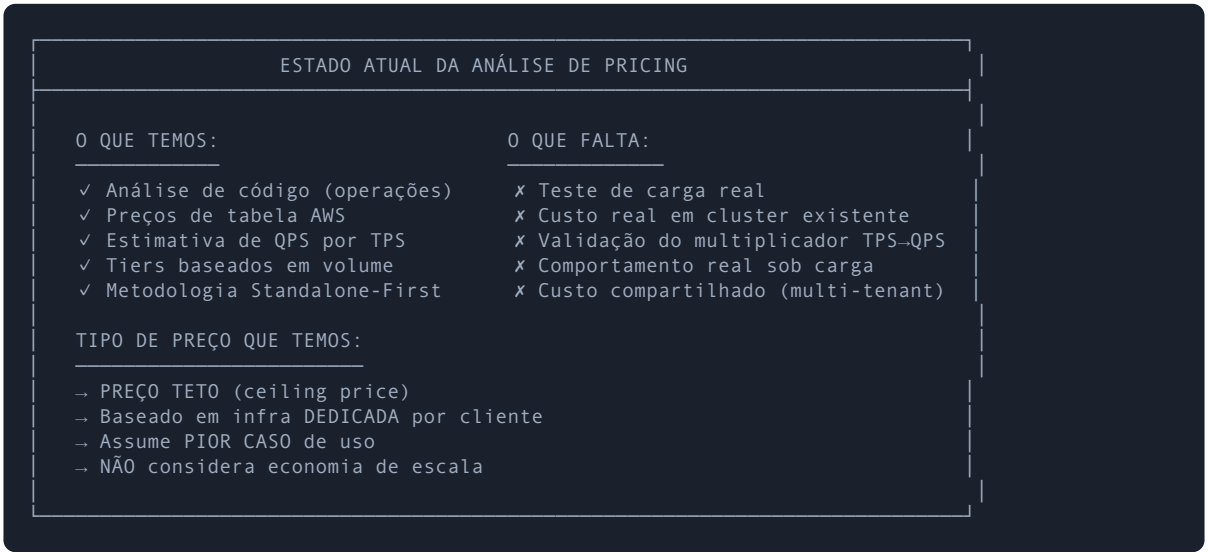
Limitações da Análise e O Que o Teste de Carga Vai Mudar

Documento: Análise de Precisão e Roadmap de Validação **Contexto:** Discussão com Jeff sobre Terraform, KubeCost e testes de carga **Data:** Janeiro 2026

1. O Que Temos Hoje vs O Que Precisamos



2. Classificação das Premissas

2.1 O Que Sabemos com Certeza (FATO)

Premissa	Fonte	Confiança
Componentes de infra (PostgreSQL, Redis, RabbitMQ)	docker-compose.yml, CLAUDE.md	100%
Operações de ingestion (SETNX, SELECT, INSERT)	Código fonte analisado	95%
RabbitMQ = 1 msg por job (não por txn)	event_publisher.go	95%
Preços de tabela AWS	aws.amazon.com/pricing	100%

2.2 O Que Estimamos (HIPÓTESE)

Premissa	Nossa Estimativa	O Que Pode Mudar	Impacto
1 TPS = 11 QPS	Análise de código	Pode ser 8-15 dependendo de otimizações	±30% no sizing
Fator de pico = 3x	Padrão da indústria	Pode ser 2x ou 5x dependendo do cliente	±40% no sizing
Match rate = 100%	Pior caso	Na prática 60-90%	-10% a -40% nas operações
Batch size = 1000 txns	Assumido	Pode ser 100 ou 10.000	Impacta RabbitMQ (irrelevante)

2.3 O Que Não Sabemos (INCÓGNITA)

Incógnita	Por Que Importa	Como Descobrir
Custo real no cluster K8S existente	Pode ser MUITO menor que standalone	KubeCost + tags
Overhead de GORM vs SQL raw	Pode adicionar 20-50% de queries	Profiling em prod
Comportamento de cache do PostgreSQL	Hit rate alto = menos I/O	Teste de carga
Latência real das operações	Pode limitar throughput	Teste de carga
Custo de rede entre componentes	Pode ser significativo em alto volume	AWS Cost Explorer

---

3. O Que o Teste de Carga Vai Responder

3.1 Perguntas que o Teste de Carga Responde

```
┌─────────────────────────────────────────────────────────────────────┐
│                  PERGUNTAS PARA O TESTE DE CARGA                    │
├─────────────────────────────────────────────────────────────────────┤
│                                                                     │
│   1. THROUGHPUT REAL                                                │
│      "Quantas transações/segundo a infra X consegue processar?"     │
│      → Hoje assumimos: db.t3.medium aguenta ~30 QPS                 │
│      → Teste vai mostrar: pode ser 20 ou 50                         │
│                                                                     │
│   2. MULTIPLICADOR TPS→QPS REAL                                     │
│      "1 TPS do cliente gera quantas queries reais?"                 │
│      → Hoje assumimos: 11 QPS                                       │
│      → Teste vai mostrar: número exato                              │
│                                                                     │
│   3. PONTO DE SATURAÇÃO                                             │
│      "A partir de qual TPS a infra começa a degradar?"              │
│      → Hoje assumimos: baseado em specs da AWS                      │
│      → Teste vai mostrar: gargalo real (CPU? Memória? I/O?)         │
│                                                                     │
│   4. COMPORTAMENTO SOB PICO                                         │
│      "O que acontece com 3x a carga normal?"                        │
│      → Hoje assumimos: fator de 3x é suficiente                     │
│      → Teste vai mostrar: se precisa de mais margem                 │
│                                                                     │
│   5. CUSTO POR TRANSAÇÃO REAL                                       │
```

```
|      "Quanto custa processar 1M transações?"                                  |
|      → Hoje estimamos: R$ X baseado em preço de tabela                        |
|      → Teste + KubeCost vai mostrar: custo real                               |
|                                                                               |
└──────────────────────────────────────────────────────────────────────────────┘
```

3.2 Cenários de Resultado do Teste

Cenário A: Teste confirma nossas estimativas (±20%)

```
Resultado: Nossa análise está correta
Ação: Manter pricing, ajustar margens se necessário
Impacto: Baixo - validação do modelo
```

Cenário B: Infra aguenta MAIS do que estimamos

```
Resultado: db.t3.medium aguenta 50 QPS, não 30
Ação: Podemos usar instâncias menores OU aumentar margem
Impacto: Positivo - custo cai, margem sobe
Exemplo: Starter pode custar R$ 800/mês ao invés de R$ 1.150/mês
         → Margem sobe de 71% para 80%
```

Cenário C: Infra aguenta MENOS do que estimamos

```
Resultado: db.t3.medium só aguenta 15 QPS
Ação: Precisamos de instâncias maiores OU reduzir volume dos tiers
Impacto: Negativo - custo sobe, margem cai
Exemplo: Starter precisa de db.t3.large
         → Custo sobe de R$ 1.150 para R$ 1.500/mês
         → Margem cai de 71% para 62%
```

Cenário D: Multiplicador TPS→QPS está errado

```
Resultado: 1 TPS = 20 QPS, não 11
Ação: Recalcular todos os tiers
Impacto: Significativo - muda todo o dimensionamento
```

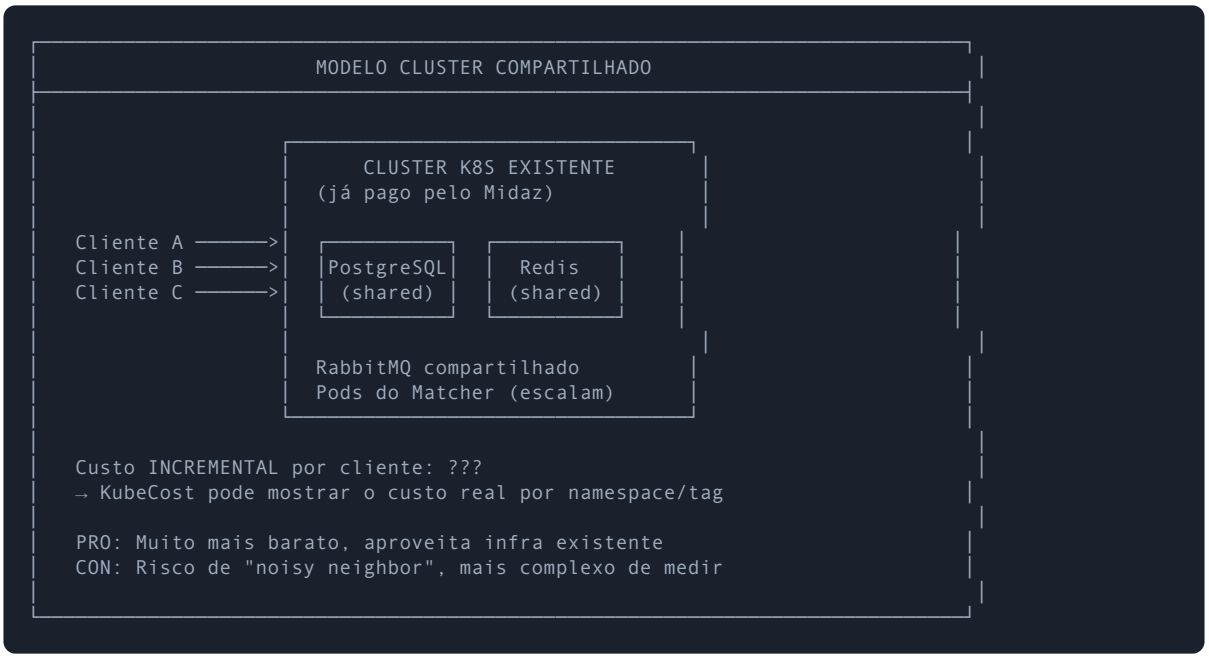
---

4. Standalone vs Cluster Compartilhado

4.1 Nossa Análise Atual: Standalone-First

```
┌──────────────────────────────────────────────────────────────────────────────┐
│                        MODELO STANDALONE-FIRST                               │
├──────────────────────────────────────────────────────────────────────────────┤
│                                                                              │
│   Cliente A                  Cliente B                  Cliente C            │
│   ┌──────────────┐           ┌──────────────┐           ┌──────────────┐     │
│   │ PostgreSQL   │           │ PostgreSQL   │           │ PostgreSQL   │     │
│   │ Redis        │           │ Redis        │           │ Redis        │     │
│   │ RabbitMQ     │           │ RabbitMQ     │           │ RabbitMQ     │     │
│   │ Fargate      │           │ Fargate      │           │ Fargate      │     │
│   └──────────────┘           └──────────────┘           └──────────────┘     │
│                                                                              │
│   Custo: R$ 1.150/mês        Custo: R$ 1.150/mês        Custo: R$ 1.150      │
│   TOTAL: R$ 3.450/mês para 3 clientes                                        │
│                                                                              │
│   PRO: Isolamento total, fácil de calcular, sem "noisy neighbor"             │
│   CON: Mais caro, não aproveita economia de escala                           │
│                                                                              │
└──────────────────────────────────────────────────────────────────────────────┘
```

4.2 Alternativa: Cluster Compartilhado (o que Jeff mencionou)



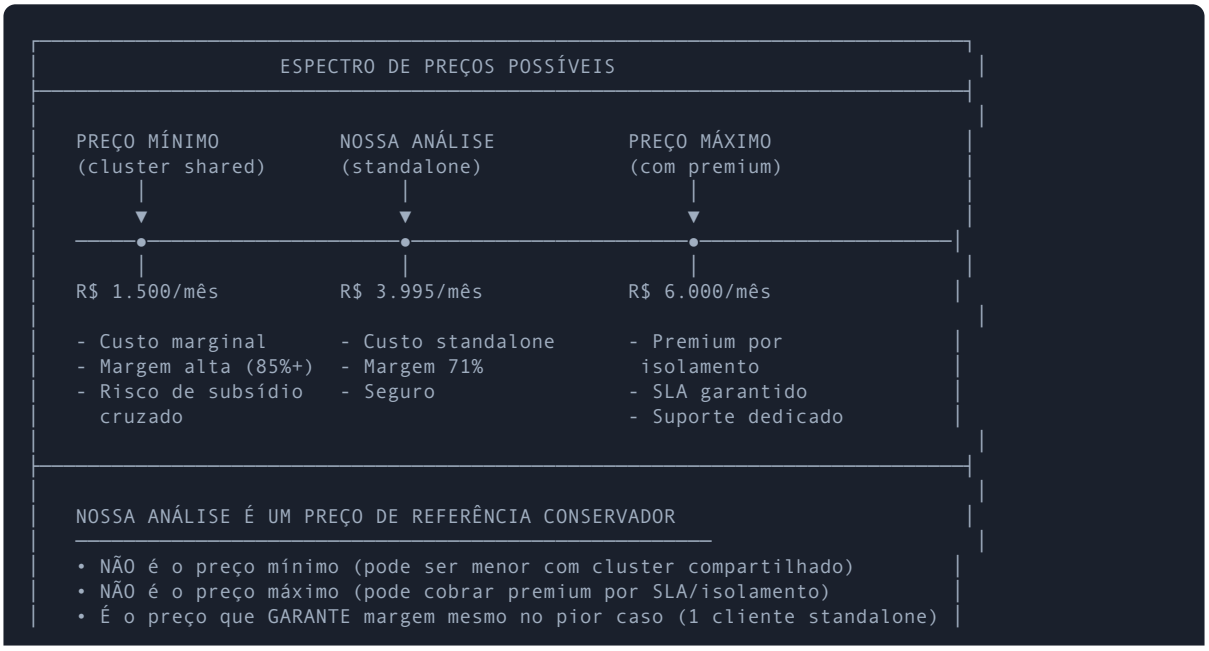
4.3 Impacto no Pricing

Modelo	Custo Estimado (Starter)	Preço Sugerido	Margem
Standalone (nossa análise)	R$ 1.150/mês	R$ 3.995/mês	71%
Cluster compartilhado (estimativa)	R$ 300-500/mês?	R$ 2.995/mês?	83-90%?

O que o KubeCost vai mostrar: - Custo REAL de CPU/memória por pod do Matcher - Custo de storage incremental - Custo de rede - Permite calcular custo marginal por cliente

---

5. Como Nosso Preço Se Posiciona



```
└──────────────────────────────────────────────────────────────────────┘
```

6. Precisão da Análise Atual

6.1 Matriz de Confiança

Componente da Análise	Confiança	Variação Possível	Após Teste de Carga
Preço de tabela AWS	100%	0%	Não muda
Operações por transação	75%	±30%	Validado
QPS suportado por instância	60%	±50%	Validado
Fator de pico	50%	±40%	Calibrado por cliente
Custo de rede/outros	40%	±100%	Medido
CUSTO TOTAL ESTIMADO	**65%**	**±35%**	**±10%**

6.2 Range de Preço Real

```
Tier Starter (nossa análise: R$ 3.995/mês)

Cenário pessimista (custo +35%):
├─ Custo real: R$ 1.550/mês
├─ Preço mantido: R$ 3.995/mês
└─ Margem: 61% (ainda OK)

Cenário otimista (custo -35%):
├─ Custo real: R$ 750/mês
├─ Preço mantido: R$ 3.995/mês
└─ Margem: 81% (excelente)

Cenário cluster compartilhado:
├─ Custo real: R$ 300-500/mês
├─ Preço pode baixar: R$ 2.495/mês
└─ Margem: 80%+ (competitivo)
```

7. Roadmap de Validação

7.1 Curto Prazo (Próximas 2-4 semanas)

```
┌──────────────────────────────────────────────────────────────────────┐
│  FASE 1: VALIDAÇÃO COM O TIME DE DEV                                 │
├──────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  ☐ Enviar documento de validação TPS→QPS para o time                 │
│  ☐ Confirmar operações por transação                                 │
│  ☐ Confirmar se há overhead não mapeado (logs, métricas, etc.)       │
│  ☐ Entender se há otimizações planejadas (batch, cache)              │
│                                                                      │
│  ENTREGÁVEL: Multiplicador TPS→QPS validado                          │
```

7.2 Médio Prazo (1-2 meses)

```
┌─────────────────────────────────────────────────────────────────────┐
│  FASE 2: TESTE DE CARGA                                             │
├─────────────────────────────────────────────────────────────────────┤
│                                                                     │
│  ☐ Configurar ambiente de teste (idealmente no cluster existente)   │
│  ☐ Definir cenários: 1K, 10K, 100K, 1M transações                   │
│  ☐ Medir:                                                           │
│    - Throughput máximo por configuração de infra                    │
│    - Latência p50, p95, p99                                         │
│    - Uso de CPU/memória                                             │
│    - QPS real nos bancos                                            │
│  ☐ Identificar gargalos                                             │
│                                                                     │
│  ENTREGÁVEL: Tabela real de capacidade por instância                │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

7.3 Médio Prazo (Paralelo)

```
┌─────────────────────────────────────────────────────────────────────┐
│  FASE 3: MEDIÇÃO DE CUSTO REAL (KubeCost)                           │
├─────────────────────────────────────────────────────────────────────┤
│                                                                     │
│  ☐ Configurar KubeCost no cluster K8S                               │
│  ☐ Criar tags/labels para o Matcher                                 │
│  ☐ Rodar workload representativo por 30 dias                        │
│  ☐ Extrair custo real:                                              │
│    - CPU/memória dos pods                                           │
│    - Storage (PVCs)                                                 │
│    - Rede                                                           │
│    - Banco de dados (por query se possível)                         │
│                                                                     │
│  ENTREGÁVEL: Custo real por transação processada                    │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

7.4 Longo Prazo (3+ meses)

```
┌─────────────────────────────────────────────────────────────────────┐
│  FASE 4: REFINAMENTO DO PRICING                                     │
├─────────────────────────────────────────────────────────────────────┤
│                                                                     │
│  ☐ Comparar custo estimado vs custo real                            │
│  ☐ Ajustar tiers se necessário                                      │
│  ☐ Definir modelo final:                                            │
│    - Standalone (preço premium, isolamento)                         │
│    - Compartilhado (preço competitivo, multi-tenant)                │
│    - Híbrido (starter compartilhado, scale standalone)              │
│  ☐ Definir overage pricing baseado em custo real                    │
│                                                                     │
│  ENTREGÁVEL: Pricing validado e pronto para go-to-market            │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

---

8. Recomendação

O que fazer AGORA com o pricing atual:

1. **Usar como referência interna** - Os números são bons o suficiente para planejamento
2. **Não publicar preços finais** - Aguardar validação do teste de carga
3. **Usar para conversas com clientes** - "Estimamos na faixa de R$ 3-4K/mês para o tier inicial"
4. **Planejar teste de carga** - É o próximo passo crítico

O que NÃO fazer:

1. ❌ Assumir que os preços estão 100% corretos
2. ❌ Fechar contratos de longo prazo sem validação
3. ❌ Ignorar a possibilidade de cluster compartilhado (pode ser muito mais barato)

Changelog

Versão	Data	Alteração
1.0	Jan 2026	Versão inicial baseada na discussão com Jeff