



UNIFSA – CENTRO UNIVERSITÁRIO SANTO AGOSTINHO
ENGENHARIA DE SOFTWARE

MIKAELLA MARTINS SILVA

DESENVOLVIMENTO DE SISTEMA CRUD
Sistema de Clientes

TERESINA – PI
2025

1 INTRODUÇÃO

Este trabalho apresenta o desenvolvimento de um sistema CRUD em Python, criado para gerenciar informações de clientes de forma simples e funcional através do terminal. CRUD é um acrônimo para Create, Read, Update e Delete, que representa as quatro operações fundamentais utilizadas na manipulação de dados em aplicações computacionais.

Uma implementação bem estruturada de CRUD garante funcionalidade, organização e confiabilidade, servindo como base sólida para aplicações maiores e mais complexas. Quando cada operação funciona adequadamente, o sistema torna-se robusto, fácil de manter e adequado para qualquer tipo de gerenciamento de dados.

Dessa forma o código criado foi otimizado para ser mais compacto, mantendo todas as funcionalidades essenciais de cadastro, listagem, atualização, remoção de clientes e geração de relatório. Sendo assim, objetivo principal é simular um pequeno sistema corporativo capaz de cadastrar, listar, atualizar e remover clientes, além de gerar um relatório básico sobre os dados armazenados.

2 DESENVOLVIMENTO

O tema escolhido foi **Sistema de Clientes**, permitindo armazenar informações importantes como:

- Nome do cliente
- Telefone
- Serviço contratado

O sistema foi pensado para atender pequenas empresas que precisam organizar sua base de clientes de maneira prática, sem interface gráfica, funcionando diretamente no terminal.

3 ESTRUTURA DO PROJETO

Seguindo os requisitos da prova, o sistema foi construído com:

- Lista de dicionários para armazenar os registros
- Funções separadas para cada operação do CRUD
- Menu interativo utilizando repetição
- Validações básicas para evitar erros comuns
- Saída de texto amigável para o usuário

Além disso, cada cliente recebe um **ID único**, garantindo controle adequado dos registros.

4 FUNCIONALIDADES IMPLEMENTADAS

4.1 Cadastrar Cliente(Create)

Permite inserir um novo cliente, solicitando nome, telefone e serviço contratado. Também impede cadastro com nome vazio ou telefone inválido.

4.2 Listar Clientes(Read)

Exibe todos os clientes registrados, incluindo o ID, facilitando consultas e atualizações.

4.3 Atualizar Cliente(Update)

Permite alterar os dados de um cliente existente. O usuário informa o ID e escolhe o que deseja modificar.

4.4 Remover Cliente>Delete)

Exclui um cliente da lista a partir de seu ID.

4.5 Resumo

O relatório exibe de forma resumida o total de clientes cadastrados e a quantidade de clientes por serviço contratado, mantendo a funcionalidade essencial mesmo com o código mais compacto.

- Total de clientes cadastrados
- Quantidade por tipo de serviço contratado

4.6 Sair do Sistema

Encerra o programa de forma segura

5 CÓDIGO-FONTE SISTEMA

```
clientes = []
proximo_id = 1

def cadastrar():
    global proximo_id
    nome = input("Nome: ").strip()
    tel = input("Telefone: ").strip()
    serv = input("Serviço: ").strip()
    if not nome or not tel:
        print("Erro: Nome e telefone são obrigatórios")
        return
    clientes.append({"id": proximo_id, "nome": nome, "telefone": tel, "servico": serv})
    print(f"Cliente cadastrado! ID: {proximo_id}")
    proximo_id += 1

def listar():
    if not clientes:
        print("Nenhum cliente cadastrado.")
        return
    for c in clientes:
        print(f"{c['id']} | {c['nome']} | {c['telefone']} | {c['servico']}")

def atualizar():
    try: idc = int(input("ID do cliente: "))
    except: print("ID inválido"); return
    for c in clientes:
        if c["id"] == idc:
            opt = input("Alterar: 1-Nome 2-Tel 3-Serviço: ")
            if opt=="1": c["nome"]=input("Novo nome: ").strip()
            elif opt=="2": c["telefone"]=input("Novo telefone: ").strip()
            elif opt=="3": c["servico"]=input("Novo serviço: ").strip()
            else: print("Opção inválida"); return
            print("Atualizado!"); return
    print("Cliente não encontrado.")

def remover():
    try: idc=int(input("ID do cliente: "))
    except: print("ID inválido"); return
    for c in clientes:
        if c["id"]==idc:
```

```
clientes.remove(c)
print("Cliente removido!"); return
print("Cliente não encontrado.")

def relatorio():
    print(f"Total: {len(clientes)}")
    cont = {}
    for c in clientes:
        cont[c['servico']] = cont.get(c['servico'],0)+1
    for s,q in cont.items():
        print(f"{s}: {q} cliente(s)")

def menu():
    while True:
        print("\n1-Cadastrar 2-Listar 3-Atualizar 4-Remover 5-Relatório 6-Sair")
        op = input("Escolha: ")
        if op=="1": cadastrar()
        elif op=="2": listar()
        elif op=="3": atualizar()
        elif op=="4": remover()
        elif op=="5": relatorio()
        elif op=="6": break
        else: print("Inválido!")

menu()
```

6 CONCLUSÃO

O desenvolvimento deste sistema permitiu aplicar na prática os conceitos fundamentais da disciplina de Programação I. Criar e organizar funções, manipular listas de dicionários e estruturar um menu interativo foram etapas importantes para reforçar a lógica de programação. Mesmo com a otimização do código para torná-lo mais compacto, todas as funções essenciais do CRUD foram preservadas, incluindo o relatório de clientes.

7 REFERÊNCIAS

- DATE, C. J. An Introduction to Database Systems. 8. ed. Boston: Addison-Wesley, 2003.
- ELMASRI, R.; NAVATHE, S. B. Fundamentals of Database Systems. 7. ed. Boston: Pearson, 2016.
- PYTHON SOFTWARE FOUNDATION. Python Documentation. Disponível em: <https://docs.python.org/>. Acesso em: 30 nov. 2025.