

# Relatório de Desenvolvimento - Lab01: Cifra de Vigenère

Este documento descreve o passo a passo do desenvolvimento do trabalho de implementação da Cifra de Vigenère, que utiliza a tabela ASCII com valores de 0 a 255.

## 1. Estudo da Cifra de Vigenère

O primeiro passo foi o estudo teórico da Cifra de Vigenère para compreender sua base matemática. A cifra é uma substituição polialfabética que utiliza uma palavra-chave para realizar a encriptação e decriptação.

Para um alfabeto de 256 caracteres (tabela ASCII), as operações são definidas como:

- **Encriptação:** O valor de cada caractere do texto cifrado ( $C_i$ ) é a soma do valor do caractere do texto claro ( $P_i$ ) com o valor do caractere correspondente da chave ( $K_i$ ), módulo 256.

$$C_i = (P_i + K_i) \pmod{256}$$

- **Decriptação:** A operação inversa. O valor do caractere do texto claro ( $P_i$ ) é obtido pela subtração do valor do caractere da chave ( $K_i$ ) do caractere cifrado ( $C_i$ ), módulo 256.

$$P_i = (C_i - K_i + 256) \pmod{256}$$

## 2. Implementação da Função de Encriptação

Seguindo a **Tarefa 1**, foi implementada a função `encriptar`. Esta função recebe como entrada o texto claro e a chave e retorna o texto encriptado. A lógica percorre cada caractere do texto original, aplicando a fórmula de encriptação com o caractere correspondente da chave (repetindo a chave, se necessário). Após a implementação, foram realizados testes manuais com valores simples para validar a lógica inicial.

## 3. Implementação da Função de Decriptação

De forma análoga, e de acordo com a **Tarefa 2**, foi implementada a função `decriptar`. Ela recebe o texto encriptado e a chave e retorna o texto claro original. A lógica aplica a fórmula de decriptação para reverter o processo. Esta função também foi validada inicialmente com testes manuais, usando os resultados gerados pela função de encriptação.

## 4. Criação e Automação do Plano de Testes

Para garantir a robustez e a correção do código, conforme a **Tarefa 3**, foi criado um plano de testes automatizado.

- 1. Criação do Arquivo de Testes:** Foi elaborado um arquivo de texto ( `test.txt` ) contendo no mínimo 5 casos de teste distintos. Estes casos foram projetados para cobrir cenários simples e extremos, como chave maior que o texto e o uso de caracteres especiais.
- 2. Implementação da Função de Testes:** Foi desenvolvida uma função em C que recebe o nome do arquivo de testes como um argumento de linha de comando. Essa função lê cada linha do arquivo, separa o texto claro e a chave, executa os procedimentos de encriptação e decriptação, e por fim, compara o resultado decriptado com o texto claro original para verificar se a implementação está correta.

## 5. Estrutura e Execução do Programa

A solução final consiste em um arquivo de código-fonte principal, `main.c`. Para a validação, um arquivo `test.txt` contendo o plano de testes é utilizado.

Para compilar o programa, basta executar o seguinte comando no terminal:

```
gcc main.c -o meu_programa
```

Após a compilação, o programa deve ser executado passando o nome do arquivo de testes como um argumento de linha de comando. Embora um arquivo `testes.txt` seja o padrão,

qualquer outro arquivo com casos de teste no formato correto pode ser utilizado.

```
./meu_programa test.txt
```