

Nomes: Nicolás Hoefling de C,

Joao Pedro Girardi

Turma: 0104N

Data Entrega: 14/10/2024

RELATORIO SISTEMA DE CÁLCULO MATEMATICO AVANÇADO UTILIZANDO PHP VS JS

Tempo de Execução:

Um valor aproximado do tempo de execução das funções;

Navegador utilizado: operaGX

- Fibonacci valor: 500;
- Fatorial valor: 500;
- Geração de Matriz (são gerados valores aleatórios), Soma e Multiplicação;
- Geração de Vetores (são gerados vetores aleatórios), Soma e Multiplicação;
- Benchmark (número de interação utilizado: 100000)

Os 500 primeiros números de **Fibonacci** em JS

- Tempo de execução: 0.20 ms. = 0,0002 segundos.

Os 500 primeiros números de **Fibonacci** em PHP =

- Tempo de execução: 16.64 ms. = 0,01664 segundos.

Conclusão: JavaScript é aproximadamente **83 vezes mais rápido** que PHP para calcular os 500 primeiros números de Fibonacci.

Fatorial de 500 em JS

- Tempo de execução: 0.40 ms. = 0,0004 segundos.

Fatorial de 500 em PHP

- Tempo de execução: 4.23 ms. = 0,00423 segundos.

Conclusão: JavaScript é aproximadamente 10,58 vezes mais rápido que PHP para calcular o fatorial de 500.

Geração de **Matriz** JS: (500)

- Tempo de execução: 67.00 ms. = 0,067 segundos.

Geração de **Matriz** PHP: (500)

- Tempo de execução: 229.08 ms. = 0,22908 segundos.

Soma de **Matriz** JS: (500)

- Tempo de execução: 7.40 ms. = 0,00740 segundos.

Soma de **Matriz** PHP: (500)

- Tempo de execução: 20.61 ms. = 0,02061 segundos.

Multiplicação de **Matriz** JS: (500)

- Tempo de execução: 602.10 ms. = 0,60210 segundos.

Multiplicação de **Matriz** PHP: (500)

- Tempo de execução 132.84 ms. = 0,13284 segundos.

Conclusões:

- **Geração de Matriz:** JavaScript é aproximadamente **3,42 vezes mais rápido** que PHP.
- **Soma de Matriz:** JavaScript é aproximadamente **2,78 vezes mais rápido** que PHP.
- **Multiplicação de Matriz:** PHP é aproximadamente **4,52 vezes mais rápido** que JavaScript.

Geração de **Vetor** JS: (15000)

- Tempo de execução: 4.80 ms. = 0,00480 segundos.

Geração de **Vetor** PHP: (15000)

- Tempo de execução: 2.57 ms. = 0,00257 segundos.

Soma de **Vetor** JS: (15000)

- Tempo de execução: 1.50 ms. = 0,00150 segundos.

Soma de **Vetor** PHP: (15000)

- Tempo de execução: 0.49 ms. = 0,00049 segundos.

Multiplicação de **Vetor** JS valor escalar: (2.5)

- Tempo de execução: 5579.90 ms. = 5,57990 segundos.

Multiplicação de **Vetor** PHP valor escalar: (2.5)

- Tempo de execução: 3.54 ms. = 0,00354 segundos.

Conclusões:

- **Geração de Vetor:** PHP é aproximadamente **1,87 vezes mais rápido** que JavaScript.
- **Soma de Vetor:** PHP é aproximadamente **3,06 vezes mais rápido** que JavaScript.
- **Multiplicação de Vetor por Escalar:** PHP é **1.574.271 vezes mais rápido** que JavaScript.

Benchmark (100000) em JS

- Tempo de execução: 3.20 ms. = 0,00320 segundos.

Benchmark (100000) em PHP

- Tempo de execução: 31.46 ms. = 0,03146 segundos.

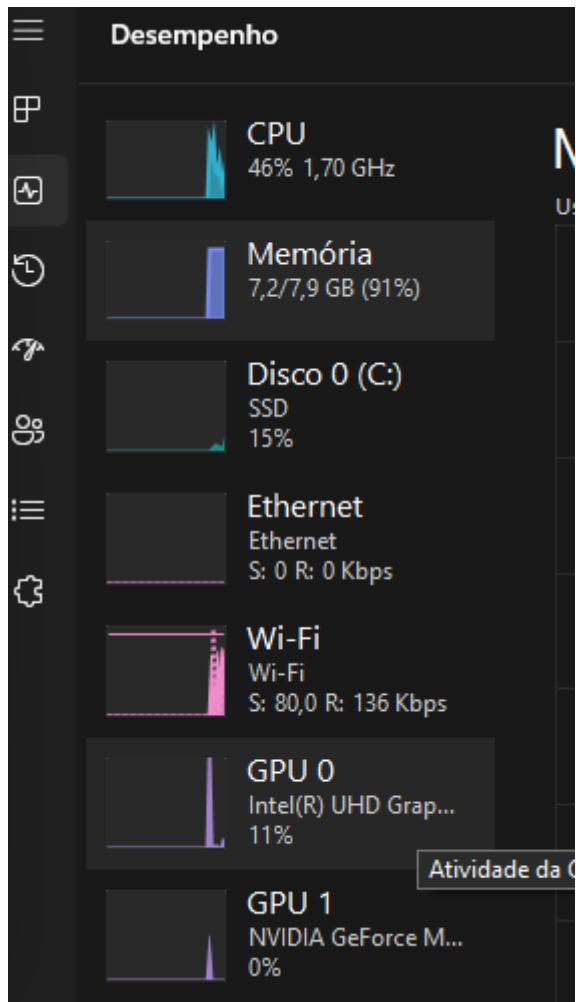
Conclusão: JavaScript é aproximadamente **9,83 vezes mais rápido** que PHP para 100.000 interações no benchmark.

JavaScript (Client-side): Geralmente apresenta tempos de execução mais rápidos para operações que não exigem comunicação constante com o servidor.

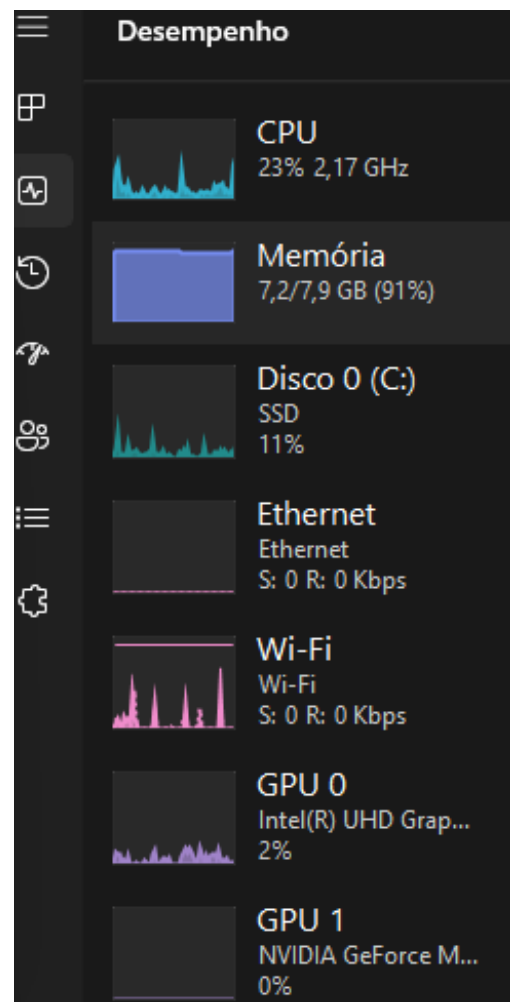
PHP (Server-side): Embora mais lento em algumas operações, é adequado para tarefas que requerem processamento no servidor, especialmente quando a consistência e a segurança dos dados são prioritárias.

Consumo de recursos (CPU e memória) – máquina (Nícolas):

Consumo PHP:



Consumo JavaScript:



Circunstâncias de desenvolvimento (ambiente de hardware e software usados para os testes)

Máquinas Utilizadas:

Nicolas:

- Processador Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
- RAM instalada 8,00 GB (utilizável: 7,88 GB)
- SSD: 256GB
- GPU: GeForce MX110 2GB

Joao Pedro:

- Processador Intel(R) Celeron N4000
- RAM instalada 4,00 GB (utilizável: 3,88 GB)

O que influencia na escolha do cliente ou servidor para cada tipo de cálculo.

- **Use Client-side (JavaScript) se:**
 - Deseja uma **experiência mais rápida e interativa**.
 - Está lidando com **cálculos simples** ou **visualizações** que não exigem muita segurança.
- **Use Server-side (PHP) se:**
 - Precisa de **maior segurança** e **controle** sobre os dados.
 - Está realizando **cálculos complexos** ou **operações com grandes conjuntos de dados**.
 - Deseja **manter o código protegido** e centralizado no servidor.

Muitas aplicações combinam as duas abordagens para aproveitar as vantagens de cada uma. Por exemplo, realizar cálculos rápidos no **client-side** para uma resposta imediata e usar o **server-side** para operações mais complexas ou para salvar dados no servidor.

Comparação entre os itens acima entre cliente-side e server-side

Eficiência de Execução

JavaScript (Lado do Cliente):

- Execução no navegador: As operações são realizadas no navegador do usuário. A eficiência depende da máquina do usuário e do navegador utilizado.
- Desempenho variável: Como a performance depende do dispositivo do usuário, pode haver grandes variações. Computadores mais antigos ou com menos recursos podem ter dificuldades em realizar cálculos complexos.
- Conexão e Latência: Opera independentemente da conexão de rede, evitando latência de rede nas operações.

Tempo de Desenvolvimento

JavaScript (Lado do Cliente):

- Interatividade: JS é ideal para criar interfaces interativas e fornecer feedback imediato ao usuário sem precisar recarregar a página.
- Complexidade: Cálculos complexos podem ser mais difíceis de implementar e depurar no navegador.
- Bibliotecas e Frameworks: Existem muitas bibliotecas e frameworks, como WebAssembly, que ajudam a melhorar o desempenho para cálculos pesados.

UTILIZAR JS

Eficiência de Execução

PHP (Lado do Servidor):

- Execução no servidor: As operações são realizadas no servidor, o que geralmente possui recursos mais robustos comparado ao dispositivo de um usuário médio.
- Desempenho constante: A performance é mais previsível, pois depende dos recursos do servidor e não varia com o dispositivo do usuário.
- Escalabilidade: Pode ser escalado vertical ou horizontalmente para lidar com uma maior carga de processamento.

Tempo de Desenvolvimento

PHP (Lado do Servidor):

- Simplicidade: PHP é uma linguagem de script de fácil aprendizado e uso para tarefas do lado do servidor.
- Bibliotecas: PHP tem diversas bibliotecas poderosas para cálculos matemáticos e manipulação de matrizes, como GMP para grandes números.
- Menos Interatividade: PHP não pode manipular a interface diretamente, então as operações do lado do cliente devem ser combinadas com AJAX ou outra tecnologia para fornecer feedback em tempo real.

UTILIZAR PHP