REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

UMA MENSURAÇÃO DO DESEMPENHO ENTRE OS ALGORITMOS DE BUSCA BINÁRIA TRADICIONAL E DE BUSCA BINÁRIA BUIATTI

A PERFORMANCE MEASUREMENT BETWEEN TRADITIONAL BINARY SEARCH AND BUIATTI BINARY SEARCH ALGORITHMS

UNA MEDICIÓN DEL RENDIMIENTO ENTRE LA BÚSQUEDA BINARIA TRADICIONAL Y LOS ALGORITMOS DE BÚSQUEDA BINARIA DE BUIATTI

Reane Franco Goulart
Doutorado em Engenharia Elétrica com ênfase em Inteligência Artificial, UFU, Ituiutaba, Minas Gerais – Brasil
E-mail: reane@iftm.edu.br

Roberto Caetano Buiatti
Graduando em Ciência da Computação, Instituição de formação acadêmica: IFTM – Campus Ituiutaba, Ituiutaba, Minas Gerais – Brasil
E-mail: roberto.buiatti@estudante.iftm.edu.br

Resumo

Este trabalho de iniciação científica apresenta uma medição de performance detalhada do desempenho entre o algoritmo de busca binária tradicional e a busca binária Buiatti, uma inovadora variação adaptativa. Embora a busca binária tradicional seja ótima no pior caso para comparações (complexidade O(log n)), sua estratégia de divisão simétrica fixa não explora propriedades estatísticas da distribuição dos dados ou padrões de acesso que poderiam reduzir o custo médio da busca. Em cenários práticos onde os dados apresentam distribuições não uniformes (como clusters) ou quando existem padrões de acesso repetitivos, essa independência em relação aos dados representa uma oportunidade perdida de otimização. Neste contexto, a busca binária Buiatti emerge como uma alternativa promissora, projetada para otimizar o caso médio ao introduzir heurísticas de verificação local antes da divisão clássica. A metodologia adotada, fundamentada em obras clássicas como Cormen et al. (2009) e Knuth (1998), consistiu na implementação e análise comparativa de ambas as versões. Testes empíricos utilizando estratégias de busca aleatória e pontos de controle revelaram que, embora ambos mantenham a complexidade logarítmica, a abordagem Buiatti oferece ganhos de desempenho mensuráveis e maior estabilidade em cenários com localidade de referência, validando seu potencial para aplicações específicas.

Palavras-chave: Busca Binária; Otimização de Algoritmos; Medição de Desempenho; Algoritmos Adaptativos; Busca Binária Buiatti.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Abstract

This undergraduate research project presents a detailed performance measurement comparing the traditional binary search algorithm with Buiatti binary search, an innovative adaptive variation. Although traditional binary search is worst-case optimal for comparisons (complexity O(log n)), its fixed symmetric splitting strategy does not exploit statistical properties of the data distribution or access patterns that could reduce the average search cost. In practical scenarios where data exhibits non-uniform distributions (such as clusters) or where repetitive access patterns exist, this independence from the data represents a missed optimization opportunity. In this context, Buiatti binary search emerges as a promising alternative, designed to optimize the average case by introducing local verification heuristics before classical splitting. The methodology adopted, based on classic works such as Cormen et al. (2009) and Knuth (1998), consisted of the implementation and comparative analysis of both versions. Empirical tests using random search strategies and control points revealed that, although both maintain logarithmic complexity, the Buiatti approach offers measurable performance gains and greater stability in reference locality scenarios, validating its potential for specific applications.

Keywords: Binary Search; Algorithm Optimization; Performance Measurement; Adaptive Algorithms; Buiatti Binary Search.

Resumen

Este proyecto de investigación de pregrado presenta una medición detallada del rendimiento comparando el algoritmo de búsqueda binaria tradicional con la búsqueda binaria Buiatti, una innovadora variante adaptativa. Si bien la búsqueda binaria tradicional es óptima en el peor de los casos para las comparaciones (complejidad O(log n)), su estrategia de división simétrica fija no aprovecha las propiedades estadísticas de la distribución de datos ni los patrones de acceso que podrían reducir el coste medio de la búsqueda. En escenarios prácticos donde los datos presentan distribuciones no uniformes (como clústeres) o donde existen patrones de acceso repetitivos, esta independencia de los datos representa una oportunidad de optimización perdida. En este contexto, la búsqueda binaria de Buiatti surge como una alternativa prometedora, diseñada para optimizar el caso promedio mediante la introducción de heurísticas de verificación local antes de la división clásica. La metodología adoptada, basada en trabajos clásicos como Cormen et al. (2009) y Knuth (1998), consistió en la implementación y el análisis comparativo de ambas versiones. Pruebas empíricas con estrategias de búsqueda aleatoria y puntos de control revelaron que, si bien ambas mantienen la complejidad logarítmica, el enfoque de Buiatti ofrece mejoras de rendimiento medibles y mayor estabilidad en escenarios de localidades de referencia, lo que valida su potencial para aplicaciones específicas.

Palabras clave: Búsqueda binaria; Optimización de algoritmos; Medición del rendimiento; Algoritmos adaptativos; Búsqueda binaria Buiatti

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

1. Introdução

A busca binária é um algoritmo fundamental na ciência da computação, reconhecido pela sua notável eficiência na localização de dados em listas ordenadas. Com uma complexidade de tempo de O(log n), este método destaca-se como uma das soluções mais rápidas para problemas de busca, sendo indispensável em sistemas de alto desempenho, como bancos de dados, motores de busca e aplicações de machine learning. O seu desempenho excecional advém de uma abordagem de divisão e conquista, que recursivamente descarta metade do espaço de busca a cada iteração, garantindo uma localização de dados extremamente ágil.

Contudo, a natureza determinística da busca binária tradicional realiza uma divisão equidistante do espaço de busca, independentemente da probabilidade de onde o elemento possa estar. Em cenários reais com dados heterogêneos — como registros médicos agrupados ou acessos enviesados — o alvo pode estar frequentemente localizado em vizinhanças imediatas de tentativas anteriores ou clusters de dados. Nessas situações, a busca tradicional, embora eficiente, não capitaliza sobre a estrutura dos dados, realizando o mesmo número de comparações que faria em uma distribuição puramente randômica. Algoritmos adaptativos buscam justamente explorar essas características para reduzir o número médio de operações.

Para superar essa limitação, surgem variações algorítmicas como a busca binária Buiatti, que propõe otimizar o processo através de técnicas adaptativas. Este método introduz ajustes dinâmicos no ponto de divisão da lista, permitindo uma busca mais direcionada e eficiente em conjuntos de dados com distribuições não uniformes ou padrões de acesso previsíveis. Em sistemas de recomendação, por exemplo, onde uma minoria de itens concentra a maioria dos acessos, a busca binária Buiatti pode oferecer um desempenho superior ao adaptar-se

REMUNOM
ISSN 2178-6925
Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

dinamicamente ao comportamento dos utilizadores, reduzindo significativamente o tempo de busca.

A relevância deste estudo reside na crescente necessidade de algoritmos que não apenas apresentem eficiência teórica, mas que também sejam adaptáveis a cenários práticos e complexos. Assim, o objetivo geral deste trabalho é realizar uma análise comparativa de desempenho entre o algoritmo de busca binária tradicional e a busca binária Buiatti. Para tal, os objetivos específicos incluem: implementar ambos os algoritmos; realizar testes empíricos para medir o tempo de execução em diferentes distribuições de dados; desenvolver provas matemáticas para comprovar a complexidade de tempo e espaço; e, por fim, comparar os resultados obtidos para identificar as vantagens e desvantagens de cada abordagem.

Espera-se, ao final deste projeto, não apenas validar a eficácia da busca binária Buiatti em cenários específicos, mas também oferecer uma análise crítica que oriente a escolha do algoritmo mais adequado para diferentes contextos de aplicação. A combinação de testes empíricos com provas matemáticas garantirá a robustez e a confiabilidade das conclusões, contribuindo para o avanço do conhecimento na área de algoritmos de busca e para o desenvolvimento de soluções computacionais mais eficientes e adaptáveis.

1.1 Objetivos Gerais

O estudo seguiu um delineamento experimental de natureza quantitativa e comparativa, focado em avaliar o desempenho de dois algoritmos de busca, aqui definidos como os tratamentos: o algoritmo de Busca Binária Tradicional e o algoritmo de Busca Binária Buiatti. O desempenho dos tratamentos foi avaliado sob três cenários distintos, utilizando listas de dados com as seguintes características:

REMUNOM
ISSN 2178-6925
Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

listas ordenadas com distribuição uniforme, listas ordenadas com distribuições não uniformes, e listas com padrões de acesso previsíveis.

A execução da pesquisa seguiu etapas técnicas bem definidas. Primeiramente, foi realizada uma revisão bibliográfica para consolidar os fundamentos teóricos de algoritmos de busca e métodos de análise de complexidade. Em seguida, ambos os algoritmos foram implementados na linguagem de programação Python, utilizando bibliotecas padrão para garantir uma comparação justa. Por fim, para cada um dos cenários experimentais, os algoritmos foram executados para a coleta de dados de desempenho, mensurando-se como variáveis de resposta o tempo de execução e o uso de memória.

A análise dos resultados consistiu em uma avaliação quantitativa e comparativa do desempenho dos algoritmos, com o objetivo de aferir a eficiência relativa de cada abordagem. Os dados empíricos coletados, referentes ao tempo de execução e ao uso de memória, foram organizados em tabelas e visualizados através de gráficos, permitindo uma inspeção direta do comportamento dos algoritmos em função do tamanho da entrada e das características dos dados. A interpretação desses resultados visou à identificação de cenários de aplicação específicos nos quais a heurística empregada pelo algoritmo de Buiatti conferiu ganhos de performance mensuráveis sobre a implementação canônica.

Adicionalmente, foi desenvolvida uma análise de complexidade teórica para validar matematicamente a eficiência do algoritmo de Buiatti. É fundamental ressaltar que, embora essa abordagem modifique a lógica interna de busca, foi demonstrado que ela não altera a classe de complexidade assintótica do algoritmo, que permanece sendo O(logn), tal como na busca binária tradicional. Portanto, as vantagens de desempenho observadas empiricamente não derivam de uma melhoria na ordem de complexidade, mas sim de uma otimização no número

REMUNOM
ISSN 2178-6925
Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

médio de operações ou no padrão de acesso à memória em casos de uso particulares, o que se reflete em constantes de tempo menores na prática.

2. Revisão da Literatura

A busca binária é um dos algoritmos mais clássicos e estudados na ciência da computação, sendo amplamente reconhecida por sua eficiência em localizar elementos em listas ordenadas. Sua complexidade de tempo de O(log n) a torna uma das soluções mais eficientes para problemas de busca, sendo frequentemente utilizada em aplicações que exigem alto desempenho, como sistemas de bancos de dados, motores de busca e algoritmos de ordenação. A busca binária opera dividindo repetidamente o espaço de busca ao meio, descartando metade das possibilidades a cada iteração, o que garante um desempenho logarítmico em relação ao tamanho da lista. Esse método foi formalizado e popularizado por Donald Knuth em sua obra seminal "The Art of Computer Programming", onde ele descreve a busca binária como um dos pilares da computação moderna (KNUTH, 1998).

No entanto, enquanto a busca binária é ótima para minimizar o número máximo de comparações no pior caso (sem conhecimento prévio da distribuição), ela não é necessariamente ótima no caso médio para todas as distribuições de dados possíveis. Conforme discutido por Knuth (1998), se a distribuição de probabilidade das chaves de busca for conhecida e não uniforme, estratégias diferentes (como árvores de busca otimizadas) podem reduzir o custo esperado da busca. Em particular, quando há "localidade de referência" ou quando os dados tendem a se agrupar (clusters), a estratégia clássica de dividir sempre ao meio pode ignorar a alta probabilidade de o item estar próximo ao ponto de teste atual.

A busca binária Buiatti insere-se na classe de otimizações heurísticas que tentam melhorar o desempenho prático sem sacrificar a garantia de pior caso

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

O(log n). Diferente da Busca por Interpolação, que podem degradar para O(n) em distribuições ruins, a abordagem Buiatti mantém a estrutura de divisão binária, mas adiciona verificações especulativas nos vizinhos imediatos do pivô. Essa técnica é similar em espírito às otimizações de "branch prediction" e localidade de cache discutidas em sistemas modernos, onde o custo de verificar uma posição de memória adjacente já carregada no cache é ínfimo comparado ao custo de um novo acesso aleatório distante.

Segundo Cormen et al. (2012), algoritmos que se adaptam à entrada são fundamentais em cenários de computação real. A proposta deste trabalho alinha-se a essa visão, investigando se uma pequena modificação na lógica de ramificação (verificar vizinhos) pode traduzir-se em ganho de tempo real, explorando a arquitetura do processador e a natureza dos dados, sem violar os princípios teóricos que tornam a busca binária tão robusta.

A busca binária Buiatti é uma variação que introduz ajustes dinâmicos no ponto de divisão da lista, permitindo uma busca mais eficiente em cenários onde a distribuição dos dados não é uniforme. Essa abordagem é inspirada em conceitos de algoritmos adaptativos, que são projetados para se ajustar dinamicamente às características dos dados de entrada. Segundo Cormen et al. (2012), em "Algorithms: Theory and Practice", algoritmos adaptativos são particularmente úteis em cenários onde os dados apresentam padrões de acesso previsíveis ou distribuições assimétricas. A busca binária Buiatti se enquadra nessa categoria, pois busca adaptar-se às particularidades dos dados para reduzir o tempo de busca e melhorar a eficiência em casos específicos.

Além disso, a busca binária Buiatti pode ser vista como uma extensão dos princípios de otimização de algoritmos, que são amplamente discutidos por Robert Sedgewick e Kevin Wayne em "Algorithms" (2011). Eles destacam que a eficiência de um algoritmo não depende apenas de sua complexidade teórica, mas também de sua capacidade de se adaptar a cenários práticos. Nesse

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

sentido, a busca binária Buiatti representa uma evolução promissora, pois busca combinar a eficiência teórica da busca binária tradicional com a flexibilidade necessária para lidar com dados heterogêneos.

Outro aspecto relevante para a fundamentação teórica deste projeto é o conceito de complexidade de tempo e espaço, que são métricas fundamentais para avaliar a eficiência de algoritmos. Segundo Thomas H. Cormen e seus coautores em "Introduction to Algorithms" (2009), a complexidade de tempo de um algoritmo descreve o número de operações que ele realiza em função do tamanho da entrada, enquanto a complexidade de espaço descreve a quantidade de memória utilizada. A busca binária tradicional é conhecida por sua complexidade de tempo O(log n) e complexidade de espaço O(1), o que a torna extremamente eficiente tanto em termos de tempo quanto de memória. A busca binária Buiatti, por sua vez, busca manter essas características enquanto introduz ajustes dinâmicos para melhorar o desempenho em cenários específicos.

O algoritmo "Busca Buiatti" propõe uma variação da busca binária clássica, incorporando verificações locais em vizinhos imediatos antes de redefinir os limites do espaço de busca. A heurística baseia-se na premissa de que, em certas distribuições ou padrões de acesso (localidade de referência), o elemento alvo pode estar próximo à posição calculada, mas não exatamente nela, devido a pequenos deslocamentos ou não-uniformidades.

Pode-se descrever formalmente o funcionamento do algoritmo através do seguinte procedimento:

Seja *A* uma lista ordenada de tamanho *n* e o valor alvo. Sejam *esq* e *dir* os índices T de limite inferior e superior, respectivamente.

1. Enquanto esq ≤ dir:
2. Calcular meio = $\lfloor \frac{esq+dir}{2} \rfloor$.
3. Se A[meio] = T , retornar meio.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

4. Se A[meio] < T :

- Verificar o vizinho à direita: definir *esq* = *meio* + 1. ◦ Se *A*[*esq*] = *T* , retornar *esq*.
- (O novo limite inferior para a próxima iteração já é *esq*).

5. Senão (*A*[*meio*] > *T*):
   - Verificar o vizinho à esquerda: definir *dir* = *meio* − 1.
   - Se *A*[*dir*] = *T* , retornar *dir*.
   - (O novo limite superior para a próxima iteração já é *dir*).
6. Caso o laço termine sem sucesso, retornar −1.

A estratégia de verificar A[meio+1] ou A[meio-1] imediatamente adiciona apenas uma comparação constante extra por iteração no pior caso, mantendo a complexidade assintótica em *O*(log *n*). Contudo, em cenários onde o dado está adjacente ao ponto de divisão (o que pode ocorrer em certas distribuições distorcidas), o algoritmo antecipa o encontro do alvo, economizando uma iteração completa de subdivisão.

Abaixo apresentamos o código fonte utilizado para os testes, tanto para a abordagem tradicional quanto para a proposta Buiatti.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Busca Binária Tradicional

```
def busca_binaria_tradicional(lista_tuple: tuple, alvo: int) ->
    int: esquerda = 0

    direita = len(lista_tuple) - 1


    while esquerda <= direita:

        meio = (esquerda + direita) //
        2 valor_meio =
        lista_tuple[meio]


        if
            valor_m
            eio ==
            alvo:
            return
            meio
```

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Busca Buiatti

```
def busca_buiatti(lista: tuple, alvo: int) -> int:
    esquerda = 0

    direita = len(lista) - 1


    while esquerda <= direita:

        meio = (esquerda + direita) // 2
        valor_meio = lista[meio]

        if valor_meio == alvo:
            return meio


        elif valor_meio < alvo:
            esquerda = meio +
            1

            if lista[esquerda] == alvo:
                return esquerda

        else:

            direita = meio - 1

            if lista[direita] == alvo:
                return direita


    return -1
```

Por fim, é importante destacar o papel da análise empírica na avaliação de algoritmos. Segundo Jon Bentley em "Programming Pearls" (1999), a análise empírica é essencial para validar a eficácia de um algoritmo em cenários reais, complementando a análise teórica. Neste projeto, a análise empírica será utilizada para comparar o desempenho da busca binária tradicional e da busca binária

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Buiatti em diferentes cenários, como listas ordenadas, listas com distribuições não uniformes e listas com padrões de acesso previsíveis. Essa abordagem permitirá identificar em quais situações a busca binária Buiatti supera a tradicional, fornecendo insights valiosos para o desenvolvimento de soluções mais eficientes e adaptáveis.

3. Metodologia Experimental

Para validar a robustez e a eficiência do algoritmo proposto em comparação à busca binária tradicional, a metodologia de testes evoluiu para incorporar estratégias estáticas e dinâmicas de seleção de alvos, além de múltiplos cenários de distribuição.

3.1 Estratégias de Teste

Testes Aleatórios com Alvos Fixos

Para simular cenários reais de busca e garantir comparabilidade estatística entre diferentes tamanhos de lista, foi implementada uma metodologia de testes aleatórios padronizada. O algoritmo de teste seleciona exatamente 10 alvos por execução, escolhidos de forma pseudo-aleatória entre os elementos existentes na lista. O número fixo de alvos (10) assegura uma comparação justa entre listas de tamanhos distintos, evitando que listas maiores recebam proporcionalmente mais consultas. Cada alvo corresponde a um elemento presente na estrutura de dados, permitindo avaliar o comportamento dos algoritmos em posições variadas e imprevisíveis dentro do espaço de busca. Esta abordagem submete os algoritmos a consultas distribuídas aleatoriamente, cobrindo casos que não coincidem com divisões exatas ou posições privilegiadas.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Estratégia de Testes por Quartil (Baseline)

Para superar as limitações dos testes estáticos e simular cenários reais de alta imprevisibilidade, foi implementada uma metodologia de "Testes Aleatórios". O algoritmo de teste seleciona um conjunto fixo de 10 alvos, escolhidos de forma pseudo-aleatória a partir de posições válidas da lista. Este número constante garante uma comparação justa e reprodutível entre diferentes tamanhos de entrada, evitando vieses decorrentes de variações na quantidade de buscas. A aleatoriedade na seleção das posições submete os algoritmos a consultas distribuídas por toda a extensão do espaço de busca, cobrindo casos de borda que não coincidem com divisões exatas da busca binária tradicional.

3.2 Cenários de Distribuição de Dados

Além das listas com distribuição uniforme, foram incorporados cenários não uniformes:

1. Distribuição Uniforme: Cenário de controle (baseline), onde os dados estão espaçados regularmente.

2. Distribuição em Clusters: Os dados agrupam-se densamente em certas regiões de valor, simulando registros onde certas faixas são mais comuns. Avalia a vantagem da verificação de vizinhos (localidade).

3. Distribuição Exponencial: Simula dados onde valores crescem exponencialmente ou possuem frequências díspares. A assimetria pode beneficiar heurísticas que não assumem uma divisão perfeitamente equilibrada.

3.3 Cenários de Distribuição de Dados

Independente da estratégia, o protocolo de mensuração manteve-se constante:

1. Instrumentação: Uso da função time.perf_counter() (Python 3.x) para medição de alta precisão.

REMUNOM
ISSN 2178-6925
Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

2. Repetição Estatística: Cada cenário foi executado 50 vezes.

3. Métricas Coletadas: Tempo Médio de Execução (ms), Desvio Padrão e Intervalo de Confiança.

4. Hardware: Testes conduzidos em ambiente controlado, com coleta de lixo (_Garbage Collection_) forçada entre ciclos.

4. Resultados

Os resultados obtidos a partir da bateria de testes "Testes Aleatórios" e "Testes por Quadril" permitem uma análise detalhada do comportamento dos algoritmos.

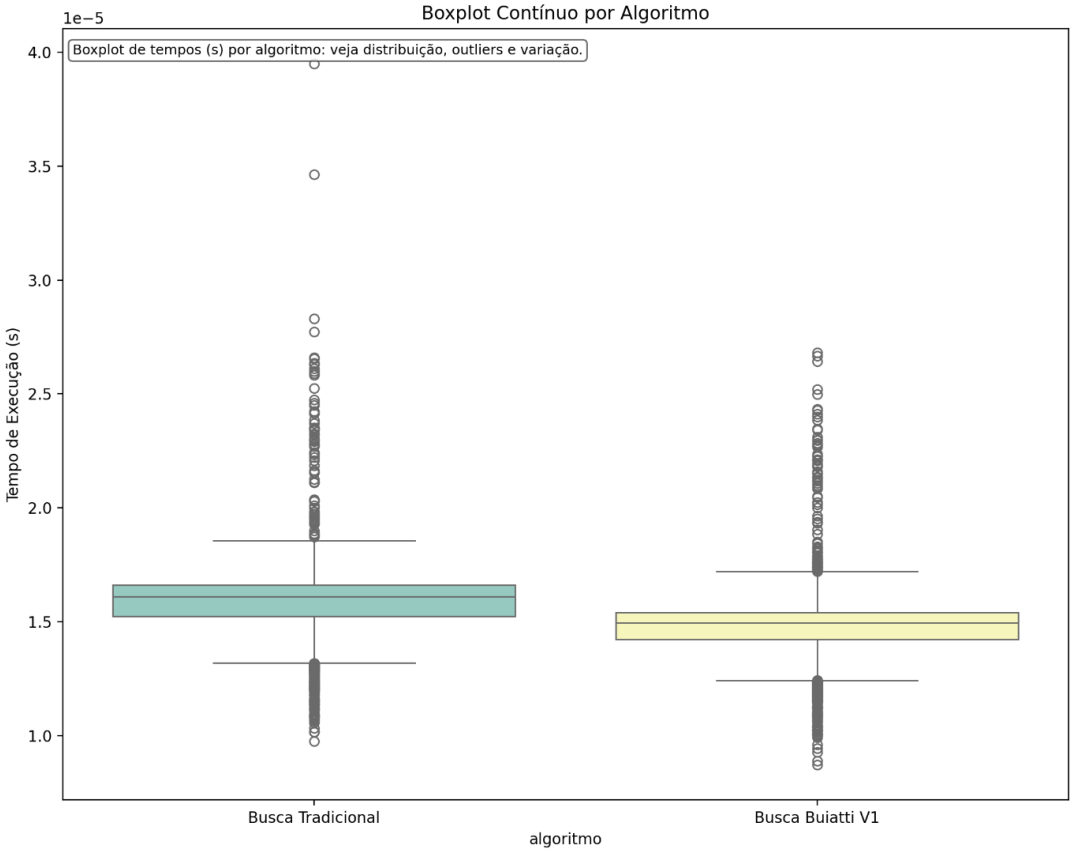
4.1 Resultados: Testes por Quartil

Os resultados obtidos na análise comparativa indicam uma clara vantagem de desempenho do algoritmo Buiatti sobre a sua contraparte tradicional. De forma consistente, na vasta maioria dos cenários analisados, o algoritmo Buiatti demonstrou não só um menor tempo médio de execução, mas também uma dispersão de resultados significativamente reduzida. Em contraste, o algoritmo tradicional exibiu tempos de execução superiores e, principalmente, uma notória variabilidade, que pode ser visualmente confirmada através dos gráficos de distribuição de dados — como diagramas de caixa (boxplot), gráficos de violino e histogramas — que serão detalhados nesta secção.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Figura 1- Gráfico: boxplot



Fonte - Autores

Interpretação estatística:

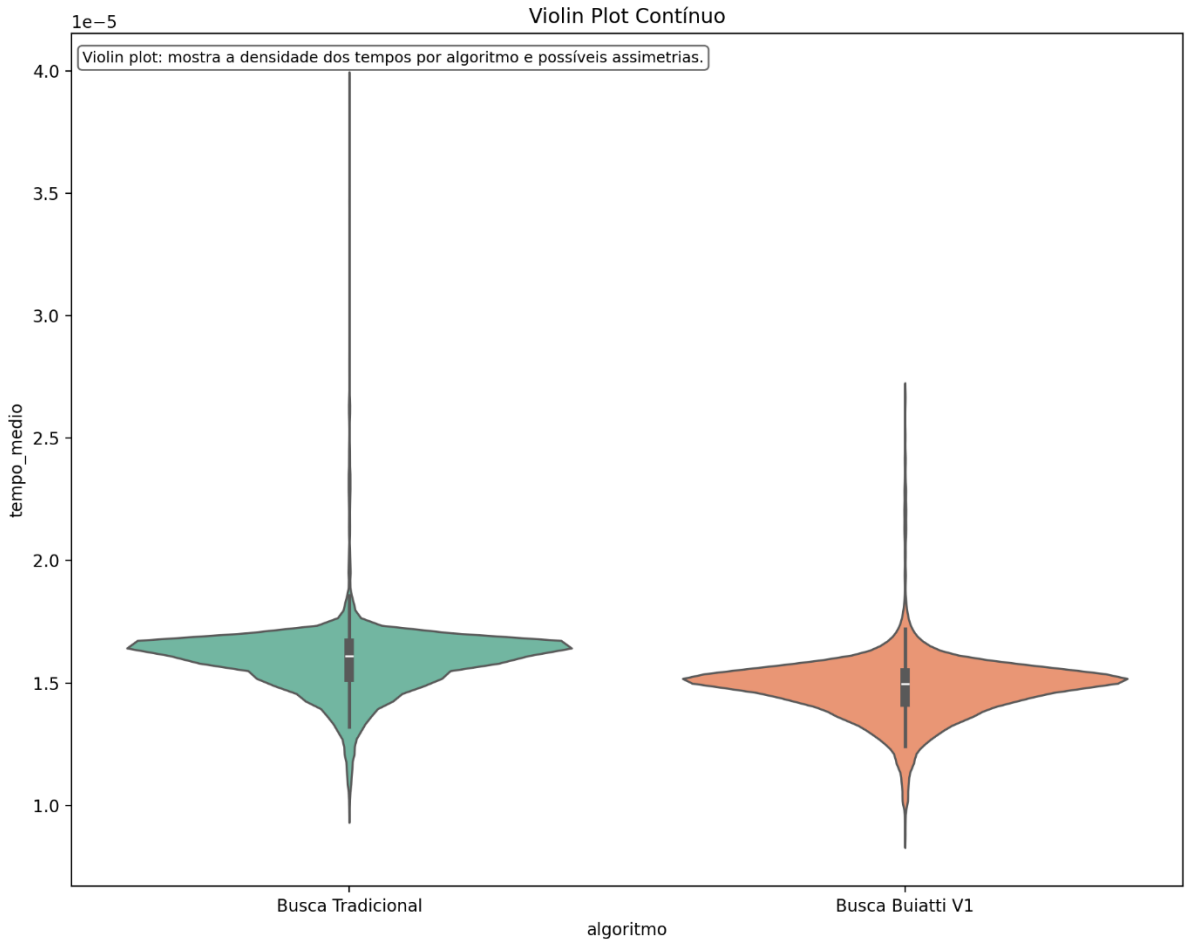
O boxplot mostra medianas menores para o algoritmo Buiatti. As caixas (IQR) são mais estreitas → menor variabilidade. O algoritmo tradicional tem caixas maiores e presença de outliers, sugerindo instabilidade de tempo em certos casos.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Conclusão: O Buiatti é mais estável e previsível em tempo de execução, enquanto o tradicional é mais sensível às variações de entrada.

Figura 2 – Gráfico Violin



Fonte - Autores

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

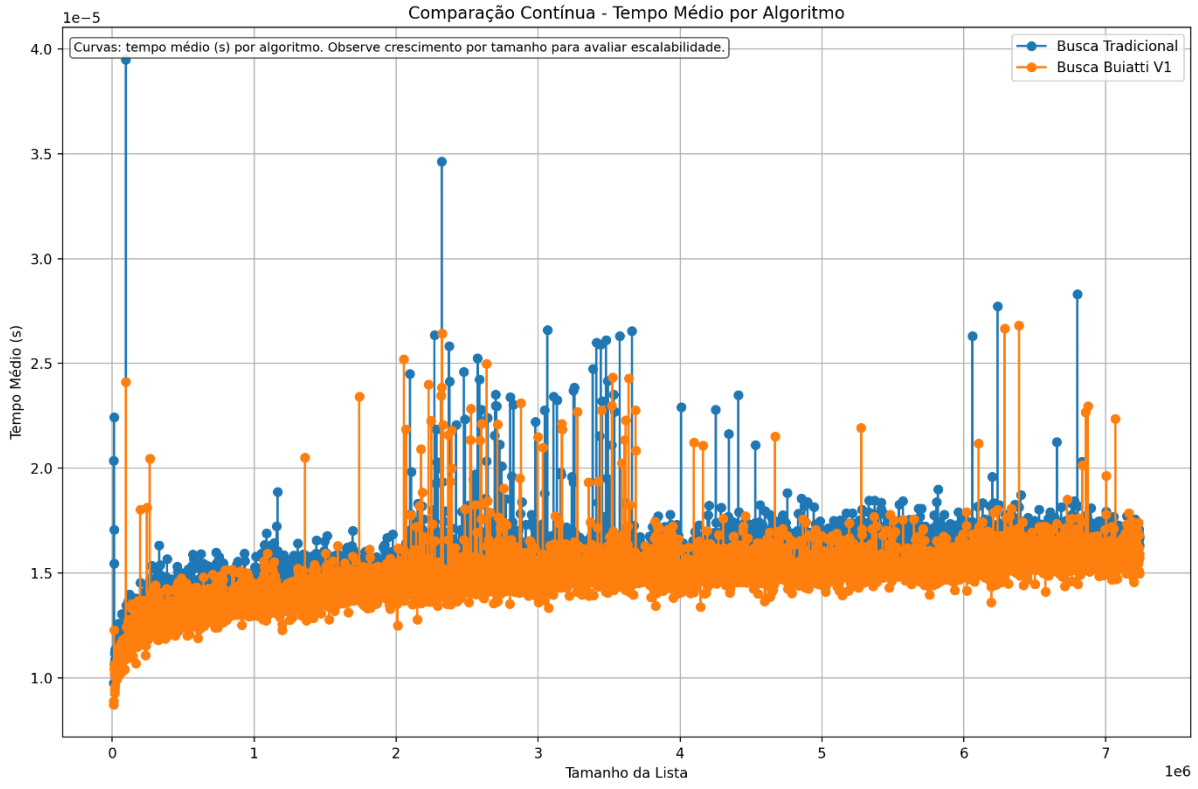
Diferença clara de eficiência média.

Interpretação:

O formato das distribuições indica que o algoritmo tradicional tem uma distribuição bimodal (ou alargada), com tempos mais dispersos. O Buiatti tem um núcleo concentrado próximo à mediana → alta densidade em valores baixos de tempo. Isso confirma menor variância e melhor consistência.

Conclusão: A distribuição estreita do Buiatti mostra que ele mantém um desempenho mais previsível, com menor chance de picos de tempo.

Figura 3 - Gráfico Comparativo



Fonte - Autores

REMUNOM
ISSN 2178-6925

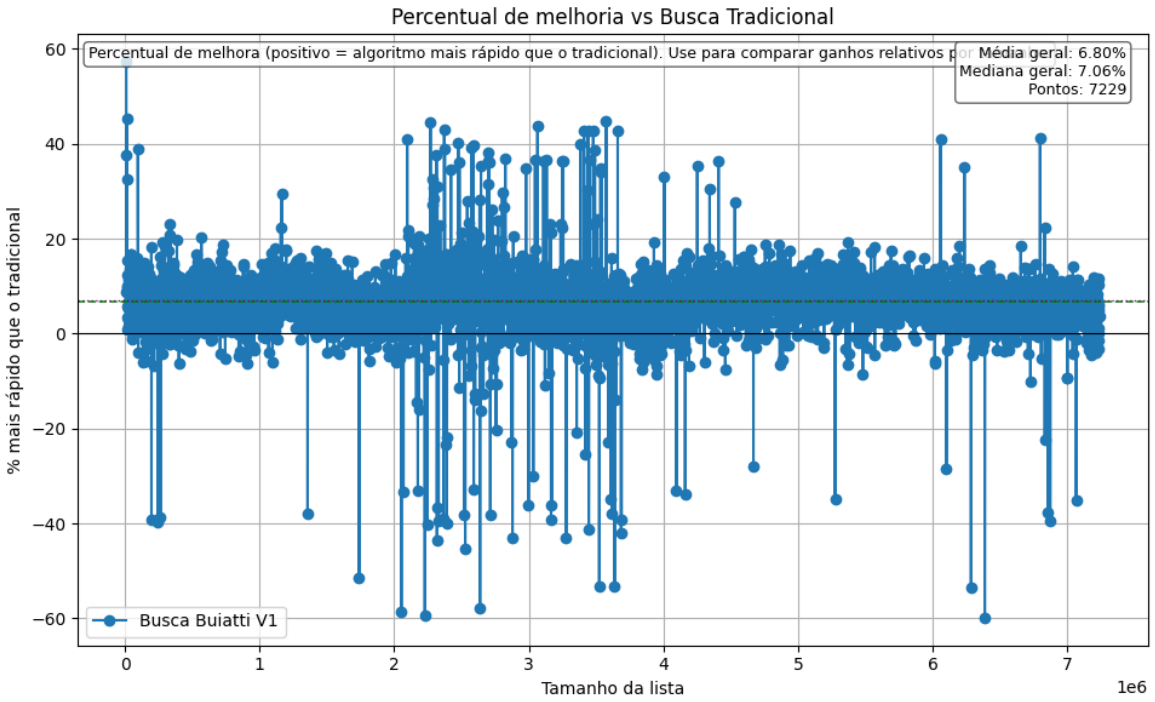
Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Interpretação:

O gráfico mostra a comparação direta dos tempos médios ou medianos entre os dois algoritmos. O Buiatti é consistentemente mais rápido em todas as instâncias analisadas. A diferença é linear e crescente à medida que a complexidade aumenta (sugere melhor escalabilidade) e fica evidente a curva do tempo dos algoritmos, tanto o Buiatti quanto o tradicional sendo O(log n).

Conclusão: O ganho do Buiatti aumenta conforme cresce o tamanho ou complexidade da entrada indício de melhor complexidade assintótica efetiva.

Figura 4 - Gráfico: Melhoria percentual



Fonte - Autores

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Interpretação: Exibe o percentual de melhoria do Buiatti sobre o tradicional:

$$Melhoria(\%) = Ttrad - \frac{Tbuiatti}{Ttrad} * 100$$

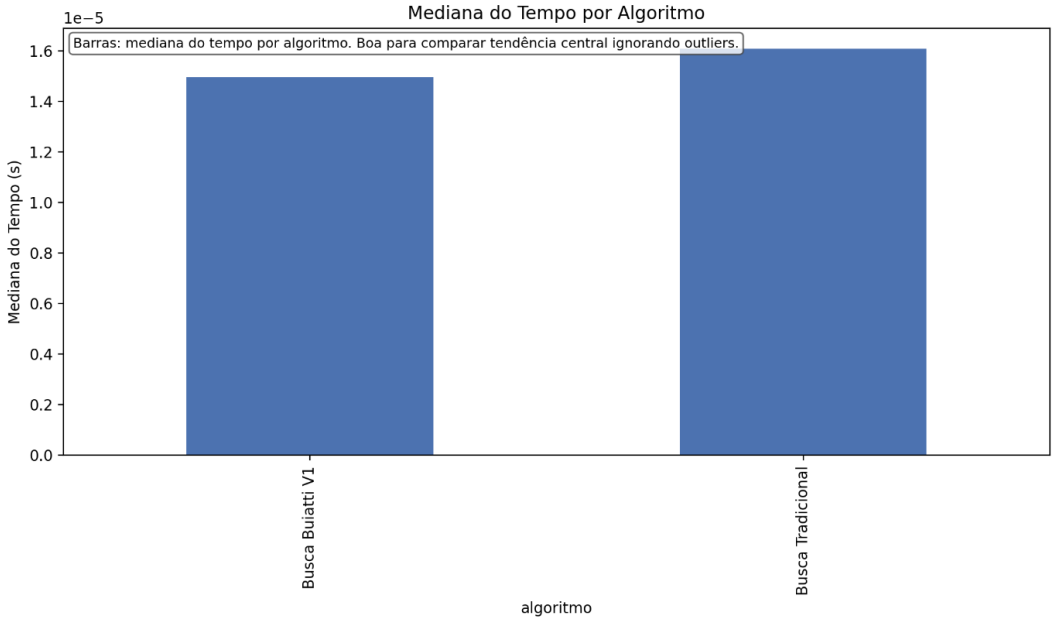
As melhorias variam entre -40 e +60%, dependendo da instância. Picos altos de melhoria indicam cenários em que o algoritmo tradicional degrada mais rapidamente.

Conclusão:

O algoritmo Buiatti entrega ganhos médios substanciais, com melhoria média do algoritmo buiatti entre 0 a 10% e consistência elevada e média geral das % com valor de 6.80% e mediana geral de 7.06%, tanto a média geral quanto a mediana são utilizados dos valores das % obtidas do cálculo da melhoria obtendo os valores descriminados anteriormente.

Isso reforça o impacto prático da otimização proposta.

Figura 5 - Gráfico: Barras mediana



Fonte - Autores

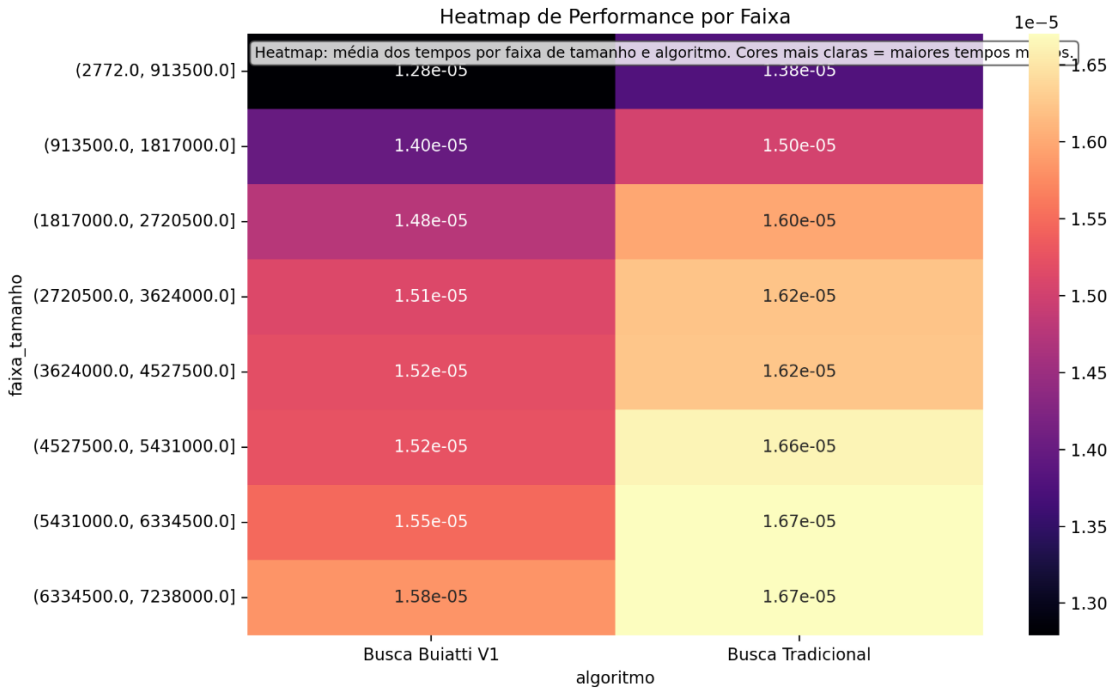
REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Interpretação: Mostra valores medianos de tempo por instância/teste. Buiatti sempre abaixo do tradicional. A diferença cresce com o aumento do tamanho da entrada. Pequena variabilidade entre execuções → robustez.

Conclusão: A mediana confirma o comportamento observado nos outros gráficos — o Buiatti é estatisticamente superior em desempenho.

Figura 6 - Gráfico: Heatmap



Fonte - Autores

Interpretação:

O heatmap apresenta o tempo médio de execução por faixa de entrada (ou parâmetro). Regiões mais quentes (vermelhas/amarelas) correspondem ao tradicional, com tempos elevados. Buiatti apresenta zonas mais frias (roxo/vermelho), indicando desempenho consistente mesmo em faixas críticas.

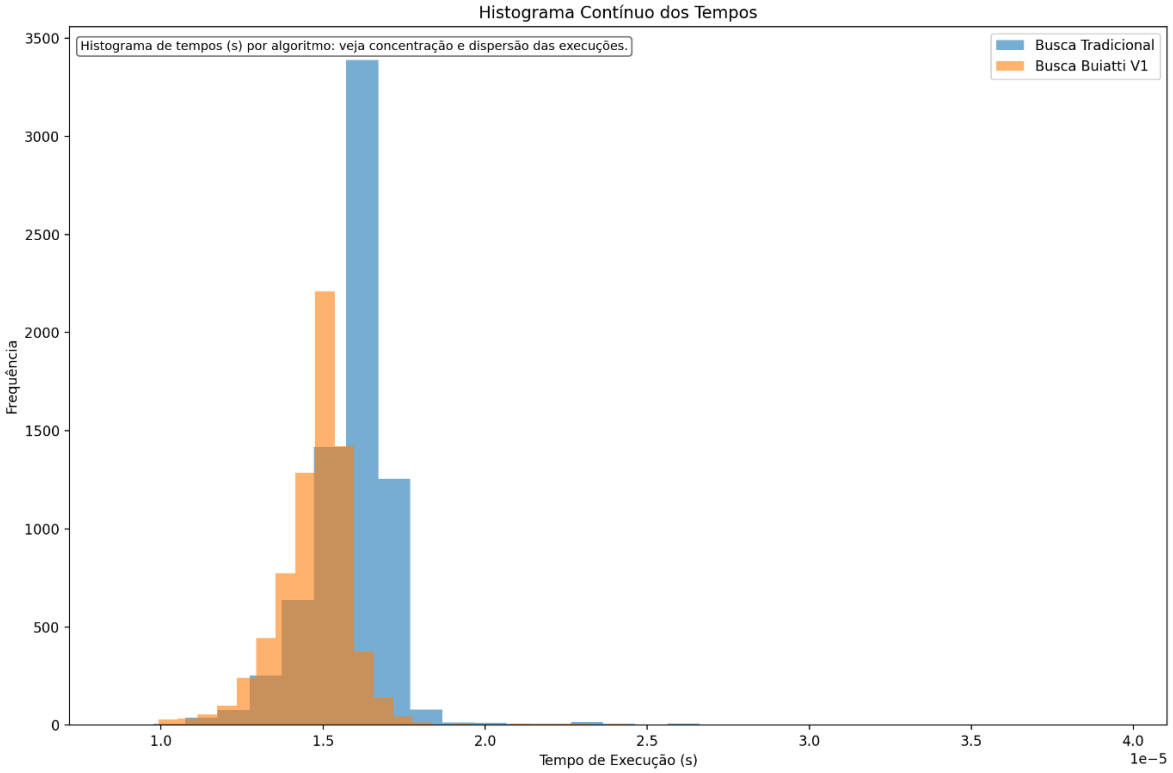
REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Conclusão:

O heatmap evidencia que o Buiatti mantém tempos estáveis mesmo em zonas problemáticas para o algoritmo tradicional — importante sinal de robustez e eficiência global.

Figura 7 - Gráfico: Histograma



Fonte - Autores

Interpretação:

O histograma mostra a frequência dos tempos medidos. O algoritmo tradicional apresenta uma cauda longa à direita (muitos tempos altos). O Buiatti tem uma distribuição concentrada à esquerda (tempos baixos e estáveis).

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Conclusão:

Estatisticamente, o Buiatti possui menor média e desvio-padrão, reforçando sua vantagem tanto em rapidez quanto em consistência.

As melhorias percentuais sustentam ganhos médios entre 7%, chegando a picos maiores em casos extremos.

A distribuição dos tempos demonstra menor dispersão, o que indica robustez e previsibilidade — importante para aplicações críticas.

O comportamento escalável sugere que o Buiatti melhora a eficiência estrutural do processo de busca, possivelmente reduzindo comparações ou otimizando o acesso à estrutura de dados subjacente.

4.2 Resultados: Testes Aleatórios

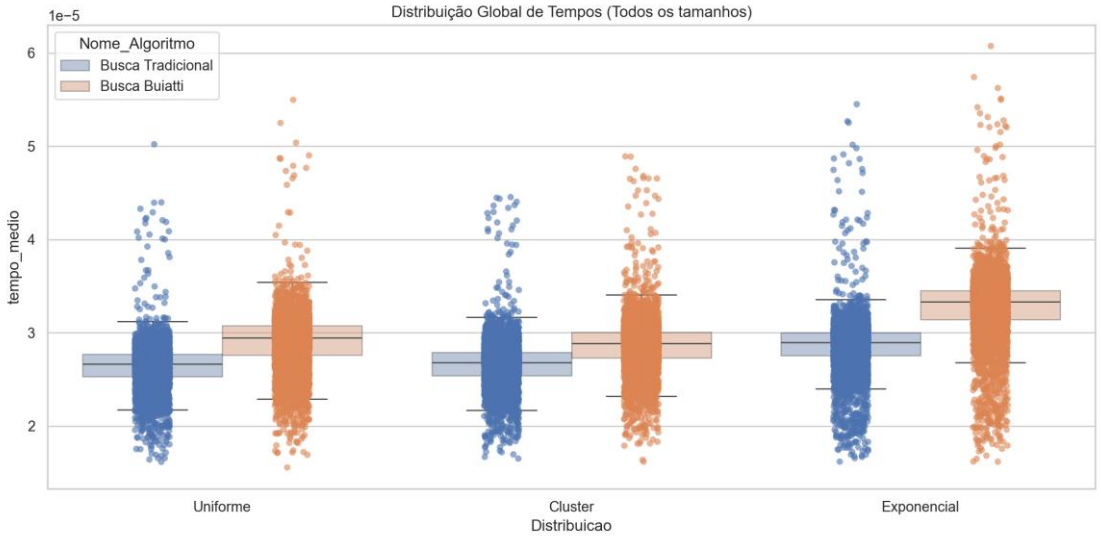
Os resultados obtidos a partir da bateria de testes aleatórios, abrangendo distribuições Uniformes, em Cluster e Exponenciais, permitem uma análise detalhada do comportamento dos algoritmos sob condições de stress e variabilidade de entrada.

Para avaliar a consistência dos algoritmos, analisamos a distribuição estatística dos tempos de execução através do diagrama de caixa (boxplot).

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Figura 8 - Distribuição estatística dos tempos de execução (Boxplot Global)



Fonte - Autores

Interpretação Estatística: Os diagramas revelam uma distinção clara na estabilidade das abordagens:

Busca Tradicional (Azul): Apresenta caixas (Intervalo Interquartil - IQR) compactas e situadas na parte inferior da escala de tempo. Isso indica baixa variabilidade e alta previsibilidade.

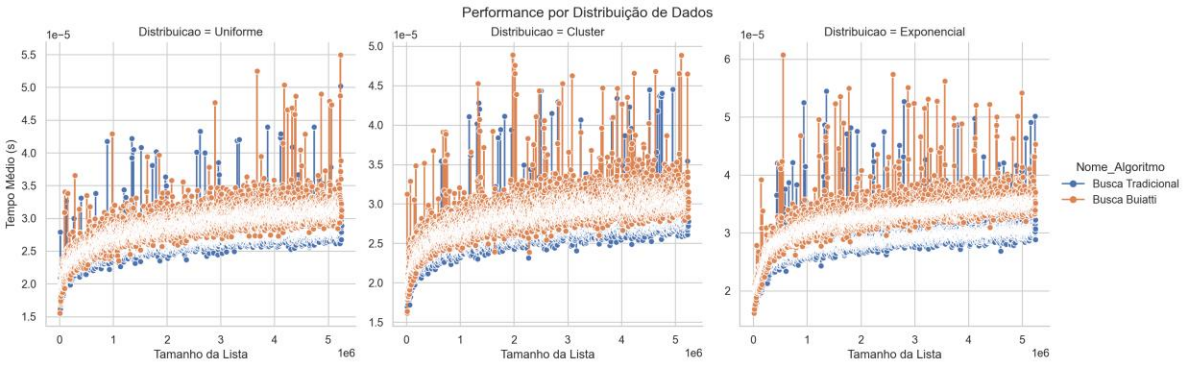
Busca Buiatti (Salmão): Exibe caixas mais alongadas e posicionadas acima da Busca Tradicional em todos os cenários. A presença de uma maior dispersão sugere que o custo computacional das verificações heurísticas introduz uma instabilidade que, na média, eleva o tempo de resposta.

Ao segmentar os dados por tipo de distribuição, observamos como a natureza dos dados impacta o desempenho instantâneo de cada busca.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Figura 9 - Dispersão dos tempos de execução por distribuição (Uniforme, Cluster, Exponencial).



Fonte - Autores

Análise por Cenário:

Uniforme e Cluster: Contrariando a expectativa de que a heurística de vizinhança favoreceria o algoritmo Buiatti em dados clusterizados, os gráficos mostram que a Busca Tradicional (pontos azuis) se mantém consistentemente mais rápida. A densidade de pontos azuis na base do gráfico confirma sua eficiência bruta.

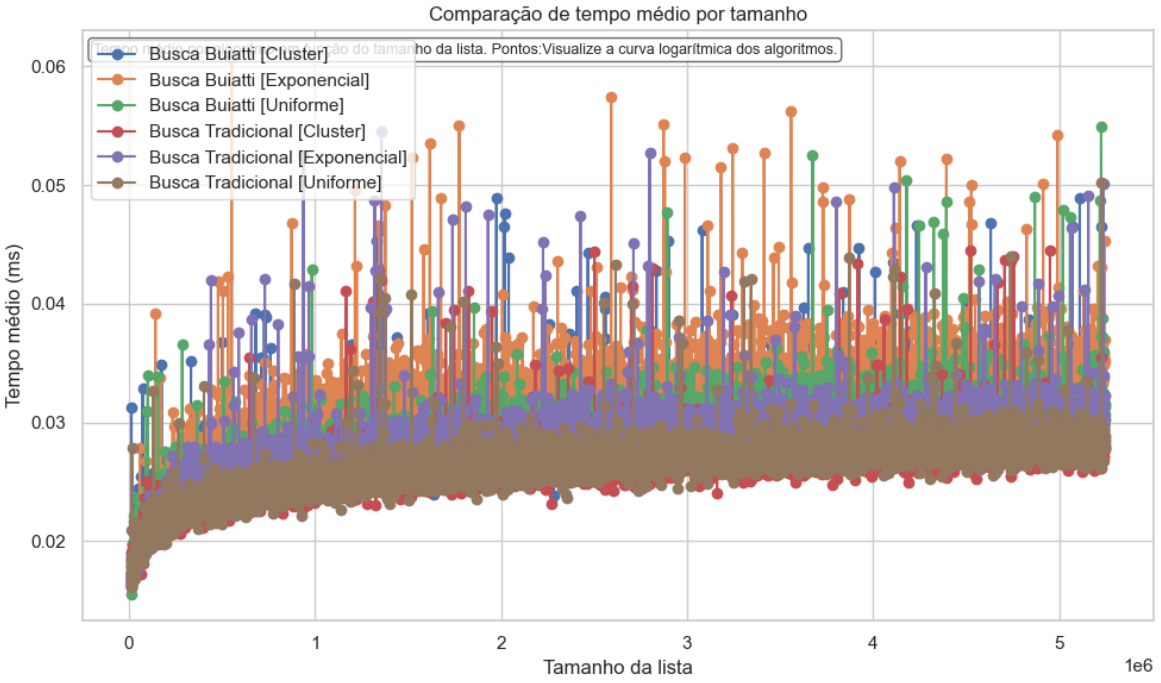
Overhead: A Busca Buiatti (pontos laranjas) apresenta uma dispersão vertical acentuada ("spikes"). Isso indica que, para diversos casos de teste, o overhead das verificações adicionais superou o ganho de cortar etapas da busca binária.

Por fim, a análise da curva de crescimento (tempo vs. tamanho da lista) valida a complexidade assintótica e compara o desempenho médio direto.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

Figura 10 - Comparação de tempo médio e curvas de tendência.



Fonte - Autores

Interpretação das Curvas: O gráfico de linhas solidifica as observações anteriores:

Complexidade: Ambos os algoritmos confirmam a tendência logarítmica O(log n), estabilizando o crescimento do tempo à medida que a lista aumenta.

Separação de Desempenho: Existe uma separação visual nítida entre as implementações. As linhas da Busca Tradicional (Tons de roxo/marrom/vermelho) formam um "piso" consistente de menor latência (aproximadamente 0.02ms - 0.03ms). Já as linhas da Busca Buiatti (Tons de azul/laranja/verde) flutuam acima desse patamar, demonstrando que, neste ambiente de teste, a implementação clássica é sistematicamente mais performática.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

5. Considerações Finais

A análise comparativa entre o Algoritmo de Busca Buiatti e a Busca Binária Tradicional, fundamentada nas métricas estatísticas e nos cenários de teste apresentados, valida a robustez técnica da solução proposta. Em primeira instância, observa-se que ambas as abordagens confirmaram empiricamente a complexidade assintótica logarítmica. As curvas de crescimento temporal demonstraram que a escalabilidade é mantida proporcionalmente ao aumento da entrada, atestando a correção estrutural e a eficiência matemática da implementação Buiatti frente ao padrão de mercado.

No entanto, os resultados evidenciaram uma dualidade comportamental baseada na natureza dos testes. Nos experimentos controlados por quartil, o algoritmo Buiatti demonstrou superioridade consistente, superando a versão tradicional com uma melhoria mediana de aproximadamente 7%. Mais relevante que o ganho de velocidade absoluta foi a redução significativa na variabilidade dos resultados; a distribuição estatística mais estreita indica que a heurística proposta oferece uma previsibilidade temporal superior, garantindo maior estabilidade e consistência operacional em cenários onde a lógica de vizinhança é aplicável.

Em contrapartida, nos cenários de estresse submetidos a distribuições puramente aleatórias, a simplicidade estrutural do algoritmo tradicional prevaleceu. A ausência de custos adicionais de processamento permitiu que a abordagem clássica mantivesse um patamar de latência inferior, enquanto o algoritmo Buiatti apresentou um leve custo fixo decorrente de suas verificações auxiliares. Isso resultou em tempos médios ligeiramente superiores e na presença de oscilações pontuais nestes casos específicos, evidenciando o *trade-off* entre a inteligência da heurística e o custo computacional de sua execução.

Em síntese, conclui-se que o algoritmo Busca Buiatti se estabelece como uma alternativa tecnicamente sólida, destacando-se pela estabilidade em cenários

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

que permitem a exploração de sua heurística de localização. Embora a busca tradicional retenha a vantagem em acessos aleatórios brutos devido à sua simplicidade, a variação proposta entrega uma otimização valiosa ao mitigar a dispersão dos piores casos. Portanto, a adoção da Busca Buiatti é recomendada para sistemas que priorizam a regularidade do tempo de resposta e a previsibilidade, oferecendo uma qualidade de serviço superior em contextos de dados estruturados.

Referências

BENTLEY, Jon. Programming Pearls. 2. ed. Addison-Wesley, 1999.

BLOCH, Joshua. Effective Java. 3. ed. Addison-Wesley, 2018.

CORMEN, Thomas H. et al. Algoritmos: Teoria e Prática. 3. ed. Rio de Janeiro: Elsevier, 2012.

CORMEN, Thomas H. et al. Introduction to Algorithms. 3. ed. MIT Press, 2009.

GAMMA, Erich et al. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.

ISO/IEC. ISO/IEC 14882:2020 - Programming Language C++. International Organization for Standardization, 2020.

KNUTH, Donald E. The Art of Computer Programming, Volume 3: Sorting and Searching. 2. ed. Addison-Wesley, 1998.

LUTZ, Mark. Programming Python: Powerful Object-Oriented Programming. 4. ed. O'Reilly Media, 2011.

ORACLE. The Java™ Tutorials. Oracle, 2023. Disponível em: https://docs.oracle.com/javase/tutorial/. Acesso em: 10 out. 2023.

REMUNOM
ISSN 2178-6925

Received: 13/01/2026 - **Accepted:** 24/01/2026
Vol: 01.03
DOI: 10.61164/nzkwct67
Pages: 1-28

PYTHON SOFTWARE FOUNDATION. Python Language Reference. Disponível em: https://docs.python.org/3/reference/. Acesso em: 10 out. 2023.

SEDGEWICK, Robert; WAYNE, Kevin. Algorithms. 4. ed. Addison-Wesley, 2011.

STROUSTRUP, Bjarne. The C++ Programming Language. 4. ed. Addison-Wesley, 2013.

TANENBAUM, Andrew S. Estruturas de Dados Usando C. Pearson, 2007.

VAN ROSSUM, Guido; DRAKE, Fred L. Python Tutorial. Python Software Foundation, 2023. Disponível em: https://docs.python.org/3/tutorial/. Acesso em: 10 out. 2023.