ESTEGANOGRAFIA PARA COMPARTILHAMENTO SEGURO DE ARQUIVOS: ENCAPSULAMENTO DE DADOS EM IMAGENS

STEGANOGRAPHY FOR SECURE FILE SHARING: DATA ENCAPSULATION INTO IMAGES

Gabriel ALBINO
gabrielalbino-@hotmail.com
Ciência da Computação, Unianchieta

Guilherme SOUZA
guilhermesouza1670@gmail.com
Ciência da Computação, Unianchieta

Pedro ALMEIDA
pedroborelialmeida@gmail.com
Ciência da Computação, Unianchieta

Vinicius SOARES
viniciussoaresanjos@gmail.com
Ciência da Computação, Unianchieta

Clayton Augusto VALDO
clayton.valdo@anchieta.br
Ciência da Computação, Unianchieta

Resumo
A esteganografia é a prática de esconder mensagens em arquivos comuns. Surgida na Grécia em 440 a.C., evoluiu até técnicas digitais modernas, como a modificação do bit menos significativo (LSB) e manipulação de transformações de cosseno discreto (DCT). Este estudo visa ocultar informações em imagens, focando nas redes sociais. Aborda também os desafios da esteganografia, como as alterações de cores dos pixels em imagens JPEG durante a compressão. Além disso, discute bibliotecas de processamento de imagem e ferramentas de criptografia que complementam a esteganografia para garantir a segurança da informação. Por fim, apresenta um software prático usando LSB para compartilhar arquivos com segurança em redes sociais, com testes em diferentes plataformas.
Palavras-Chave
Esteganografia; Redes Sociais; Segurança da Informação; LSB; DCT.

Abstract
Steganography is the practice of hiding messages in ordinary files. It originated in Greece in 440 BC and has evolved into modern digital techniques, such as least significant bit (LSB) modification and discrete cosine transform (DCT) manipulation. This study aims to hide information in images, focusing on social networks. It also addresses the challenges of steganography, such as pixel color changes in JPEG images during compression. In addition, it discusses image processing libraries and cryptography tools that complement steganography to ensure information security. Finally, it presents practical software using LSB to securely share files on social networks, with tests on different platforms.
Keywords

Steganography; Social Networks; Information Security; LSB; DCT.

INTRODUÇÃO

Este trabalho visa tratar o uso da esteganografia em redes sociais, um ambiente repleto de visibilidade onde manter uma informação oculta pode ser difícil. Segundo Klaid, Samanta e Khan (2021), esteganografia é um método de ocultar uma mensagem dentro de arquivos normais. Devido a este processo, os arquivos servem como portadores de comunicação oculta. Dessa forma, a principal dificuldade de design da esteganografia é a não detectabilidade, isto é, a percepção de que, quando o segredo está oculto, ninguém suspeitará dele.

Segundo Klaid, Samanta e Khan (2021), a esteganografia se tornou conhecida pelo uso de malwares. Todavia, dados informam que antes de 440 a.C., o povo da Grécia já utilizava essa prática. Os gregos reais raspavam a cabeça dos escravos e tatuavam mensagens secretas em seus crânios. Posteriormente, esperavam que o cabelo do escravo crescesse novamente antes de enviá-los ao destino, de forma que os inimigos não conseguissem identificar o que havia sido feito.

"Com o advento da utilização dos computadores, [...] ocorre também com esteganografia o aumento por seu interesse. O que antes se restringia a textos e tintas invisíveis ganha, agora, novas possibilidades com a esteganografia digital. Pode-se usar desde os meios mais comuns como imagem, vídeo e áudio até os mais surpreendentes como, por exemplo, descritores de arquivos e superbloco de sistemas de arquivos." (ANDERSON; NEEDHAM; SHAMIR, 1998 apud ALBUQUERQUE, 2008)

Frente os avanços tecnológicos, o uso da esteganografia continuou permeando práticas e ocupando espaço no meio virtual. Tendo como foco o uso de imagens para transmitir uma mensagem, verifica-se que existem preocupações com o formato mais usado em redes sociais, o JPE. Para Schaathun (2012 apud Polachini, 2022) este formato de imagem tem como característica a compressão com perda que descarta dados não essenciais ao sistema visual humano, visando reduzir o tamanho do arquivo da imagem. Dessa forma, técnicas de esteganografia no domínio espacial não podem ser utilizadas diretamente em arquivos desse formato.

O presente artigo, visa transmitir arquivos de forma oculta em arquivos de mídia nas redes sociais. À vista disso, o software desenvolvido neste trabalho possibilita que diferentes pessoas façam a gravação ou leitura de estego-objetos.

FUNDAMENTAÇÃO TEÓRICA

Como mencionado anteriormente, embora seja possível utilizar diversos tipos de arquivos para esteganografia, este estudo se concentra no uso de imagens para delimitar o campo de pesquisa. De acordo com Julio, Brazil e Albuquerque (2007), as imagens são a mídia de cobertura mais popular para esteganografia, podendo ser armazenadas em formato bitmap direto (como BMP) e formato comprimido (como JPEG). Em razão disso, considera-se plausível fundamentar o funcionamento de um arquivo de imagem.

Formatos de Imagem

Morkel, Eloff e Oliver (2005) explicam que, para um computador, uma imagem é uma coleção de números que representam diferentes intensidades de luz em todas as suas partes. Essa representação numérica forma uma grade que conhecemos como pixels. Na internet, a maioria das imagens é representada por uma grade retangular de pixels, onde cada pixel está associado a uma cor.

"Existem dois formatos principais para imagens de computador: raster [conhecido também como bitmap], baseado em pontos coloridos, que são quase sempre armazenados em uma matriz retangular e geralmente são compactados tão próximos que pontos individuais não são mais distinguíveis, e vetor, baseado em linhas, círculos e outros elementos 'primitivos' que normalmente cobrem uma área considerável e são facilmente distinguíveis uns dos outros. Muitas imagens

podem ser representadas em qualquer formato; de fato, qualquer imagem baseada em vetor pode ser aproximada por uma imagem raster (muitos pontos).” (ROELOFS, 1999)

- BMP

Os arquivos BMP são divididos em 3 ou 4 partes. A primeira é o cabeçalho (header), seguido pela seção de informações (info header). Caso a imagem seja indexada por cor haverá uma parte de paleta (optional palette) e, por fim, os dados de pixel (image data). A posição dos dados da imagem em relação ao início do arquivo está contida no cabeçalho. A largura e altura, tipo de compactação e número de cores ficam armazenadas no cabeçalho de informações, conforme dito por Bouke (1998). Tal estrutura pode ser observada na Figura 1.

Figura 1. Estrutura BMP (BOUKER, 1998)

- JPEG

Como dito por Mike Pound no vídeo JPEG [...] (2005), um ponto importante sobre o JPEG é que ele não é um formato propriamente dito. O JPEG é um método de compressão. No geral os arquivos são na realidade JFIFs (JPEG File Interchange Format), que trabalham como warppers dos dados compactados. Sendo assim, ao se referir a um arquivo JPEG, na realidade estamos nos referindo a um arquivo JFIF.

A compressão do JPEG é vantajosa pelo fato do ser humano não distinguir tão bem as cores quanto os tons de cinza. Assim como mudanças de alta frequência na intensidade da imagem. Pound, no vídeo JPEG [...] (2005), afirma que pedaços com tais frequências, como um arbusto ao fundo da imagem, podem ser borrados e não haverá diferença visual a não ser que se utilize o zoom.

“Então, para começar, falaremos apenas sobre os aspectos de cor do JPEG. [...] O que fazemos primeiro é mudar o espaço de cor. Nós o transformamos no espaço de cor Y-cb-cr. [...] O que estamos tentando fazer com Y-cb-cr é separar a luminosidade de uma imagem para que a intensidade de cada pixel forme a cor real. Depois de convertermos, fazemos o downsample e, essencialmente, reduzimos a quantidade de cor em nossa imagem. Então, aplicamos uma transformação discreta de cosseno, [...] E por fim a quantizamos, que é a parte com perdas real da compressão JPEG. Então, nós a codificamos e esse é o nosso arquivo.” (JPEG [...], 2015)

Algoritmos de Esteganografia

Como citado por Possatti et al. (2019), no contexto atual, a esteganografia é a prática de esconder um arquivo digital dentro de outro. Assim, um arquivo é o arquivo de cobertura (cover-message ou carrier-medium) e o outro é um segredo que se deseja esconder, chamado de dado embutido ou embedded data. Após a inserção do dado embutido na mensagem de cobertura, se obtém o estego-objeto ou esteganograma. Arquitetura representada na Figura 2.

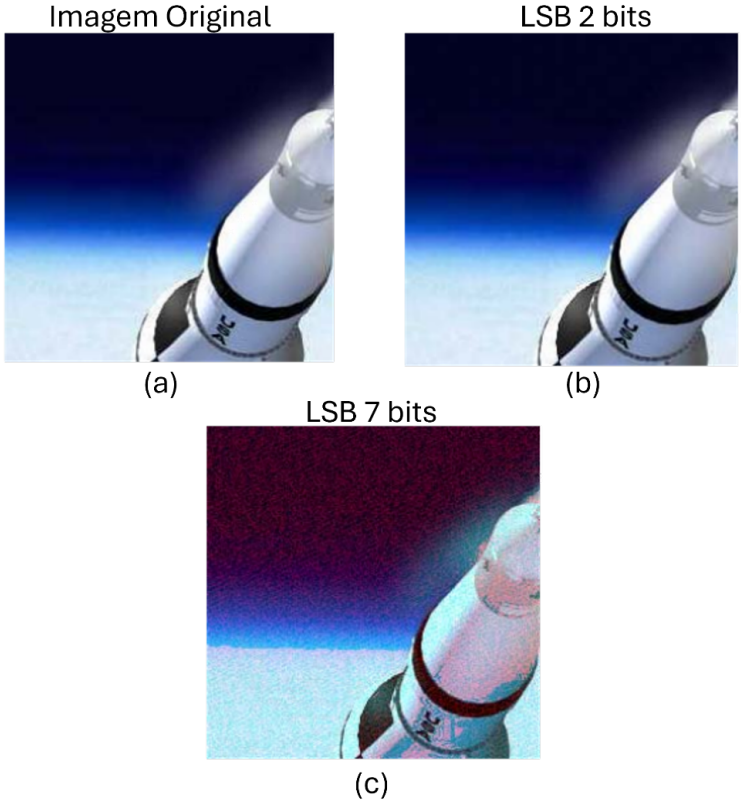
Figura 2. Arquitetura genérica de um sistema de esteganografia. (POSSATTI et al., 2019)

- LSB (Least Significant Bit)

A tática LSB é a mais comum em imagem de bitmap, tomando vantagem da estrutura da imagem para esconder a informação desejada. Em consonância com Albuquerque (2008), a estrutura de uma imagem, matematicamente, pode ser vista por 3 matrizes onde cada célula possui um valor que varia de 0 a 255, conhecido como RGB. Assim, o olho humano não consegue perceber alterações mínimas nos tons das cores em imagens, como uma variação de 1 na tonalidade azul.

Segundo Albuquerque (2008), o LSB consiste, basicamente, em modificar o bit menos significativo de uma das bandas de cor (R, G ou B) de uma célula da matriz de imagem com a finalidade de inserir a mensagem. Dentro do processo existem versões que utilizam 1, 2 ou n bits para gerar a estego-imagem. Posto isso, quanto maior o número de bits, maior o impacto na visualização da imagem e sua capacidade de armazenamento, como mostrado na Figura 3. Ademais, Albuquerque (2008) cita o baixo custo computacional como uma das características do LSB, especialmente em caso que utilizam apenas 1 bit, o que mantém uma alta fidelidade a imagem original. A modificação de apenas 1 bit garante uma dificuldade muito grande para se notar a diferença entre as imagens a 'olho nu'.

Figura 3. Exemplo LSB adaptado de (ALBUQUERQUE, 2008)



Com o baixo custo computacional surgem desvantagens. Com base no texto de Albuquerque (2008), destacam-se dois pontos principais: 1. a limitação na capacidade de armazenamento, que geralmente utilizam 1 bit por pixel, exigindo 8 pixels para armazenar um único caractere (ou, em versões modificadas, 3 bits por pixel); 2. por ser uma técnica simples e pioneira, existem muitos algoritmos capazes de detectar seu uso e extrair a mensagem oculta através de técnicas estatísticas de estegoanálise.
Alburquerque (2008) ressalta que ainda existem as versões cíclica – na qual o bit menos significativo a ser modificado é alternado entre as bandas de maneira cíclica – e com salto – em que um salto de pixels é feito entre cada bloco de n bits inseridos, sendo n a quantidade dos bits LSB escolhido no algoritmo.

- Algoritmos e Transformações

"As técnicas de esteganografia baseadas em algoritmos e transformações conseguem tirar proveito de um dos principais problemas da inserção no canal LSB que é a compressão. Para isso são utilizadas: a transformada de Fourier discreta, a transformada de cosseno discreta e a transformada Z. "(GONZALEZ; WOODS, 2002 apud JULIO; BRAZIL; ALBUQUERQUE, 2007)
Para Julio, Brazil e Albuquerque (2007) dados escondidos no domínio de transformação são mais robustos, espalhados pela imagem e possuem resistência contra o processamento de sinal. No geral, as técnicas são conhecidas pela sua sofisticação no mascaramento de informações, entretanto, a sofisticação não implica diretamente em maior robustez a ataques.
Ainda no trabalho de Julio, Brazil e Albuquerque (2007), algoritmos baseados em transformações, no geral, aplicam uma transformação em blocos de 8x8 pixel na imagem. Cada bloco tem seus coeficientes mais redundantes ou de menor importância selecionados e posteriormente recebem a mensagem secreta. Este processo ocorre através da atribuição da mensagem e cada coeficiente.

- Jesteg

Como definido por Polachini (2022), JSteg é um algoritmo utilizado em imagens JPEG baseando-se no domínio de frequência. Como o JPEG é um formato de imagem com perda que descarta dados não essenciais para os seres-humanos com o objetivo de reduzir o tamanho do arquivo da imagem, técnicas de esteganografia no domínio espacial (LSB na matriz de cor), não podem ser diretamente aplicadas. O processo de compressão com perda ocorre em blocos de 8x8 de cada componente cor. Cada bloco de pixel é mapeado através da Transformada de Cosseno Discreta bidimensional, resultando em um bloco 8x8 de coeficientes DCT.
"O algoritmo de esteganografia JSteg consiste em realizar a substituição de bits menos significativos dos coeficientes DCT de forma a não causar mudança perceptível quando a imagem é convertida para o domínio espacial." (SHEISI; MESGARIAN; RAHMANI, 2012 apud POLACHINI, 2022)

- Transformada de Cosseno Discreta (DCT)

Segundo Julio, Brazil e Albuquerque (2022), DCT é uma sigla do inglês Discrete Cosine Transform que em português significa Transformada de Cosseno Discreta. Essa é baseada em cossenos, utilizada no processamento digital de imagens e compressão de dados. O valor da função da DCT de um vetor p de pixels de comprimento n é:

$$G_f = \frac{1}{2} C_f \sum_{t=0}^{n-1} p_t \cos\left(\frac{(2t+1)f\pi}{2n}\right),$$

$$onde: C_f \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases} \quad para\ f = 0, 1, \ldots, n-1.$$

Julio, Brazil e Albuquerque (2007) explicam que a matriz dessa transformada é composta por vetores ortonormais, caracterizando-a como uma matriz de rotação. Essa transformada é amplamente utilizada na compressão de dados, pois concentra a maior parte da informação nos primeiros elementos do vetor, otimizando o armazenamento e facilitando a quantização dos valores para compressão com ou sem perdas.
Seguindo com o trabalho de Julio, Brazil e Albuquerque (2022) é dito que, os dados transformados podem ser recuperados realizando a operação inversa, conhecida como IDCT (Inverse Discrete Cosine Transform) dada pela fórmula abaixo. A maioria das compressões de imagens e vídeos usa a DCT do vetor p com tamanho n = 8.

$$p_t = \frac{1}{2}\sum_{j=0}^{n-1} C_f G_j \cos\left(\frac{(2t+1)j\pi}{2n}\right), para\ t = 0, 1, \dots, n-1.$$

“Sabendo que os pixels de uma imagem têm correlação com seus vizinhos nas duas dimensões da imagem, e não apenas em uma dimensão, a DCT para ser usada na compressão de imagens também deve ser uma transformada bidimensional. A fórmula para uma matriz (ou seja, uma imagem) p de tamanho n x n é:” (JULIO; BRAZIL; ALBUQUERQUE, 2007).

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum y = 0^{n-1} p_{xy} \cos\left(\frac{(2t+1)j\pi}{2n}\right) \cos\left(\frac{(2t+1)j\pi}{2n}\right), para\ 0 \le i,j \le n-1;$$

$$onde\ C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases}$$

“Essa transformada pode ser considerada como uma rotação (ou duas rotações consecutivas, uma em cada dimensão), ou ainda como uma base ortogonal em um espaço vetorial de n dimensões. A recuperação dos dados transformados pode ser feita usando a transformação inversa, conhecida como IDCT bidimensional:” (JULIO; BRAZIL; ALBUQUERQUE, 2007).

$$p_{xy} = \frac{1}{4}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} \cos\left(\frac{(2t+1)j\pi}{2n}\right) \cos\left(\frac{(2t+1)j\pi}{2n}\right)$$

Para Julio, Brazil e Albuquerque (2007), similarmente à transforma unidimensional, a transformada bidimensional gera uma matriz na qual os coeficientes mais significativos se concentram no canto superior esquerdo, enquanto os coeficientes de menor valor se acumulam na extremidade oposta. Essa característica facilita tanto o armazenamento de dados quanto a quantização em casos de compressão com perdas.

- Algoritmo F5

De acordo com Qiao et al. (2015 apud Possatti et al., 2019), o F5 foi desenvolvido com base no método utilizado pelo JSteg, inicialmente denominado F3. Mesmo que o Jsteg tenha resistência a ataques visuais e boa capacidade esteganográfica, o algoritmo não é resistente a ataques estatísticos, característica que também afetou o F3. Em resposta a essas vulnerabilidades, foi desenvolvida uma versão chamada F4, que obteve como objetivo corrigir esses problemas. Posteriormente, o F5 foi desenvolvido para aprimorar a eficiência de embutimento e a distribuição das alterações.
Como dito por Possatti et al. (2019), a vulnerabilidade do JSteg a ataques decorre da substituição de bits, criando uma dependência entre coeficientes que diferem apenas no LSB e igualando suas frequências. O algoritmo F3 corrige essa falha evitando a substituição de bits. Em vez disso, o F3

decrementa o valor absoluto do LSB do coeficiente quando este não coincide com o bit a ser embutido. Por exemplo, ao embutir o bit 1 no coeficiente 0000111, nenhuma alteração é feita. No entanto, ao embutir o bit 1 no coeficiente 0000110, o coeficiente é decrementado para 0000101, ajustando o LSB para o valor 1.
Possatti et al. (2019) explica que o algoritmo F4 aprimora o F3 ao corrigir a fraqueza conhecida como "shrinkage". Para isso, mapeia os coeficientes negativos com o valor esteganográfico invertido, onde coeficientes negativos pares representam um bit 1 e coeficientes negativos ímpares representam um bit 0, ao contrário do F3. O F4 opera da mesma forma que o F3 sobre os coeficientes positivos, resultando em um histograma que não difere significativamente do histograma de uma imagem original.
No que diz respeito ao F5, ele melhora dois aspectos do F4: 1. a eficiência do embutimento e a distribuição uniforme das alterações no esteganograma, ao aumentar a eficiência do embutimento, menos alterações são feitas nos coeficientes da imagem. 2. A distribuição uniforme das alterações torna mais difícil a detecção por técnicas de esteganálise, uma vez que as alterações não se concentram em uma única parte da imagem. O algoritmo F5 inicialmente permuta todos os coeficientes e, em seguida, começa o embutimento na sequência de coeficientes permutados. A permutação depende de uma chave derivada da senha usada, permitindo ao receptor, com a chave correta, repetir a permutação e ler a mensagem. Esse processo de permutação é eficiente devido à sua complexidade linear.

Bibliotecas de Processamento de Imagem
Como dito por Albuquerque (2008), o que anteriormente se restringia a textos e tintas invisíveis ganha, agora, novas possibilidades com a esteganografia digital. Os métodos mais comuns são variações de técnicas de modificação do bit menos significativo (LSB) devido ao baixo custo computacional requerido. Dessa forma, é necessário o uso de ferramentas para manipulação das imagens e aplicação dos algoritmos.

- GDI+

“GDI+ é a parte do sistema operacional Windows que fornece gráficos vetoriais bidimensionais, imagens e tipografia. O GDI+ melhora o GDI (a Interface de Dispositivo Gráfico incluída em versões anteriores do Windows) adicionando novos recursos e otimizando os recursos existentes. A interface de classe gerenciada GDI+ (um conjunto de wrappers) faz parte do .NET Framework, um ambiente para criar, implantar e executar XML Web Services e outros aplicativos.” (SANTOS; GEORGE, 2023)

- OpenCV

“OpenCV (Open Source Computer Vision Library) é uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto. OpenCV foi construído para fornecer uma infraestrutura comum para aplicações de visão computacional e para acelerar o uso da percepção de máquina nos produtos comerciais. [...] A biblioteca tem mais de 2.500 algoritmos otimizados, que inclui um conjunto abrangente de algoritmos clássicos e de última geração de visão computacional e aprendizado de máquina. [...] Possui interfaces C++, Python, Java e MATLAB e suporta Windows, Linux, Android e Mac OS.” (OPENCV, 2024)

- AForge.NET

“AForge.NET é uma estrutura C# de código aberto projetada para desenvolvedores e pesquisadores nas áreas de Visão Computacional e Inteligência Artificial - processamento de imagens, redes neurais, algoritmos genéticos, lógica fuzzy, aprendizado de máquina, robótica etc. [...] A estrutura é fornecida não apenas com diferentes bibliotecas e suas fontes, mas também com muitos aplicativos de exemplo, que demonstram o uso desta estrutura, e com arquivos de ajuda de documentação, que são fornecidos no formato HTML Help.” (AFORGE.NET, 2012)

Ferramenta de Criptografia

A segurança de uma mensagem não pode ser garantida somente com a esteganografia, já que, como dito por Olivera (2007), a esteganografia é a arte de esconder uma mensagem, enquanto a criptografia é a arte de cifrar uma mensagem. Apesar de ambas terem a intenção de proteger uma informação, os princípios são bem diferentes. Na criptografia, a existência da mensagem é conhecida, mas não o conteúdo. Com esteganografia há incerteza da existência de uma mensagem.

Silva (2017) destaca que a transmissão de uma mensagem pode estar sujeita a várias ameaças, incluindo interceptação, alteração, fabricação de mensagens falsas e interrupção da transmissão. A criptografia atua na proteção das três primeiras ameaças. Contudo, a interrupção da transmissão se mantém como um riso e possibilita o avanço gradual da esteganografia.

Tendo como exemplo a situação citada por Klaib, Samanta e Khan (2021), onde um remetente visa compartilhar informações através de canais públicos com um destinatário, mesmo na presença de um invasor. O remetente utiliza uma mensagem de cobertura para enviar o segredo ao destinatário sem que o invasor perceba. Em vez de interromper a transmissão, o intruso pode tentar encontrar alguma mensagem no texto enviado.

Silva (2017) também explica que o Processamento Digital de Imagens (PDI), a criptografia e a esteganografia podem ser combinados para garantir a confidencialidade das informações. Visto que a criptografia torna o conteúdo de uma mensagem secreto, impedindo que pessoas não autorizadas tenham acesso a esse conteúdo.

- OpenSSL

“A biblioteca de software OpenSSL é um kit de ferramentas robusto, de nível comercial e completo para criptografia de uso geral e comunicação segura. É desenvolvido sob a missão OpenSSL com o apoio da OpenSSL Foundation e OpenSSL Corporation. A biblioteca de software OpenSSL da versão 3.0 é licenciada sob a licença Apache, o que significa que é gratuito obtê-la e usá-la para fins comerciais e não comerciais, sujeita a algumas condições de licença simples.” (OPENSSL, 2024)

O Uso Negativo da Esteganografia

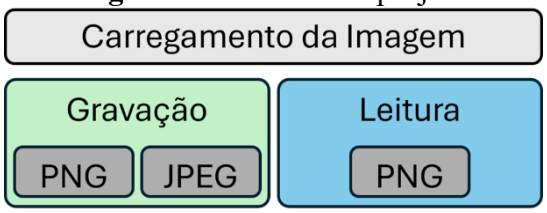
Cardoso et al. (2010) aponta que o ciberespaço é repleto de diversos aspectos atraentes ao usuário, como mensagens, sites, vídeos, links, imagens sedutoras, ofertas de produtos gratuitos e notícias de acontecimentos reais. Não obstante, por trás dessas informações aparentemente inofensivas, pode haver a presença de vírus e outras formas de informações maliciosas, utilizando a técnica conhecida como esteganografia.

Além disso, segundo Cardoso et al. (2010), a esteganografia, como qualquer outra técnica, pode ser usada tanto para atividades legais e ilegais. Muitos criminosos aproveitam programas de esteganografia disponíveis na internet para esconder fotos das vítimas em objetos aparentemente inofensivos, como imagens, vídeos e músicas, a fim de evitar a detecção em caso de apreensão dos dispositivos de armazenamento onde essas fotos se encontram.

RESULTADOS E DISCUSSÃO

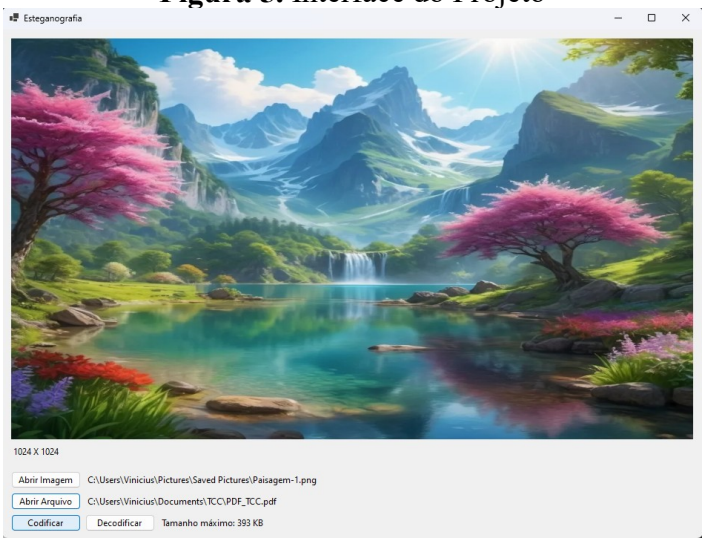
Inicialmente, é importante demonstrar como o programa é estruturado para leitura e gravação dos arquivos usados no processo de esteganografia. Dividimos o programa em três partes distintas que atuam em conjunto para entregar o estego-objeto e/ou o arquivo extraído de um estego-objeto. A Figura 4 demonstra que para o processo de gravação é possível usar qualquer tipo de imagem no processo de leitura somente PNGs são aceitos.

Figura 4. Estrutura do projeto



O carregamento da imagem modifica principalmente a interface do usuário. Ao escolher uma imagem, essa é exibida na tela em conjunto de suas dimensões e tamanho máximo do arquivo segredo a ser embutido. Caso queira inserir um segredo codificado, é possível selecionar qual o arquivo a ser embutido. Para casos de decodificação o usuário deverá seguir somente com a seleção da imagem. Tal interface é apresentada na Figura 5.

Figura 5. Interface do Projeto



Gravação

O processo de gravação usa a técnica de LSB para implantação dos bits do arquivo segredo. Antes de tudo o arquivo é convertido em vetor de bytes e tal vetor é somado a um outro com mais 8 bytes de tamanho. Dessa forma podemos usar os 4 primeiros bytes para dizer ao processo de decodificação qual a quantidade de bytes que ele precisa resgatar e mais 4 para salvar a extensão do arquivo.

Como a tática do LSB modifica os bits do canal RGB toda imagem de cobertura é convertida em bitmap dentro do programa, por esse motivo a saída sempre será um PNG. O programa passa por cada pixel da esquerda para direita e modifica o último bit de cada faixa de cor de acordo com o byte inserido. No caso de acabar com os bytes e a imagem ainda possuir pixels não utilizados o loop é cortado como demonstrado no código da Figura 7.

Figura 7. Inserção no canal RGB

```
private Bitmap LSB(Bitmap img, byte[] dataBytes)
{
    int dataIndex = 0;

    for (int i = 0; i < img.Width; i++)
    {
        for (int j = 0; j < img.Height; j++)
        {
            if (dataIndex < dataBytes.Length * 8)
            {
                Color pixel = img.GetPixel(i, j);

                // Inserir o bit do arquivo nos menores bits significativos dos componentes R, G e B
                int bitIndex = dataIndex % 8;
                int byteIndex = dataIndex / 8;
                byte bit = (byte)((dataBytes[byteIndex] >> (7 - bitIndex)) & 1);

                int r = (pixel.R & 0xFE) | ((bit >> 2) & 1);
                int g = (pixel.G & 0xFE) | ((bit >> 1) & 1);
                int b = (pixel.B & 0xFE) | (bit & 1);

                img.SetPixel(i, j, Color.FromArgb(r, g, b));
                dataIndex++;
            }
            else
            {
                break;
            }
        }
    }

    return img;
}
```

Com a imagem totalmente gerada inicia-se o processo de salvamento do estego-objeto. Como dito anteriormente a imagem foi convertida para Bitmap logo a única maneira de salvar disponível é de formatos com esta base, sem ela não é possível manipular a banda de cor da forma como foi feito. A Figura 8 demonstra que o resultado só pode ser salvo em PNG.

Figura 8. Processo de gravação do resultado

```
// Salva a imagem com o arquivo embutido
SaveFileDialog saveFile = new SaveFileDialog
{
    Filter = "PNG Image (*.png)|*.png",
    InitialDirectory = @"C:\Users\Guilherme\Desktop"
};

if (saveFile.ShowDialog() == DialogResult.OK)
{
    string savePath = saveFile.FileName;
    img.Save(savePath, System.Drawing.Imaging.ImageFormat.Png);
    MessageBox.Show("Arquivo embutido com sucesso!");
}
```

Neste processo não é necessário o uso de nenhuma biblioteca adicional além das já disponíveis no C#. Sendo principal a System.Drawing.Imaging que permite salvar a imagem resultante como PNG após todas as modificações.

Decodificação

No processo de decodificação, realizamos o inverso do processo de codificação. Isso significa que, ao colocar um PNG temos a leitura do LSB na matriz de cor. Este processo reverso consiste em ler todos os bits menos significativos e montar um vetor de bytes. Logo após é realizada a extração dos 4 primeiros bytes para delimitar o tamanho dos arquivos a serem resgatado. Com isso é feita a extração da extensão do arquivo e convertida em texto. Por fim, o conteúdo do arquivo é resgatado e uma caixa de diálogo é aberta para salvar o arquivo já com a extensão correta. Processo exibido na Figura 9.

Figura 9. Extração via LSB

```
// Recuperar o comprimento do arquivo dos primeiros 4 bytes
int fileLength = BitConverter.ToInt32(dataBytes, 0);

// Verificar se fileLength é razoável
if (fileLength < 0 || fileLength > dataBytes.Length - 8)
{
    MessageBox.Show("Erro ao recuperar o arquivo: comprimento dos dados não é válido.");
    return;
}

// Verificar se temos bytes suficientes para a extensão e o conteúdo do arquivo
if (dataBytes.Length < 8 + fileLength)
{
    MessageBox.Show("Erro ao recuperar o arquivo: o comprimento dos dados não é suficiente.");
    return;
}

// Recuperar a extensão do arquivo dos próximos 4 bytes
fileExtension = Encoding.UTF8.GetString(dataBytes, 4, 4).Trim('\0');

// Recuperar o conteúdo do arquivo
byte[] fileBytes = new byte[fileLength];
Array.Copy(dataBytes, 8, fileBytes, 0, fileLength);

// Salvar o arquivo extraído
SaveFileDialog saveFile = new SaveFileDialog
{
    Filter = $"Arquivo Recuperado (*.{fileExtension})|*.{fileExtension}",
    DefaultExt = fileExtension,
    InitialDirectory = @"C:\Users\UserName\Desktop"
};
```

A extração do tamanho do arquivo passa por uma checagem para garantir que o valor resgatado esteja dentro dos parâmetros possíveis, sendo eles: o tamanho do arquivo é menor que o da imagem menos os 8 bytes do programa e o tamanho do arquivo mais os bytes da extensão é menor do que a imagem. Com tais verificações prontas a extensão pode ser extraída e usada para criar o arquivo.

Teste e Resultados

Para realizar os testes no programa utilizamos um arquivo PNG de 1024X1024 com tamanho original de 1.22MB. O arquivo segredo inserido na imagem é uma cópia da versão atual deste trabalho, um PDF de 327KB. Após realizada a codificação do PDF o arquivo resultante é um PNG de 1024X1024 com tamanho de 2MB, as duas imagens podem ser vistas na Figura 10.

Figura 10 – Demonstração imagem original e estego-objeto

Como é possível perceber, ambas as imagens parecem idênticas a olho nu, alcançando assim o objetivo de qualquer estego-objeto. Essa imagem será utilizada para espalhar o nosso arquivo segredo. As redes utilizadas serão o Discord, um mensageiro utilizado pelo público de jogos eletrônicos, WhatsApp e Instagram.

- Discord

Ao publicar uma imagem no Discord, um preview é gerado como mostrado na figura 11. Qualquer usuário com acesso a este canal pode realizar o download da imagem em seu computador. Ao baixá-la o arquivo resultante possui os mesmos 2MB do estego-objeto enviado indicando que a imagem não passou por qualquer tipo de modificação no envio. Executar o módulo de leitura na imagem resulta em uma janela para salvar um PDF demonstrado na Figura 12. O PDF resultante pode ser aberto e exibe corretamente o PDF original.

Figura 11. Publicação Discord

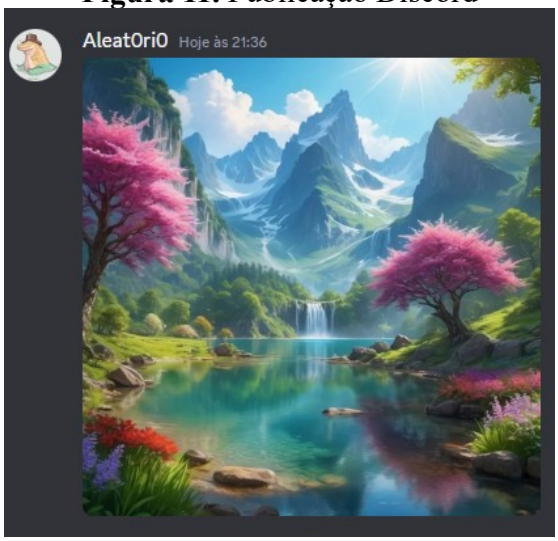
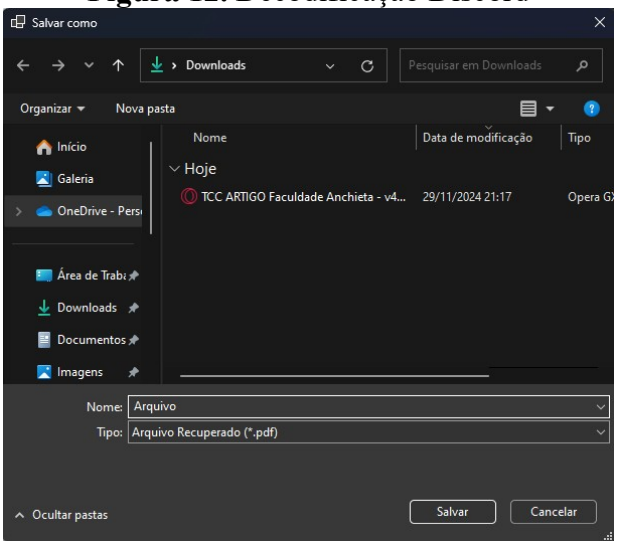


Figura 12. Decodificação Discord



- WhatsApp

Imagens enviadas no WhatsApp exibem um preview igualmente ao Discord, exemplificado na figura 13. O Download desta imagem resulta em um arquivo JPG com 228KB isso significa que uma grande compressão foi utilizada. Ao tentar ler este arquivo com o método LSB o programa retorna erro pois os bytes encontrados no início da imagem resultam em um tamanho de arquivo maior do que o disponível. Uma forma de contornar a conversão é enviar a imagem como um documento, resultando na visualização da figura 14. O download desta visualização retorna o estego-objeto original.

Figura 13. Publicação WhatsApp

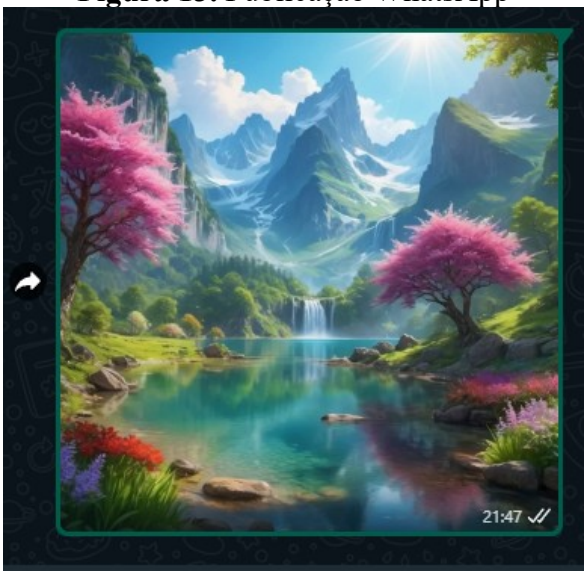
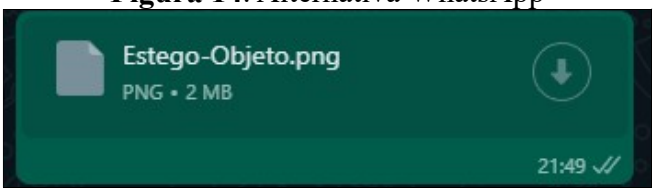


Figura 14. Alternativa WhatsApp

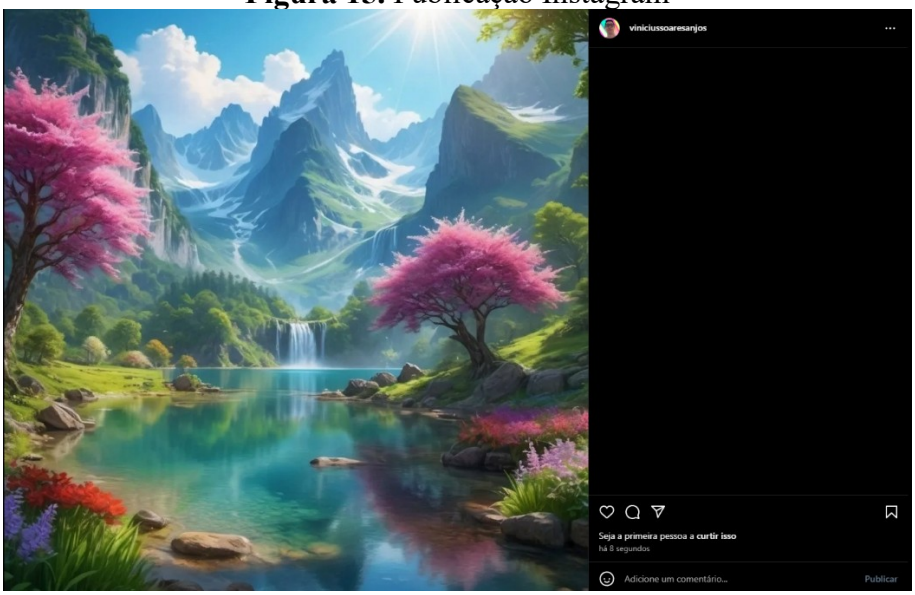


O PDF resultante do download da visualização da figura 15 pode ser aberto e exibe corretamente o PDF original.

- Instagram

Imagens publicadas no Instagram sofrem do mesmo problema do WhatsApp. Ao postar uma imagem ela é convertida para JPEG resultando em uma imagem de 1080X1080 com 220KB. Realizar a decodificação em tal imagem gera um erro de overflow na criação do vetor de bytes do arquivo segredo. O uso do Instagram e redes semelhantes se torna impossível pelo motivo dessa conversão. A publicação pode ser vista na figura 15.

Figura 15. Publicação Instagram



CONSIDERAÇÕES FINAIS

Tendo como exemplo os mensageiros WhatsApp e Discord, podemos enviar a imagem gerada e quem a recebe pode ler o arquivo secreto. Contudo, redes como Facebook e Instagram transformam automaticamente o PNG em JPEG o que distorce os bits e corrompe o arquivo secreto.
À vista disso, o uso de redes sociais para compartilhamento de arquivos é possível com certas exceções. Redes que permitem o compartilhamento de arquivos sem conversão automática possibilitam que o processo de esteganografia permaneça na imagem até seu destino, como no caso o Discord e WhatsApp.
Para redes sociais focadas na distribuição de imagem, como Facebook e Instagram não será possível utilizar o programa gerado. Arquivos de imagem passam por compressão ao serem convertidos para JPEG como explicado na fundamentação deste trabalho, essa compressão é com perda o que resulta na alteração dos valores da matriz RGB. Por esse motivo o compartilhamento não funciona na versão atual de nossa aplicação.
Como proposta para trabalhos futuros, deixamos a implementação por completo da técnica de JSteg ou F5 o que abriria espaço para compartilhamento do estego-objeto nas redes mencionadas no parágrafo anterior. Como encontrado durante nossas pesquisas, tais técnicas não manipulam o domínio espacial da imagem, resultando em uma codificação já embutida na compressão do arquivo.

REFERÊNCIAS BIBLIOGRÁFICAS

AFORGE.NET. *AForge.NET Framework*. [S. l.], 2012. Disponível em: <https://www.aforgenet.com/framework/>. Acesso em: 25 out. 2024.

ALBUQUERQUE, Rafael Bezerra. Esteganografia: *Análise de Algoritmos Baseada em Comparação entre Imagens***.** Orientador: Carlos Alexandre Barros de Mello. 2008. 54 p. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação) - Escola Politécnica de Pernambuco, [S. l.], 2008. Disponível em: <https://www.ecomp.poli.br/ListaTCC/20082/TCC_Rafael-Esteganografia_vf.pdf>. Acesso em: 19 out. 2024.

BOURKE, Paul. *BMP Image Format*. 1998. Disponível em: https://www.ubytujsa.sk/files/BMP_files.pdf. Acesso em: 15 nov. 2024.

CARDOSO, Nágila Magalhães; HASHIMOTO, Yuri Campos; SILVA, Keith Maila Domingos; MAIA, Anderson Trindade. Redes sociais a nova arma do crime cibernético: O efeito do uso da engenharia social e da esteganografia. In: *THE SIXTH INTERNATIONAL CONFERENCE ON FORENSIC COMPUTER SCIENCE.* [S. l.: s. n.], 2010. p. 195-201. Disponível em: <http://icofcs.org/2011/ICoFCS2011-PP23.pdf>. Acesso em: 25 out. 2024.

JPEG ‘files‘ & Colour (JPEG Pt1)- Computerphile. Direção: Sean Riley. Gravação de Sean Riley. University of Nottingham: [s. n.], 2005. Disponível em: https://www.youtube.com/watch?v=n_uNPbdenRs. Acesso em: 5 dez. 2024.

JULIO, Eduardo Pagani; BRAZIL, Wagner Gaspar; ALBUQUERQUE, Célio Vinicius Neves. Esteganografia e suas Aplicações. In: PIRMEZ, Luci; DELICATO, Flávia Coimbra; CARMO, Luiz Fernando Rust da Costa. *Minicursos do VII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais.* [S. l.: s. n.], 2007. cap. 2, p. 54-102. Disponível em: <http://www.dcc.ic.uff.br/~celio/papers/minicurso-sbseg07.pdf>. Acesso em: 19 out. 2024.

KLAIB, Mohammad Fadel Jamil; SAMANTA, Debabrata; KHAN, Mohammad Zubair. *Social*

Media and Steganography: Use, Risks and Current Status. [S. l.], 2 nov. 2021. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9599677>. Acesso em: 26 set. 2024.

MORKEL, T; ELOFF, JHP; OLIVIER, MS. AN OVERVIEW OF IMAGE STEGANOGRAPHY. *In*: FIFTH ANNUAL INFORMATION SECURITY SOUTH AFRICA CONFERENCE (ISSA2005), 2005, Sandton, África do Sul. *Proceedings* [...]. [*S. l.*: *s. n.*], 2005. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/30900669/stegoverview-libre.pdf?1392204634=&response-content-disposition=inline%3B+filename%3DAn_overview_of_image_steganography.pdf&Expires=1731689802&Signature=NEjUmRy98UMJDR-mMlb~DU97Hc4thjjMrzC5GqQqtu30fTWh0XX~UQQ0BRKYkXTvrLP4DTjHyHenF9b1nXud~nlCXU8dBL2JzWF~tLYqnuqyyDXBkTEjZ78IZr39iIxqzFaq7xzkzUu1nkO8dRD05SSPNO10lEYvPxTJEa6TRFj0Wd7Np19SE~L5Q0vJPS882PfNXxsh0cFVJD4LkiIZK5kI9v4hXwjA~4alW9X0XexyyzBr1gJemU8u30~asjuOR9p7y79nOTegcLGghvrAX~tduxOZzpJVxD99eH0r8Q8c-rOIi~~OPLIA8A6j-7B-ZMmrb1l1~kKqiqJYy-n82A__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>. Acesso em: 15 nov. 2024.

OLIVEIRA, Fábio Borges. *Análise da segurança de criptografia e esteganografia em seqüências de imagens***.** Orientador: Renato Portugal & Jauvane Cavalcante de Oliveira. 2007. Dissertação (Mestrado em Modelagem Computacional) - Laboratório Nacional de Computação Científica, [S. l.], 2007. Disponível em: <https://www.lncc.br/~borges/doc/dissertação.pdf>. Acesso em: 19 out. 2024.

OPENCV. *Sobre o Código Gerenciado no GDI+*. [S. l.], 2024. Disponível em: <https://opencv.org/about/>. Acesso em: 25 out. 2024.

OPENSSL. *OpenSSL - Library*. [S. l.], 2024. Disponível em: <https://openssl-library.org>. Acesso em: 25 out. 2024.

POLACHINI, Matehus Esquinelato. *Detecção de Esteganografia em Imagens Utilizando Aprendizado de Máquina*. Orientador: Kelton Augusto Pontara da Costa. 2022. 54 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Estadual Paulista Júlio de Mesquita Filho, [S. l.], 2022. Disponível em: <https://repositorio.unesp.br/server/api/core/bitstreams/0a78a27f-9a40-4bde-9390-d4658458a682/content>. Acesso em: 19 out. 2024.

POSSATTI, Lucas Caetano; FILHO, Gilberto Neves Sudré; RESENDO, Leandro Colombi; ANDRADE, Jefferson Oliveira; KOMATI, Karin Satie. *Um Estudo de Técnicas de Esteganálise em Estego-Imagens com Texto Embutido com LSB*. Brazilian Journal of Development, [S. l.], v. 5, n. 10, p. 20702-20738, 20 out. 2019. DOI 10.34117/bjdv5n10-251. Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/3964/3745>. Acesso em: 19 out. 2024.

ROELOFS, Greg. *PNG: The Definitive Guide*. [*S. l.*]: O'Reilly & Associates, Inc., United States, 1999. 321 p. ISBN 1565925424.

SANTOS, Brittany; GEORGE, Andy. *Sobre o Código Gerenciado no GDI+*. In: MICROSOFT. Elementos Gráficos e Desenhos nos Windows Forms. [S. l.], 18 out. 2023. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/desktop/winforms/advanced/about-gdi-managed-code?view=netframeworkdesktop-4.8>. Acesso em: 25 out. 2024.

SILVA, João Paulo de Freitas Costa. *Criptografia e esteganografia aplicadas a imagens digitais.*

Orientador: Leandro Carlos de Souza. 2017. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal Rural do Semi-árido, [S. l.], 2017. Disponível em: <https://repositorio.ufersa.edu.br/server/api/core/bitstreams/ac01e233-e187-4a06-b0a4-1cfc58235402/content>. Acesso em: 19 out. 2024.

AGRADECIMENTOS

Gostaríamos de expressar nossa profunda gratidão a todos que contribuíram para a realização deste trabalho. Primeiramente, agradecemos ao nosso orientador, Prof. Me. Clayton Valdo e coordenador de curso Vanderlei Lenne, pelo suporte, paciência e valiosas orientações durante todo o processo. À nossas famílias, por todo o amor, compreensão e apoio incondicional, especialmente nos momentos mais desafiadores. Agradecemos também aos amigos, pelo encorajamento e companhia nos momentos de descontração. Por fim, agradeço a todas as instituições e pessoas que, direta ou indiretamente, contribuíram para que este trabalho fosse realizado.