

# Tailan de Souza Oliveira, 553381, Sistemas de Informação

## Lab01

Descrição passo a passo;

Primeiramente você precisa rodar o código fonte, se for no Linux;

`python3 vigenere.py`

Apos rodar o código, você vera esse terminal

```
→ Lab01 git:(main) ✗ python3 vigenere.py
```

```
Escolha uma opção:  
1 - Cifrar (encriptar)  
2 - Decifrar (decriptar)  
3 - Executar testes manuais  
0 - Sair  
Digite sua escolha: █
```

Se escolher a opção 1, vai pedir a palavra para você cifrar exemplo “**Tailan**” e posteriormente vai pedir para digitar a chave, que pode ser “**Ola**”

```
Digite sua escolha: 1  
Digite o texto claro: Tailan  
Digite a chave: Ola  
Texto cifrado (base64): o83Ku83P
```

```
Escolha uma opção:  
1 - Cifrar (encriptar)  
2 - Decifrar (decriptar)  
3 - Executar testes manuais  
0 - Sair  
Digite sua escolha: █
```

Dado isso, vemos a palavra cifrada “**o83Ku83P**” isso, copie simplesmente o conteúdo em “**o83Ku83P**” rode o código novamente escolha 2 para decifrar cole o texto cifrado, e depois coloque a palavra chave **Ola**, e vera “**Tailan**”, novamente

```
Escolha uma opção:  
1 - Cifrar (encriptar)  
2 - Decifrar (decriptar)  
3 - Executar testes manuais  
0 - Sair  
Digite sua escolha: 2  
Digite o texto cifrado (base64): o83Ku83P  
Digite a chave: Ola  
Texto decifrado: Tailan
```

Para que esse código fosse produzido foi pegado ideias desses [site](#), além disso todo o código foi feito em Python, segue código abaixo

```
import base64

def encriptar(texto_claro: str, chave: str) -> str:

    if not chave:
        raise ValueError("Chave não pode ser vazia.")
    pt_bytes = texto_claro.encode("latin-1")
    key_bytes = chave.encode("latin-1")
    out = bytearray()
    for i, b in enumerate(pt_bytes):
        k = key_bytes[i % len(key_bytes)]
        out.append((b + k) % 256)
    return base64.b64encode(bytes(out)).decode("ascii") # string
imprimível

def decriptar(texto_encriptado: str, chave: str) -> str:
    if not chave:
        raise ValueError("Chave não pode ser vazia.")
    try:
        cipher_bytes = base64.b64decode(texto_encriptado)
    except Exception:
        raise ValueError("Texto encriptado inválido: espere uma string
base64.")
    key_bytes = chave.encode("latin-1")
    out = bytearray()
    for i, c in enumerate(cipher_bytes):
        k = key_bytes[i % len(key_bytes)]
        out.append((c - k + 256) % 256)
    return bytes(out).decode("latin-1")

def executar_testes_manualmente():
    try:
        n = int(input("Quantos casos de teste deseja executar? "))
    except ValueError:
        print("Número inválido.")
        return

    for i in range(n):
```

```

print(f"\n--- Caso {i+1} ---")
texto = input("Digite o texto claro: ")
chave = input("Digite a chave: ")
try:
    cifrado = encriptar(texto, chave)
    decifrado = decriptar(cifrado, chave)
except Exception as e:
    print("Erro:", e)
    continue

print("Cifrado (base64):", cifrado)
print("Decifrado:", decifrado)
print("Resultado:", "OK - decifrou corretamente" if decifrado ==
texto else "FALHOU")

if __name__ == "__main__":
    menu = (
        "Escolha uma opção:\n"
        "1 - Cifrar (encriptar)\n"
        "2 - Decifrar (decriptar)\n"
        "3 - Executar testes manuais\n"
        "0 - Sair\n"
    )
    while True:
        print()
        opcao = input(menu + "Digite sua escolha: ").strip()
        if opcao == "1":
            texto = input("Digite o texto claro: ")
            chave = input("Digite a chave: ")
            try:
                cif = encriptar(texto, chave)
                print("Texto cifrado (base64):", cif)
            except Exception as e:
                print("Erro:", e)

        elif opcao == "2":
            texto_cifrado = input("Digite o texto cifrado (base64): ")
            chave = input("Digite a chave: ")
            try:
                dec = decriptar(texto_cifrado, chave)
                print("Texto decifrado:", dec)
            except Exception as e:

```

```
        print("Erro:", e)

    elif opcao == "3":
        executar_testes_manualmente()

    elif opcao == "0":
        break

    else:
        print("Opção inválida. Tente novamente.")
```

## Plano de Testes

Caso 1 – Teste básico

Texto claro: TANK

Chave: WAR

Caso 2 – Texto maior que a chave

Texto claro: BOMBERPLANE

Chave: FIGHT

Caso 3 – Chave maior que o texto

Texto claro: PAN ZER

Chave: STRATEGY

Caso 4 – Texto com caracteres especiais

Texto claro: SPITFIRE-1940!

Chave: AIRFORCE

Caso 5 – Texto e chave longos

Texto claro: TIGERTANKANDMUSTANGAIRCRAFT

Chave: SECONDWORLDWAR

## Como executar o programa Python

1. Salve o código em um arquivo chamado `vigenere.py`.
2. Abra o terminal na pasta onde o arquivo está salvo.
3. Execute o programa com:  
`python3 vigenere.py`
4. O menu exibirá opções:
  - 1 - Para cifrar (encriptar).
  - 2 - Para decifrar (decriptar).
  - 3 - Para executar testes manuais.
5. Para cada caso 1 a 5 (cifrar):
  - Digite o Texto claro do caso.
  - Digite a Chave correspondente.
  - O programa mostrará o resultado em Base64. Copie este valor.
6. Para cada caso 1 a 5 (decifrar):
  - Digite o Texto cifrado em Base64 (copiado na etapa anterior).
  - Digite a mesma Chave usada.
  - O programa exibirá o Texto decifrado, que deve ser idêntico ao Texto claro do caso.

### Observação importante:

- Foi utilizado o formato Base64 para representar o texto cifrado, garantindo que todos os bytes da tabela ASCII (0–255) possam ser manipulados de forma segura no terminal.
- Sempre utilize o valor Base64 gerado pelo programa para realizar a decriptação corretamente.