

Cifragem de César

Mário Leite

...

Um dos métodos mais simples de encriptar (codificar) mensagens é o clássico “Cifra de Cesar” ou “Cifragem de Cesar”. Neste método uma letra é substituída por outra do alfabeto depois de um *deslocamento* de determinado tamanho; por exemplo, a letra “K” seria substituída por “N”; “P” seria substituída por “S”, e assim por diante, considerando um deslocamento de tamanho três, por padrão. O nome deste método é creditado ao imperador romano Julio Cesar (100 AC – 44 AC) que o criou para enviar mensagens a seus generais. E com medo de que o mensageiro pudesse estar compactuado com o inimigo, ele misturava as letras nas mensagens de modo que só o destinatário sabia o que realmente estava escrito. Então, suponha que o imperador confiasse cegamente num de seus generais e quisesse ordenar sua volta à Roma para uma missão secreta; observe abaixo como poderia ser esta mensagem: A **figura 1a** mostra um esquema de como é feita a codificação da palavra “VOLTE” pelo método “Cifra de Cesar”.

- Mensagem normal: VOLTE A ROMA PARA VIGIAR BRUTUS
- Mensagem codificada: **YROWH D URPD SDUD YLJLDU EUXWXV**

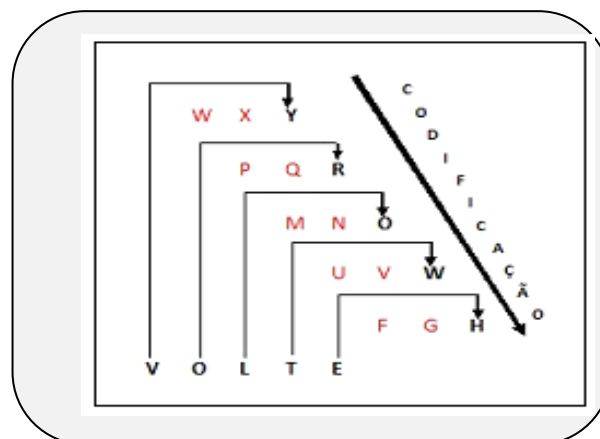


Figura 1a - Encriptação pela “Cifragem de Cesar”

Observe que a encriptação/cifragem ocorreu de forma bem simples: cada letra da mensagem normal foi substituída pela terceira letra APÓS ela; no caso a letra “V” foi substituída por “Y”, a letra “O” foi substituída por “R”, e assim por diante. O modo como o general descobria o teor real da mensagem era através da “chave” que lhe foi dada pelo imperador, da seguinte maneira: “*cada letra deve ser substituída pela terceira letra ANTES dela; de trás para frente*”. Em resumo era este o processo de decriptar/decodificar a mensagem. Veja esquema de decriptação da palavra “YROWH” na **figura 1b**.

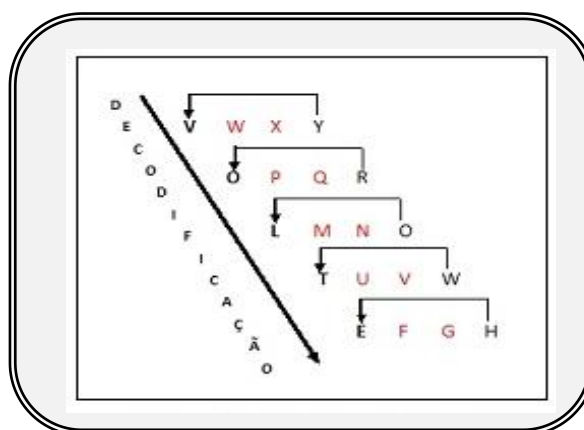


Figura 1b - Decriptação pela “Cifragem de Cesar”

O programa “**ProgCifraCesar**”, codificado em **Visualg**, apresenta uma solução mais genérica para o método “Cifra de Cesar”, aceitando deslocamentos de 3 a 25, com duas funções: **CifraCesar()** e **DecifraCesar()**. A primeira recebe o texto normal e o retorna cifrado; a segunda recebe um o cifrado e o retorna decifrado.

Nota: Na “Cifra de Cesar” com o deslocamento de tamanho 3 (três) das letras podemos ter apenas 26 (vinte e seis) possibilidades de “chaves” em potencial, o que torna a encriptação muito fácil de ser “quebrada”; mas este tamanho de deslocamento dos caracteres pode ser tratado como uma variável de entrada (como no programa “ProgCifraCesar”); assim o programa ficou mais geral. Normalmente, todos os métodos de deslocamento de caracteres monoalfabéticos são conhecidos como “Cifra de Cesar”. Entretanto, para se ter uma segurança maior nas cifragens de deslocamento é fortemente aconselhável utilizar um algoritmo mais geral de substituição, podendo ser conseguido um total de 403291461126605635584000000 “chaves” possíveis; bem mais seguro, né?!!

Código em Visualg 3.01

Algoritmo "ProgCifraCesar"

```
//Encripta/Decripta mensagem com o método "Cifragem de Cesar"
//Autor: Mário Leite
//data: 14/05/2023
//-----
//Variáveis globais
Var Op, MsgOriginal, MsgCifrada, MsgDecifrada: caractere
    TamMsgOriginal, TamMsgCifrada, Desloc: inteiro
//-----
funcao CifraCesar(MsgOrig:caractere; TamMsg,Desloc:inteiro): caractere
var j, NumDecif: inteiro
    VetMsgOrig, VetMsgCif: vetor[1..200] de caractere
    NumDecif, MsgCif, StrDecif : caractere
inicio
Para j De 1 Ate TamMsg Faca
    VetMsgOrig[j] <- Copia(MsgOrig,j,1)
    VetMsgOrig[j] <- Maiusc(VetMsgOrig[j]) //converte em maiúsculas
FimPara
{Monta a mensagem cifrada}
MsgCif <- ""
Para j De 1 Ate TamMsg Faca
    StrDecif <- VetMsgOrig[j]
    NumDecif <- Asc(StrDecif)
    Se((VetMsgOrig[j]=" ") ou (VetMsgOrig[j]=" ")) Entao
        VetMsgCif[j] <- " "
    Senao
        NumDecif <- NumPCarac(NumDecif) //variável estrutura Caractere
        Escolha (NumDecif) //trata exceções para cifrar X Y Z
            Caso "88"
                VetMsgCif[j] <- Carac(65)
            Caso "89"
                VetMsgCif[j] <- Carac(66)
            Caso "90"
                VetMsgCif[j] <- Carac(67)
            OutroCaso
                VetMsgCif[j] <- Carac(NumDecif+Desloc)
        FimEscolha //fim do tratamento das exceções X Y Z
    FimSe
    MsgCif <- MsgCif + VetMsgCif[j]
FimPara
Retorne MsgCif
fimfuncao //fim da função "CifraCesar"
//-----
```

```

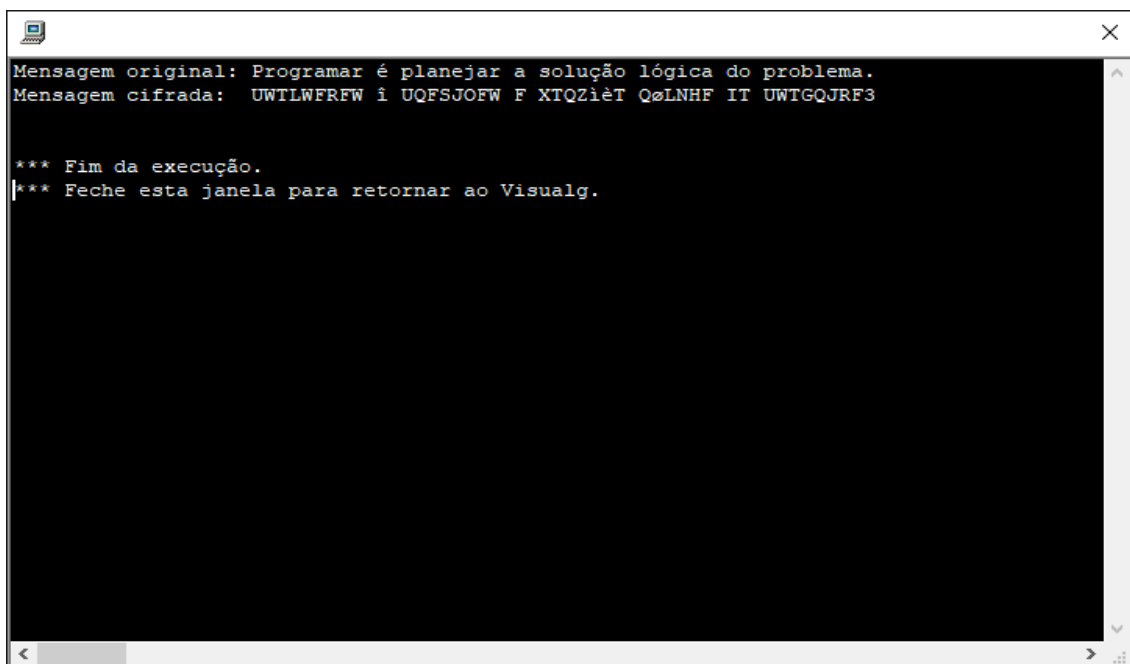
funcao DecifraCesar(MsgCif:caractere;TamMsg,Desloc:inteiro): caractere
    var j, NumCif: inteiro
    VetMsgCif, VetMsgDecif: vetor[1..200] de caractere
    NumFic, MsgDecif, StrCif : caractere
inicio
    Para j De 1 Ate TamMsg Faca
        VetMsgCif[j] <- Copia(MsgCif,j,1)
    FimPara
    {Monta a mensagem decifrada}
    MsgDecif <- ""
    Para j De 1 Ate TamMsg Faca
        StrCif <- VetMsgCif[j]
        NumCif <- Asc(StrCif)
        Se((VetMsgCif[j]=" ") ou (VetMsgCif[j]=" ")) Entao
            VetMsgDecif[j] <- " "
        Senao
            Se(NumCif<65) Entao
                NumCif <- NumCif+32 //garante caracteres normais
            Se(NumCif=81) Entao //exceção para o ponto (.) na mensagem
                VetMsgDecif[j] <- "."
            Senao
                VetMsgDecif[j] <- Carac(NumCif)
            FimSe
        Senao
            NumFic <- NumPCarac(NumCif) //variável estrutura Caractere
            Escolha (NumFic) //trata exceções para decifrar A B C
                Caso "65"
                    VetMsgDecif[j] <- Carac(88)
                Caso "66"
                    VetMsgDecif[j] <- Carac(89)
                Caso "67"
                    VetMsgDecif[j] <- Carac(90)
                OutroCaso
                    VetMsgDecif[j] <- Carac(NumCif-Desloc)
            Fimescolha //fim do tratamento das exceções de A B C
            FimSe
        FimSe
        MsgDecif <- MsgDecif + VetMsgDecif[j]
    FimPara
    Retorne MsgDecif
fimfuncao //fim da função "DecifraCesar"
//=====
//Programa principal
Inicio
    Op <- "X"
    Enquanto ((Op<>"C") e (Op<>"D")) Faca
        Escreva("Para cifrar [C] - Para decifrar [D]: ")
        Leia(op)
        Op <- Maiusc(Op)
    FimEnquanto
    Escreval("") //salta linha
    Desloc <- 0
    Enquanto ((Desloc<3) ou (Desloc>25)) Faca
        Escreva("Digite o tamanho do deslocamento [3 a 25]: ")
        Leia(Desloc)
    FimEnquanto
    Escreval("")
    TamMsgOriginal <- 2

```

```

Se(Op="C") Entao
    {Lê a mensagem original}
    Enquanto ((TamMsgOriginal<3) ou (TamMsgOriginal>200)) Faca
        Escreval("Digite a mensagem original [de 3 até 200 caracteres]: ")
        Leia(MsgOriginal)
        TamMsgOriginal <- Compr(MsgOriginal)
    FimEnquanto
    MsgCifrada <- CifraCesar(MsgOriginal,TamMsgOriginal,Desloc)
Senao
    TamMsgCifrada <- 2
    Enquanto ((TamMsgCifrada<3) ou (TamMsgCifrada>200)) Faca
        Escreval("Digite a mensagem cifrada [de 3 até 200 caracteres]: ")
        Leia(MsgCifrada)
        TamMsgCifrada <- Compr(MsgCifrada)
    FimEnquanto
    MsgDecifrada <- DecifraCesar(MsgCifrada,TamMsgCifrada,Desloc)
FimSe
LimpaTela
Escreval("")
Escreval("")
Escreval("Cifragem de Cesar")
Escreval("-----")
Escreval("")
Se(Op="C") Entao
    {Exibe a mensagem cifrada}
    Escreval("Mensagem original: ", MsgOriginal)
    Escreval("Mensagem cifrada: ", MsgCifrada)
Senao
    {Exibe a decifrada}
    Escreval("Mensagem cifrada: ",MsgCifrada)
    Escreval("Mensagem decifrada: ",MsgDecifrada)
FimSe
Escreval("")
FimAlgoritmo //fim do programa "ProgCifraCesar"

```



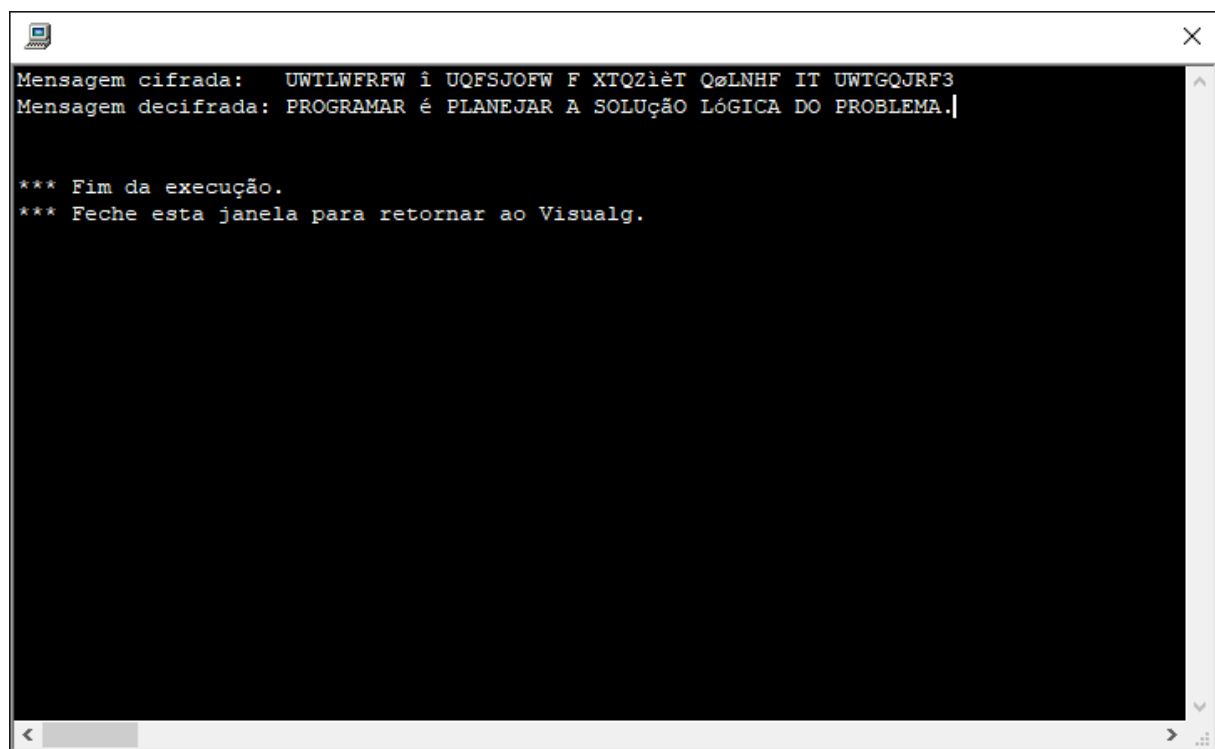
```

Mensagem original: Programar é planejar a solução lógica do problema.
Mensagem cifrada: UWTLWFRFW î UQFSJOFW F XTQZiêT QøLNHF IT UWIGQJRF3

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.

```

Figura 1a - Codificando a mensagem com deslocamento de cinco caracteres



A screenshot of a terminal window with a black background and white text. The window has a standard macOS-style title bar with a close button in the top right corner. The text inside the terminal is as follows:

```
Mensagem cifrada:  UWTLWFRFW i UQFSJOFW F XTQZièT QøLNHF IT UWIGQJRF3
Mensagem decifrada: PROGRAMAR é PLANEJAR A SOLUção LÓGICA DO PROBLEMA.|

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

The terminal also features a horizontal scrollbar at the bottom.

Figura 1b - Decodificando a mensagem com deslocamento de cinco caracteres