A importância e a necessidade da criptografia visando a restrição de informações em um desastre ambiental

Nomes: Isael Gomes de Oliveira Junior	N319445
Raphael Alves Fiore	T5241A9
Mattheus Rodrigues Alonso	D3134D-3
Jhonatas Ferreira Paschalis	D757GC9

SANTOS - SP

(2018)

SUMÁRIO

1. INTRODUÇÃO

O desejo de enviar mensagens de forma segura, ou seja, sem que ninguém mais, a não ser o destinatário, consiga ler, existe há muito tempo, até mesmo quando não havia computadores e muito menos o avanço tecnológico dos tempos atuais. Era conhecida e praticada há muito tempo, principalmente associadas as práticas militares.

As primeiras mensagens criptografadas consistiam na substituição de simples caracteres de uma mensagem ou de um determinado modo de adulterar as posições do caracteres de modo a que não fosse possível ler o texto sem o reordenar de uma dada forma. Em guerras primitivas tem-se relato de generais que raspavam a cabeça de um escravo e sobre a cabeça escreviam mensagens secretas de guerra e, após o cabelo crescer, o emissário era enviado para passar a informação, era uma técnica conhecida como *esteganografia*.

1.1 Histórico e Conceitos gerais

A definição de criptografia é bastante simples. Consiste no uso de técnicas e métodos para modificar texto ou dados legíveis, tornando esses dados inelegíveis, com exceção do destinatário. Podemos classificar a criptografia em Clássica, Medieval e Moderna.

A criptografia Clássica é a mais antiga conhecida pelo ser humano. Seu primeiro uso foi encontrado em hieróglifos irregulares esculpidos em monumentos do Antigo Império do Egito (a cerca de 4500 anos). Porém, não podem ser considerados como tentativas sérias de comunicações secretas, mas sim de ser mensagens misteriosas, intrigas ou mesmo diversão para os alfabetizados. A criptografia Medieval, que era relacionada quase que exclusivamente a uma consequência da competição política e revolução religiosa.

Por fim, temos a Criptografia Moderna, que começou durante a segunda guerra mundial, onde o matemático britânico Alan Turing quebrou a criptografia

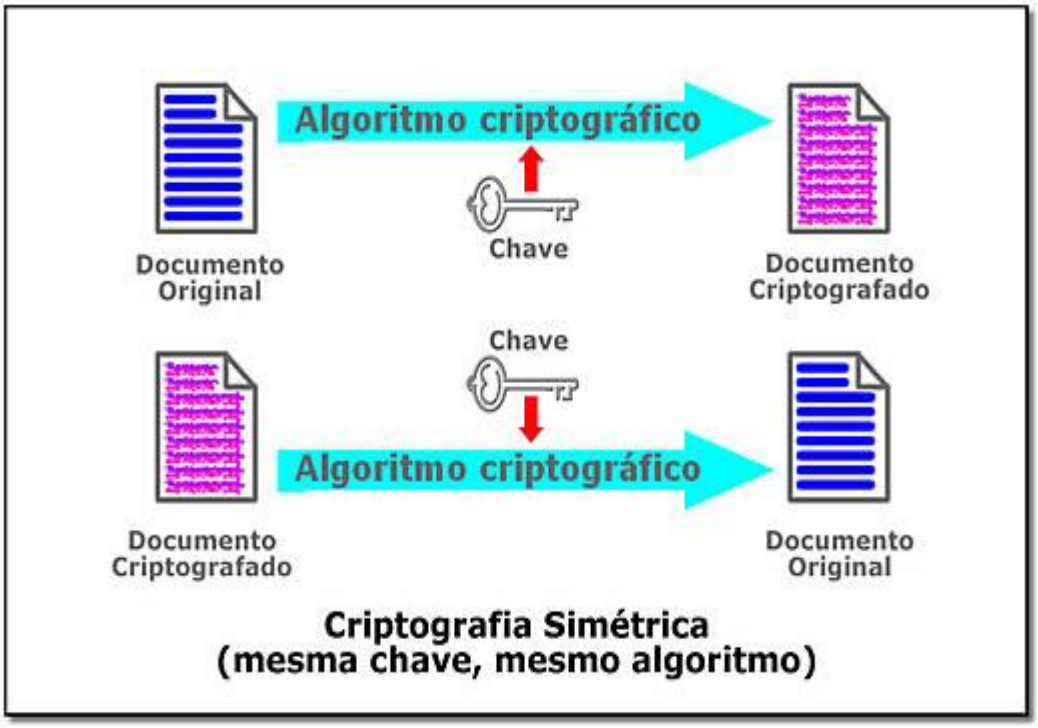
do exército alemão (máquina Enigma). Com o crescente aumento no uso da internet, apareceu a necessidade de aprimorar os mecanismos que promovem a segurança das transações de informações confidenciais.

O acúmulo de uma grande quantidade de dados e a conexão constante por meio de diferentes dispositivos são duas certezas que temos sobre o futuro. Empresas, do mais baixo porte até o maior, perceberam o quão importante é a segurança da informação no mundo da tecnologia. A segurança da informação e a criptografia estão totalmente relacionadas. Todos os dias, as empresas perdem milhares de dólares, seja com invasões propriamente ditas, onde ocorrem roubos de informações, como número de senhas de banco, ou até mesmo com um DDOS, que pode derrubar um servidor e deixar indisponível um serviço para milhões de usuários. Sendo assim, a necessidade também de melhorar os mecanismos que promovem a segurança das transações de informações credenciais.

Atualmente a criptografia se concentra no estudo de algoritmos que possam ser implementados em computadores. Esses algoritmos são baseados em bits e fazem o embaralhamento da informação a partir de uma chave ou um par de chaves. Os primeiros métodos utilizavam apenas um algoritmo, mas acabava que era difícil manter o sigilo. Sendo assim, quanto mais bits são aplicados no algoritmo, mais segura será sua criptografia. Através dela obtemos diversas propriedades importantes como a confidencialidade (sigilo da informação), integridade (garantia que a mensagem não foi alterada), autenticidade (quem foi a autor da mensagem) e irretratabilidade ou não repúdio (capacidade de não negar a construção da mensagem).

Na criptografia temos chaves simétricas e assimétricas. As chaves simétricas são os tipos mais simples de chaves, no qual tanto o emissor quanto receptor usam a mesma chave para codificar e decodificar a mensagem, como é possível ver na *figura 1*.

Figura 1: Figura demonstrando a atuação da chave simétrica



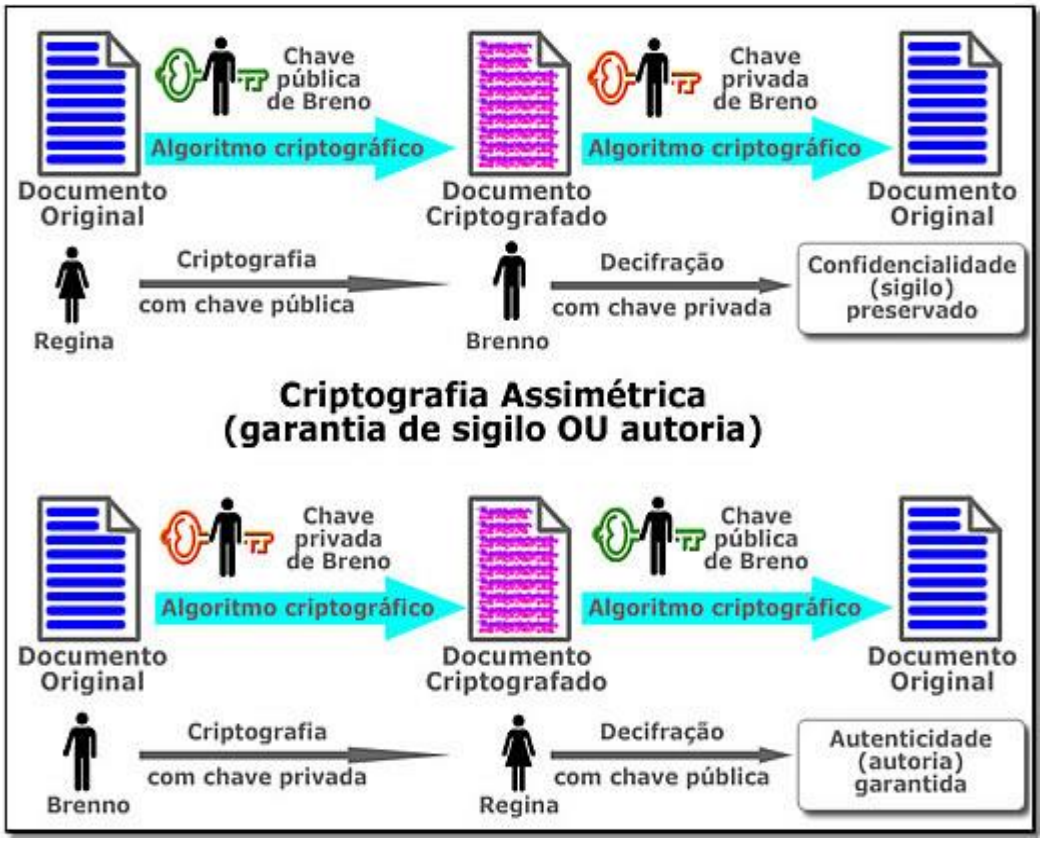
https://www.bpiropo.com.br/fpc20071203.htm

Alguns dos principais algoritmos que lidam com esse método são: DES (Data Encryption Standard, criado pela IBM em 1977, que oferece um chave de 56 bits, o que equivale a 72 quatrilhões de combinações, RES (Ron's Code) criado por Ron Rivest, possui chaves de 8 a 1024 bits, e IDEA (International Data Encryption Algorithm, criado em 1991, que faz uso de uma chave de 128 bits.

Nas assimétricas, serão idênticas as chaves utilizadas, tanto pelo fornecedor quanto pelo receptor, e assim, é melhor aproveitada para situações em que conseguimos enviar a chave pessoalmente, já que, quando enviamos pelo web, corremos um certo risco de estar exposta *(figura 2)*. Alguns exemplos são: ElGamal, que opera com um problema matemático conhecido como 'logaritmo discreto", sendo frequente em assinaturas digitais. Outro exemplo é o RSA (River, Shamir e Andleman), criado em 1977, sendo atualmente um dos algoritmos mais usados. Representa a tentativa de descobrir a chave pública multiplicando dois números primos para obtenção um terceiro valor. Através da criptografia obtemos diversas propriedades importantes como a

confidencialidade (sigilo da informação), integridade (garantia que a mensagem não foi alterada), autenticidade (quem foi a autor da mensagem) e irretratabilidade ou não repúdio (capacidade de não negar a construção da mensagem).

Figura 2: Demonstração de como ocorre a chave assimétrica



https://www.bpiropo.com.br/fpc20071203.htm

Outra classificação para os algoritmos é em relação aos métodos de operação que podem ser dois: de substituição e de transposição. Mais uma importante classificação é em relação ao modo de processamento que pode ser: os cifradores de bloco e cifradores de fluxo. Cifradores de Bloco operam sobre 8 bits ou 16 bits e funcionam com complementos para que todos blocos tenham o mesmo tamanho. Cifradores de Fluxo é onde a cifragem ocorre bit a bit continuo. Essas classificações são importantes e nos permitem melhor organizar cada algoritmo criptográfico.

1.2 Técnicas criptográficas mais utilizadas

Na criptografia há diversos tipos de técnicas, todas com o mesmo objetivo, de esconder a mensagem de qualquer um que não seja o destinatário. A primeira que será citada nesse estudo é a criptografia de *Hash*. O *Hash* nada mais é que uma sequência de bits que tem como objetivo identificar um arquivo. Isso significa que se realizarmos um processamento num arquivo será gerado um *hash* que é único, e dessa forma, alcançamos a garantia da integridade.

Utilizando o *hash* também temos a propriedade da unidirecionalidade onde o caminho de volta não é possível. Além disso, utilizando Hash não há a necessidade de chaves e, temos a garantia da consistência, pois se introduzirmos a mesma mensagem de Hash teremos exatamente o mesmo Hash sendo gerado. Por fim, o *hash* também nos oferece as propriedades de randomicidade e unicidade onde nunca temos a mesma mensagem de Hash para diferentes mensagens. Conforme é possível visualizar na figura 3, ela permite que através de uma *string* de qualquer tamanho, seja calculado um identificador digital de tamanho fixo, chamado de valor *hash*.

Talvez o grande problema do *hash* é que que estamos vulneráveis a ataques de dicionário, onde um atacante tem um banco de dados de senhas prováveis com seus respectivos *hashs*. Como uma solução para este problema tem-se o SALT que é uma porção aleatória de texto que é concatenado com a senha original, e assim sendo, por exemplo, pessoas com mesma senha obterão *hashs* diferentes.

O valor *hash* geralmente é formado por 16 bytes (no caso do MD-2, MD-4 e MD- 5) ou 20 bytes (no caso do SHA-1), mas pode se estender, embora não passe de 512 bytes. Seja uma função *hash* H, e x uma string qualquer, teremos que H(x) será o valor *hash* para a string X.

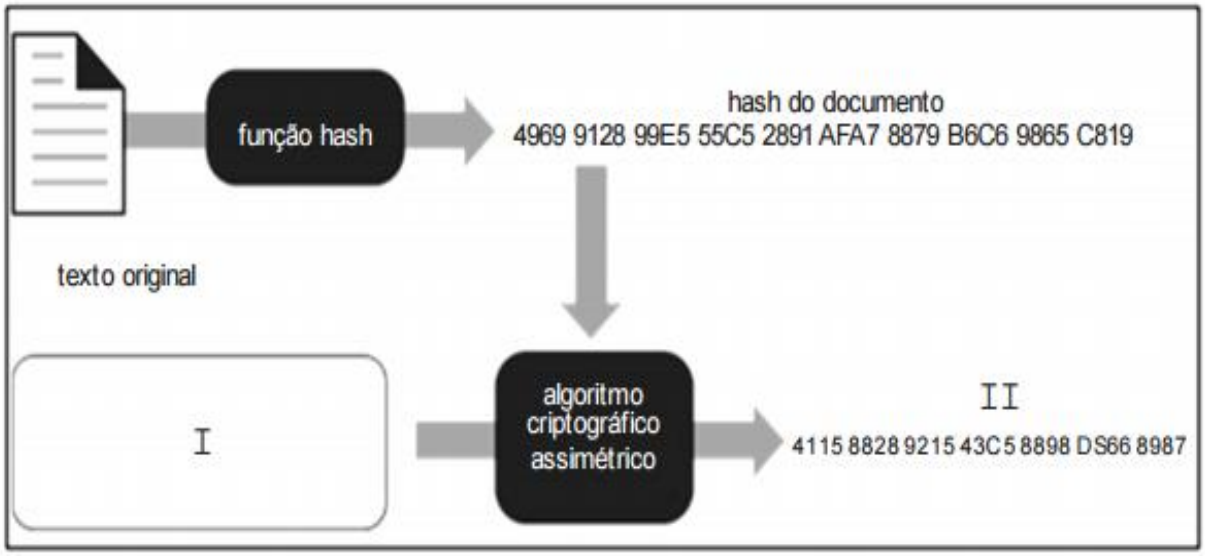
As características básicas de uma função *hash* são:

- O valor de entrada da função possui qualquer tamanho;

- O valor de saída da função possui tamanho fixo;

- *H*(*x*) é relativamente fácil de ser computado, para qualquer valor de *x*;
- *H*(*x*) é uma função “*one-way*”;
- *H*(*x*) é livre de colisão.

Figura 3: Criptografia de *Hash*



https://www.qconcursos.com/questoes-de-concursos/questoes/45805c17-ec

Uma função *hash* é dita "*one-way*” pois uma vez obtido o valor *hash* h para uma string *x*, é computacionalmente impossível fazer o processo inverso, ou seja, encontrar um valor *x* tal que *H*(*x*) = *h.*

Percebe-se ainda que H(x) é livre de colisão, significando que as funções *hash* devem garantir uma probabilidade mínima de que duas strings diferentes acabem por resultar no mesmo valor *hash*. Qualquer alteração na string original que deu origem ao identificador digital, mesmo que de um único bit, acabará por gerar uma alteração significativa no valor *hash* final.

A segunda técnica mencionada no trabalho é a criptografia monoalfabéticas (Hebreus). Antes de entrar a fundo nela, é necessário saber o

conceito de permutação. Uma permutação, na criptografia, um conjunto finito de elementos S em uma sequência ordenada de todos os elementos de S, com cada um aparecendo exatamente uma vez.

Por exemplo, se S = {a, b, c}, existem seis permutações de S: abc, acb, bac, bca, cab, cba, em geral, existem n! permutações de um conjunto de n elementos, pois o primeiro deles pode ser escolhido de n maneiras, o segundo, de n – 1 maneiras, o terceiro, de n – 2 maneiras, e assim por diante.

Entretanto, as cifras monoalfabéticas são fáceis de se quebrar porque refletem os dados de frequência do alfabeto original. Uma contramedida é oferecer vários substitutos, conhecidos como homófonos, para uma única letra. Por exemplo, a letra e poderia ser atribuída a diversos símbolos de cifra diferentes, como 16, 74, 35 e 21, com cada homófono usado em rodízio, ou aleatoriamente. Se o número de símbolos atribuídos a cada letra for proporcional à frequência relativa dela, então a informação de frequência de única letra é completamente extinta.

O terceiro tipo de cifra citado é a Cifra Playfar, que trata os diagramas no texto claro como unidades isoladas e as traduz para diagramas de texto cifrado. O algoritmo presente é baseado na utilização de uma matriz 5 x 5 de letras construídas usando uma palavra-chave (figura 4).

A Cifra Playfar Playfair é relativamente fácil de ser quebrada, pois ainda deixa intacta grande parte da estrutura da linguagem de texto claro, no entanto, ela é um grande avanço em relação às cifras monoalfabéticas simples. Além do mais, as frequências relativas das letras individuais exibem um intervalo muito maior do que o dos digramas, tornando a análise de frequência muito mais difícil. Por esses motivos, a cifra Playfair foi, por muito tempo, considerada indecifrável. Ela foi usada como sistema de campo padrão pelo Exército britânico na Primeira Guerra Mundial, e ainda gozava de um uso considerável pelo Exército dos Estados Unidos e outras forças aliadas durante a Segunda Guerra Mundial.

Figura 4: Cifra Playfar

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

http://wiki.stoa.usp.br/images/c/cf/Stallings-cap2e3.pdf

1.3 Tema APS

O presente trabalho foi sobre um vazamento óleo diesel que ocorreu na área industrial de Cubatão, no estado se São Paulo. Esse vazamento provocou a contaminação de um córrego e a morte de vários peixes. Os responsáveis Técnicos da Companhia Ambiental do Estado de São Paulo (Cetesb) descobriram que o vazamento de óleo diesel ocorreu, por volta das 19h de sábado, na área de tancagem da base da empresa Ipiranga, localizada no bairro Pilões, na área industrial de Cubatão.

Segundo os técnico, vazaram aproximadamente 15 metros cúbicos de óleo, que equivalem a 15 mil litros, e que atingiram um córrego localizado na frente do terminal. Entretanto, na verdade, a grande preocupação da equipe é que este vazamento foi muito mais grave e danoso ao meio ambiente do que está sendo informado. Os trabalhadores que cercam o local informam, a pedido do governo estadual de São Paulo, estão escondendo a verdadeiro situação que preocupa os morados não só de Cubatão, mas de toda baixada Santista.

Figura 5: Vazamento óleo diesel em Cubatão

https://g1.globo.com/sp/santos-regiao/noticia/2018/11/13/cetesb-diz-que-nao-houve-vazamento-de-oleo-diesel-em-corrego-de-cubatao.ghtml

1.3 Objetivo

O objetivo desse trabalho foi de criar um programa de criptografia, utilizando a técnica de Cifra de César, para permitir a comunicação, de forma sigilosa, entre os responsáveis técnicos da Cetesb e o governo estadual de São Paulo, com o intuito de impedir que qualquer outro indivíduo consiga ter acesso aos dados e informações que estão sendo trafegados entre os dois.

2. JUSTIFICATIVA

Com o agravamento do impacto de óleo diesel na região de Cubatão, houve a necessidade da criação do programa de criptografia, pois há a chance de afetar outras regiões da baixada Santista, e causar um alarde ainda maior entre os moradores, caso alguém tenha acesso a informações sigilosas.

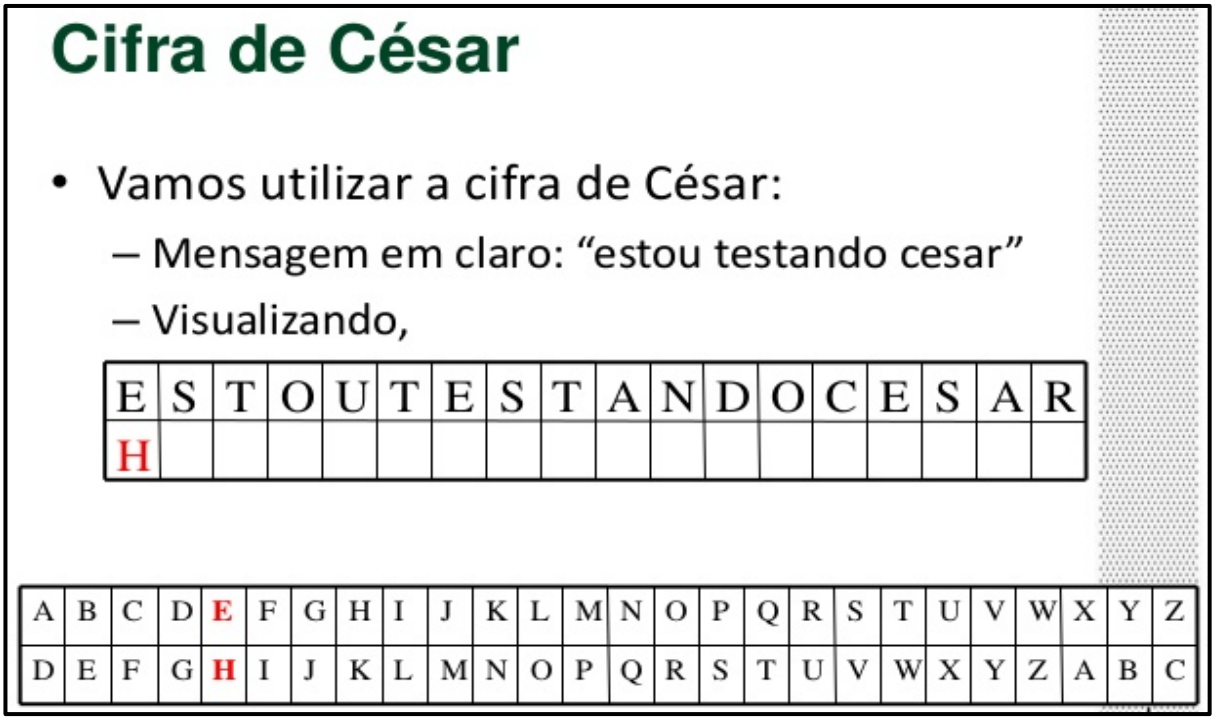
3. CIFRA DE CÉSAR

3.1 Estruturação, Conceitos e Fundamentação

A Criptografia de César, utilizada neste estudo, é uma das técnicas mais simples e conhecidas já utilizadas. Ela foi criada por um imperador romano conhecido como Júlio César, com o intuito de se comunicar com seus generais.

A técnica consiste em substituir uma letra qualquer do alfabeto pela sua terceira vizinha, ou seja, a letra “A” por exemplo, é transformada em “D”, em um alfabeto circular, onde o “Z” vira “C”, conforme mostra a figura 6.

Figura 6: Demonstração, de forma simples, como funciona a técnica.



https://pt.slideshare.net/anchises/criptografia-simetrica-e-assimtrica

Entretanto, deve-se notar que há duas operações distintas:

I - cifrar: Já foi informado que nessa técnica, para cifrar uma mensagem, deve-se substituir cada letra pela terceira vizinha subsequente. Na linguagem de programação C, por exemplo, usando variável do tipo string, isto poderia ser feito com: strCIF[i] = strTXT[i] + 3.

Uma pessoa que conhece bem a técnica, verá erro na fórmula acima: ela só funciona até o "W" que somado com 3 virará "Z". Para as letras "X", "Y" e "Z" a fórmula é falha! Aí entraria em ação a operação de módulo, muito utilizada em criptografia.

II - decifrar: Já para recuperar a mensagem outro método matemático precisa ser usado. Ao invés de trocar pela terceira subsequente, deve-se trocar pela terceira anterior ("D" deve virar "A").

Percebe-se nitidamente duas operações distintas para cifrar ou decifrar. Uma variação interessante da cifra de césar é a Rot13, onde se troca pela décima terceira vizinha. Ela é interessante pois como o alfabeto tem exatos 26 letras, o processo de cifrar e decifrar é o mesmo! Para cifrar basta achar a vizinha 13 subsequente e para decifrar também (a vizinha de "A" é "M" e a vizinha de "M" é o "A").

Outros exemplos utilizando essa técnica são:

Palavra 1: **claro**

Tabela 1: Palavra **claro**

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

Palavra 2: **cifra**

Tabela 2: Palavra **cifra**

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

3.2 Benefícios em relação as técnicas anteriores

As técnicas anteriores usavam o mesmo método de substituição, como a usada pelos hebreus, porém eles trocavam a primeira letra do alfabeto pela última, a segunda pela penúltima e assim invertendo o alfabeto. Nesse contexto, é perceptível que, naquela época, a cifra de César era mais segura.

Benefícios específicos da técnica:

- **Simplicidade para desenvolvimento de softwares que utilizem criptografia (como o do presente estudo).**
- **Facilidade para ensinar os conceitos básicos de criptografia.**
- **Velocidade e algoritmos rápidos.**
- **Chaves simples já são capazes de gerar textos cifrados.**
- **Facilidade de implementação de hardware por fazer uso de chaves simétricas.**

3.3 Aplicações que fizeram uso da técnica

A cifra, criada por Júlio César, foi primeiramente usada com o objetivo de proteger mensagens de caráter militar. Como já mencionada, era considerada na época a técnica mais segura de criptografia, até porque, existem relatos que

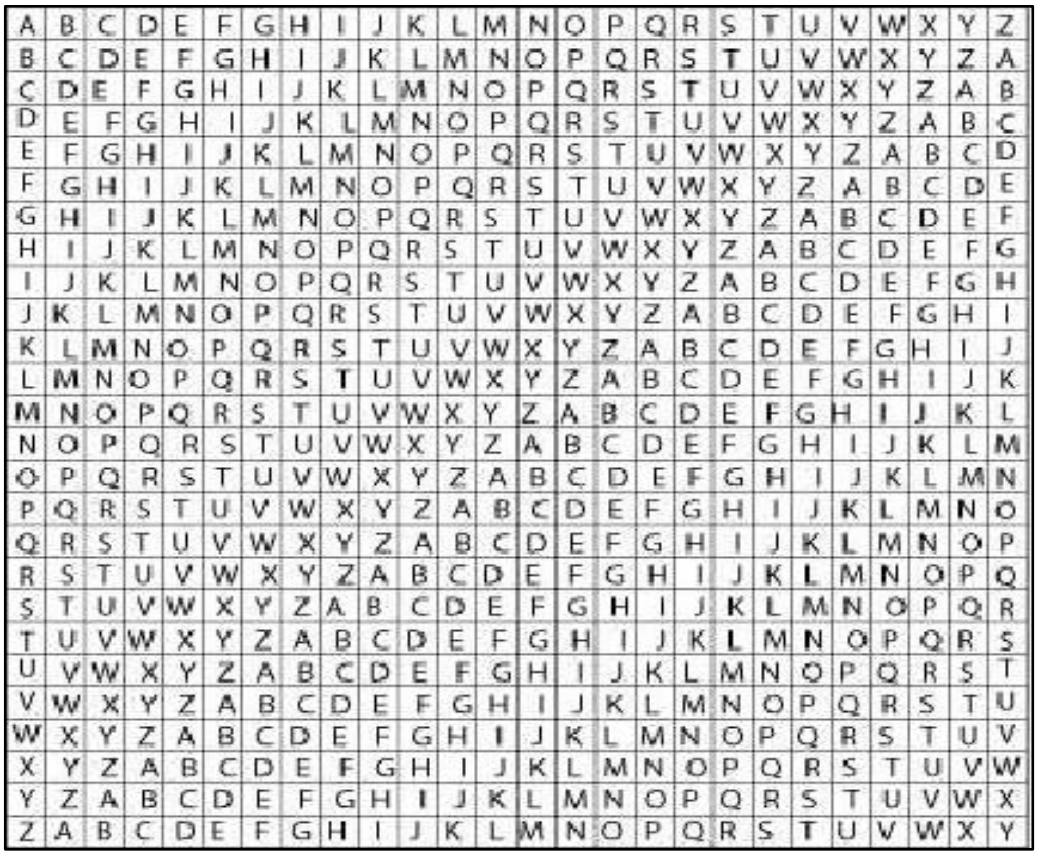
muitos inimigos de Júlio eram analfabetos, e outros imaginavam que ele estava escrevendo em uma linguagem estrangeira, desconhecida por eles.

No século XIX, a seção de anúncios pessoais nos jornais era por vezes utilizada para trocar mensagens criptografadas usando esquemas simples de criptografia. Alguns exemplos de amantes utilizando comunicações secretas criptografadas usando a cifra de César no *The Times*. Até mesmo em 1915, a cifra de César continuava em uso: o exército russo empregou-a em substituição às cifras mais complicadas que provaram serem muito difíceis de suas tropas entenderem; no entanto, criptoanalistas alemães e austríacos tiveram pouca dificuldade em descriptografar suas mensagens.

De acordo com Jonathan Strickland (2007), outro exemplo de uso foi por Johannes Trimethius, que se propôs a colocar o alfabeto em uma matriz. A matriz possuía 26 linhas e colunas. Sendo que, a primeira linha tinha o alfabeto escrito normalmente. A linha seguinte usava a cifra de César para mover o alfabeto sobre um espaço. Cada linha alterava o alfabeto em um outro ponto para que a linha final iniciasse com a letra “z” e terminasse com a letra “y, conforma a figura 7.

Nos dias atuais, as cifras de César ainda são encontradas, principalmente em brinquedos infantis, como os anéis decodificadores. Outro exemplo conhecido, é a presença dela no algoritmo ROT13, método este considerado simples, visando ofuscar o texto. Esse algoritmo está presente no *Unix.*

Figura 7: Quadro de Trimethius



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Fonte: http://ciencia.hsw.uol.com.br/cracker2.htm

3.3 Discussão Comparativa

É uma técnica, como já dito, muito simples comparado aos métodos de criptografia que existem atualmente, onde há algoritmos de chaves assimétricas, como por exemplo o RSA, já citado no estudo. Entretanto, principalmente com o objetivo de usar para fins didáticos, como estudantes, é uma técnica para iniciar os estudos sobre criptografia. Permitindo assim, a partir dela, se aprofundar e compreender os diversos modos de se criptografar e descriptografar um texto.

3.4 Vulnerabilidades, falhas e melhorias

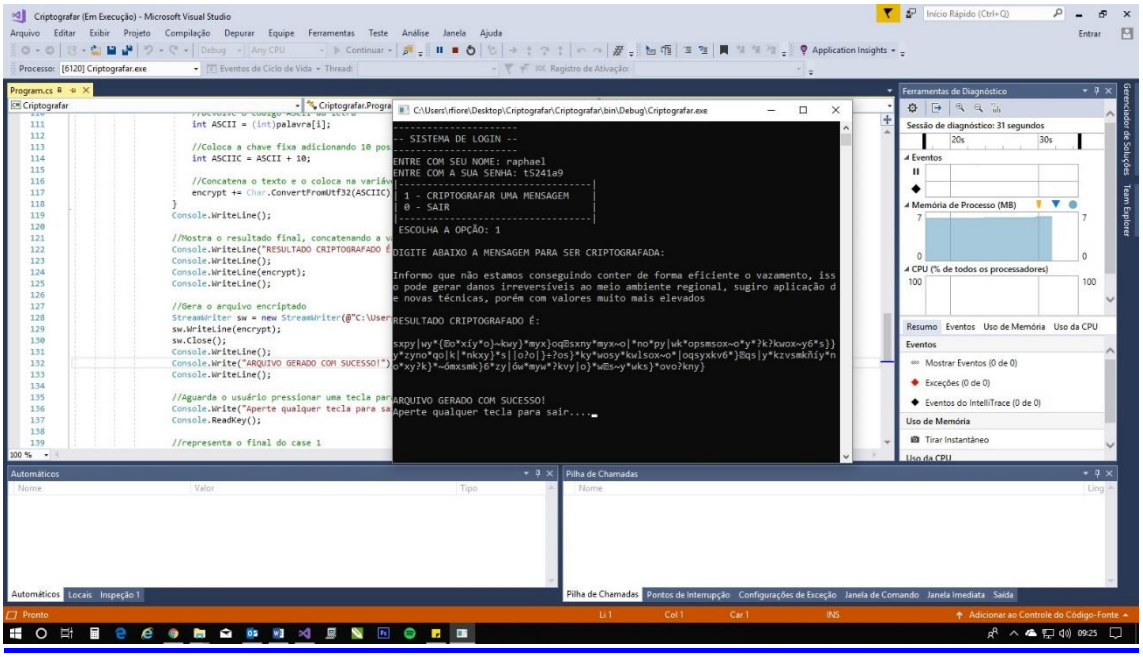
No que diz respeito as falhas, podem ser citados:

- **Possui um número limitado e pequeno de chaves.**
- **Pode sofrer diversos tipos de criptoanálise.**
- **Não permite a autenticação.**
- **Não permite a irretratabilidade do remetente.**
- **Dificuldade de gerenciamento e transmissão de chaves.**

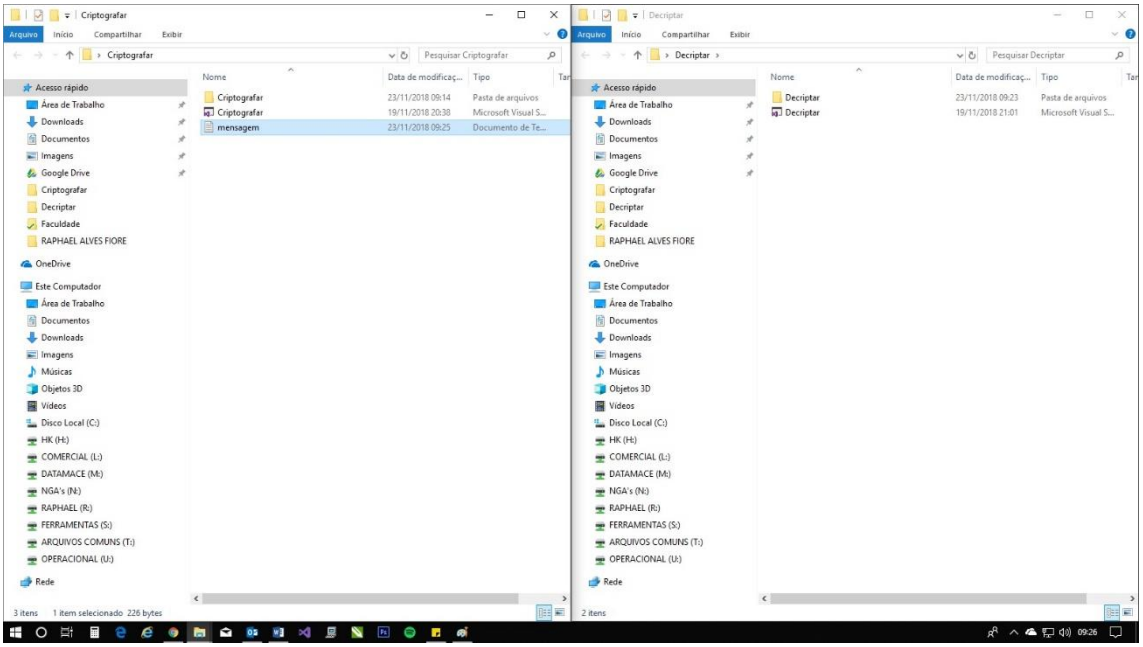
Pode ser citado, em relação a melhorias, um número maior de símbolos no texto que está com a criptografia, pois não existiram somente letras, mas também números e caracteres considerados especiais dentro dele, tornando assim mais difícil a descriptografia por alguns tipos de criptoanálise. Tudo isso, graças a ajuda dos computadores e implementação da tabela ASCII.

4. PROGRAMA FUNCIONANDO

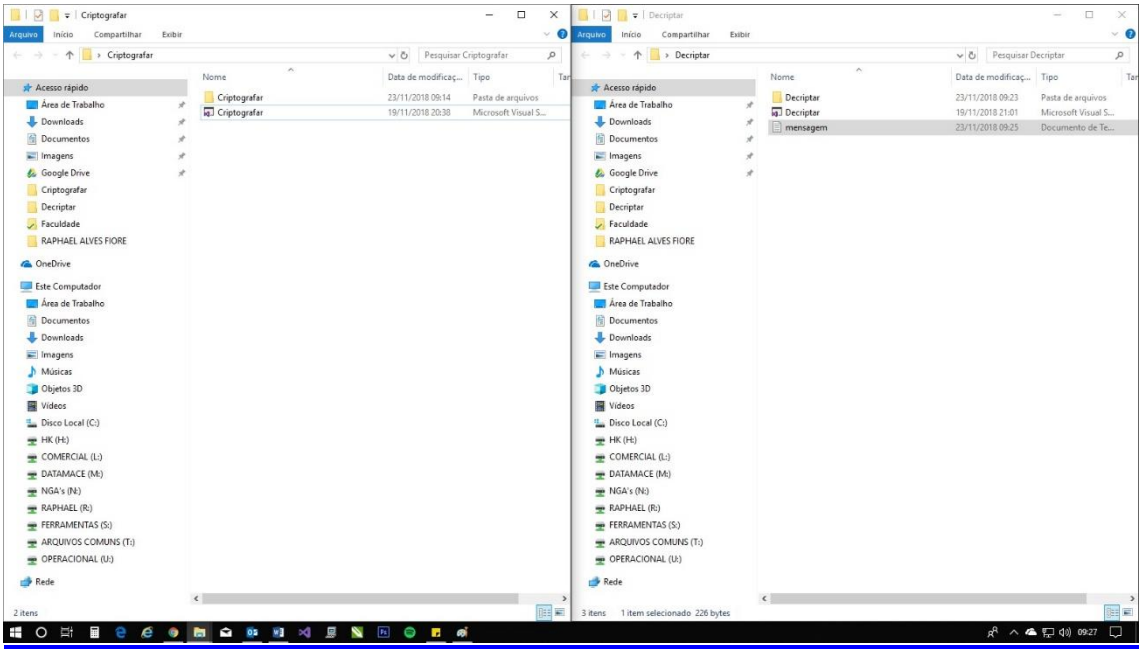
Encriptando o Arquivo:



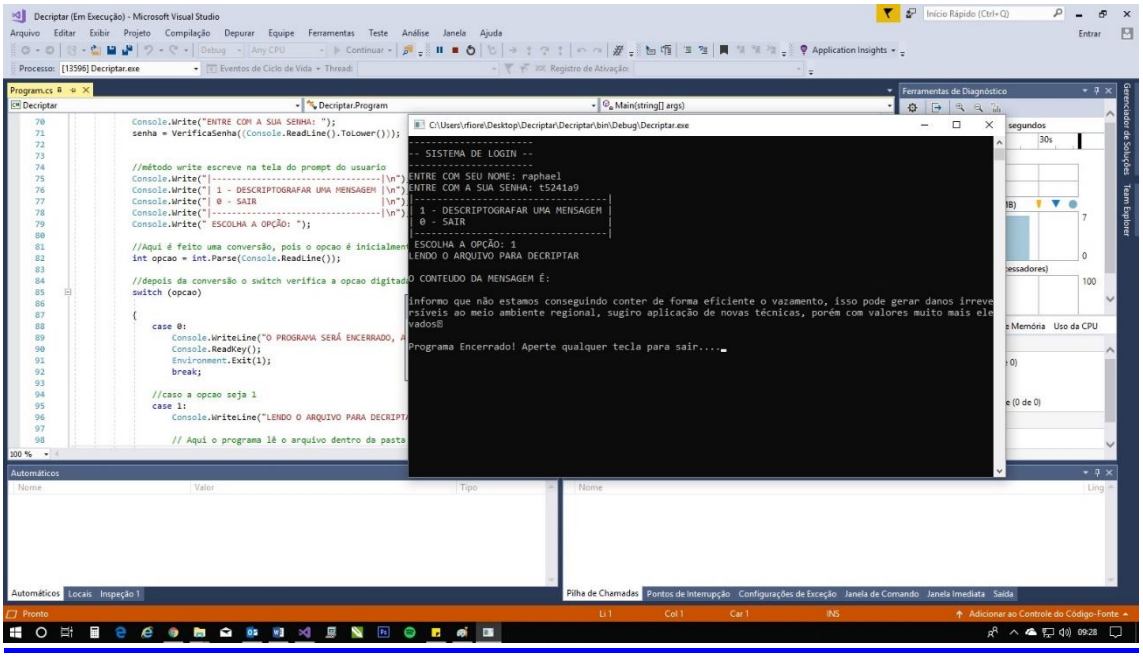
Arquivo Gerado:



Arquivo Enviado:



Arquivo Decriptado:



4.1 LINHAS DE CÓDIGO

C:\Users\Raphael\Desktop\Criptografar\Criptografar\Program.cs 1

```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.IO;
 4  using System.Linq;
 5  using System.Text;
 6  using System.Threading.Tasks;
 7
 8  namespace Criptografar
 9  {
10      class Program
11      {
12          /*Desenvolvido por:
13            Raphael Alves Fiore            - R.A: T5241A9  - CC2941
14            Mattheus Rodrigues Alonso      - R.A: D3134D-3 - CC2P41
15            Isael Gomes de Oliveira Junior - R.A: N319445 - SI2P41
16            Jhonatas Ferreira Paschalis    - R.A: D757GC9 - CC2P41
17          */
18          static string VerificaNome(string name)
19          {
20              int countSenha = 0;
21              while ((name != "raphael") && (name != "isael") && (name !=
                  "mattheus") && (name != "jhonatas") && countSenha < 3)
22              {
23                  Console.WriteLine("USUÁRIO NÃO AUTORIZADO!");
24                  Console.Write("ENTRE COM SEU NOME: ");
25                  countSenha++;
26                  name = Console.ReadLine().ToLower();
27              }
28              if (countSenha == 3)
29              {
30                  Console.WriteLine();
31                  Console.WriteLine("EXCEDEU O MÁXIMO DE TENTATIVAS, VOCÊ NÃO
                      ESTÁ AUTORIZADO E O PROGRAMA SERÁ ENCERRADO, APERTE QUALQUER
                       TECLA..");
32                  Console.ReadKey();
33                  Environment.Exit(1);
34              }
35              return name;
36          }
37
38          static string VerificaSenha(string pass)
39          {
40              int countSenha = 0;
41              while ((pass != "t5241a9") && (pass != "n319445") && (pass !=
                  "d3134d3") && (pass != "d757gc9")
42                  && countSenha < 3)
43              {
44                  Console.WriteLine("SENHA NÃO AUTORIZADA!");
45                  Console.Write("ENTRE COM A SUA SENHA: ");
46                  countSenha++;
47                  pass = Console.ReadLine();
48              }
49              if (countSenha == 3)
50              {
51                  Console.WriteLine();
52                  Console.WriteLine("EXCEDEU O MÁXIMO DE TENTATIVAS, O PROGRAMA
```

C:\Users\Raphael\Desktop\Criptografar\Criptografar\Program.cs 2

```
                    SERÁ ENCERRADO, APERTE QUALQUER TECLA..");
53                  Console.ReadKey();
54                  Environment.Exit(1);
55              }
56              return pass;
57          }
58
59          static void Main(string[] args)
60          {
61              //declaração das variáveis
62              string palavra, encrypt = "";
63              string nome, senha = "";
64
65              //método write escreve na tela do prompt do usuario
66              Console.WriteLine("----------------------");
67              Console.WriteLine("-- SISTEMA DE LOGIN --");
68              Console.WriteLine("----------------------");
69
70              Console.Write("ENTRE COM SEU NOME: ");
71              nome = VerificaNome((Console.ReadLine().ToLower()));
72
73              Console.Write("ENTRE COM A SUA SENHA: ");
74              senha = VerificaSenha((Console.ReadLine().ToLower()));
75
76
77              //método write escreve na tela do prompt do usuario
78              Console.Write("|--------------------------------|\n");
79              Console.Write("| 1 - CRIPTOGRAFAR UMA MENSAGEM    |\n");
80              Console.Write("| 0 - SAIR                         |\n");
81              Console.Write("|--------------------------------|\n");
82              Console.Write(" ESCOLHA A OPÇÃO: ");
83
84              //Aqui é feito uma conversão, pois o opcao é inicialmente uma
                  string
85              int opcao = int.Parse(Console.ReadLine());
86
87              //depois da conversão o switch verifica a opcao digitada
88              switch (opcao)
89
90              {
91                  case 0:
92                      Console.WriteLine("O PROGRAMA SERÁ ENCERRADO, APERTE
                        QUALQUER TECLA..");
93                      Console.ReadKey();
94                      Environment.Exit(1);
95                      break;
96
97                  //caso a opcao seja 1
98                  case 1:
99                      Console.Write("Entre com a mensagem para ser
                        criptografada: ");
100
101                     //palavra é a variavel que o usuario vai digitar
102                     //O método .ToLower() transforma qualquer letra maiúscula
                        em minúscula
103                     palavra = Console.ReadLine().ToLower();
```

C:\Users\Raphael\Desktop\Criptografar\Criptografar\Program.cs 3

```
104
105                     //enquanto a palavra for menor que i
106                     for (int i = 0; i < palavra.Length; i++)
107
108                     {
109                         //Devolve o codigo ASCII da letra
110                         int ASCII = (int)palavra[i];
111
112                         //Coloca a chave fixa adicionando 10 posições no
                        numero da tabela ASCII
113                         int ASCIIC = ASCII + 10;
114
115                         //Concatena o texto e o coloca na variável
116                         encrypt += Char.ConvertFromUtf32(ASCIIC);
117                     }
118
119                     //Mostra o resultado final, concatenando a variável em que
                         está o texto cifrado
120                     Console.WriteLine("Resultado Criptografado é: " +
                    encrypt);
121
122                     //Gera o arquivo encriptado
123                     StreamWriter sw = new StreamWriter(@"C:\Users\Raphael
                    \Desktop\Criptografar\mensagem.txt");
124                     sw.WriteLine(encrypt);
125                     sw.Close();
126                     Console.WriteLine();
127                     Console.Write("Arquivo Gerado com Sucesso!\n");
128
129
130                     //Aguarda o usuário pressionar uma tecla para sair
131                     Console.Write("Aperte qualquer tecla para sair....");
132                     Console.ReadKey();
133
134                     //representa o final do case 1
135                     break;
136
137                 default:
138                     if ((opcao < 0)|| (opcao > 1))
139                     {
140                         Console.WriteLine("NUMERO INVÁLIDO, O PROGRAMA SERÁ
                    ENCERRADO, APERTE QUALQUER TECLA..");
141                         Console.ReadKey();
142                         Environment.Exit(1);
143                     }
144
145                     break;
146
147             }
148
149         }
150     }
151 }
152
```

C:\Users\Raphael\Desktop\Decriptar\Decriptar\Program.cs 1

```
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace Decriptar
9  {
10     class Program
11     {
12         /*Desenvolvido por:
13           Raphael Alves Fiore            - R.A: T5241A9  - CC2941
14           Mattheus Rodrigues Alonso      - R.A: D3134D-3 - CC2P41
15           Isael Gomes de Oliveira Junior - R.A: N319445 - SI2P41
16           Jhonatas Ferreira Paschalis    - R.A: D757GC9 - CC2P41
17         */
18         static string VerificaNome(string name)
19         {
20             int countSenha = 0;
21             while ((name != "raphael") && (name != "isael") && (name !=
                 "mattheus") && (name != "jhonatas") && countSenha < 3)
22             {
23                 Console.WriteLine("USUÁRIO NÃO AUTORIZADO!");
24                 Console.Write("ENTRE COM SEU NOME: ");
25                 countSenha++;
26                 name = Console.ReadLine().ToLower();
27             }
28             if (countSenha == 3)
29             {
30                 Console.WriteLine("EXCEDEU O MÁXIMO DE TENTATIVAS, VOCÊ NÃO
                     ESTÁ AUTORIZADO E O PROGRAMA SERÁ ENCERRADO, APERTE QUALQUER
                      TECLA..");
31                 Console.ReadKey();
32                 Environment.Exit(1);
33             }
34             return name;
35         }
36
37         static string VerificaSenha(string pass)
38         {
39             int countSenha = 0;
40             while ((pass != "t5241a9") && (pass != "n319445") && (pass !=
                 "d3134d3") && (pass != "d757gc9")
41                 && countSenha < 3)
42             {
43                 Console.WriteLine("SENHA NÃO AUTORIZADA!");
44                 Console.Write("ENTRE COM A SUA SENHA: ");
45                 countSenha++;
46                 pass = Console.ReadLine();
47             }
48             if (countSenha == 3)
49             {
50                 Console.WriteLine("EXCEDEU O MÁXIMO DE TENTATIVAS, O PROGRAMA
                     SERÁ ENCERRADO, APERTE QUALQUER TECLA..");
51                 Console.ReadKey();
```

C:\Users\Raphael\Desktop\Decriptar\Decriptar\Program.cs 2

```
52                  Environment.Exit(1);
53              }
54              return pass;
55          }
56          static void Main(string[] args)
57          {
58              //declaração das variáveis
59              string palavra, encrypt = "";
60              string nome, senha = "";
61
62              //método write escreve na tela do prompt do usuario
63              Console.WriteLine("----------------------");
64              Console.WriteLine("-- SISTEMA DE LOGIN --");
65              Console.WriteLine("----------------------");
66
67              Console.Write("ENTRE COM SEU NOME: ");
68              nome = VerificaNome((Console.ReadLine().ToLower()));
69
70              Console.Write("ENTRE COM A SUA SENHA: ");
71              senha = VerificaSenha((Console.ReadLine().ToLower()));
72
73
74              //método write escreve na tela do prompt do usuario
75              Console.Write("|---------------------------------|\n");
76              Console.Write("| 1 - DESCRIPTOGRAFAR UMA MENSAGEM |\n");
77              Console.Write("| 0 - SAIR                         |\n");
78              Console.Write("|---------------------------------|\n");
79              Console.Write(" ESCOLHA A OPÇÃO: ");
80
81              //Aqui é feito uma conversão, pois o opcao é inicialmente uma
                  string
82              int opcao = int.Parse(Console.ReadLine());
83
84              //depois da conversão o switch verifica a opcao digitada
85              switch (opcao)
86
87              {
88                  case 0:
89                      Console.WriteLine("O PROGRAMA SERÁ ENCERRADO, APERTE
                        QUALQUER TECLA..");
90                      Console.ReadKey();
91                      Environment.Exit(1);
92                      break;
93
94                  //caso a opcao seja 1
95                  case 1:
96                      Console.WriteLine("Lendo o arquivo para decriptar");
97
98                      // Aqui o programa lê o arquivo dentro da pasta do
                        programa
99                      palavra = File.ReadAllText(@"C:\Users\Raphael\Desktop
                        \Decriptar\mensagem.txt");
100                     Console.WriteLine();
101
102                     for (int i = 0; i < palavra.Length; i++)
103
```

C:\Users\Raphael\Desktop\Decriptar\Decriptar\Program.cs 3

```
104                     {
105
106                         int ASCII = (int)palavra[i];
107
108                         //Coloca a chave fixa retirando 10 posições no numero da tabela ASCII
109                         int ASCIIC = ASCII - 10;
110
111                         encrypt += Char.ConvertFromUtf32(ASCIIC);
112
113                     }
114
115                     //exibe a mensagem descriptografada
116                     Console.WriteLine("O Conteudo da mensagem é: " + encrypt);
117
118                     //Aguarda o usuário pressionar uma tecla para sair
119                     Console.WriteLine();
120                     Console.Write("Programa Encerrado! Aperte qualquer tecla para sair....");
121                     Console.ReadKey();
122                     break;
123
124                     default:
125                     if ((opcao < 0) || (opcao > 1))
126                     {
127                         Console.WriteLine("NUMERO INVÁLIDO, O PROGRAMA SERÁ ENCERRADO, APERTE QUALQUER TECLA..");
128                         Console.ReadKey();
129                         Environment.Exit(1);
130
131                     }
132                     break;
133             }
134         }
135     }
136 }
137
```

6. REFERÊNCIAS BIBLIOGRÁFICAS

http://www.math.stonybrook.edu/~moira/mat331-spr10/papers/1987%20LucianoCryptology%20From%20Caesar%20Ciphers%20to.pdf

https://pplware.sapo.pt/informacao/conhea-a-historia-da-criptografia/

https://www.devmedia.com.br/criptografia-conceitos-padroes-e-um-pouco-de-historia-net-magazine-63/13540

https://cryptoid.com.br/banco-de-noticias/a-historia-da-criptografia/

https://www.estudopratico.com.br/criptografia/

https://pt.khanacademy.org/computing/computer-science/cryptography

https://www.psafe.com/blog/encriptacao-dados/

https://www.devmedia.com.br/criptografia-assimetrica-criptografando-e-descriptografando-dados-em-java/31213

https://www.kaspersky.com.br/resource-center/definitions/encryption

https://www.bpiropo.com.br/fpc20071203.htm

https://www.devmedia.com.br/como-funciona-a-criptografia-hash-em-java/31139

http://www.academia.edu/30198771/Criptografia_-_Cifra_de_C%C3%A9sar

https://www.infowester.com/criptografia.php

https://br.ccm.net/contents/131-o-que-e-criptografia

https://www.psafe.com/blog/o-que-e-criptografia/

https://www.ccmtecnologia.com.br/blog/criptografia-de-rede-questao-de-seguranca-importancia-para-uma-vpn

https://pt.stackoverflow.com/questions/68300/criptografia-e-seus-bits-como-explicar

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, Iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME: ISAEL GOMES DE OLIVEIRA JUNIOR

RA: N319445 **CURSO:** SISTEMAS DA INFORMAÇÃO

CAMPUS: UNIP - SANTOS **SEMESTRE:** SI2P41 **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
10/2018	REUNIÃO E ESCOLHA DO TEMA	10 HORAS		
10/2018	PESQUISA E ESTUDO SOBRE O TEMA	10 HORAS		
11/2018	DESENVOLVIMENTO DO PROJETO EM IDE	20 HORAS		
11/2018	REVISÃO E DIAGRAMAÇÃO DA APS	10 HORAS		

TOTAL DE HORAS: 50 HORAS

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, Iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME: JHONATAS FERREIRA PASCHALIS

RA: D757GC9 **CURSO:** CIENCIAS DA COMPUTAÇÃO

CAMPUS: UNIP - SANTOS **SEMESTRE:** CC2P41 **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
10/2018	REUNIÃO E ESCOLHA DO TEMA	10 HORAS		
10/2018	PESQUISA E ESTUDO SOBRE O TEMA	10 HORAS		
11/2018	DESENVOLVIMENTO DO PROJETO EM IDE	20 HORAS		
11/2018	REVISÃO E DIAGRAMAÇÃO DA APS	10 HORAS		

TOTAL DE HORAS: 50 HORAS

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, Iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME: MATTHEUS RODRIGUES ALONSO

RA: D3134D-3 **CURSO:** CIENCIAS DA COMPUTAÇÃO

CAMPUS: UNIP - SANTOS **SEMESTRE:** CC2P41 **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
10/2018	REUNIÃO E ESCOLHA DO TEMA	10 HORAS		
10/2018	PESQUISA E ESTUDO SOBRE O TEMA	10 HORAS		
11/2018	DESENVOLVIMENTO DO PROJETO EM IDE	20 HORAS		
11/2018	REVISÃO E DIAGRAMAÇÃO DA APS	10 HORAS		

TOTAL DE HORAS: 50 HORAS

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, Iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME: RAPHAEL ALVES FIORE

RA: T5241A9 **CURSO:** CIENCIAS DA COMPUTAÇÃO

CAMPUS: UNIP - SANTOS **SEMESTRE:** CC2P41 **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
10/2018	REUNIÃO E ESCOLHA DO TEMA	10 HORAS		
10/2018	PESQUISA E ESTUDO SOBRE O TEMA	10 HORAS		
11/2018	DESENVOLVIMENTO DO PROJETO EM IDE	20 HORAS		
11/2018	REVISÃO E DIAGRAMAÇÃO DA APS	10 HORAS		

TOTAL DE HORAS: 50 HORAS