



DIRETRIZES PARA O PROJETO INTEGRADOR III-B (16E3A M5/6)

TEMA:

Desenvolvimento de um Sistema em Python com Operações em Banco de Dados

Aluno: Luan Vitor Costa Silva Lopes

Email: luanvito2005@gmail.com

1-Planilha de Requisitos

Requisitos de Função (RF)

- *RF01: O sistema precisa possibilitar a inscrição de clientes.*
- *RF02: O software deve apresentar uma lista de todos os clientes registrados.*
- *RF03: O sistema deve ter a capacidade de eliminar um cliente.*
- *RF04: O sistema precisa possibilitar a alteração das informações de um cliente.*
- *RF05: O sistema precisa guardar o nome, email, número de telefone e endereço do cliente.*

Não Funcionais (INF)

- *RNF01: A implementação do sistema deve ser feita em Python.*
- *RNF02: É necessário que o sistema use um banco de dados relacional, como MySQL ou PostgreSQL.*
- *RNF03: É necessário que o sistema possua persistência de dados (armazenamento contínuo).*
- *RNF04: O sistema deve empregar conceitos fundamentais de programação, tais como listas, dicionários, condicionais, laços e funções.*

2-Identificação das Entidades Centrais

Instituição: Cliente

Campo	Tipo de dado	Descrição
<i>id</i>	<i>Inteiro</i>	<i>Identificador único (PK)</i>
<i>nome</i>	<i>Texto (varchar)</i>	<i>Nome completo do cliente</i>
<i>email</i>	<i>Texto (varchar)</i>	<i>Email do cliente</i>
<i>telefone</i>	<i>Texto (varchar)</i>	<i>Telefone do cliente</i>
<i>endereco</i>	<i>Texto (varchar)</i>	<i>Endereço do cliente</i>

3-Estrutura Conceitual (ER)

Cliente: Entidade

- **PK: id**
- **nome**
- **email**
- **telefone**
- **endereco**

(Diagrama básico de uma única entidade, já que o contexto é inicial e direto. Se desejar, posso criar uma representação gráfica do diagrama ER.)

4-Estrutura Lógica (Tabelas)

Tabela: clientes

```
CREATE TABLE clientes (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    telefone VARCHAR(20),  
    endereco VARCHAR(150)  
);
```

O modelo já foi normalizado (1FN, 2FN e 3FN), uma vez que:

- Todos os campos possuem natureza atômica.
- Existe chave primária
- Não existem dependências temporárias.

5-Programa inicial em Python (ainda sem banco de dados)

Segue o esquema inicial contendo listas e dicionários:

```
clientes = []
```

```
def cadastrar_cliente():
```

```
    nome = input("Nome: ")
```

```
    email = input("Email: ")
```

```
    telefone = input("Telefone: ")
```

```
    endereco = input("Endereço: ")
```

```
    cliente = {
```

```
        "id": len(clientes) + 1,
```

```
        "nome": nome,
```

```
        "email": email,
```

```
        "telefone": telefone,
```

```
        "endereco": endereco
```

```
    }
```

```
    clientes.append(cliente)
```

```
    print("Cliente cadastrado com sucesso!\n")
```

```
def listar_clientes():
```

```
    for cliente in clientes:
```

```
        print(f"ID: {cliente['id']} | Nome: {cliente['nome']} | Email: {cliente['email']}")
```

```
def excluir_cliente():
    id_excluir = int(input("Digite o ID do cliente para excluir: "))
    for cliente in clientes:
        if cliente['id'] == id_excluir:
            clientes.remove(cliente)
            print("Cliente excluído com sucesso!\n")
            return
    print("Cliente não encontrado.\n")
```

```
def menu():
    while True:
        print("1. Cadastrar Cliente")
        print("2. Listar Clientes")
        print("3. Excluir Cliente")
        print("4. Sair")
        opcao = input("Escolha uma opção: ")

        if opcao == "1":
            cadastrar_cliente()
        elif opcao == "2":
            listar_clientes()
        elif opcao == "3":
            excluir_cliente()
        elif opcao == "4":
            break
        else:
            print("Opção inválida!")
```

```
menu()
```

Anderson Soares Costa
Professor Responsável