

NEVA: Programa para aumento de performance de sites web usando o algoritmo de esteganografia

MÁRCIO PANTOJA LOBATO¹,
PEDRO VICTOR PONTES PINHEIRO¹,
ROBERTO YURI DA SILVA FRANCO¹,
ALESSANDRA NATASHA ALCANTARA B. BAGANHA²

Resumo. Atualmente, qualidade pode já não ser considerado um diferencial competitivo e sim uma exigência. O mesmo vale para o cenário da internet, por isso não se pode deixar de considerar um aspecto importante: a velocidade. O tempo de carregamento de um site é alvo de estudo e testes constantes para obter a melhor performance possível. Este artigo descreve o desenvolvimento do programa NEVA, que busca melhorar a performance dos sites web. Para isso, o NEVA faz uso do algoritmo de esteganografia, o qual auxilia na redução de arquivos a serem enviados a uma máquina cliente. O funcionamento do NEVA, explicações mais detalhadas sobre o protótipo e alguns conceitos teóricos serão apresentados.

Palavras-chave: neva, esteganografia, performance de sites web.

NEVA: Software to increase the performance of web sites using the steganography algorithm

Abstract. Nowadays, quality can no longer be considered a competitive advantage but a requirement. The same applies to the internet scenario, therefore we cannot fail to consider one important aspect: speed. The loading time of a web site is the object of study and constant testing to get the best possible performance. This article describes the development of the NEVA software, which aim to improve the performance of web sites. To do that, NEVA makes use of the steganography algorithm, which helps to reduce the number of files to be sent to a client machine. The functioning of NEVA, more detailed explanations on the prototype and some theoretical concepts will be presented.

Key words: neva, steganography, web sites performance.

¹ Bacharel em Ciência da Computação - Centro Universitário do Estado do Pará (CESUPA)
e-mail: {marcio.lobato02, pedrovictor.pinheiro, roberto.yuri.franco}@gmail.com

² Mestre em Ciência da Computação. Docente do Centro Universitário do Estado do Pará (CESUPA).
e-mail: alessandra.natasha@gmail.com

1. INTRODUÇÃO

O HTTP (*HyperText Transfer Protocol*) é um protocolo de aplicação comumente utilizado na *web* para a comunicação entre servidores e os navegadores. (TANENBAUM, 2003)

Em termos de funcionamento cliente-servidor, o tráfego de informações entre uma estação cliente e um servidor na *web* é realizado a partir do navegador (*browser*) do cliente quando este envia o nome da página para o servidor, o qual transmite a página de volta.

Desta forma, nessa comunicação pode-se encontrar um problema, pois segundo Souders (2007) apenas 10 a 20% do tempo de envio da página é destinado ao documento HTML (*HyperText Markup Language*), enquanto que 80 a 90% do tempo é destinado aos outros componentes (imagens, folhas de estilo, *scripts*, *flash*, etc.).

A partir desses dados, um grupo de estudantes do Centro Universitário da Pará (CESUPA) foi motivado a encontrar maneiras de diminuir o tempo nesses envios dos componentes da página, assim surgiu a idéia do NEVA. O acrônimo NEVA significa: “Nunca Esteganografia Vista Assim”, pois seu principal objetivo reside na associação do algoritmo de esteganografia como uma maneira de aumentar a performance do carregamento dos *sites web*.

O restante do artigo está estruturado de forma a apresentar na Seção 2 as principais características do projeto enquanto a Seção 3 aborda aspectos a respeito do desenvolvimento do projeto, ficando o modo de funcionamento descrito na Seção 4. A Seção 5 faz uma demonstração de uso do protótipo e comentários finais e trabalhos futuros são parte da Seção 6.

2. CARACTERÍSTICAS DO PROJETO

O NEVA procura usar a esteganografia³ devido a característica de embutir, ou seja, “ocultar” um dado dentro de outro. A técnica da esteganografia, assim como a criptografia, comumente usa o conceito de chave, entretanto esse conceito não é usado no NEVA para poder diminuir ao máximo o tempo de extração do objeto ocultado.

³ A esteganografia significa “a arte da escrita escondida”, ela é capaz de embutir um dado dentro de um determinado objeto chamado estego-objeto. A esteganografia é usada na área de segurança. (ALBUQUERQUE, 2007)

Dessa forma, o NEVA busca diminuir o número de requisições HTTP, pois ao embutir dados dentro de outros, pode-se enviar mais dados por um custo de requisições HTTP menor, o que implica em menos atividade na rede. Essa forma de redução de requisições HTTP é similar a forma utilizada pela técnica de compressão Gzip⁴.

3. DESENVOLVIMENTO DO PROJETO

O processo de desenvolvimento do NEVA iniciou pelo levantamento dos requisitos funcionais e não funcionais em forma de *user stories*, necessários para a construção do projeto. Também foram definidas estratégias para cumprir objetivos específicos mostrados conforme cada item de desenvolvimento do projeto:

1. Garantia do não aumento de tamanho dos arquivos depois da esteganografia, ou seja, quando um ou mais dados ao serem embutidos dentro de um objeto, esse resultado final deve possuir o mesmo tamanho do objeto inicial.
2. Garantia da qualidade da imagem final, ou seja, a imagem não sofre alterações perceptíveis.
3. O dado retirado do objeto final apresenta-se exatamente igual ao dado inicial tanto em tamanho, quanto em nome e formato, assim a integridade dos arquivos é garantida mesmo com o não aumento de tamanho descrito no item 1.
4. Desenvolvimento de uma versão do projeto a ser utilizada em cada lado do ambiente cliente-servidor. Uma versão para realizar a esteganografia no lado do servidor, e outra para retirar os objetos esteganografados no lado do cliente.

A comunicação entre as duas versões, citadas no item 4, está baseada no conceito de *sockets* e *streams*, e ambas foram desenvolvidas na linguagem de programação JAVA (DEITEL, 2005).

O item 1 garante uma grande diferenciação do NEVA pros softwares atuais que realizam a esteganografia, pois estes não se preocupam com o tamanho final do arquivo.

4. MODO DE FUNCIONAMENTO

A figura 1 mostra um diagrama de seqüência que ilustra os passos da comunicação e a troca de arquivos entre os protótipos.

⁴ O Gzip é atualmente um método de compressão popular e efetivo, utilizado para comprimir os documentos HTML e também *scripts* e folhas de estilo, ficando a cargo do lado do cliente a tarefa da descompressão desses arquivos. (SOUDEERS, 2007)

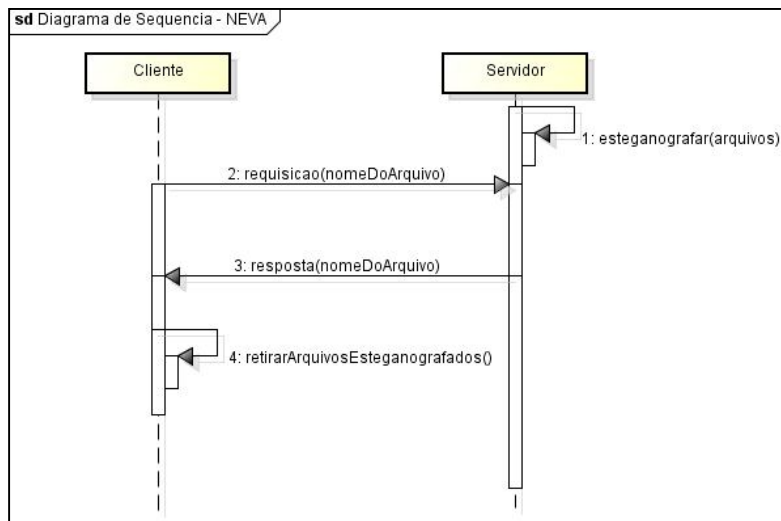


Figura 1. Diagrama de seqüência do protótipo

1. É realizada a esteganografia dos arquivos existentes no servidor;
2. O cliente faz uma requisição ao servidor por um determinado arquivo;
3. O servidor responde ao cliente com esse arquivo;
4. O cliente realiza o processo inverso da esteganografia, ou seja, extrai o objeto oculto.

5. APRESENTAÇÃO DO PROTÓTIPO

Na demonstração a seguir, foi utilizado um arquivo do tipo HTML e um arquivo do tipo BMP.

No protótipo atual é realizado manualmente a esteganografia através de outra versão do projeto, na figura 2 a aplicação pede por um arquivo do tipo imagem e na figura 3 por um do tipo texto.

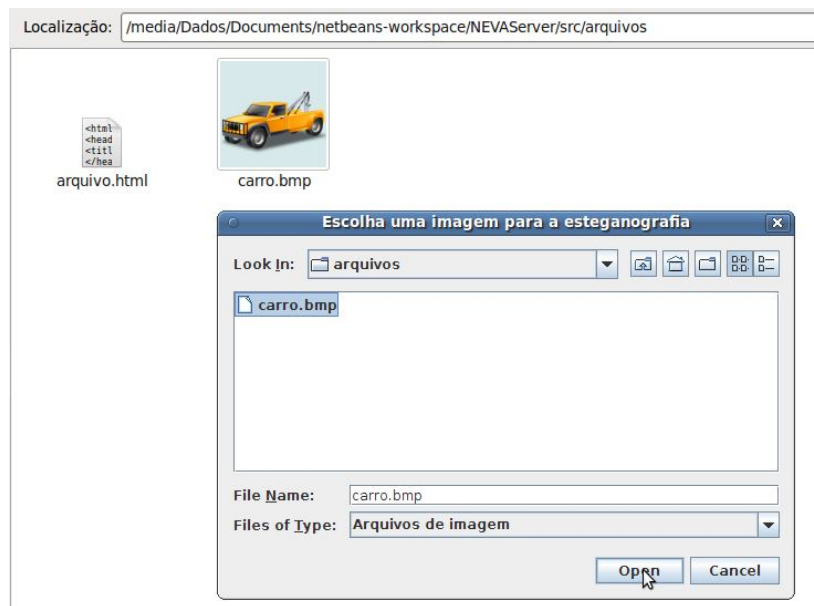


Figura 2. Escolha de arquivo do tipo imagem

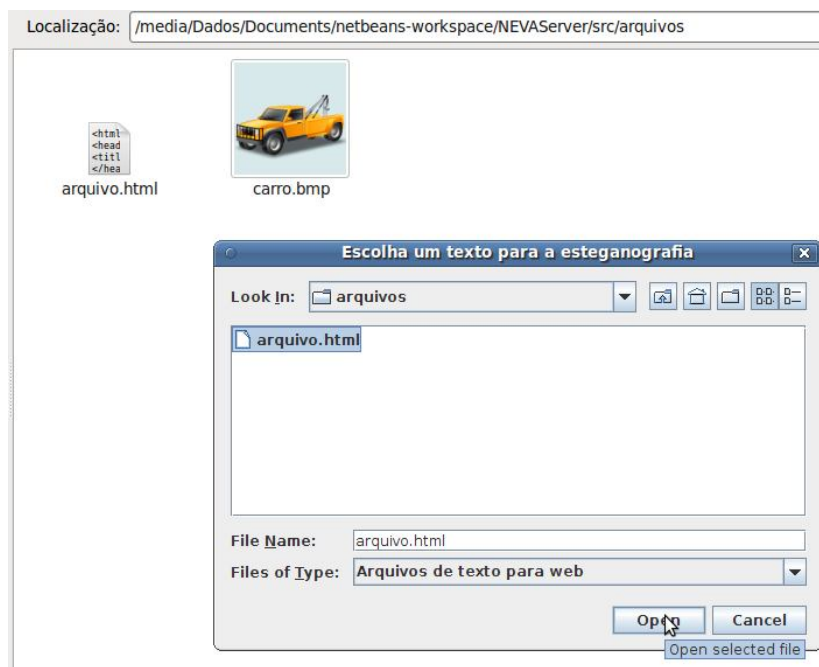


Figura 3. Escolha do arquivo do tipo texto

A figura 4 a seguir mostra as propriedades desses arquivos.

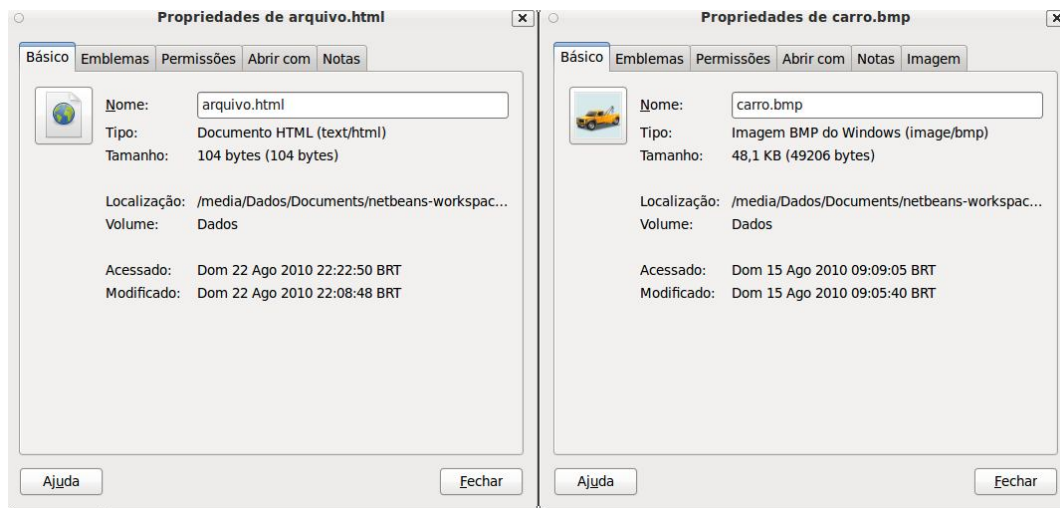


Figura 4. Propriedades dos arquivos no lado do servidor

Um aspecto interessante é que ao realizar a esteganografia o *software* analisa o arquivo texto e gera um cabeçalho a ser inserido na imagem junto com o texto, essa estratégia foi adotada para armazenar três informações importantes: nome do arquivo, formato e tamanho.

Na figura 5 observa-se o arquivo final gerado, o qual possui “Modificada” acrescentado ao final de seu nome, apenas para identificação do mesmo.

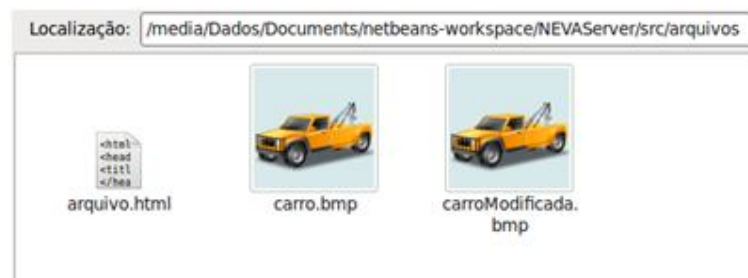


Figura 5. Criação da imagem “carroModificada.bmp” com o “arquivo.html” esteganografado

Na figura 6, a seguir, pode-se observar que as duas figuras são de mesmo tamanho e que as diferenças nas imagens são imperceptíveis.

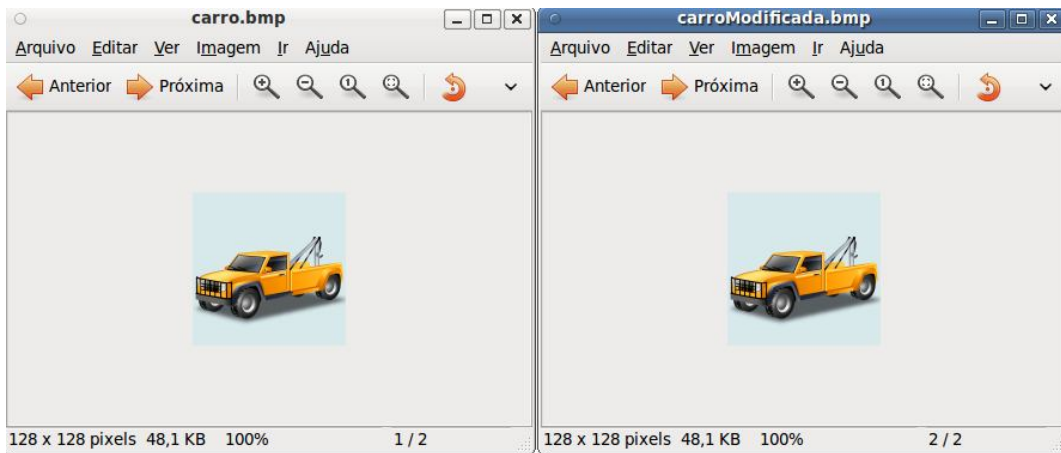


Figura 6. Comparação entre as imagens normal (esquerda) a esteganografada (direita)

Em seguida com o servidor já iniciado, deve-se iniciar o processo cliente fazer a requisição do arquivo “carroModificada.bmp”. Encontrando o arquivo, este será enviado para o cliente, onde será realizado a extração do objeto esteganografado.

Ao verificar a pasta de arquivos recebidos do cliente o “arquivo.html” é encontrado, porém teve o acréscimo de “Extraído” ao seu nome para o propósito de identificação. A figura 7 mostra a pasta com esses arquivos, e a figura 8 mostra a comparação do “arquivo.html” antes e depois de ser esteganografado.

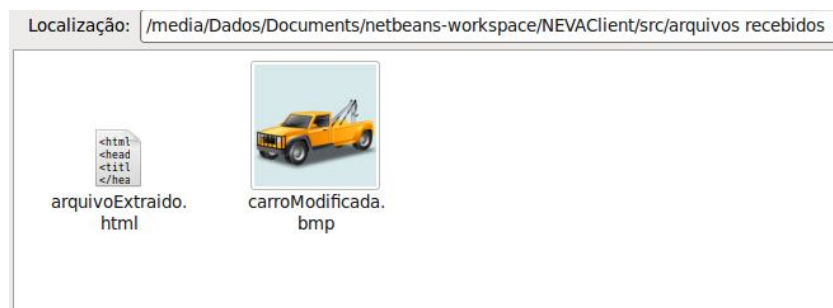


Figura 7. Processo de extração do arquivo concluída

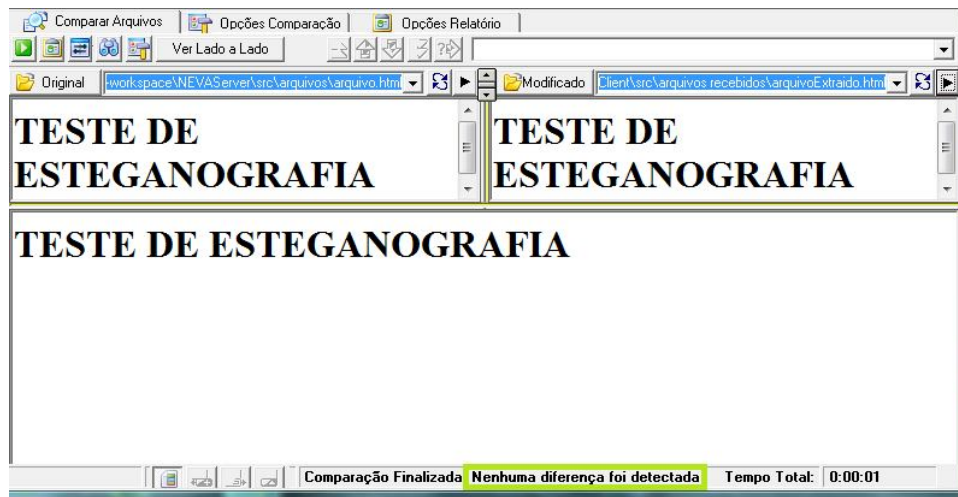


Figura 8. Nenhuma diferença encontrada entre os arquivos de texto

Pode-se verificar que o tamanho dos arquivos não variou ao comparar a figura 4 com a figura 9.

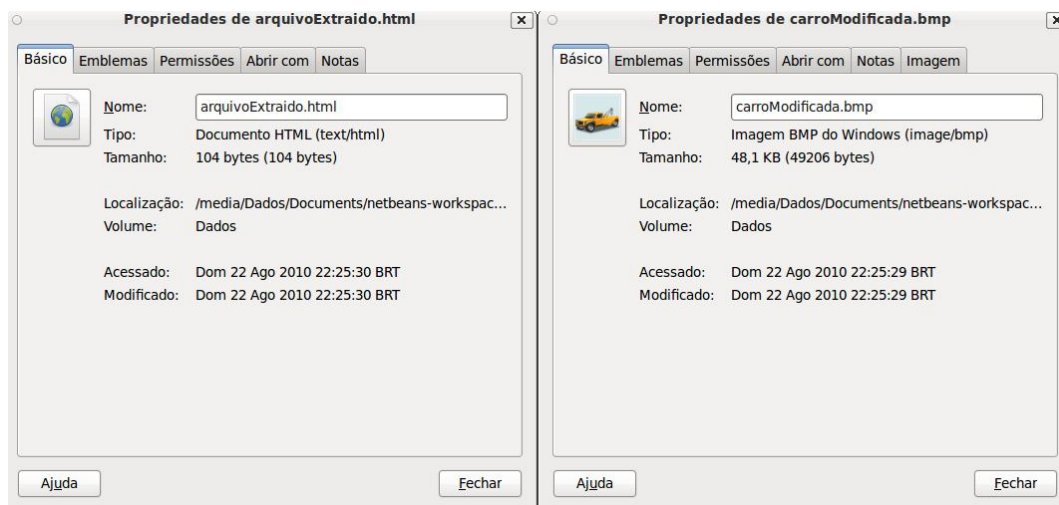


Figura 9. Propriedades dos arquivos no lado do cliente

6. CONCLUSÃO E TRABALHOS FUTUROS

Pode-se perceber que com o NEVA é possível abordar a esteganografia a partir de um novo contexto, associando o aumento de desempenho uma vez que foi possível o envio de um arquivo texto e uma imagem *bitmap* pelo mesmo “custo” em *bytes* do arquivo *bitmap*.

Entretanto, ainda são necessários diversos testes de carga e desempenho para saber até onde é válido o emprego da técnica, uma vez que foi observado durante o processo o aumento de consumo de processamento.

Como trabalhos futuros será associado esteganografia a um método de compressão, e empregada a técnica de aferição, através da monitoração e coleta de dados, bem como medido o intercâmbio entre o aumento de performance de transmissão e o uso de recursos como memória e CPU.

REFERÊNCIAS

ALBUQUERQUE, C. V. N. **Esteganografia: A Arte das Mensagens Ocultas**. Fonte, v. 4, p. 101-109. Belo Horizonte, 2007.

DEITEL, H. M., DEITEL P. J. **JAVA: Como Programar**. 6 ed. Prentice-Hall, 2005.

MARQUES, A. C. R.; SOARES, A. G. M.; TORRES, R. H. **Aplicando Engenharia de Software Experimental em um Estudo Comparativo entre Algoritmos de Compressão de Dados e Esteganografia**. Monografia (Bacharelado em Ciência da Computação). Centro Universitário do Estado do Pará, 2009.

SOUDERS, S. **High Performance Web Sites - Essential Knowledge for Front-End Engineers**. 1 ed. O'Reilly Media, 2007.

TANENBAUM, A. S. **Redes de Computadores**. 4. ed. São Paulo: Campus, 2003.