UNIVERSIDADE PAULISTA

CURSO DE CIENCIAS DA COMPUTACAO

BRUNO DA SILVA PIMENTEL – N844132

GUILHERME VIEIRA ABBENANTE GOMES – N8959E0

HITALO CHAVES DOS SANTOS – N841419

JOÃO VICTOR CRISCI – F3445H0

MURILO HENRIQUE MARTINS REIS – N843276

AS TÉCNICAS CRIPTOGRÁFICAS, CONCEITOS, USOS E APLICAÇÕES

SÃO PAULO

2022

BRUNO DA SILVA PIMENTEL – N844132

GUILHERME VIEIRA ABBENANTE GOMES – N8959E0

HITALO CHAVES DOS SANTOS – N841419

JOÃO VICTOR CRISCI – F3445H0

MURILO HENRIQUE MARTINS REIS – N843276

AS TÉCNICAS CRIPTOGRÁFICAS, CONCEITOS, USOS E APLICAÇÕES

Trabalho de APS apresentado ao Curso de Ciências da computação da Faculdade UNIP de Tatuapé, como requisito parcial para obtenção da nota final.

SÃO PAULO

2022

SUMÁRIO

1 OBJETIVO DO TRABALHO 4

2 INTRODUÇÃO 4

3 CRIPTOGRAFIA CONCEITOS GERAIS 6

4 TÉCNICAS CRIPTOGRÁFICAS MAIS UTILIZADAS E CONHECIDAS 10

4.1 Sistema Binário 10

4.2 Máquina Enigma 10

4.3 Cifra de César 11

4.4 Cifra de Vigenère 11

4.5 Código Morse 13

4.6 Advanced Encryption Standard (AES) 13

4.7 Rivest-Shamir-Adleman (RSA) 14

5 DISSERTAÇÃO (CIFRA DE CÉSAR) 15

5.1 Estruturação, conceitos e fundamentação 15

5.2 Benefícios em relação às técnicas anteriores 17

5.3 Aplicações que fazem/fizeram uso da técnica 18

5.3.1 Cifra de Vigenère. 18

5.4 Discussão comparativa entre esta técnica e outras conhecidas/utilizadas 19

5.5 Vulnerabilidades e falhas 20

5.6 Melhorias propostas e/ou implementadas 21

6 PROJETO (ESTRUTURA) DO PROGRAMA 22

7 RELATÓRIO COM AS LINHAS DE CÓDIGO DO PROGRAMA 30

8 APRESENTAÇÃO DO PROGRAMA EM FUNCIONAMENTO EM UM COMPUTADOR, APRESENTANDO TODAS AS FUNCIONALIDADES PEDIDAS E EXTRAS. 36

9 CONCLUSÃO 37

10 BIBLIOGRAFIA 39

11 FICHAS DE ATIVIDADE PRÁTICA SUPERVISIONADA (APS) 40

1 OBJETIVO DO TRABALHO

O nosso objetivo é demonstrar as possibilidades que temos quando fazemos algoritmos, procedimentos, e funções específicas para adquirirmos um resultado. Nesse caso, estudamos sobre criptografia, algo muito presente no nosso dia a dia, mesmo implicitamente, em programas de computador/celular, que envolvem troca de mensagens, troca de dados, e até mesmo em partes mais críticas da sociedade como instituições financeiras e sistemas controlados pelo governo, devido à necessidade de reforçar o sigilo nas transações cruciais de dados entre eles. E para nós, foi proposto de arquitetar uma lógica estruturada e uma simples interface para simular uma comunicação entre a guarda costeira brasileira e um navio transportando lixo tóxico originado da Ásia.

2 INTRODUÇÃO

Nessa Atividade Prática Supervisionada (APS), cuja disciplina vinculada é Introdução a programação estruturada (IPE), nos foi atribuído o tema "As Técnicas Criptográficas, conceitos, usos e aplicações". Diante disso, discutiremos a importância da Criptografia, uma ferramenta utilizada para o controle de troca de dados sigilosos em vários segmentos da tecnologia. Ademais, em conjunto, desenvolvemos um programa de Criptografia, cujo objetivo é possibilitar a representação de uma comunicação entre a guarda costeira brasileira com um navio que transporta lixo tóxico proveniente da Ásia.

Após discutirmos sobre qual tipo de criptografia usar em nosso programa, chegamos no consentimento de usar uma variação da Cifra de Cesar, que basicamente consiste em 2 variáveis guardadas na memória com todos os possíveis caracteres existentes em uma mensagem (letras, números e símbolos), cada um em ordem diferente (inversa um do outro), e então, o usuário (emissor ou receptor) tem a opção de criar uma chave para ser usada como base da codificação e decodificação, e assim, podermos somente ter a mensagem ocultada e depois revelada corretamente, se a chave digitada foi a mesma em ambos lados. Isso também gera a opção de criptografar uma mesma mensagem várias vezes, utilizando diversas chaves diferentes, assim contribuindo para uma melhor segurança na hora da troca de informações.

A partir disso, conseguimos analisar em nosso código todos os fatores lógicos e matemáticos que funcionam individualmente no programa, como também em conjunto. Com o objetivo de estruturar um programa simples, porém eficaz, o que o torna um excelente modo de entender como seguir o fluxo dos processos que ocorrem no mesmo, e assim, aumentando nossa capacidade intuitiva e cognitiva para resolução de problemas futuros, uma vez que eles podem ser bem maiores e complicados.

Mesmo sendo atribuída a disciplina de IPE, podemos ver a presente interdisciplinaridade do nosso sistema e incógnita proposta em geral com Lógica de programação e algoritmos (LPA). Tendo em conta que para o bom funcionamento e arquitetação do nosso código, precisamos entender sobre fluxos, ordem de

procedimentos, onde encaixarmos as funções a serem feitas, o caminho em que os dados irão percorrer, e a forma correta de agruparmos os módulos para que eles sejam de fácil entendimento e flexíveis para futuras alterações.

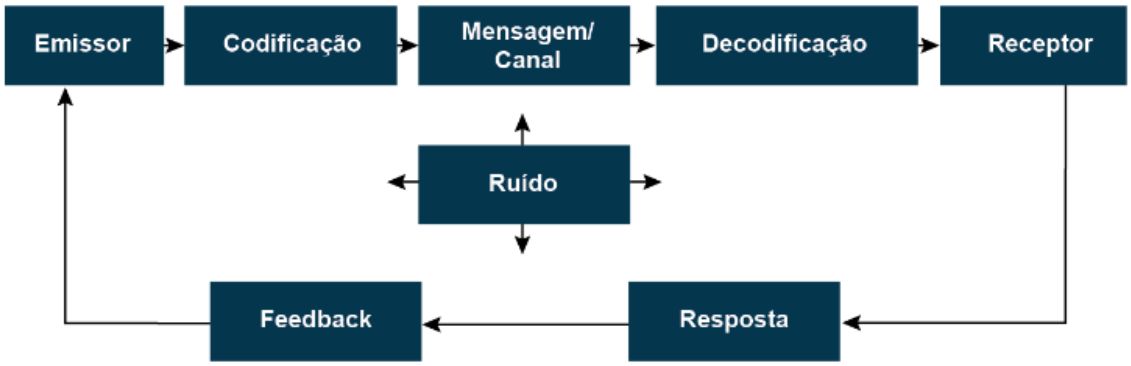
Além disso, abordaremos profundamente sobre nossa técnica criptográfica escolhida, também como suas principais características e método de aplicação para o bom funcionamento do código. Por fim, iremos expor nossas linhas de código, elucidaremos sua funcionalidade por módulos, e apresentaremos o programa funcionando com uma chave aleatória.

3 CRIPTOGRAFIA CONCEITOS GERAIS

O termo “Criptografia” que por definição é a transformação de um texto normal em um texto cifrado, é atualmente muito conhecido devido à segurança que o mesmo oferece em transações bancárias, trocas de mensagens e até mesmo de informações de usuários, além de estar presente em uma das áreas principais da Tecnologia da Informação, sendo ela a Segurança da Informação. Logo, o seu objetivo principal é fazer que uma mensagem se torne secreta, para que dificulte a sua decifração e aumente a proteção de dados e informações sigilosas.

Portanto, pode-se dizer que um dos principais objetivos, se não o principal da Criptografia, é a segurança que ela impõe na comunicação entre alguém que emite a informação e alguém que a recebe. Dentro disso, vale ressaltar a relação desta temática com a disciplina online AVA da UNIP, Comunicação e Expressão, quando é abordado o tema Funções da Linguagem. Com base no esquema abaixo, é possível notar esta relação, visto que há um emissor que emite uma mensagem codificada ao receptor, por meio de algum canal, e este tem que decodificá-la para poder compreender e interpretar a mensagem. Logo após decodificar, o receptor devolve uma resposta ao emissor e a comunicação se encerra. Caso contrário, se houver algum ruído na transmissão da informação, a resposta será diferente ou até mesmo negada.

Imagem 1: Representação linear do processo de comunicação



Fonte: AVA Unip Comunicação e Expressão

Porém, diferentemente do que se imagina, apesar da sua grande correlação com as grandes tecnologias e a sua importância no atual mercado global, a atividade

de criptografar não é exclusiva dos tempos presentes. A palavra Criptografia, vem do grego, na junção de duas palavras: *kryptós*, que significa escondido; e *gráphein*, que significa escrita. Alguns historiadores abordam o seu surgimento no Egito antigo, quando em algumas escritas, substituíam hieroglifos simples por outros mais raros de serem usados. Ademais, o ato de criptografar, ou melhor, cifrar, era fortemente utilizado nos textos religiosos, como até mesmo na Bíblia.

Por conta de ser uma ferramenta de transmissão de dados e mensagens de maneira escondida (cifrada) entre um emissor e um receptor, diversas nações utilizaram desse fator para que as técnicas criptográficas fossem aproveitadas em seu favor nas guerras, durante a história. Neste sentido, foi criado uma das cifras mais conhecidas: *A Cifra de César,* que apesar de não ser tão complexa, foi muito útil para que o imperador Júlio César conseguisse enviar mensagens e recebê-las entre os generais durante as guerras que ocorriam na época.

Imagem 2: Representação de anel Decodificador para cifra de César

Outrossim, durante a Segunda Guerra Mundial, a Alemanha com o objetivo de ganhar a guerra com o seu partido nazista, seu exército utilizou de uma ferramenta criada pelo engenheiro elétrico e alemão Arthur Scherbius, muito complexa e bem elaborada para a época, chamada máquina Enigma. Muitos acreditavam ser impossível de decifrá-la, visto que ela conseguia criar até 380 bits, o que gera um número praticamente incontável de combinações possíveis. Portanto, os nazistas tiveram muito controle de trocas de mensagens criptografadas e seguras.

Imagem 3: Máquina Enigma usada na Segunda Guerra Mundial

De outra forma, para que seja possível entender de fato como é o funcionamento de uma Criptografia, é indubitável a necessidade em conhecer as chaves criptográficas, estas sendo utilizadas na hora de criptografar uma mensagem e, na hora de decriptar. Além disso, as chaves são formadas por bits (0 ou 1), uma chave de 1 bit, por exemplo, equivale a 2 combinações de 0 e 1, logo a fórmula para realizar este cálculo é: 2^n, onde “n” equivale a quantos bits a chave tiver. Quanto maior o número de bits a chave possuir, mais seguro ela será, visto o grande número que deverá ser calculado. E é a chave que tem o controle do algoritmo que converterá a mensagem cifrada.

As chaves podem ser classificadas em duas formas: chaves simétricas (privadas) e chaves assimétricas. A primeira é usada em criptografias básicas, em que utilizando uma mesma chave, é possível criptografar e decriptar. Porém, por utilizar a mesma, é ocasionado uma vulnerabilidade, dado que o emissor tem que passar a sua chave para os receptores, podendo haver uma interceptação no meio, derrubando toda a segurança da Criptografia. Portanto, há a necessidade de trocar constantemente a sua chave e a sua principal vantagem é a simplicidade.

Diferentemente da apresentada anteriormente, agora, as chaves assimétricas são tratadas em criptografias mais avançadas, e utilizam um par de chaves, sendo elas chaves públicas e chaves privadas. De maneira separada, a pública é usada para a criptografar uma informação e a privada para decriptar. Com base nisso, o emissor da mensagem envia a chave pública para o destinatário e assegura a chave privada para si próprio. Logo, agora o sistema de Criptografia é bem mais seguro e bem menos

vulnerável, pois apenas quem tiver a chave privada irá conseguir acessar a mensagem, portanto se houver uma interceptação da chave entre o emissor e o receptor, esta chave será a pública, que não é capaz de decriptar os dados.

4 TÉCNICAS CRIPTOGRÁFICAS MAIS UTILIZADAS E CONHECIDAS

Assim como já foi apresentado anteriormente, a Criptografia é uma ciência em que codifica uma mensagem de maneiras que a dificulta de ser decodificada, o que garante uma maior segurança na transmissão de mensagens importantes e até mesmo “secretas”, entre indivíduos, instituições, governos, etc. Nesse cenário, pelo decorrer de toda a história da humanidade, foi necessário a utilização de técnicas criptográficas para poderem garantir vantagem nas transmissões entre emissores e receptores. Portanto, ao tratar-se dos dias atuais, algumas técnicas que foram criadas há muito tempo atrás, ainda são utilizadas, assim como, outras que foram criadas recentemente também são.

4.1 Sistema Binário

Indiscutivelmente, o sistema binário é uma das técnicas de criptografia mais conhecidas, principalmente na área de Computação, já que é a linguagem das máquinas. Ela é composta por 0 (zero) ou 1 (um). Com o objetivo de padronizar a representação dos caracteres alfanuméricos nos computadores, foi desenvolvido a ASCII (American Standard Code for Information Interchange), que traduzido para o português é, Código Padrão Americano para Intercâmbio de Informação. Este padrão americano, transforma um caractere alfanumérico em 7 (sete) dígitos binários e mais 1 (um) para verificar se há algum tipo de erro, portanto formando 8 (oito) dígitos binários. Para poder haver uma melhor compreensão, os exemplos a seguir mostrarão alguns caracteres e as suas respectivas maneiras de serem representadas em binário. A letra “A” corresponde à sequência 0100 0001, já a letra “a”, agora minúscula corresponde à 0110 0001.

4.2 Máquina Enigma

Esta, que já foi abordada no tópico anterior, não poderia faltar nas técnicas criptográficas mais usadas, visto a sua importância na história. A “Máquina Enigma”, foi uma ferramenta muito importante desenvolvida pelo alemão Arthur Scherbius, que foi utilizada pela Alemanha e os outros países que formavam o Eixo durante a Segunda Guerra Mundial, sendo eles a Itália e o Japão. O seu principal objetivo era gerar a comunicação exclusiva entre os países do Eixo, e lançar ataques surpresas contra os países Aliados. Para a época,

o Enigma era fácil de ser interceptado, porém, considerado impossível de decodificar as mensagens, dado que o seu número de combinações possíveis era praticamente impossível de ser contado. Estes tão grandes, que se tentasse adivinhar as mensagens por força bruta, levaria aproximadamente toda a eternidade do infinito para decodificá-las.

4.3 Cifra de César

A Cifra de César, é uma cifra monoalfabética no qual foi usada pelo Imperador Romano, Júlio César, para se comunicar com os seus generais durante as guerras da época, sem que seus inimigos captassem as suas mensagens transmitidas. Esta técnica consiste em utilizar um alfabeto romano e cifrá-lo através de uma troca de cada letra deste alfabeto pela terceira letra que a segue, por exemplo, a letra “A”, agora criptografada seria a letra “D”, e assim por diante.

Imagem 4: Técnica de cifragem pela Cifra de César

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S

Q	R	S	T	U	V	W	X	Y	Z
T	U	V	W	X	Y	Z	A	B	C

Fonte: StackoverFlow, 2017

Exemplo da aplicação da Cifra de César, utilizando a frase “Atividade Prática Supervisionada”:

(Atividade Prática Supervisionada) = DWLYLGDGH SUDWLFA VXSHUYLVLRQDGD

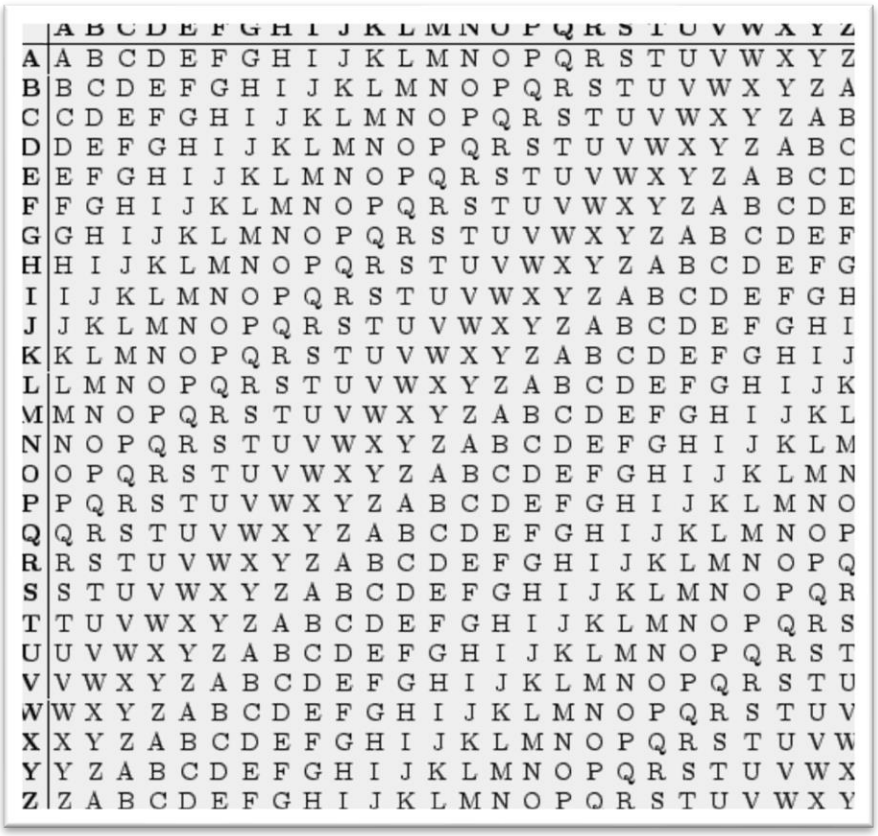
Pode parecer uma cifra simples, entretanto na época a grande maioria das pessoas eram analfabetas, o que dificultava ainda mais a decifragem das palavras.

4.4 Cifra de Vigenère

A cifra de Vigenère, foi inventada em 1553, pelo criptologista italiano Giovan Batista Bellaso. No entanto, o nome da cifra foi dado de maneira errada para o Blaise de Vigenère. Este método criptográfico é um sistema polialfabético, em que se usa uma tabela composta pelas 26 (vinte e seis) possibilidades da Cifra de César. Para criptografar uma mensagem

utilizando essa técnica, é necessário escolher uma palavra-chave e uma mensagem que deseja criptografar. Após isto, coloca-se a palavra-chave embaixo da mensagem e, através da tabela comparar a letra da palavra-chave que deve ser localizada na linha com a letra da mensagem que se localiza na coluna, ou o inverso.

Imagem 5: Tabela para aplicação da Cifra de Vigenère.



	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
A	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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: Eduardo Popovici, 2011

Exemplo de aplicação da Cifra de Vigenère, utilizando a palavra “trabalho” e a palavra-chave “logica”:

Mensagem	t r a b a l h o
Palavra-Chave	l o g i c a l o
Mensagem Criptografada	e f g j c l s c

Por fim, a Mensagem Criptografada, o resultado da mensagem “trabalho” com a palavra-chave “logica” é “efgjclsc”.

4.5 Código Morse

Este que é um dos mais famosos métodos criptográficos, o Código Morse é um sistema que utiliza sinais elétricos com o objetivo de reproduzir algarismos, pontuação e letras. O seu funcionamento é simples, devido à utilização de dois modos, o ligado e o desligado, o pulso e o não pulso, ou seja, ele também é binário. Essa técnica, pode ser usada através de pulsos elétricos veiculados por ondas mecânicas, ondas eletromagnéticas, luzes, vibrações, cabos, etc. Para compreender como se faz a cifragem das mensagens, é necessário conhecer a tabela das cifragens para código Morse. Para isso segue a tabela abaixo:

Imagem 6: Tabela de conversão para Código Morse

A •-	N -•	0 -----
B -•••	O ---	1 •----
C -•-•	P •--•	2 ••---
D -••	Q --•-	3 •••--
E •	R •-•	4 ••••-
F ••-•	S •••	5 •••••
G --•	T -	6 -••••
H ••••	U ••-	7 --•••
I ••	V •••-	8 ---••
J •---	W •--	9 ----•
K -•-	X -••-	• •-•-•-
L •-••	Y -•--	, --••--
M --	Z --••	? ••--••

Como é possível visualizar, as criptografias geradas das letras são representadas por pontos e por traços. Tendo eles em conta, o receptor consegue captar a mensagem criptografada através de quatro estados, sendo eles: pulso longo (traço), pulso curto (ponto), pulso desligado longo (espaço entre as palavras e os caracteres) e pulso desligado curto (espaço entre traços e pontos).

4.6 Advanced Encryption Standard (AES)

É uma técnica criptográfica adotada pelo governo dos Estados Unidos, muito conhecida, devido a sua alta velocidade, fácil execução e baixo uso de memória, que utiliza o modelo de chave simétrica, no qual permite que com a mesma chave possa criptografar e

descriptografar uma mensagem, e esta apresenta um tamanho de 128 bits, variando para 192 ou para 256 bits, caso necessite de uma proteção maior. O único ataque que pode captar a mensagem desse modelo é o de força bruta, entretanto por conta dos altos números de bits e combinações possíveis, fica extremamente difícil conseguir isto, nos tempos atuais.

4.7 Rivest-Shamir-Adleman (RSA)

O RSA, inventado em 1978, é o sistema criptográfico mais conhecido e usado atualmente. Este tem esse nome devido às iniciais dos seus inventores, R. L. Rivest, A. Shamir e L. Adleman. Dentro disso, vale dizer que o RSA utiliza o modelo de chave assimétrica, que consiste em um par de chaves, uma pública e outra privada. A primeira com o objetivo de criptografar, e a segunda que deve ser mantida em segredo, com o objetivo de descriptografar.

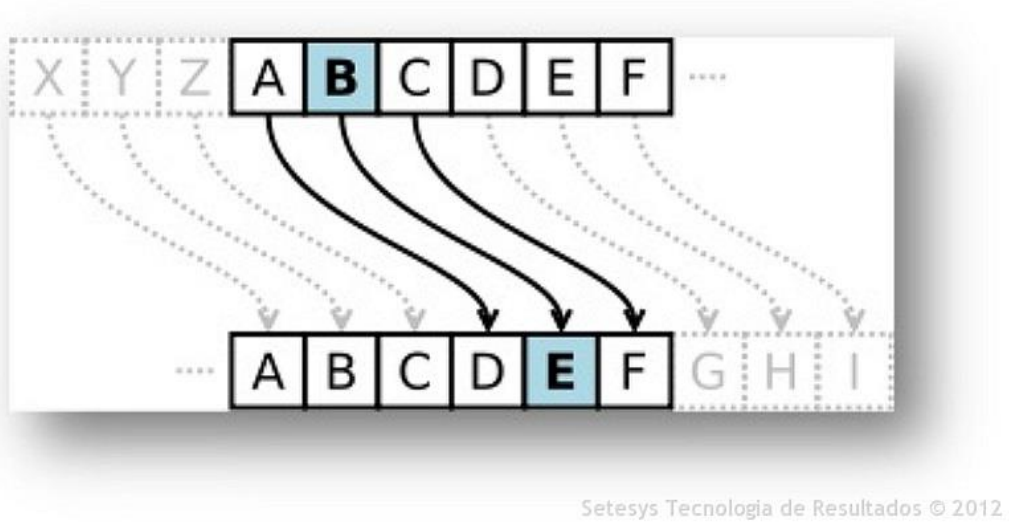
Esse, que é considerado um dos mais seguros do mercado, se não o mais seguro, foi quem permitiu a criptografia nas assinaturas digitais. Portanto, é aplicado nas compras online, troca de e-mails, transferências digitais e dentre outros.

5 DISSERTAÇÃO (CIFRA DE CÉSAR)

5.1 Estruturação, conceitos e fundamentação

A Cifra de César é uma das mais antigas, simples e de longe a mais conhecida entre várias que existem. Seu nome é dado por ser criada pelo Imperador Romano Júlio César, que Segundo Tanenbaum e Wetherall (2011, p.483), Júlio César usava para proteger suas correspondências particulares e de cunho militar um código de substituição no qual cada três posições a frente da letra do texto original era substituída no alfabeto: a letra "a" era substituída por “d”, a “b” por “e”, e assim sucessivamente.

Imagem 7: Funcionamento básico da Cifra de César



Fonte: http://setesys.com.br/blog/como-produzir-senhas-criativas-utilizando-a-cifra-de-cesar/ (2012)

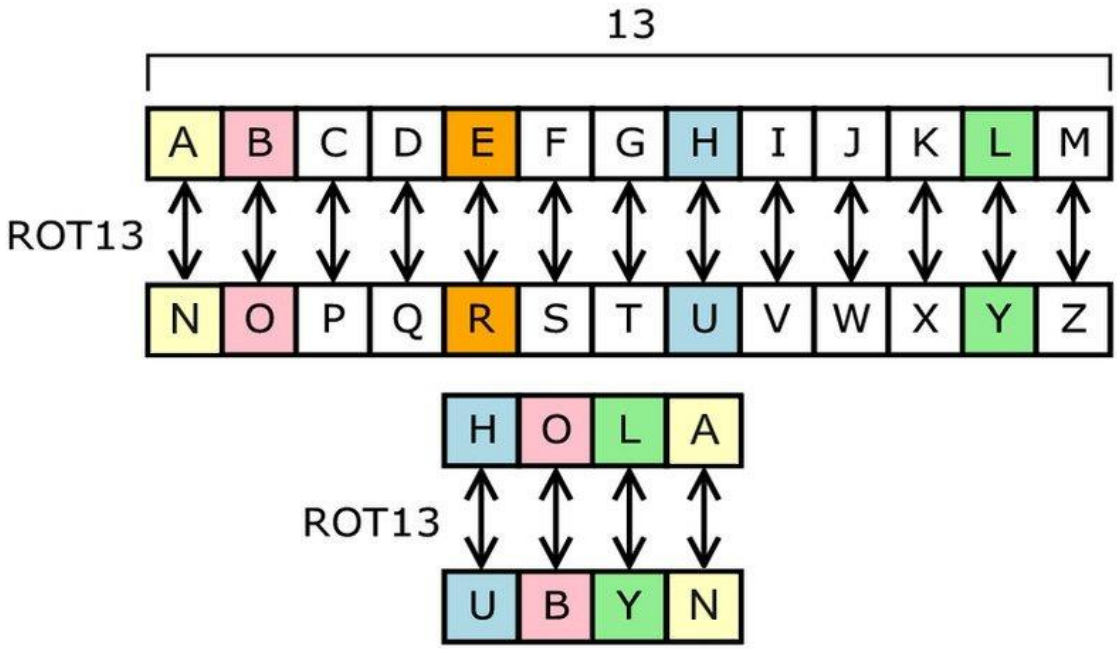
Júlio César criou algo que mais para frente seria chamado de cifra de substituição, onde desde o início já se tem um sistema de substituição pré-definido, normalmente uma letra X do alfabeto é substituída por outra letra Y, podendo ser deslocada quantas casas necessárias para a direita ou esquerda.

Tanenbaum e Wetherall (2011) explica que esse método de criptografia usa um sistema simples de substituição monoalfabética, onde cada letra é substituída por outra letra de acordo com a chave utilizada, podendo ser ela qualquer uma das 26

letras do alfabeto, criando assim um texto criptografado, mas mantendo a ordem dos símbolos do texto original.

Sabendo disso, podemos fazer uma pequena simulação:

Imagem 8: Substituição Monoalfabética



Fonte: https://www.neoteo.com/el-cifrado-del-cesar (2010).

Na imagem podemos ver um exemplo da cifragem ROT13, nela substituímos cada letra por 13 letras adiante de onde se encontra no alfabeto, sendo assim o A se torna N, e vice e versa, e para decodificar é necessário que seja feito o processo ao contrário, voltando as 13 casas.

Assim caso uma mensagem enviada fosse interceptada seria de maior dificuldade de ser lida, por exemplo, caso a mensagem fosse: “Estamos escondidos no campo” ficaria “Rfgnzbf rfpbaqvqbf ab pnzcb”.
Pontos positivos:

- Facilidade para introdução aos conceitos da criptografia;
- Fácil implementação em sistemas;
- Chaves simples.

5.2 Benefícios em relação às técnicas anteriores

Sendo uma das primeiras a serem criadas, por volta de 100 a.C., a Cifra de Cesar é uma das mais antigas cifragens existentes, porém de longe não é a mais antiga.

A Cifragem de Khnumhotep II, é uma técnica datada por volta de 1900 a.C., no Egito antigo, estes vestígios foram encontrados na tumba de um grande chefe egípcio, onde seus escribas decidiram substituir símbolos mais comuns em sua cultura por outros mais complexos e desconhecidos. Baseado no conhecido acredita-se que seus motivos foram para esconder e preservar seus segredos de sua cultura e religião.

Imagem 9: O Mural de Beni-Hasan

Fonte: https://www.easyvoyage.de/aegypten/graeber-von-beni-hassan-6143 (2012)

Porém, o grande problema desta ideia é que apesar de serem símbolos menos conhecidos não são criados somente para este propósito ou com algum tipo de quebra de padrão, César criou uma cifra onde cada uma de suas mensagens poderiam conter uma quantidade diferente de casas a serem puladas, assim dificultando muito mais a visibilidade, pois caso percebesse que suas cartas estavam sendo lidas muito rápido bastava somente alterar o número de casas para desacelerar a força inimiga.

5.3 Aplicações que fazem/fizeram uso da técnica

5.3.1 Cifra de Vigenère.

Jonathan Strickland (2007) Argumenta que a Cifra de Vigenère utilizava um sistema polialfabético particularmente difícil de decifrar, o método era de uma combinação do Quadro de Trimethius, neste quadro a primeira linha seria o alfabeto conhecido e a cada linha a primeira casa é substituída pela que estava adiante, logo se na primeira linha e coluna está a letra “a” na próxima será a “b”, e uma chave. Onde a chave determina qual dos alfabetos da tabela será utilizado.

Imagem 10: Tabela da Cifra de Vigenère

	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
A	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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: https://www.egress.com/blog/email-encryption/encryption-101-the-vigenere-cipher (2015)

Segundo Jonathan Strickland (2007) O criptógrafo liga a coluna da primeira letra do texto original com a linha da primeira letra da chave, a coluna da segunda letra do texto original com a linha da segunda letra da chave e assim sucessivamente, sempre que a chave acabar deve se voltar a sua primeira letra até criptografar todo o texto. A intersecção entre linha e coluna será a letra correspondente ao texto criptografado.

5.4 Discussão comparativa entre esta técnica e outras conhecidas/utilizadas

A Cifra de César apesar de ser bem versátil e utilizada para fins educacionais, comparada as criptografias presentes nos dias de hoje é muito simples e quebrável, por exemplo, a DES (Data Encryption Standart) que foi um dos primeiros modelos a serem utilizados por computadores para criptografia com uma proteção simples de pôr volta 56 bits, tendo 16 ciclos de codificação, apesar de os números não serem astronômicos, a quantidade de combinações resultantes é 72 quatrilhões de combinações geradas.

E sua evolução deixa a César ainda mais obsoleto, caso não fosse o bastante a 3DES (Triple DES) com a Ideia de substituir o DES original, a técnica trabalha com 3 chaves de 56 bits ao invés de uma, gerando uma proteção de 168 bits, sendo considerada até hoje, por especialistas, suficiente para proteger os dados considerando que o padrão seria 112 bits.

5.5 Vulnerabilidades e falhas

Quebrar a cifra de César é muito simples, sendo assim em alguns minutos de tentativa e erro qualquer pessoa é capaz de descobrir o texto cifrado. Um grande problema são as pistas deixadas pela cifragem, uma brecha é o espaçamento entre as palavras que facilita a sua compreensão, deste modo o atacante pode começar pelas palavras menores, juntando em outro problema, à final na cifra original todas as letras iguais serão substituídas igualmente.

Por exemplo, na palavra “banana” o “a” seria substituído pela mesma letra todas às vezes, sendo assim depois de cifrada se tornando “ldfdfa”.

Vulnerabilidades:

- Espaçamento entre as palavras;
- Padrão de substituição das letras;
- Técnica muito conhecidas;
- Possui um pequeno número de chaves.

5.6 Melhorias propostas e/ou implementadas

Apesar da Cifra de César ser facilmente quebrada, há um meio de melhorá-la. Com uma modificação no cálculo de substituição é possível complicar, e muito essa quebra. A Cifra de César estendida segue o mesmo fundamento da original, porém a mudança propõe dois parâmetros (A e B), e utilizando a equação Y = A * X + B, onde Y é a posição inicial do alfabeto onde a letra que será cifrada está e X a posição onde ela será substituída, sendo assim somente este deslocamento que A e B irão proporcionar aumentarão a dificuldade do atacante de descobrir qual era a frase original escrita no texto.

6 PROJETO (ESTRUTURA) DO PROGRAMA

Para iniciar será necessária a instalação de duas bibliotecas fundamentais para o desenvolvimento do programa como um todo.

```
# PySimpleGUI bibliotec || pip install pysimplegui
# Pyperclip bibliotec || pip install pyperclip
```

Fonte: Própria (2022)

Elas são a PySimpleGUI e a Pyperclip

```
import PySimpleGUI as sg
import pyperclip
```

Fonte: Própria (2022)

Neste bloco importamos ambas as bibliotecas e utilizamos o “as” para renomear a biblioteca PySimpleGUI buscando maior velocidade durante o código.

```
alphabet = 'abcdefghijklmnopqrstuvwxyz,.?çéãõáâàêèóòô!@#$%¨&*()-_+=º 0123456789'
alphabet2 = '9876543210 ,.?çéãõáâàêèóòô!@#$%¨&*()-_+=ºzyxwvutsrqponmlkjihgfedcba'
```

Fonte: Própria (2022)

O alfabeto 1 e 2 são as chaves privadas do programa, a utilizamos junto com a chave pública do usuário para criar uma mensagem criptografada.

```
# Criptografar
def encrypt(message, key):
    key = int(key)
    if key:
        message = message.lower()
        caracter = ''
        for slot in message:
            c_index = alphabet.index(slot)
            caracter += alphabet2[(c_index + key) % len(alphabet)]
        if(event != 'Copiar'):
            msCripto()
        return caracter
    else:
        if(event != 'Copiar'):
            error()
        return ''
```

Fonte: Própria (2022)

Na função “encrypt”, solicitamos uma chave ao usuário composta somente por números inteiros, logo após a mensagem que deseja ser criptografado sendo assim o texto é manipulado de maneira que a letra-a-letra da frase seja alterada pela chave pública, utilizando a chave privada.

Por exemplo, supondo a chave privada seja (r, s, f, e, d, c, b, a) e a publica escolhida pelo usuário seja três, sendo a mensagem “Cesar”, pegamos a primeira letra e a identificamos em nossa chave privada, a letra “C” está na sexta casa da chave sendo assim adicionamos a chave pública do usuário, três, logo o que antes era um “C” se tornará o “R”, continuando por toda a palavra chegaremos a o resultado de “Rbdfe”.

```
# Desencriptografar
def decrypt(message, key):
    key = int(key)
    if key:
        message = message.lower()
        caracter = ''
        for slot in message:
            c_index = alphabet2.index(slot)
            caracter += alphabet[(c_index - key) % len(alphabet)]
        if(event != 'Copiar'):
            msDescripto()
        return caracter
    else:
        if(event != 'Copiar'):
            error()
        return ''
```

Fonte: Própria (2022)

Na função “decrypt”, faremos o processo contrário antes visto na outra função, solicitaremos o texto criptografado pela chave privada junto com a chave pública antes fornecida pelo usuário, assim voltando o texto ao seu original, caso a chave privada esteja incorreta será retornada uma mensagem totalmente sem sentido, com a ideia de confundir o invasor e induzindo-o ao erro.

Revertendo o processo feito no exemplo anterior, o usuário colocaria a palavra já cifrada “Rbdfe” e voltaríamos três casas por letra seguindo a chave privada (r, s, f, e, d, c, b, a) tendo a letra “f” como exemplo, a resposta seria a letra “a”, seguindo o processo de descriptografar, chegaríamos novamente ao nome “Cesar”

Todas as imagens a seguir são relacionadas a interface do programa, por este motivo não serão detalhadas uma a uma.

```
# Interface
def interface():
    sg.theme('DarkBlue')
    layout = [
        [sg.Text(
            'Bem vindo ao sistema de criptografia da guarda costeira brasileira!',
            font=("Helvetica", 21),
            pad=(0, 50)
        )],

        [sg.Image(source="image/logo.png",)],

        [sg.Text(
            'Todo e qualquer contato com a tripulação do seu navio e o lixo toxico será controlado a partir de agora. \nPara se comunicar conosco criptografe e envie a sua men
            font=("Arial", 13),
            pad=(0, 50)

        ), ],
```

Fonte: Própria (2022)

```
    [sg.Button(
        button_text="Cifrar",
        size=(30, 2),
        button_color=('white', '#009444')
    ),

        sg.Button(
            button_text='Decifrar',
            size=(30, 2),
            button_color=('white', '#009444')
    )
    ],
]
return sg.Window('Bem vindo!', size=(1000, 600), element_justification='c', layout=layout, finalize=True)
```

Fonte: Própria (2022)

```
def criptografar():
    sg.theme('DarkBlue')
    col1 = [
        [sg.Text(
            'Digite a menssagem:',
            font=("Arial", 11),),

         sg.Multiline(
            key='message',
            size=(150, 0),
            no_scrollbar=True,
            font = ("Arial", 12)
        )],

        [sg.Text(
            'Digite a chave em números:',
            font=("Arial", 11),
        ),

            sg.Multiline(
                key='key',
                size=(150, 0),
                no_scrollbar=True,
                font = ("Arial", 12)
        )],
```

Fonte: Própria (2022)

```
    [sg.Button(
        'Voltar',
        size=(15, 1),
        pad=(0, 15),
        button_color=('white', '#009444')
    ),
        sg.Button(
            'Enviar',
            size=(15, 1),
            button_color=('white', '#009444')
    )],
]

layout = [
    [sg.Image(
        source="image/logo100x.png",
        pad=(0, 40)
        ),

        sg.Column(col1)
    ],

    [sg.Output(
        size=(150, 8),
        font = ("Arial", 15)
    )],

    [sg.Button(
        'Copiar',
        size=(15, 1),
        pad=(0, 15),
        button_color=('white', '#009444')
    )],
]
return sg.Window('Criptografar', size=(800, 450), layout=layout, finalize=True)
```

Fonte: Própria (2022)

```
def desencriptografar():
    sg.theme('DarkBlue')
    col1 = [
        [sg.Text('Digite a menssagem:'),

         sg.Multiline(
            key='message',
            size=(150, 0),
            no_scrollbar=True,
            font = ("Arial", 12)
        )],

        [sg.Text(
            'Digite a chave em números:'
            ),

            sg.Multiline(
            key='key',
            size=(150, 0),
            no_scrollbar=True,
            font = ("Arial", 12)
        )],

        [sg.Button(
            'Voltar',
            size=(15, 1),
            pad=(0, 15,),
            button_color=('white', '#009444')
            ),

            sg.Button(
            'Enviar',
            size=(15, 1),
            button_color=('white', '#009444')
        )],
    ]
```

Fonte: Própria (2022)

```
layout = [
    [sg.Image(
        source="image/logo100x.png",
        ),
    sg.Column(col1)],

    [sg.Output(
        size=(150, 8),
        font = ("Arial", 15)
    )],

    [sg.Button(
        'Copiar',
        size=(15, 1),
        pad=(0, 15),
        button_color=('white', '#009444')
    )],
]
return sg.Window('Desencriptografar', size=(800, 450), layout=layout, finalize=True)
```

Fonte: Própria (2022)

```
def error():
    sg.theme('DarkBlue')
    return sg.Window('Erro',[
            [sg.Text('Digite uma chave válida!',  font=("Arial", 11), pad=(0, 15),)],
            [sg.OK(size=(30), pad=(0, 25), button_color=('white', '#009444')), ]
        ], element_justification='c', size=(300, 150)).read(close=True)


def msCripto():
    sg.theme('DarkBlue')
    return sg.Window('Enviada',[
            [sg.Text('Mensagem Criptografada Enviada com sucesso!',  font=("Arial", 11), pad=(0, 15),)],
            [sg.OK(size=(30), pad=(0, 25), button_color=('white', '#009444')), ]
        ], element_justification='c', size=(350, 150)).read(close=True)

def msDescripto():
    sg.theme('DarkBlue')
    return sg.Window('Enviada',[
            [sg.Text('Mensagem Descriptografada com sucesso!',  font=("Arial", 11), pad=(0, 15),)],
            [sg.OK(size=(30), pad=(0, 25), button_color=('white', '#009444')), ]
        ], element_justification='c', size=(350, 150)).read(close=True)
```

Fonte: Própria (2022)

```
# Janelas iniciais
janela1, janela2, janela3, janela4, janela5, janela6 = interface(), None, None, None, None, None

# Loop  de leitura de eventos
while True:
    window, event, values = sg.read_all_windows()
    # Quando janela for fechada
    if window == janela1 and event == sg.WIN_CLOSED:
        break

    if window == janela2 and event == sg.WIN_CLOSED:
        break

    if window == janela3 and event == sg.WIN_CLOSED:
        break
```

Fonte: Própria (2022)

```
# Ir para próxima janela
if window == janela1 and event == 'Cifrar':
    janela1.hide()
    janela2 = criptografar()

if window == janela2 and event == 'Copiar':
    copy = encrypt(values['message'], values['key'])
    pyperclip.copy(copy)

if window == janela3 and event == 'Copiar':
    copy = decrypt(values['message'], values['key'])
    pyperclip.copy(copy)

if window == janela1 and event == 'Decifrar':
    janela1.hide()
    janela3 = desencriptografar()

if window == janela2 and event == 'Voltar':
    janela2.hide()
    janela1.un_hide()

if window == janela3 and event == 'Voltar':
    janela3.hide()
    janela1.un_hide()

if window == janela2 and event == 'Enviar':
    # Mandando para função
    print(encrypt(values['message'], values['key']))

if window == janela3 and event == 'Enviar':
    # Mandando para função
    print(decrypt(values['message'], values['key']))
```

Fonte: Própria (2022)

7 RELATÓRIO COM AS LINHAS DE CÓDIGO DO PROGRAMA

```
# PySimpleGUI bibliotec || pip install pysimplegui
# Pyperclip bibliotec || pip install pyperclip

import PySimpleGUI as sg
import pyperclip

alphabet = 'abcdefghijklmnopqrstuvwxyz,.?çéãõáâàêèóòô!@#$%¨&*()-_+=°
0123456789'
alphabet2 = '9876543210 ,.?çéãõáâàêèóòô!@#$%¨&*()-
_+=°zyxwvutsrqponmlkjihgfedcba'

# Criptografar
def encrypt(message, key):
    key = int(key)
    if key:
        message = message.lower()
        caracter = ''
        for slot in message:
            c_index = alphabet.index(slot)
            caracter += alphabet2[(c_index + key) % len(alphabet)]
        if(event != 'Copiar'):
            msCripto()
        return caracter
    else:
        if(event != 'Copiar'):
            error()
        return ''


# Desencriptografar
def decrypt(message, key):
    key = int(key)
    if key:
        message = message.lower()
        caracter = ''
        for slot in message:
            c_index = alphabet2.index(slot)
            caracter += alphabet[(c_index - key) % len(alphabet)]
        if(event != 'Copiar'):
            msDescripto()
        return caracter
    else:
        if(event != 'Copiar'):
            error()
        return ''
```

```
# Interface
def interface():
    sg.theme('DarkBlue')
    layout = [
        [sg.Text(
            'Bem vindo ao sistema de criptografia da guarda costeira
brasileira!',
            font=("Helvetica", 21),
            pad=(0, 50)
        )],

        [sg.Image(source="image/logo.png",)],

        [sg.Text(
            'Todo e qualquer contato com a tripulação do seu navio e o lixo
toxico será controlado a partir de agora. \nPara se comunicar conosco
criptografe e envie a sua mensagem.',
            font=("Arial", 13),
            pad=(0, 50)

        ), ],

        [sg.Button(
            button_text="Cifrar",
            size=(30, 2),
            button_color=('white', '#009444')
        ),

            sg.Button(
                button_text='Decifrar',
                size=(30, 2),
                button_color=('white', '#009444')
        )
        ],
    ]
    return sg.Window('Bem vindo!', size=(1000, 600),
element_justification='c', layout=layout, finalize=True)


def criptografar():
    sg.theme('DarkBlue')
    col1 = [
        [sg.Text(
            'Digite a menssagem:',
            font=("Arial", 11),),

         sg.Multiline(
            key='message',
            size=(150, 0),
```

```
            no_scrollbar=True,
            font = ("Arial", 12)
        )],

        [sg.Text(
            'Digite a chave em números:',
            font=("Arial", 11),
        ),

            sg.Multiline(
                key='key',
                size=(150, 0),
                no_scrollbar=True,
                font = ("Arial", 12)
        )],

        [sg.Button(
            'Voltar',
            size=(15, 1),
            pad=(0, 15),
            button_color=('white', '#009444')
        ),
            sg.Button(
                'Enviar',
                size=(15, 1),
                button_color=('white', '#009444')
        )],
    ]

    layout = [
        [sg.Image(
            source="image/logo100x.png",
            pad=(0, 40)
            ),

            sg.Column(col1)
        ],

        [sg.Output(
            size=(150, 8),
            font = ("Arial", 15)
        )],

        [sg.Button(
            'Copiar',
            size=(15, 1),
            pad=(0, 15),
            button_color=('white', '#009444')
        )],
```

```
    ]
    return sg.Window('Criptografar', size=(800, 450), layout=layout,
finalize=True)


def desencriptografar():
    sg.theme('DarkBlue')
    col1 = [
        [sg.Text('Digite a menssagem:'),

         sg.Multiline(
            key='message',
            size=(150, 0),
            no_scrollbar=True,
            font = ("Arial", 12)
        )],

        [sg.Text(
            'Digite a chave em números:'
            ),

            sg.Multiline(
            key='key',
            size=(150, 0),
            no_scrollbar=True,
            font = ("Arial", 12)
        )],

        [sg.Button(
            'Voltar',
            size=(15, 1),
            pad=(0, 15,),
            button_color=('white', '#009444')
            ),

            sg.Button(
            'Enviar',
            size=(15, 1),
            button_color=('white', '#009444')
        )],
    ]
    layout = [
        [sg.Image(
            source="image/logo100x.png",
            ),
        sg.Column(col1)],

        [sg.Output(
            size=(150, 8),
```

```
            font = ("Arial", 15)
        )],

        [sg.Button(
            'Copiar',
            size=(15, 1),
            pad=(0, 15),
            button_color=('white', '#009444')
        )],
    ]
    return sg.Window('Desencriptografar', size=(800, 450), layout=layout,
finalize=True)


def error():
    sg.theme('DarkBlue')
    return sg.Window('Erro',[
            [sg.Text('Digite uma chave válida!',  font=("Arial", 11), pad=(0,
15),)],
            [sg.OK(size=(30), pad=(0, 25), button_color=('white', '#009444')),
]
        ], element_justification='c', size=(300, 150)).read(close=True)


def msCripto():
    sg.theme('DarkBlue')
    return sg.Window('Enviada',[
            [sg.Text('Mensagem Criptografada Enviada com
sucesso!',  font=("Arial", 11), pad=(0, 15),)],
            [sg.OK(size=(30), pad=(0, 25), button_color=('white', '#009444')),
]
        ], element_justification='c', size=(350, 150)).read(close=True)

def msDescripto():
    sg.theme('DarkBlue')
    return sg.Window('Enviada',[
            [sg.Text('Mensagem Descriptografada com sucesso!',  font=("Arial",
11), pad=(0, 15),)],
            [sg.OK(size=(30), pad=(0, 25), button_color=('white', '#009444')),
]
        ], element_justification='c', size=(350, 150)).read(close=True)


# Janelas iniciais
janela1, janela2, janela3, janela4, janela5, janela6 = interface(), None,
None, None, None, None

# Loop  de leitura de eventos
while True:
```

```
    window, event, values = sg.read_all_windows()
    # Quando janela for fechada
    if window == janela1 and event == sg.WIN_CLOSED:
        break

    if window == janela2 and event == sg.WIN_CLOSED:
        break

    if window == janela3 and event == sg.WIN_CLOSED:
        break


    # Ir para próxima janela
    if window == janela1 and event == 'Cifrar':
        janela1.hide()
        janela2 = criptografar()

    if window == janela2 and event == 'Copiar':
        copy = encrypt(values['message'], values['key'])
        pyperclip.copy(copy)

    if window == janela3 and event == 'Copiar':
        copy = decrypt(values['message'], values['key'])
        pyperclip.copy(copy)

    if window == janela1 and event == 'Decifrar':
        janela1.hide()
        janela3 = desencriptografar()

    if window == janela2 and event == 'Voltar':
        janela2.hide()
        janela1.un_hide()

    if window == janela3 and event == 'Voltar':
        janela3.hide()
        janela1.un_hide()

    if window == janela2 and event == 'Enviar':
        # Mandando para função
        print(encrypt(values['message'], values['key']))

    if window == janela3 and event == 'Enviar':
        # Mandando para função
        print(decrypt(values['message'], values['key']))
```

8 APRESENTAÇÃO DO PROGRAMA EM FUNCIONAMENTO EM UM COMPUTADOR, APRESENTANDO TODAS AS FUNCIONALIDADES PEDIDAS E EXTRAS.

Com o intuito de facilitar a visualização dos arquivos necessários em geral, fizemos upload dos mesmos em uma pasta no Google Drive. Lá contém dois vídeos mostrando o programa funcionando inserindo tanto a chave correta para a criptografia, como a chave incorreta, e assim, mostrando a mensagem certa e a errada.

O programa em Python, um readme e os arquivos de imagem necessários para o bom funcionamento do programa estão lá anexados também.

Segue o link:

https://drive.google.com/drive/folders/1F4q0Pfj-2lHeKQRW5NypXy4CmAR5O3yK

9 CONCLUSÃO

Após analisarmos nossas pesquisas, percebemos que a criptografia foi uma grande e importante ferramenta para o desenvolvimento tecnológico da humanidade. Se não fosse nossa vontade e necessidade de precisar cada vez mais espaço, mais praticidade e mais rapidez na manipulação de dados, nunca teríamos chegado ao ponto de hoje, a partir do momento que começamos a compartilhar informações importantes, também precisamos ocultá-las para podermos preservá-las, e assim criando a segurança de dados.

Com a evolução da tecnologia, tivemos grande sucesso em proteger informações preciosas, seja elas coisas simples como privacidade de usuários em redes sociais, até mesmo instruções essenciais entre exércitos durante uma guerra. Porém, em simultâneo, por termos criptografias tão complicadas e tão desordenadas, que vimos o desafio de decifrá-las, esses desafios nos trouxeram cada vez mais métodos, lógicas e algoritmos possíveis para usarmos e ampliá-los na época, testando nossa capacidade cognitiva, a ponto de os aliados conseguirem ganhar a Segunda Guerra Mundial com a mente brilhante de Alan Turing.

“A engenhosidade humana não pode arquitetar uma escrita secreta que a própria engenhosidade humana não possa resolver” (Edgar Allan Poe)

Ademais, com esse trabalho, conseguimos ter uma visão ampla de como funciona a troca de dados, de seu fundamento, como também todos os ramos que podemos seguir para essa troca poder ficar mais rápida, mais limpa, e mais flexível. Com a divisão correta de módulos em nosso código, podemos ver com clareza todos os elementos, processos e funções nele incluído, deixando assim, as futuras modificações necessárias muito mais fáceis de serem feitas.

No começo cada membro do grupo teve uma proposta de como seria o funcionamento da criptografia, mas após nos juntarmos e discutirmos, tivemos a ideia de usar a Cifra de Cesar como base, e com ideias de cada membro, começamos a arquitetar a lógica base (fluxo). Com a Ideia básica de chave pública e chave privada em mente, produzimos um sistema simples, porém eficaz para a ocultação de uma

mensagem, este algoritmo nos permite usar qualquer número para servir como chave, esta chave será utilizada em uma função com os índices de nossas variáveis (variável consiste no alfabeto, números e outros caracteres possíveis de nossa mensagem), assim criando um novo índice e substituindo-o na mensagem principal.

Essa técnica, além de prática para mensagens rápidas, também nos permite criar codificações mais complicadas que envolvem mais de uma chave numérica, desta maneira, deixando a comunicação muito mais segura.

Percebemos também que o mais importante de toda a construção do programa é o fluxo definido, pois a partir dele, podemos fazer este programa em qualquer linguagem, e se a divisão dos módulos estiver nítida, podemos fazer alterações sem prejudicar o programa como um todo.

Portanto, concluímos que essa atividade foi muito importante para nossa formação, pois trabalhou em vários aspectos de raciocínio lógico, como o planejamento do fluxo de processos de nosso programa, estruturar a relação entre funções e variáveis possíveis no mesmo, e também, raciocinar sobre os modos de execução possíveis a partir do resultado desses processos e funções. Assim desenvolvendo nossa capacidade cognitiva e nos deixando mais abertos a novas ideias e próximos projetos.

10 BIBLIOGRAFIA

DA SILVA, Alexandre Ferreira; MARTINS, Renato Marinho. Criptografia: aspectos históricos e matemáticos. <https://ccse.uepa.br/downloads/tcc/2011/silva_martins_2011.pdf>

PORTO, Victor Monteiro Ferreira et al. Criptografia: Da origem aos dias atuais. 2015. <https://www.bdtd.uerj.br:8443/bitstream/1/4855/1/Dissertacao_%20VictorPDF.pdf>

ANDRADE, Ewerton R.; LUNARDI, Roben C.; RAMOS, Nicolas U. Conceitos basicos de Criptografia. <https://www.researchgate.net/profile/Roben-Lunardi/publication/359514048_Conceitos_basicos_de_Criptografia/links/624214e17931cc7ccf009b99/Conceitos-basicos-de-Criptografia.pdf>

CARMO, Fernando J.; LEMES, Pedro A.; FREITAS, Tiago H. Criptografia e PGP. <https://we.riseup.net/assets/317791/Criptografia+e+PGP.pdf>

MOURAO, Fábio Pires. Criptografia. <https://ime.ufg.br/bienal/2006/poster/fabio.pdf>

OLIVEIRA, Ronielton Rezende. Criptografia simétrica e assimétrica-os principais algoritmos de cifragem. **Segurança Digital [Revista online]**, v. 31, p. 11-15, 2012. <https://www.ronielton.eti.br/publicacoes/artigorevistasegurancadigital2012.pdf>

BONFIM, Daniele Helena. **Criptografia RSA**. 2017. Tese de Doutorado. Universidade de São Paulo. <https://www.teses.usp.br/teses/disponiveis/55/55136/tde-06042017-164507/publico/DanieleHelenaBonfim_revisada.pdf>

PALAZZESI, Ariel. El cifrado del César. 31 de agosto de 2018. <https://www.neoteo.com/el-cifrado-del-cesar>

ÅHLÉN AB, Johan. Caesar Cipher Decoder, Solver and Encoder. < https://www.boxentriq.com/code-breaking/caesar-cipher>

GEEKSFORGEEKS. Caesar Cipher in Cryptography. 25 de setembro de 2022 < https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/>

DOS SANTOS GREGÓRIO, Moisés Bruno et al. CriptoMat1: ensinando Matemática utilizando conceitos de Criptografia-cifra de César e César estendida. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2014. p. 84. < http://ojs.sector3.com.br/index.php/wcbie/article/view/3175>

RAMO ESTUDANTIL IEEE-UEL. Criptografia: Origem e História. 16 de abr. de 2020. <https://www.ieeeuel.org/post/criptografia-origem-e-hist%C3%B3ria>

TENENBAUM, A. S; WETHERALL, D. Redes de Computadores. 5. ed. São Paulo: Pearson Education, 2011. 582p.

MEDEIROS, Fávio. Uma breve história sobre Criptografia. 6 de julho de 2015. <https://cryptoid.com.br/criptografia-identificacao-digital-id-biometria/a-historia-da-criptografia/>

STRICKLAND, Jonathan "How Code Breakers Work" 25 de outubro de 2007. HowStuffWorks.com. <https://science.howstuffworks.com/code-breaker.htm> Acesso em: 16 de novembro de 2022.

11 FICHAS DE ATIVIDADE PRÁTICA SUPERVISIONADA (APS)

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: Bruno da Silva Pimentel

RA: N844132 **CURSO:** Ciências da Computação CC2P33

CAMPUS: UNIP - TATUAPÉ **SEMESTRE:** 2º Semestre **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
05/11	Discutir sobre o que iremos fazer no trabalho	20 Horas	Bruno	Luciano
07/11	Dividir as funções do trabalho	20 Horas	Bruno	Luciano
10/11	Programar a lógica e interface do programa	20 Horas	Bruno	Luciano
14/11	Finalizar e revisar a Estrutura por completo	20 Horas	Bruno	Luciano

TOTAL DE HORAS: 80 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: Guilherme Vieira Abbenante Gomes

RA: N8959E0 **CURSO:** Ciências da Computação CC2P33

CAMPUS: UNIP - TATUAPÉ **SEMESTRE:** 2° Semestre **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
05/11	Discutir sobre o que iremos fazer no trabalho	20 Horas	Guilherme V.	Luciano
07/11	Dividir as funções do trabalho	20 Horas	Guilherme V.	Luciano
10/11	Escrever sobre conceitos gerais e outras técnicas conhecidas	20 Horas	Guilherme V.	Luciano
14/11	Finalizar e revisar a Estrutura por completo	20 Horas	Guilherme V.	Luciano

TOTAL DE HORAS: 80 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: Hitalo Chaves dos Santos

RA: N841419 **CURSO:** Ciências da Computação CC2P33

CAMPUS: UNIP - TATUAPÉ **SEMESTRE:** 2º Semestre **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
05/11	Discutir sobre o que iremos fazer no trabalho	20 Horas	Hitalo Chaves	Luciano
07/11	Dividir as funções do trabalho	20 Horas	Hitalo Chaves	Luciano
10/11	Programar Lógica e Interface do Programa	20 Horas	Hitalo Chaves	Luciano
14/11	Finalizar e revisar a Estrutura por completo	20 Horas	Hitalo Chaves	Luciano

TOTAL DE HORAS: 80 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: João Victor Crisci

RA: F3445H0 **CURSO:** Ciências da Computação CC2P33

CAMPUS: UNIP - TATUAPÉ **SEMESTRE:** 2° Semestre **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
05/11	Discutir sobre o que iremos fazer no trabalho	20 Horas	João Crisci	Luciano
07/11	Dividir as funções do trabalho	20 Horas	João Crisci	Luciano
10/11	Produzir Objetivo, Introução e Conclusão do trabalho	20 Horas	João Crisci	Luciano
14/11	Finalizar e revisar a Estrutura por completo	20 Horas	João Crisci	Luciano

TOTAL DE HORAS: 80 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: Murilo Henrique Martins Reis

RA: N843276 **CURSO:** Ciências da Computação CC2P33

CAMPUS: UNIP - TATUAPÉ **SEMESTRE:** 2°Semestre **TURNO:** NOTURNO

			ASSINATURA	
DATA	**ATIVIDADE**	**TOTAL DE HORAS**	**ALUNO**	**PROFESSOR**
05/11	Discutir sobre o que iremos fazer no trabalho	20 Horas	Murilo	Luciano
07/11	Dividir as funções do trabalho	20 Horas	Murilo	Luciano
10/11	Fazer a dissertação e Projeto (estrutura) do programa	20 Horas	Murilo	Luciano
14/11	Finalizar e revisar a Estrutura por completo	20 Horas	Murilo	Luciano

TOTAL DE HORAS: 80 HORAS