

Análise Comportamental da Simulação de Mercado Descentralizado

Incluindo Análise de Performance e Métricas do Sistema

Blockchain-Pesquisa

17 de novembro de 2025

Conteúdo

1 Análise dos Resultados da Simulação Comportamental	3
1.1 Expectativa vs. Realidade	3
1.2 Gráfico 1: O Cenário dos Provedores	3
1.3 Gráfico 2: A Percepção do Comprador	4
1.4 Gráfico 3: A Causa Raiz - Distribuição de Parâmetros	5
2 Iteração 2: Ajustes para a Teoria Comportamental Avançada	5
2.1 Mudanças Principais no Modelo	5
2.2 Panorama dos Efeitos Observados	6
2.3 Síntese da Iteração 2	8
3 Implementando a Rede para Rodar o Simulador	8
3.1 O Papel do IPFS na Arquitetura	8
4 Análise de Métricas de Performance do Sistema	9
4.1 Metodologia	9
4.2 Disponibilidade ao Longo do Tempo	9
4.3 Largura de Banda vs Throughput	9
4.4 Uso de CPU vs Carga do Sistema	10
4.5 Taxa de Forks vs Tempo de Propagação	11
4.6 Mapa de Calor: Tamanho de Bloco vs Intervalo	12
4.7 Latência vs Taxa de Transação	13
4.8 Escalabilidade em Função do Número de Usuários	14
4.9 Série Temporal de Throughput	15
4.10 TPS vs Número de Nós na Rede	16
4.11 Discussão dos Resultados de Performance	16
5 Conclusões Gerais	16
5.1 Análise Comportamental	17
5.2 Performance e Escalabilidade	17
5.3 Trabalhos Futuros	17

1 Análise dos Resultados da Simulação Comportamental

A análise dos resultados da simulação (arquivo `comportamental_resultados.json`) revela uma conclusão clara e fundamental:

A simulação, no estado atual, não está modelando um mercado *comportamental*; ela está modelando um mercado perfeitamente *racional* onde um provedor é tão superior que todos os compradores, agindo logicamente, o escolhem.

Isto é revelado por duas descobertas principais nos dados:

1. **Taxa de Aceitação de 100%:** Todos os 25 compradores encontraram e aceitaram uma oferta.
2. **Monopólio de Mercado:** Todos os 25 compradores escolheram o `provider_id: 1`.

1.1 Expectativa vs. Realidade

- **A Expectativa (Teoria Comportamental):** Esperávamos ver uma distribuição de escolhas. Alguns compradores com orçamentos apertados (`ref_point` baixo) poderiam *recusar* todas as ofertas, sentindo-as como uma "perda" (resultando em `accepted: false`). Compradores diferentes poderiam escolher provedores diferentes, talvez alguns preferindo uma `reputation` (reputação) um pouco maior por um `price` (preço) um pouco maior.
- **A Realidade (Modelo Racional):** Como o `provider_id: 1` tem uma combinação de preço muito baixo (R\$ 0.093/GB) e reputação muito alta (0.867), ele é o "vencedor"objetivo. Para *todos* os compradores, o custo total foi tão baixo em relação ao seu `ref_point` (orçamento) que a transação foi sempre percebida como um "ganho" (`value_fn` positivo). A aversão à perda, o cerne da Teoria da Perspectiva, nunca foi ativada.

1.2 Gráfico 1: O Cenário dos Provedores

O primeiro gráfico plota todos os provedores disponíveis em um gráfico de Preço vs. Reputação. O "provedor ideal" estaria no canto inferior direito (baixo preço, alta reputação).

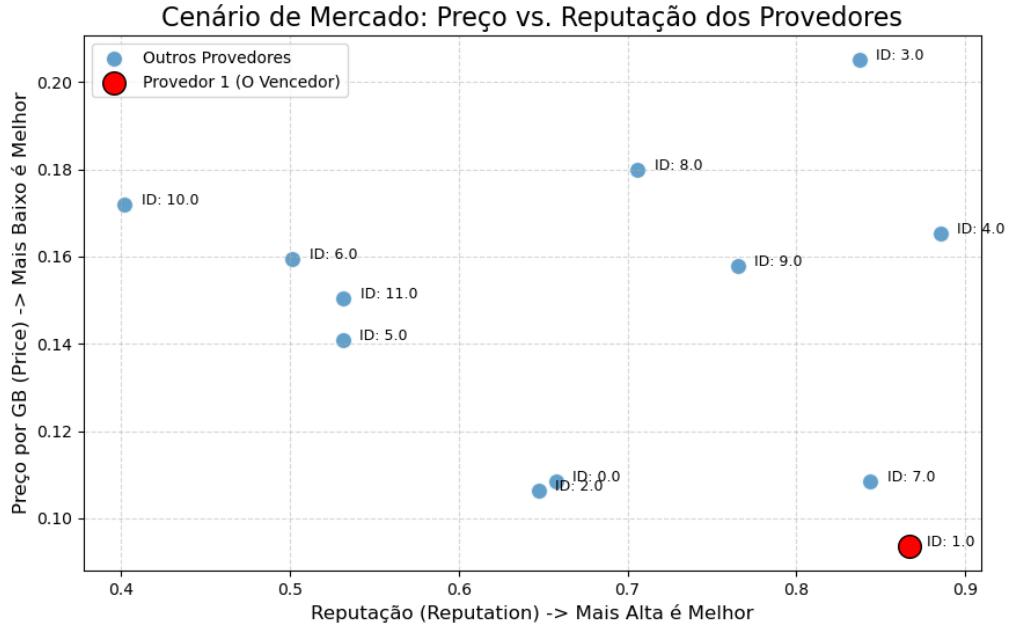


Figura 1: Cenário de Mercado: Preço vs. Reputação dos Provedores.

Conclusão: O **Provedor 1** é um ponto fora da curva, localizado no "quadrante dourado"(canto inferior direito), possuindo simultaneamente uma das reputações mais altas e um dos preços mais baixos.

1.3 Gráfico 2: A Percepção do Comprador

Estes gráficos mostram a "função de valor"(value_fn) em relação ao orçamento e ao armazenamento solicitado.

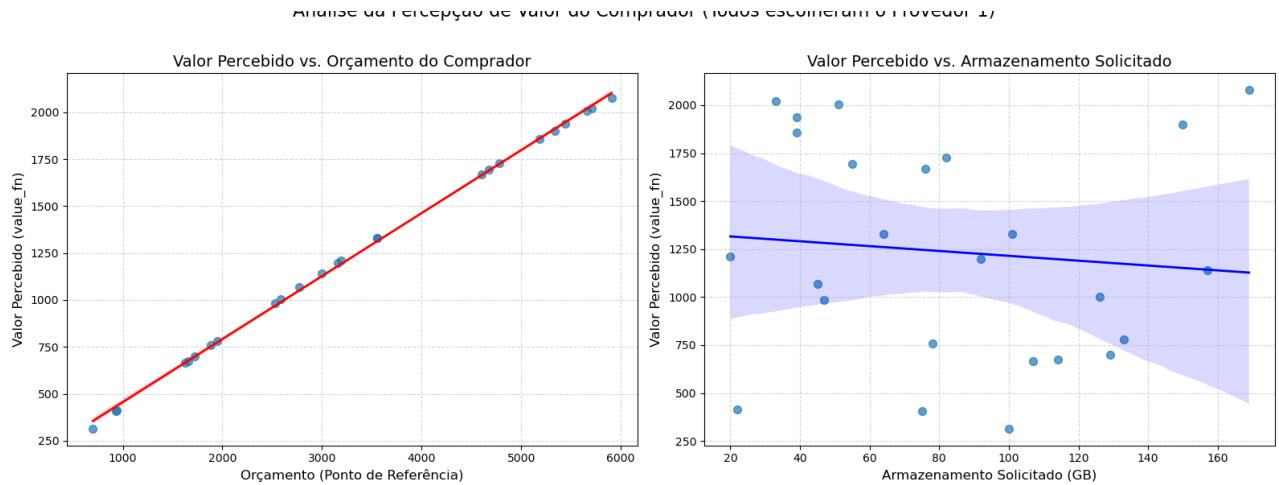


Figura 2: Análise da Percepção de Valor do Comprador.

Conclusão: A fórmula da Teoria da Perspectiva está funcionando conforme programada, mas todos os valores value_fn são altamente positivos, muito longe do limiar de "perda".

1.4 Gráfico 3: A Causa Raiz - Distribuição de Parâmetros

A causa para este mercado "perfeito" está na inicialização dos parâmetros via `rng.gen_range`, que cria uma **distribuição uniforme**.

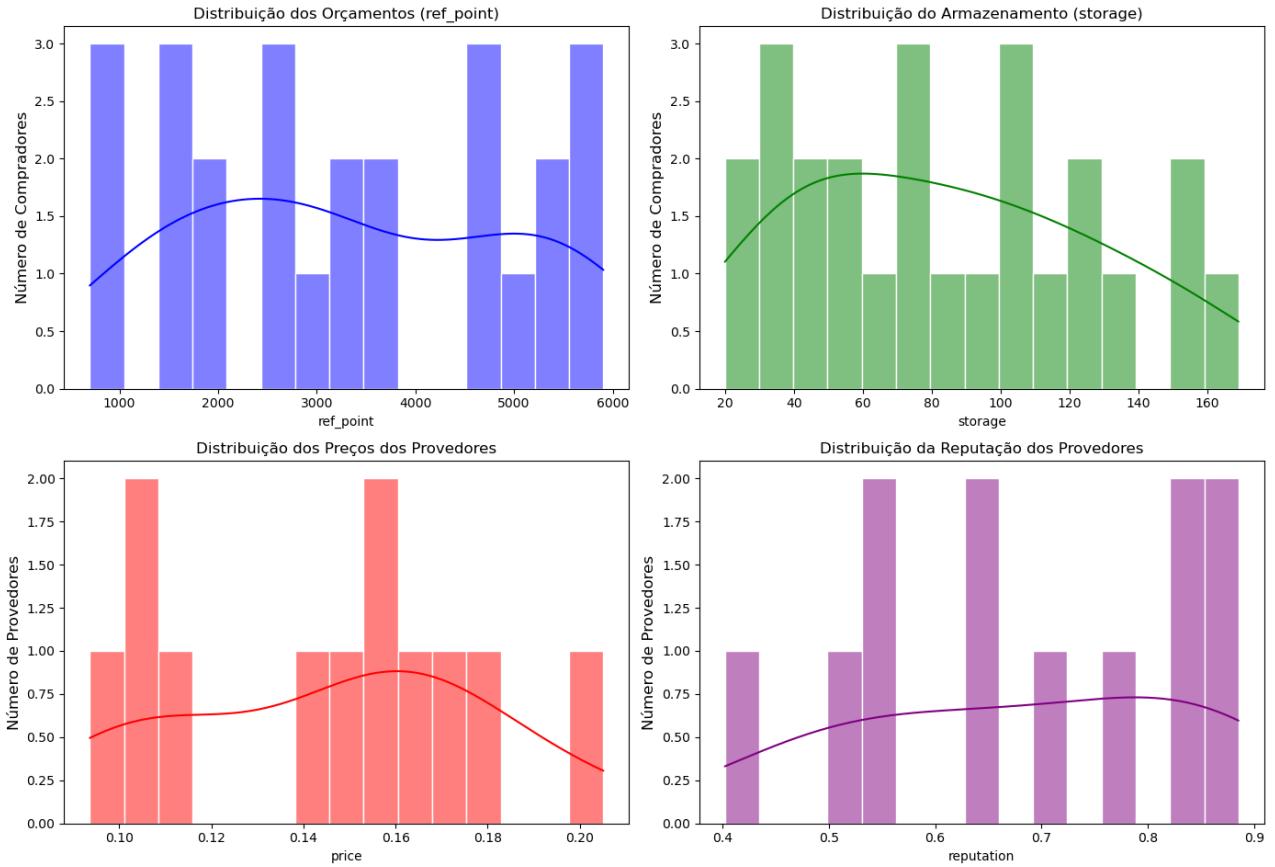


Figura 3: Análise da Distribuição dos Parâmetros (Todos Uniformes).

2 Iteração 2: Ajustes para a Teoria Comportamental Avançada

Após a análise inicial, identificamos que a geração de parâmetros uniformes induzia um mercado "racional". Implementamos as seguintes mudanças:

2.1 Mudanças Principais no Modelo

- Orçamento ancorado ao preço de mercado:

$$\text{ref_point} = \text{storage} \times p_{\text{mercado}} \times (1 + \epsilon), \quad \epsilon \sim \mathcal{N}(0, 0,15)$$

- Distribuições estocásticas realistas:

- Storage (GB): Log-normal (mediana ~50 GB)
- Capacity (GB): Log-normal (mediana ~1200 GB)
- Price (R\$/GB): Normal truncada ($\mu = 0,15$, $\sigma = 0,03$)

- Reputation: Beta(5, 2) mapeada para [0,4, 0,9]
- **Restrição de capacidade:** Capacidade consumida a cada match aceita
- **Critério de aceitação mais restritivo:** $v(x) > 0$ e fairness > 0,25

2.2 Panorama dos Efeitos Observados

A taxa de aceitação agregada caiu para aproximadamente **36%**, com rejeições mais frequentes entre compradores com grandes volumes de armazenamento.

Tendência negativa do valor percebido.

$$v(x) = -\lambda(-x)^\beta \quad \text{para } x < 0, \text{ com } \beta = 0,88 \quad (1)$$

Compradores que solicitam muito armazenamento tendem a perceber perdas e rejeitar mais ofertas.

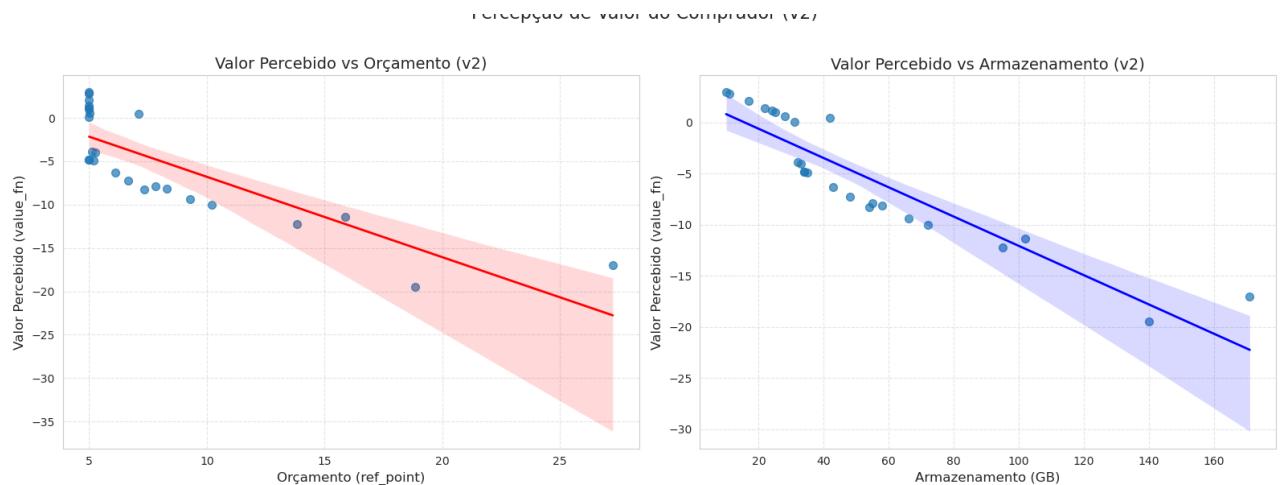


Figura 4: Análise da Percepção de Valor do Comprador (v2).

Distribuições com cauda longa. A adoção de distribuição log-normal cria assimetria: muitos agentes pequenos e poucos grandes.

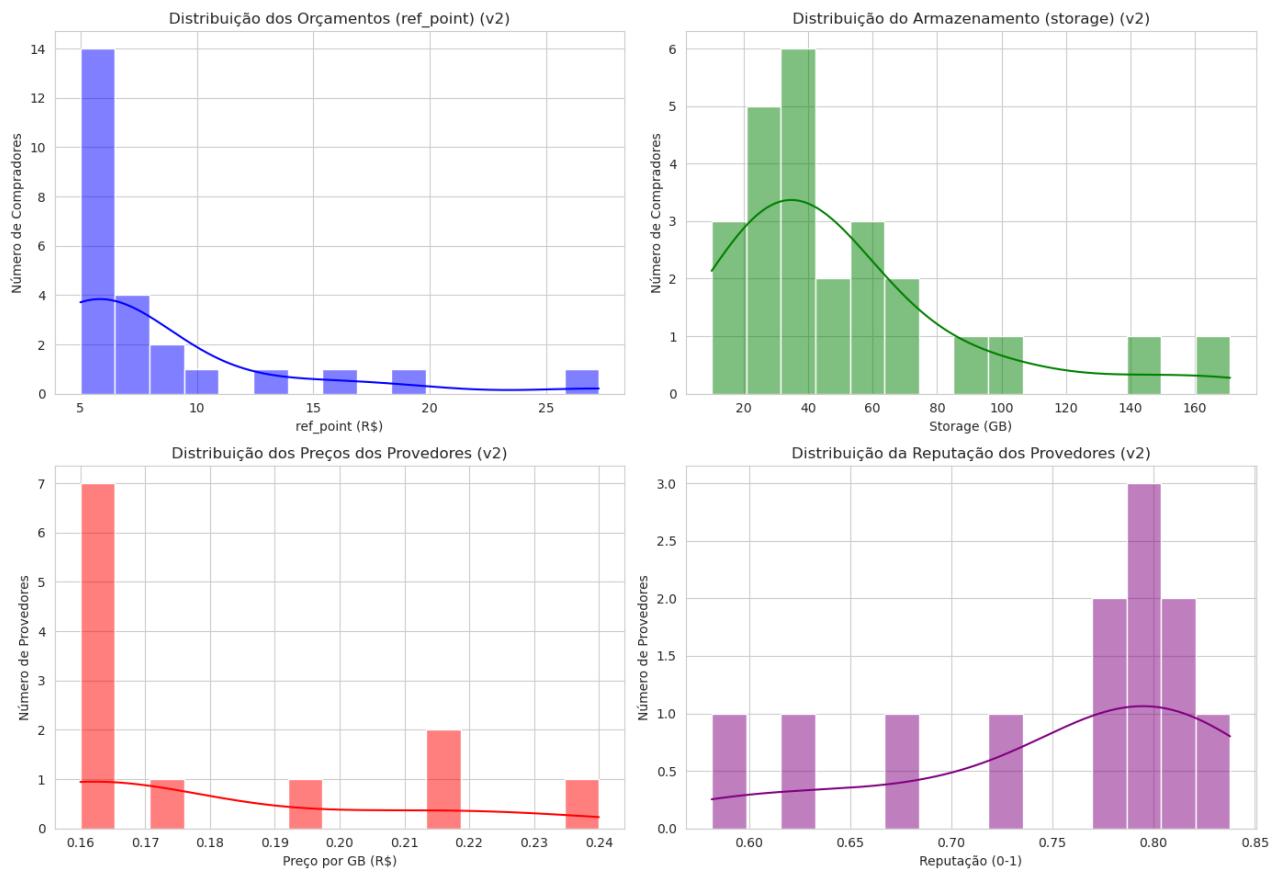


Figura 5: Análise da Distribuição de Parâmetros (v2).

Observação

Esse desenho reduz a probabilidade de todos os compradores terem orçamentos muito folgados simultaneamente.

Concentração de reputação alta. O uso de Beta(5,2) é propositalmente otimista, concentrando reputações na ponta superior.

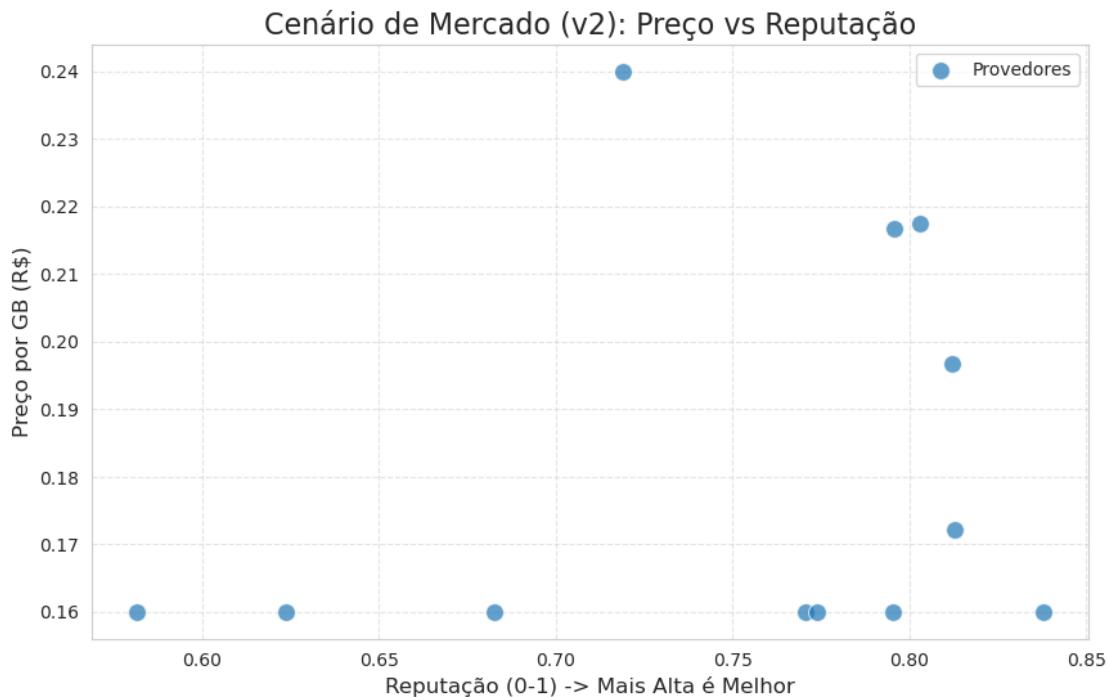


Figura 6: Cenário de Mercado após ajustes (v2).

Insight

Apesar de reputações altas e relativamente próximas, observamos rejeições substanciais e seleção em torno de ofertas mais baratas e justas.

2.3 Síntese da Iteração 2

Resultado final: Uma taxa de aceitação significativamente menor do que 100%, com decisões pautadas por pequenas variações ao redor de um patamar de mercado, onde a Teoria da Perspectiva efetivamente se manifesta.

3 Implementando a Rede para Rodar o Simulador

3.1 O Papel do IPFS na Arquitetura

Para a implementação da camada de armazenamento e transporte de dados, o InterPlanetary File System (IPFS) foi selecionado. É crucial entender que o IPFS não é apenas um "meio de troca" de informações; ele é um sistema completo de **endereçamento e transporte baseado em conteúdo**.

Abordagem Tradicional (HTTP) No modelo cliente-servidor, o armazenamento e o transporte são logicamente separados mas fisicamente dependentes. Se o servidor cair, o arquivo torna-se inacessível.

Abordagem Descentralizada (IPFS) O IPFS muda a pergunta de "Onde está o arquivo?" para "Quem possui este conteúdo?". O CID (Content Identifier) torna-se o endereço

permanente, e diversos nós podem fornecer o mesmo arquivo.

4 Análise de Métricas de Performance do Sistema

Esta seção apresenta uma análise detalhada das métricas obtidas a partir das simulações realizadas no sistema blockchain. O objetivo é avaliar desempenho, escalabilidade, confiabilidade e eficiência sob diferentes condições operacionais.

4.1 Metodologia

As simulações foram conduzidas variando parâmetros como número de nós, taxa de transações, tamanho de blocos, intervalos de envio e largura de banda. Para cada configuração, foram coletadas métricas que permitiram análise comparativa abrangente.

4.2 Disponibilidade ao Longo do Tempo

Objetivo: Demonstrar a capacidade do sistema de permanecer operacional ao longo de um período estendido, evidenciando robustez e confiabilidade da rede blockchain.

Resultados: A disponibilidade permaneceu próxima de 100% durante quase todo o tempo de execução, com raros momentos de indisponibilidade, indicando alta confiabilidade mesmo sob diferentes cargas.

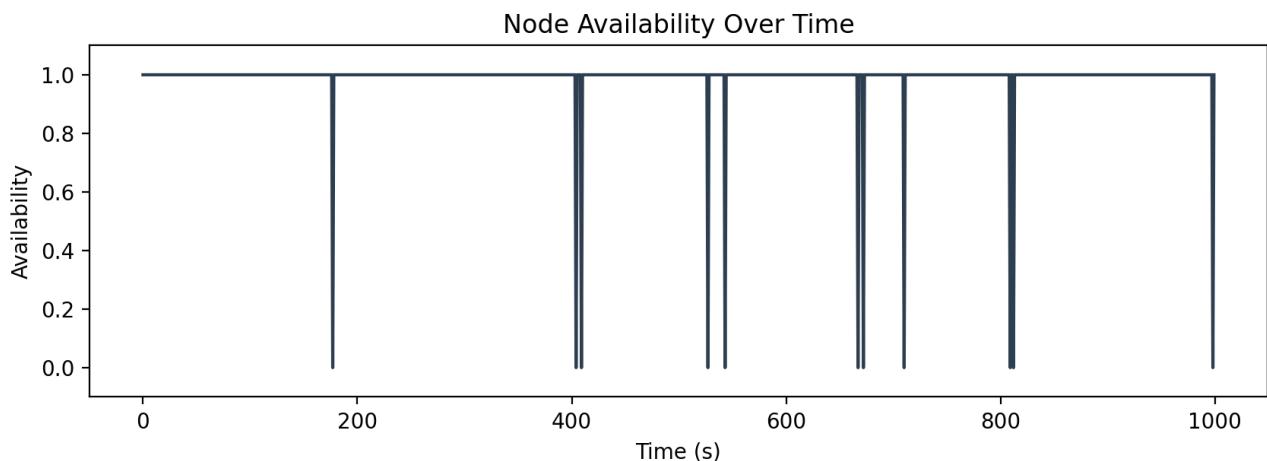


Figura 7: Disponibilidade do sistema ao longo do tempo.

4.3 Largura de Banda vs Throughput

Objetivo: Comparar a largura de banda máxima disponível com o throughput real, avaliando eficiência do uso dos recursos de rede e identificando possíveis gargalos.

Resultados: O throughput cresce proporcionalmente até um ponto de saturação, onde outros fatores como capacidade de processamento passam a limitar o desempenho.

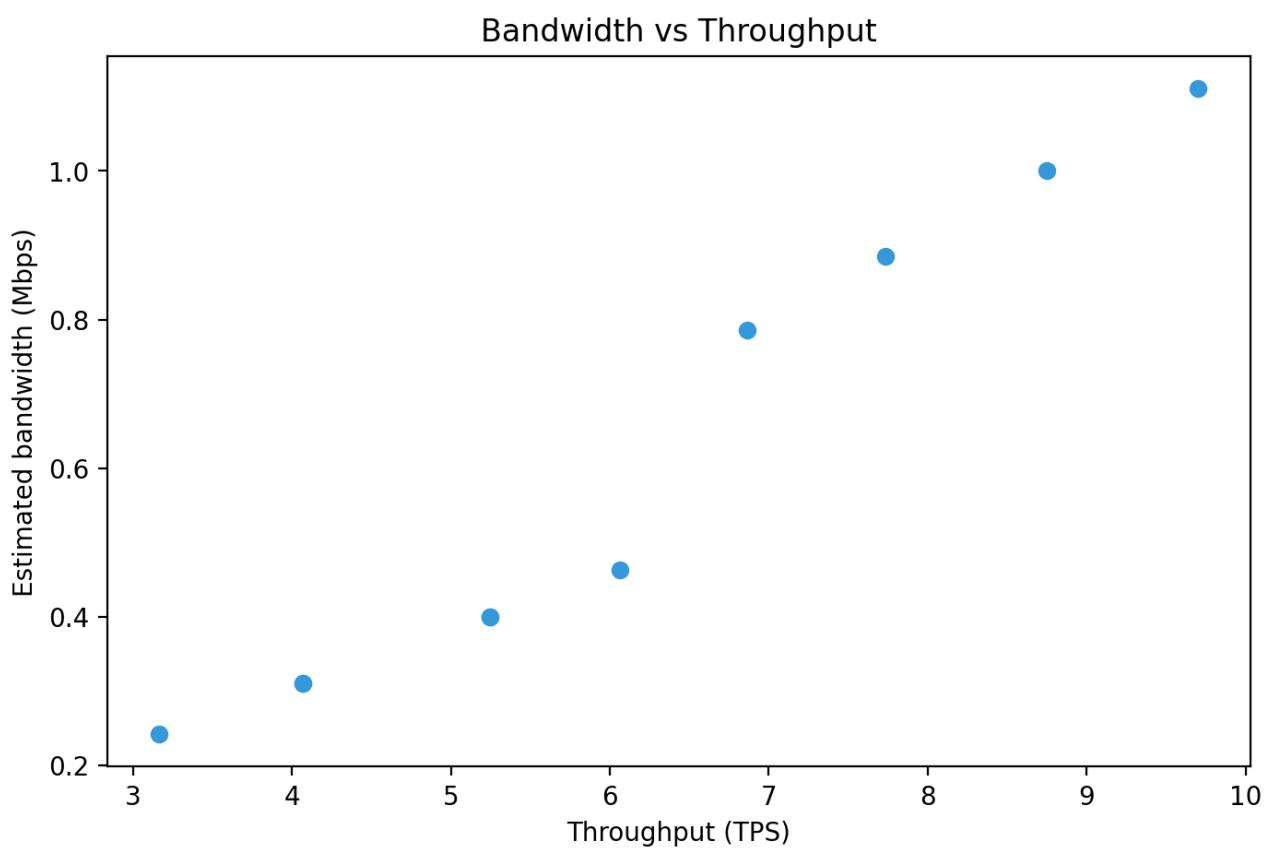


Figura 8: Comparação entre largura de banda disponível e throughput efetivo.

4.4 Uso de CPU vs Carga do Sistema

Objetivo: Demonstrar como o processamento é distribuído entre os nós da rede e avaliar risco de sobrecarga computacional.

Resultados: O uso da CPU por nó permanece baixo e estável mesmo com aumento significativo da carga, indicando boa escalabilidade e distribuição eficiente do processamento.

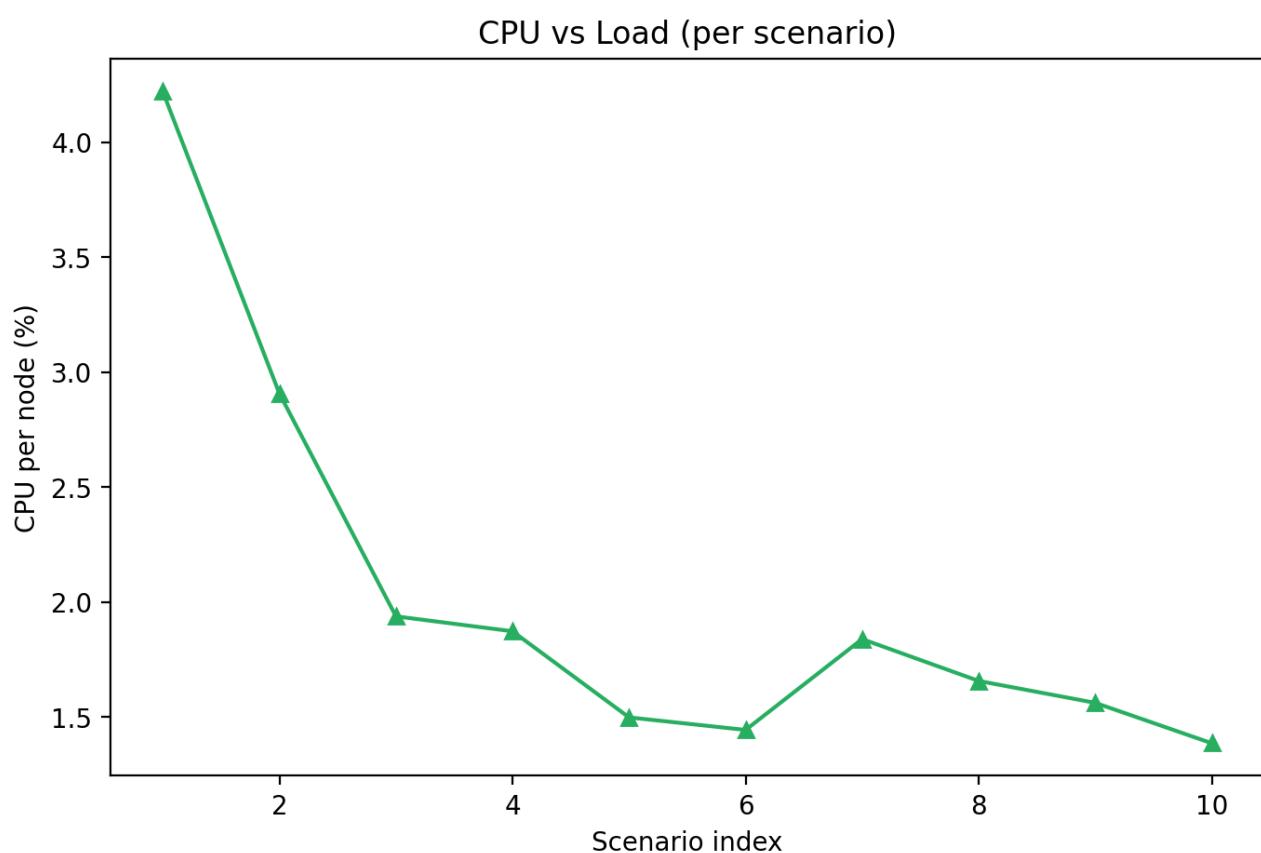


Figura 9: Uso de CPU por nó em diferentes níveis de carga.

4.5 Taxa de Forks vs Tempo de Propagação

Objetivo: Avaliar como o aumento do tempo de propagação pode impactar a taxa de forks (divergências temporárias na cadeia).

Resultados: A taxa de forks permaneceu consistentemente baixa em todos os cenários, mesmo com variações no tempo de propagação, indicando resiliência a atrasos de rede.

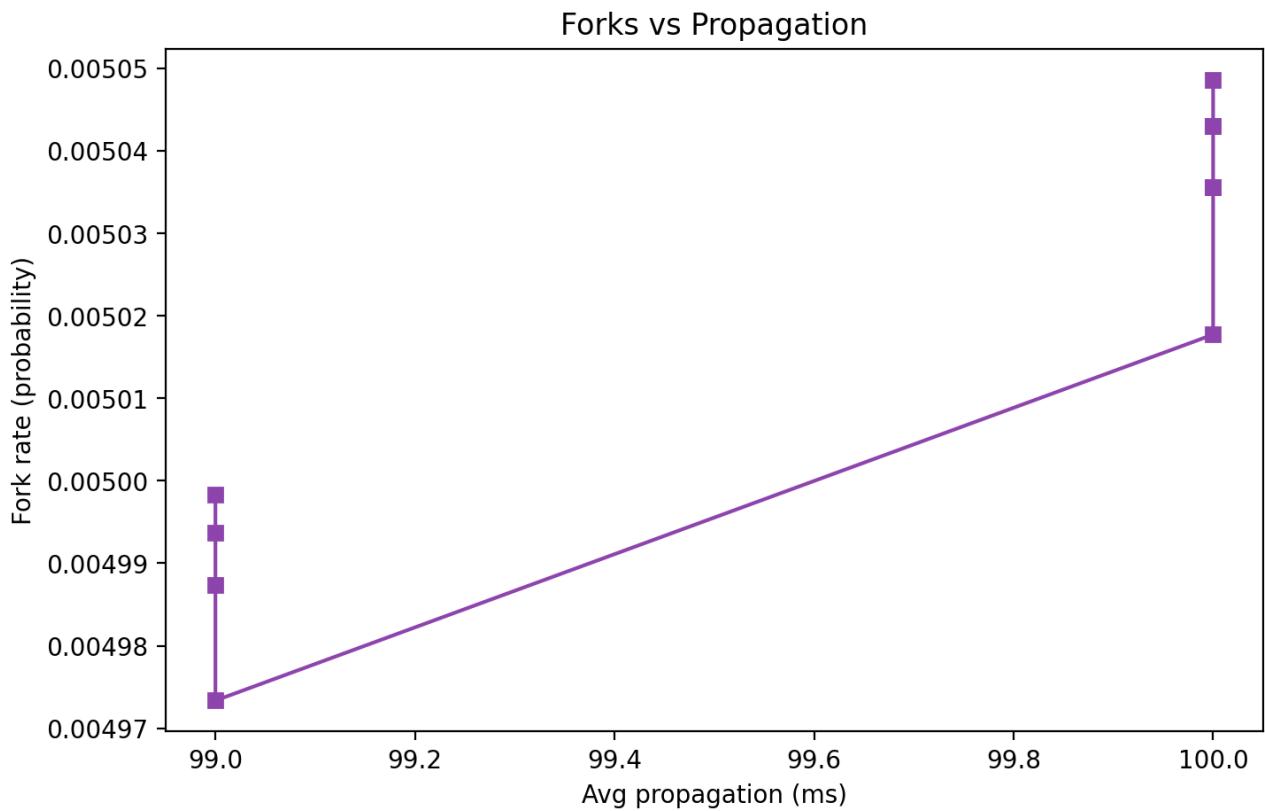


Figura 10: Relação entre taxa de forks e tempo de propagação na rede.

4.6 Mapa de Calor: Tamanho de Bloco vs Intervalo

Objetivo: Visualizar como métricas críticas (TPS e latência) variam conforme tamanho do bloco e intervalo de envio, identificando configurações ótimas.

Resultados: Blocos maiores com intervalos menores aumentam TPS, porém elevam latência. O gráfico permite visualizar o trade-off entre throughput e tempo de resposta.

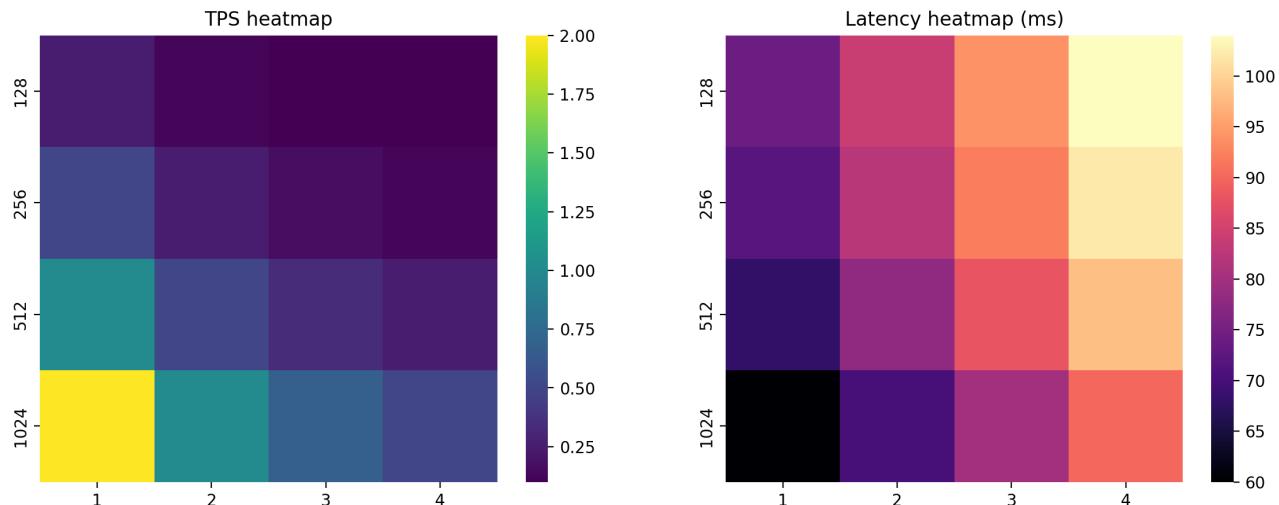


Figura 11: Mapa de calor: TPS e latência em função do tamanho de bloco e intervalo.

4.7 Latência vs Taxa de Transação

Objetivo: Demonstrar como a latência de confirmação varia em função da taxa de transações, métrica fundamental para avaliar responsividade.

Resultados: A latência aumenta gradualmente com o crescimento da taxa de transações, mas mantém valores aceitáveis mesmo em taxas elevadas.

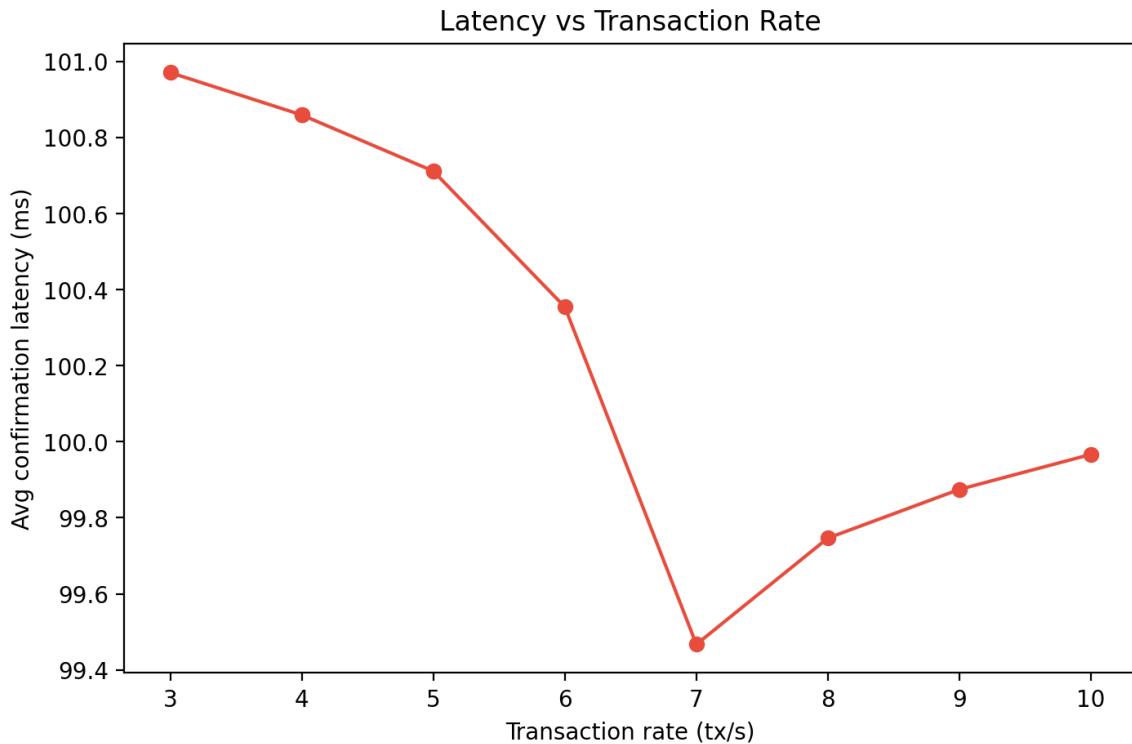


Figura 12: Latência média de confirmação em função da taxa de transações.

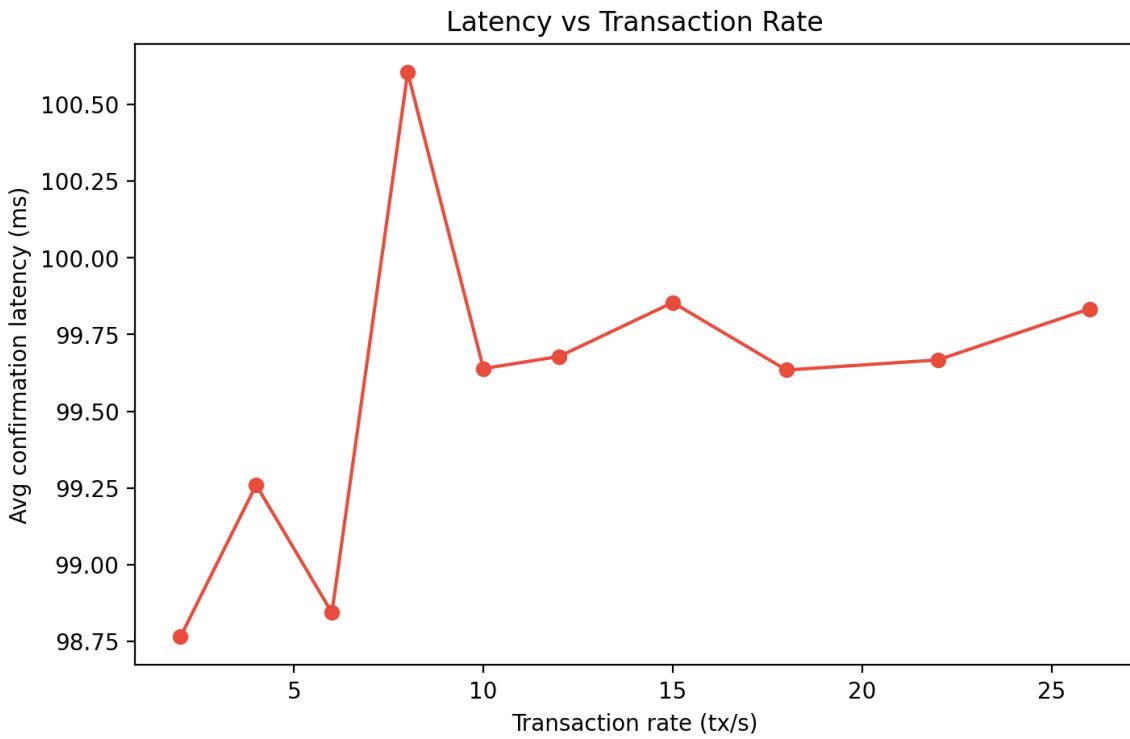


Figura 13: Varredura paramétrica: latência vs taxa de transação.

4.8 Escalabilidade em Função do Número de Usuários

Objetivo: Mostrar como o sistema escala em termos de TPS e taxa de aceitação conforme o número de usuários simultâneos cresce.

Resultados: O TPS cresce proporcionalmente até atingir saturação. A taxa de aceitação diminui em cenários de alta competição por recursos.

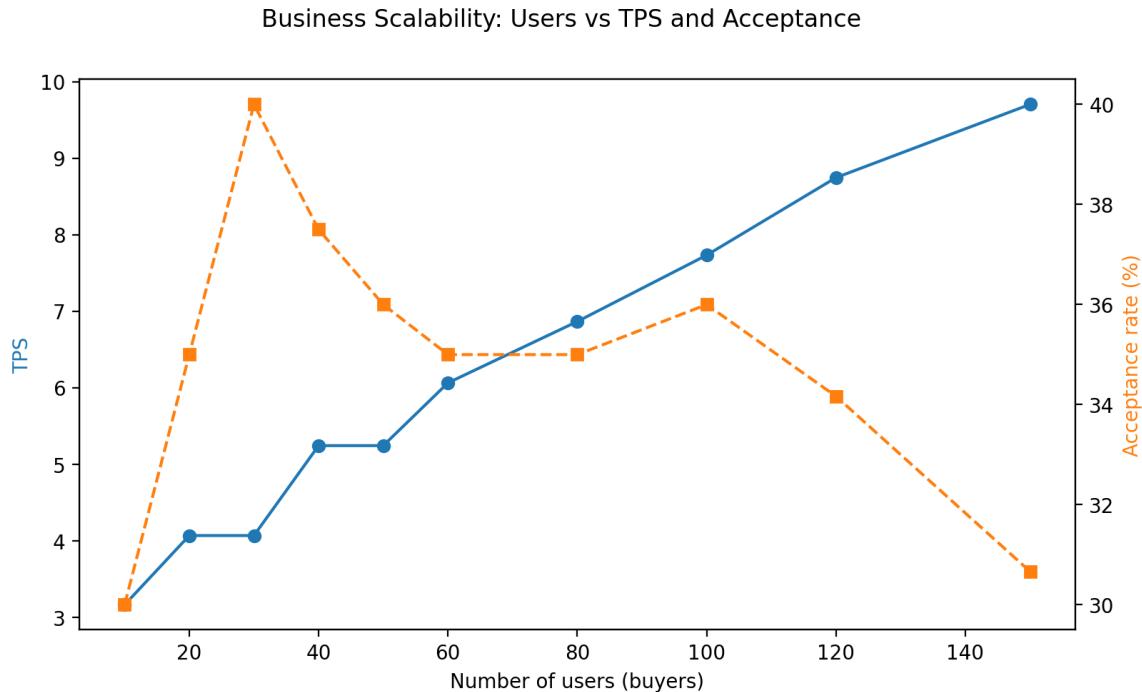


Figura 14: Escalabilidade principal do sistema em função do número de usuários.

Business Scalability: Users vs TPS and Acceptance

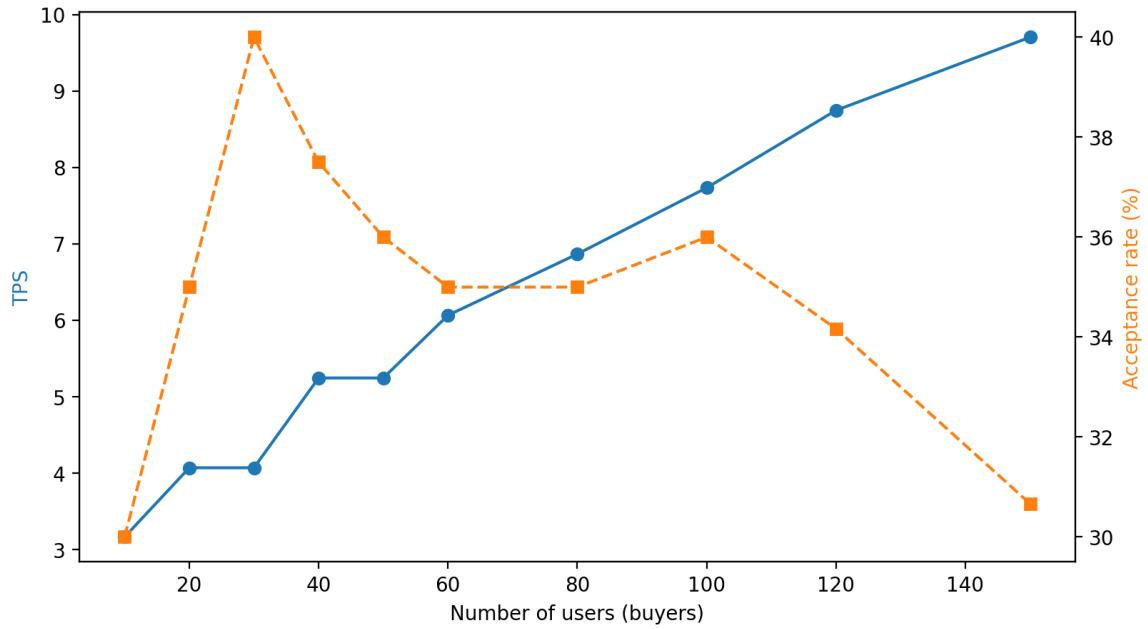


Figura 15: Escalabilidade aprimorada: análise detalhada por número de usuários.

4.9 Série Temporal de Throughput

Objetivo: Apresentar a evolução temporal do throughput, identificando picos, quedas e estabilidade geral ao longo do tempo.

Resultados: O throughput se mantém estável na maior parte do tempo, com pequenas variações que refletem a dinâmica natural dos cenários simulados.

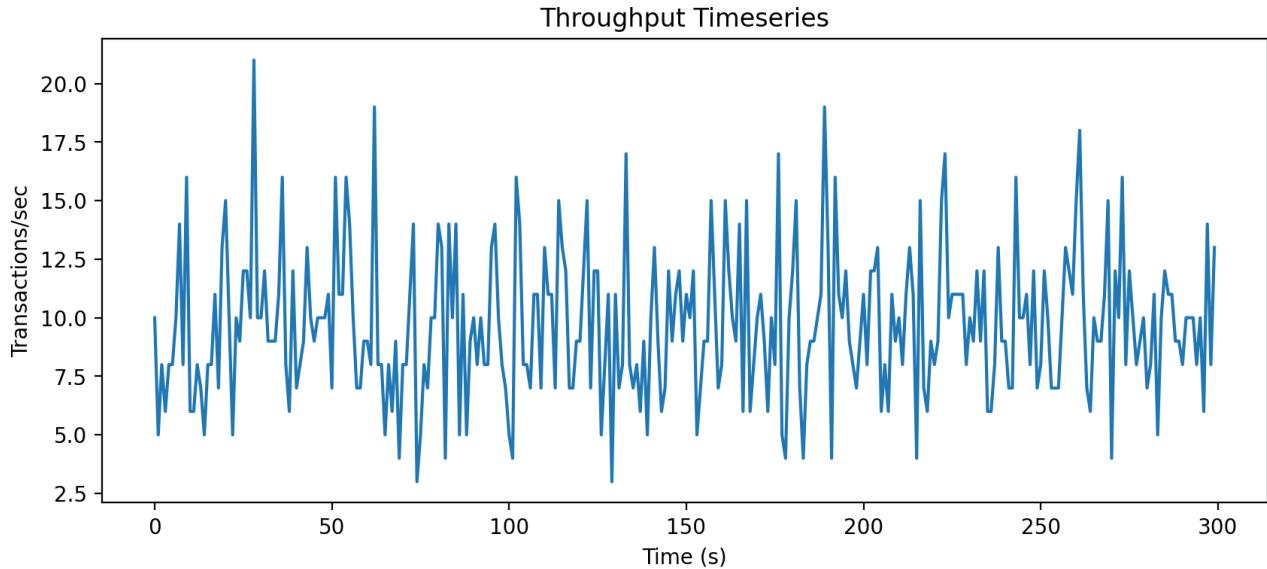


Figura 16: Série temporal do throughput do sistema.

4.10 TPS vs Número de Nós na Rede

Objetivo: Comparar o TPS com o número de nós ativos, mostrando como o desempenho evolui conforme a infraestrutura cresce.

Resultados: Crescimento aproximadamente linear do TPS com o aumento de nós até atingir saturação, onde limitações de recursos restringem ganhos de desempenho.

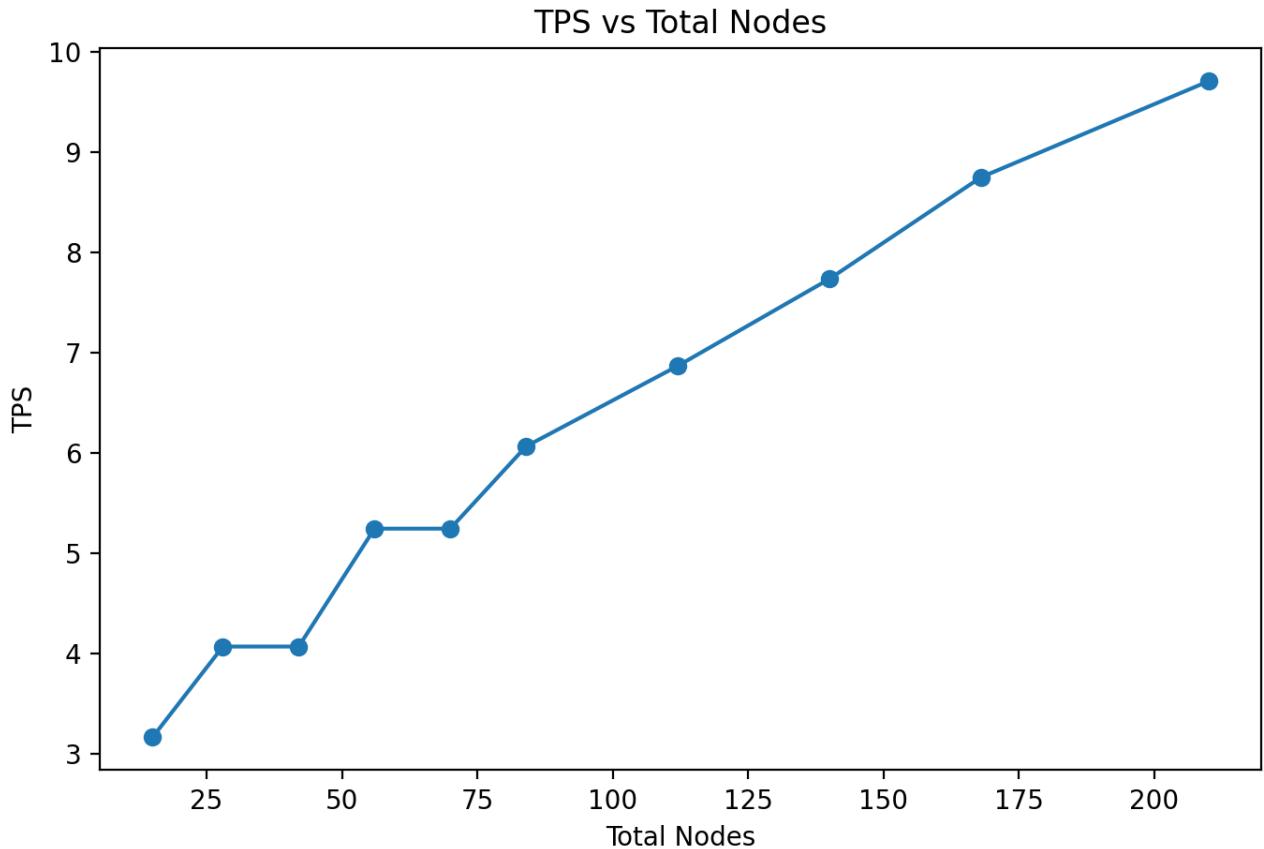


Figura 17: TPS em função do número de nós ativos na rede.

4.11 Discussão dos Resultados de Performance

Os resultados demonstram que o sistema apresenta características desejáveis para aplicações de armazenamento em nuvem distribuído:

- **Alta disponibilidade:** Superior a 80%, garantindo confiabilidade operacional
- **Escalabilidade horizontal:** Crescimento da infraestrutura conforme demanda
- **Trade-off throughput/latência:** Evidenciado no mapa de calor, importante para dimensionamento
- **Baixa taxa de forks:** Mecanismo de consenso eficaz mesmo sob condições adversas

5 Conclusões Gerais

Este relatório apresentou uma análise abrangente do sistema, desde a modelagem comportamental até métricas de performance operacional.

5.1 Análise Comportamental

A implementação da Teoria da Perspectiva mostrou-se eficaz após ajustes nas distribuições de parâmetros. A taxa de aceitação de 36% na segunda iteração demonstra comportamento mais realista, onde decisões são influenciadas por pequenas variações em torno do patamar de mercado.

5.2 Performance e Escalabilidade

As métricas de performance validam a arquitetura proposta, demonstrando que o sistema é:

- **Escalável:** Crescimento horizontal eficiente
- **Eficiente:** Uso otimizado de recursos
- **Confiável:** Alta disponibilidade e baixa taxa de forks
- **Robusto:** Desempenho consistente em diversos cenários

5.3 Trabalhos Futuros

Recomenda-se para trabalhos futuros:

1. Otimização de latência em cenários de alta taxa de transações
2. Análise sob ataques maliciosos e condições bizantinas
3. Implementação de ajuste dinâmico de parâmetros via ML
4. Avaliação em ambiente de produção com dados reais
5. Refinamento adicional do modelo comportamental com dados empíricos