



**UNIVERSIDADE DE FORTALEZA - UNIFOR**  
**Mestrado em Informática Aplicada - MIA**

**Rener Silva de Menezes**

**AMBIENTES EM NUVEM PARA CIÊNCIA DE DADOS E  
INTELIGÊNCIA ARTIFICIAL**

**Trabalho V – Teste de Capacidade**

Fortaleza

2025

## 1. Dados para o Relatório (Copie e Cole/Adapte)

Aqui estão as informações técnicas exatas baseadas no sucesso com 150 Usuários.

### Estratégia Adotada

- Caminho Escolhido: Misto (Scale-out + Tuning Intensivo de Software).
- Justificativa:
  - Inicialmente, a escalabilidade horizontal (5 máquinas) não foi suficiente devido a gargalos de I/O no Banco de Dados e configuração padrão do Apache.
  - Optamos por manter as instâncias c5.large (Scale-out) mas aplicamos Tuning de Software agressivo:
    1. Banco de Dados: Alteração de parâmetros do InnoDB (`innodb_flush_log_at_trx_commit=2`) para contornar a limitação de IOPS do disco EBS.
    2. Apache: Ajuste de `MaxRequestWorkers` para 150 (evitando swap de memória) e `KeepAliveTimeout` para 5s (otimização para Load Balancer).
    3. Logs: Desativação dos logs de acesso (`CustomLog`) para reduzir a latência de disco na aplicação.

### Arquitetura Final

- Servidores de Aplicação: 5 instâncias c5.large.
- Banco de Dados: 1 instância c5.large (Fixa).
- Sistema Operacional: Amazon Linux 2.
- Parâmetros Chave Alterados:
  - MariaDB: `max_connections=2000, innodb_buffer_pool_size=2G, innodb_flush_log_at_trx_commit=2`.
  - Apache (`httpd.conf`): `ServerLimit 150, MaxRequestWorkers 150, KeepAlive On, KeepAliveTimeout 5`.

### Resultados Obtidos (Teste de 150 Usuários)

- RPS Máximo Estável: ~25.6 req/s.
- Latência P95: 6.800 ms (6.8 segundos).
  - Obs: O SLO era de 10.000ms. O resultado ficou ~32% abaixo do limite, aprovado.
- Taxa de Erro: 0.00% (Zero erros 5xx).
  - Obs: O limite era 1%. Aprovado.

## 2. Análise de Custo (A Prova dos US\$ 0.50)

A camada de aplicação deve custar menos de \$0.50/hora.

A instância c5.large na região us-east-1 (Norte da Virgínia) custa aproximadamente \$0.085/hora (Preço On-Demand padrão da AWS).

Item	Tipo de Instância	Unitário (USD/h)	Quantidade	Custo Total (USD/h)
App Server	c5.large	\$0.085	5	\$ 0.425
<b>TOTAL</b>				<b>\$ 0.425</b>

Tabela 1: Custos (Application Layer).

### 3. Determinação da Capacidade Máxima Sustentada via Testes de Carga Escalonada

Para determinar a capacidade real da infraestrutura, adotamos uma abordagem iterativa de carga ('Step-up Test'). Iniciamos com cargas conservadoras (100 usuários) e incrementamos progressivamente. Observou-se que até 150 usuários o sistema mantém a latência estável abaixo de 7 segundos. Ao cruzar a barreira de 150 usuários (testes com 180 e 200), a latência degrada exponencialmente, indicando que o gargalo de I/O (IOPS do disco e throughput do banco) foi atingido, mesmo sem gerar erros 5xx. Portanto, definimos 150 usuários como a capacidade nominal segura.

Cenário (Usuários)	Throughput (Req/s)	P95 (SLA < 10s)	Taxa de Erro	Status	Conclusão
100	~27.1 req/s	6.1s	0%	<input checked="" type="checkbox"/> Aprovado	Sistema ocioso, subutilizado.
150	<b>~25.6 req/s</b>	<b>6.8s</b>	<b>0%</b>	<input checked="" type="checkbox"/> Aprovado	<b>Ponto Ótimo (Capacidade Máxima).</b>
180	~16.8 req/s	15.0s	0%	<input type="checkbox"/> Reprovado	Degradação de performance (Saturação).
200	~15.7 req/s	17.0s	0%	<input type="checkbox"/> Reprovado	Violação grave de SLA.
300	~5.6 req/s	26.0s	0%	<input type="checkbox"/> Reprovado	Colapso de I/O (Bottleneck).

Tabela 2: Comparativo de Latência e Throughput por Cenário de Usuários Simultâneos.

A ausência de erros (0%) mesmo sob alta latência demonstra a eficácia do tuning de MaxRequestWorkers. Ao limitar a concorrência para 150 processos, evitamos o esgotamento de memória (OOM) e o colapso dos serviços. O sistema degradou de forma controlada (aumentando o tempo de resposta devido ao enfileiramento) em vez de falhar catastroficamente (retornando erros 5xx), respeitando o timeout padrão do Load Balancer (60s).

Conclusão do Custo:

O custo total da camada de aplicação foi de \$ 0.425/h, que é inferior ao limite de \$ 0.50/h estabelecido no projeto.