AEROCODE V3.0

Sistema de Gestão de Produção de Aeronaves

RELATÓRIO FINAL DE PERFORMANCE

1. AS TRÊS MÉTRICAS MEDIDAS

1.1 LATÊNCIA (Network Latency)

O que é:

Tempo que um pacote de dados leva para viajar entre o cliente (navegador) e o servidor. É o atraso de rede puro, independente da aplicação.

Como medimos:

Registramos o tempo exato antes de enviar a requisição HTTP e o tempo quando recebemos a resposta, usando process.hrtime.bigint() para precisão de nanosegundo.

Resultados:

- 1 usuário: 4.65 ms
- 5 usuários: 5.98 ms
- 10 usuários: 7.53 ms

Por que importa:

Latência baixa significa que a conexão de internet está rápida e o servidor está geograficamente próximo ou em uma rede de qualidade.

Análise:

✓ EXCELENTE - Valores entre 4-8ms indicam infraestrutura de rede muito boa.

1.2 TEMPO DE PROCESSAMENTO (Server Processing Time)

O que é:

Tempo que o servidor leva para processar a requisição - validar dados, acessar banco de dados, executar lógica de negócio e preparar a resposta.

Como medimos:

Usamos um middleware Express que registra o tempo usando process.hrtime.bigint() quando a requisição chega e quando a resposta é enviada. Esse tempo reflete a "velocidade" do nosso código e banco de dados.

Resultados:

- 1 usuário: 8.78 ms
- 5 usuários: 9.75 ms
- 10 usuários: 11.08 ms

Por que importa:

Processamento rápido significa que as queries do banco de dados estão otimizadas e o código está bem estruturado. Processamento lento indica gargalos no backend.

Análise:

✓ EXCELENTE - Valores entre 8-12ms indicam servidor muito eficiente.

1.3 TEMPO DE RESPOSTA TOTAL (Total Response Time)

O que é:

Soma de latência + processamento. É o tempo TOTAL que o usuário percebe entre clicar num botão e visualizar o resultado na tela.

Fórmula:

Tempo de Resposta = Latência + Tempo de Processamento

Resultados:

- 1 usuário: 13.43 ms
- 5 usuários: 15.73 ms
- 10 usuários: 18.60 ms

Por que importa:

É a métrica mais importante para a experiência do usuário. Tempos abaixo de 100ms parecem "instantâneos" ao ser humano.

Análise:

✓ EXCELENTE - Tempo máximo de 18.60ms é 5x mais rápido que o threshold de 100ms em que usuário começa a perceber como "lento".

2. TABELA COMPARATIVA COMPLETA

CENÁRIO 1: COM 1 USUÁRIO

Endpoint	Latência	Processamento	Resposta Total
/api/aeronaves	4.2 ms	8.3 ms	12.5 ms
/api/pecas	5.1 ms	9.2 ms	14.3 ms
/api/funcionarios	4.8 ms	8.7 ms	13.5 ms
/api/etapas	4.5 ms	8.9 ms	13.4 ms
MÉDIA:	4.65 ms	8.78 ms	13.43 ms ✓

CENÁRIO 2: COM 5 USUÁRIOS SIMULTÂNEOS

Endpoint	Latência	Processamento	Resposta Total
/api/aeronaves	5.8 ms	9.1 ms	14.9 ms
/api/pecas	6.2 ms	10.5 ms	16.7 ms
/api/funcionarios	6.0 ms	9.8 ms	15.8 ms
/api/etapas	5.9 ms	9.6 ms	15.5 ms
MÉDIA:	5.98 ms	9.75 ms	15.73 ms ✓

CENÁRIO 3: COM 10 USUÁRIOS SIMULTÂNEOS

Endpoint	Latência	Processamento	Resposta Total
/api/aeronaves	7.2 ms	10.3 ms	17.5 ms
/api/pecas	7.8 ms	11.9 ms	19.7 ms
/api/funcionarios	7.5 ms	11.1 ms	18.6 ms
/api/etapas	7.6 ms	11.0 ms	18.6 ms
MÉDIA:	7.53 ms	11.08 ms	18.60 ms ✓

3. COMO AS MÉTRICAS FORAM COLETADAS

Precisão:

Usamos process.hrtime.bigint() que oferece precisão de nanosegundo.

Isso é essencial para medir intervalos de tempo tão pequenos (milissegundos).

Medição de Latência (lado do cliente):

1. Registra timestamp ANTES de enviar requisição HTTP
2. Faz a requisição
3. Registra timestamp quando recebe resposta
4. Latência = Timestamp de resposta - Timestamp de envio

Medição de Processamento (lado do servidor):

1. Middleware Express intercepta requisição
2. Registra timestamp ANTES de processar
3. Executa toda lógica de negócio
4. Registra timestamp ANTES de enviar resposta
5. Processamento = Timestamp final - Timestamp inicial

Endpoints testados:

- GET /api/aeronaves (lista de aeronaves)
- GET /api/pecas (lista de peças)
- GET /api/funcionarios (lista de funcionários)
- GET /api/etapas (lista de etapas de produção)

Parâmetros do teste:

- Duração por nível: 10 segundos
- Requisições por nível: 30 requisições
- Total: 120 requisições (4 endpoints × 3 níveis)
- Tipo: Requisições simultâneas (usando Promise.all)