quantidade: {produto['quantidade']}")

# Função para buscar um produto (Read)
def buscar_produto(produtos):
    # Entrada de dados
    nome = input("Digite o nome do produto que deseja buscar: ")

    # Processamento e Saída de dados
    for produto in produtos:
        if produto['nome'] == nome:
```

```
        print(f"Produto encontrado: Nome: {produto['nome']}, Preço: R${produto['preço']:.2f}, Quantidade: {produto['quantidade']}")
        return

    print(f"Produto '{nome}' não encontrado.")

# Função principal do programa
def main():
    carregar_usuarios()
    produtos = carregar_produtos()

    usuario_logado = fazer_login()

#IMPORTANTE >> PARTE PRINCIPAL DO CÓDIGO FONTE:
    while True:
        # Saída de dados
        print("\nBem-vindo ao comanda digital")
        print("Escolha o tipo de solução desejado:")
        print("1 - Pagamento igual")
        print("2 - Pagamento igualitário (proporcional ao consumo individual)")
        print("3 - Adicionar novo usuário (apenas admin)")
        print("4 - Adicionar novo produto")
        print("5 - Listar produtos")
        print("6 - Buscar produto")
        print("0 - Encerrar programa")

        # Entrada de dados
        solucao_escolhida = int(input("Digite o número da opção desejada: "))

        if solucao_escolhida == 0:
            print("Encerrando o programa...")
            sys.exit(0)
        elif solucao_escolhida == 1:
            # Entrada de dados
            valor_total = float(input("Qual o valor total a ser pago? "))
            n_pessoas = int(input("Qual o número de pessoas que pagarão a conta? "))

            # Processamento e Saída de dados
```

```
        if n_pessoas <= 0:
            print("Número inválido de pessoas. Encerrando o
programa.")

            sys.exit(1)

        valor_individual = valor_total / n_pessoas
        print(f"O valor a ser pago em uma divisão de pagamento igual
será R${valor_individual:.2f}")

    elif solucao_escolhida == 2:
        # Entrada de dados
        valor_total = float(input("Digite o valor total gasto no
restaurante: "))
        n_comidas = int(input("Digite o número de tipos diferentes de
comidas consumidas: "))

        total_calculado = 0.0

        # Processamento e Saída de dados
        for i in range(n_comidas):
            nome_comida = input(f"Digite o nome da comida {i+1}: ")
            valor_comida = float(input(f"Digite o valor gasto com a
comida '{nome_comida}': "))
            pessoas_comeram = int(input(f"Digite quantas pessoas
comeram a comida '{nome_comida}': "))

            if pessoas_comeram <= 0:
                print(f"Número inválido de pessoas para a comida
'{nome_comida}'. Encerrando o programa.")
                sys.exit(1)

            valor_individual_comida = valor_comida / pessoas_comeram
            print(f"O valor a ser pago por pessoa pela comida
'{nome_comida}' é R${valor_individual_comida:.2f}")

            total_calculado += valor_comida

        if abs(valor_total - total_calculado) < 0.01:
            print(f"Tudo certo >> Valor total calculado = Valor total
digitado: R${total_calculado:.2f}")
```



```
        else:
            print(f"Algo deu errado >> Valor total fornecido
(R${valor_total:.2f}) foi diferente do calculado
(R${total_calculado:.2f})")

        elif solucao_escolhida == 3:
            if usuario_logado == "admin":
                adicionar_usuario()
            else:
                print("Apenas o administrador pode adicionar novos
usuários.")

        elif solucao_escolhida == 4:
            adicionar_produto(produtos)

        elif solucao_escolhida == 5:
            listar_produtos(produtos)

        elif solucao_escolhida == 6:
            buscar_produto(produtos)

        else:
            print("Opção inválida. Por favor, escolha uma opção válida.")

if __name__ == "__main__":
    main()
```