

Análise de Pricing: Flowker

Data: Janeiro 2025 **Versão:** 2.0 **Autor:** Análise assistida por IA (Claude/Gauchito) **Status:** Preliminar - Aguardando validação técnica

1. Objetivo

Desenvolver um modelo de pricing para o **Flowker** (workflow orchestration) que seja:

- **Sustentável:** Cubra custos mesmo no pior cenário (único cliente, infra dedicada)
- **Competitivo:** Alinhado com mercado de orquestração de pagamentos
- **Validável:** Com premissas claras que possam ser testadas com dados reais

2. Contexto Técnico e Limitações

2.1 O que temos disponível

Recurso	Status	Uso na Análise
Código fonte do Flowker	✓ Disponível	Mapeamento de operações por workflow
docker-compose.yml	✓ Disponível	Identificação de componentes de infra
.env.example	✓ Disponível	Configurações padrão
Terraform do Flowker	✗ Não existe	N/A
Terraform do Midaz	✓ Existe	Referência para sizing de produção
Dados de produção	✗ Não disponíveis	N/A
KubeCost	✗ Não instalado	Validação futura

2.2 Limitações identificadas

1. **Sem Terraform específico do Flowker:** Não temos definição de infra de produção
2. **Sem dados de custo real:** Não há histórico de billing para o Flowker
3. **Infra compartilhada na prática:** Produtos Larian compartilharão clusters K8s
4. **Sem load tests:** Não validamos throughput real por configuração

2.3 Estratégias adotadas para cada limitação

Limitação	Estratégia Adotada	Grau de Confiança
Sem Terraform	Usar docker-compose + cloud pricing público	🟡 Médio
Sem dados de custo real	Estimar via preços públicos AWS/MongoDB Atlas	🟡 Médio
Infra compartilhada	Precificar como standalone (worst case)	🟢 Alto (metodologia)
Sem load tests	Usar análise de código para estimar ops/workflow	🟠 Baixo-Médio

3. Metodologia

3.1 Princípio Adotado: “Standalone-First Pricing”

PREMISSA FUNDAMENTAL
Cada produto deve ser precificado assumindo que ele é o ÚNICO produto usando a infraestrutura. Isso porque:
1. Um dia podemos ter um cliente usando apenas o Flowker
2. Produtos precisam se pagar sozinhos, sem subsídio cruzado
3. Se infra for compartilhada, a margem extra é BÔNUS, não dependência

Grau de Confiança na Metodologia: 🟢 ALTO

Justificativa: Este é um princípio de negócio conservador e amplamente aceito em pricing de SaaS multi-produto. Não depende de estimativas técnicas.

3.2 Fontes de Dados Utilizadas

Dado	Fonte	Confiança
Componentes de infra	<code>docker-compose.yml</code> do Flowker	● Alto
Operações por workflow	Código fonte (<code>validation_workflow.go</code> , <code>activities.go</code>)	● Médio
Preços de cloud	AWS Pricing Calculator (Janeiro 2025)	● Alto
Preços MongoDB Atlas	Pricing público MongoDB (Janeiro 2025)	● Alto
Preços Temporal Cloud	Pricing público Temporal (Janeiro 2025)	● Alto

4. Análise de Custos

4.1 Componentes de Infraestrutura (Standalone)

Baseado em `docker-compose.yml` do Flowker, os componentes necessários são:

Componente	Serviço no Docker	Equivalente Produção
PostgreSQL	<code>postgres:17</code>	AWS RDS PostgreSQL
MongoDB	<code>mongo:8</code>	MongoDB Atlas
Valkey (Redis)	<code>valkey/valkey:8</code>	AWS ElastiCache
Temporal	<code>temporalio/auto-setup:1.29</code>	Temporal Cloud
RabbitMQ	<code>rabbitmq:4</code>	Amazon MQ
Vault	<code>hashicorp/vault:1.18</code>	HashiCorp Cloud

4.2 Sizing de Produção (Estimado)

Premissa: Cliente típico com ~25.000 workflows/mês (Growth tier)

Componente	Config Mínima Produção	Justificativa
PostgreSQL	<code>db.r6g.large</code> (2 vCPU, 16GB)	Tenancy + metadados, RLS habilitado

Componente	Config Mínima Produção	Justificativa
MongoDB	M30 (2 vCPU, 8GB)	Storage principal de workflows
Valkey	cache.r6g.large (2 vCPU, 13GB)	Locks + cache de alta frequência
Temporal	Cloud - Base	Orquestração de workflows
RabbitMQ	mq.m5.large (2 vCPU, 8GB)	Event processing
Compute	2x c6g.large (2 vCPU, 4GB each)	Workers + API

Grau de Confiança no Sizing: 🟡 BAIXO-MÉDIO

Justificativa: Sem Terraform e sem load tests, o sizing é baseado em: - Configurações típicas para workloads similares - Análise do código (bounded contexts, complexidade) - Margem de segurança conservadora

⚠️ **Ação recomendada:** Validar sizing com load test antes de ir para produção.

4.3 Custos Mensais Estimados (USD)

Componente	Preço/mês (USD)	Fonte	Confiança
PostgreSQL (RDS db.r6g.large)	\$108	AWS Pricing Calculator	🟢 Alto
MongoDB (Atlas M30)	\$389	MongoDB Pricing Page	🟢 Alto
Valkey (ElastiCache cache.r6g.large)	\$94	AWS Pricing Calculator	🟢 Alto
Temporal Cloud (base)	\$25	Temporal Pricing	🟢 Alto
RabbitMQ (Amazon MQ mq.m5.large)	\$180	AWS Pricing Calculator	🟢 Alto
Compute (EKS 2x c6g.large)	\$122	AWS Pricing Calculator	🟢 Alto
TOTAL	\$918	-	🟢 Alto

Conversão: $918 \times R\ 5.00 = \text{R\$ } 4.590/\text{mês}$

Grau de Confiança no Custo Total: 🟢 ALTO

Justificativa: Os preços são públicos e verificáveis. A incerteza está no **sizing**, não no preço unitário.

4.4 Custo Variável por Workflow

Análise baseada no código fonte:

Arquivo: internal/bounded_contexts/execution_management/infrastructure/temporal/workflows/validation_workflow.go

Operações típicas por workflow:

- └─ 1x Workflow start (Temporal)
- └─ Nx Activities em paralelo (1-10 típico, assumindo 5)
 - └─ 1x Lock acquire (Valkey)
 - └─ 1x Provider validation call
 - └─ 1x Audit write (MongoDB)
 - └─ 1x Lock release (Valkey)
- └─ 1x Aggregation result
- └─ 1x Workflow complete

Operação	Quantidade/ Workflow	Custo Unitário	Custo Total
Temporal actions	~9	\$0.000025	\$0.000225
MongoDB writes	~6	~\$0.000001	\$0.000006
Valkey operations	~12	~\$0.0000001	\$0.0000012
TOTAL	-	-	~\$0.000235

Conversão: $0.000235 \times R\ 5.00 = \sim R\$ 0.0012/\text{workflow}$

Grau de Confiança no Custo Variável: 🟡 BAIXO-MÉDIO

Justificativa: - ✅ Estrutura de operações vem do código (confiável) - ⚠️
Quantidade de activities por workflow é estimada (5 assumido) - ⚠️ Custo
de MongoDB/Valkey é aproximado (sem dados de billing)

⚠️ **Ação recomendada:** Executar benchmark script para validar operações reais.

5. Análise de Break-Even

5.1 Fórmula

Break-even (workflows) = $\text{Custo Fixo Mensal} / (\text{Preço por Workflow} - \text{Custo Variável})$

5.2 Cenários por Preço

Preço/Workflow	Margem Unitária	Break-even	Clientes Equiv.*
R\$ 0.05	R\$ 0.0488	94.057 wf	~4 clientes
R\$ 0.08	R\$ 0.0788	58.249 wf	~2-3 clientes
R\$ 0.10	R\$ 0.0988	46.457 wf	~2 clientes
R\$ 0.15	R\$ 0.1488	30.847 wf	~1-2 clientes
R\$ 0.20	R\$ 0.1988	23.091 wf	~1 cliente

*Assumindo cliente médio com 25.000 workflows/mês

Grau de Confiança no Break-Even: 🟡 **MÉDIO**

Justificativa: Depende das estimativas de custo fixo (alto) e variável (baixo-médio).

6. Modelo de Pricing Recomendado

6.1 Estrutura: Base + Variável

Tier	Volume/mês	Preço Base	Preço/Workflow	Total (25K wf)
Starter	Até 1.000	Grátis	Grátis	R\$ 0
Growth	1K - 25K	R\$ 2.990	R\$ 0.08	R\$ 4.990
Scale	25K - 200K	R\$ 9.990	R\$ 0.04	R\$ 13.990 (100K)
Enterprise	200K+	Custom	Negociável	Custom

6.2 Racional do Pricing

Tier Starter (Grátis) - Objetivo: Aquisição e experimentação - Limite baixo para evitar abuso - Custo absorvido como CAC

Tier Growth (R\$ 2.990 + R\$ 0.08/wf)

Validação para cliente com 25K workflows:

— Receita:	$R\$ 2.990 + (25.000 \times R\$ 0.08) = R\$ 4.990$
— Custo:	$R\$ 4.590 \text{ (fixo)} + R\$ 30 \text{ (var)} = R\$ 4.620$
— Margem:	$R\$ 370 \text{ (7.4\%)}$

- Margem positiva mesmo com 1 cliente standalone
- Competitivo vs. mercado (Spreadly ~\$500-1000 base)

Tier Scale (R\$ 9.990 + R\$ 0.04/wf)

Validação para cliente com 100K workflows:

— Receita:	$R\$ 9.990 + (100.000 \times R\$ 0.04) = R\$ 13.990$
------------	--

— Custo: R\$ 4.590 (fixo) + R\$ 120 (var) = R\$ 4.710
— Margem: R\$ 9.280 (66.3%)

- Margem saudável para escala
- Desconto de volume incentiva crescimento

Grau de Confiança no Modelo de Pricing: 🟡 MÉDIO

Justificativa: - ✅ Metodologia standalone é sólida - ✅ Estrutura (base + variável) é padrão de mercado - ⚠️ Valores específicos dependem de validação de custos

7. Cenários de Margem

7.1 Worst Case (Standalone)

Premissas:

- 1 único cliente Growth (25K workflows)
- Infraestrutura 100% dedicada ao Flowker
- Preço: R\$ 2.990 + R\$ 0.08/wf

Resultado:

— Receita: R\$ 4.990/mês
— Custo: R\$ 4.620/mês
— Margem: R\$ 370/mês (7.4%)
— Status: ✅ Sustentável (margem positiva)

7.2 Best Case (Compartilhado)

Premissas:

- Múltiplos clientes
- Infraestrutura compartilhada (Flowker = 25% do cluster)
- Preço: R\$ 2.990 + R\$ 0.08/wf

Resultado (por cliente Growth):

— Receita: R\$ 4.990/mês
— Custo: R\$ 1.155/mês (25% da infra)
— Margem: R\$ 3.835/mês (76.9%)
— Status: ✅ Excelente

7.3 Comparativo Visual

MARGEM BRUTA POR CENÁRIO





Grau de Confiança nos Cenários: 🟡 MÉDIO

Justificativa: Depende da validação dos custos, mas a lógica é sólida.

8. Pontos em Aberto (Para Validação)

8.1 Prioridade Alta

Item	Ação Necessária	Responsável Sugerido	Impacto
Sizing de produção	Load test com 25K-100K workflows	Engenharia	Alto
Custo variável real	Executar benchmark script	Engenharia	Médio
KubeCost	Instalar e configurar tags	Engenharia/DevOps	Alto

8.2 Prioridade Média

Item	Ação Necessária	Responsável Sugerido	Impacto
Terraform do Flowker	Criar IaC de produção	Engenharia	Médio
Validação de mercado	Pesquisa com prospects	Produto/Vendas	Médio
Análise competitiva detalhada	Cotações reais de concorrentes	Produto	Baixo

8.3 Benchmark Script Disponível

Foi criado um script de benchmark em:

```
/monorepo/apps/flowker/scripts/benchmark_workflow_cost.sh
```

Este script: - Coleta métricas de baseline (MongoDB, Valkey, Temporal) - Executa N workflows - Calcula operações reais por workflow - Estima custo baseado em cloud pricing

⚠ **Requer:** Docker running + infra local do Flowker

9. Resumo de Confiança

Por Componente da Análise

Componente	Confiança	Justificativa
Metodologia (standalone-first)	● Alto	Princípio de negócio, não técnico
Componentes de infra	● Alto	Baseado em docker-compose real
Preços de cloud (unitários)	● Alto	Pricing público verificável
Sizing de produção	● Baixo-Médio	Sem Terraform/load tests
Custo variável por workflow	● Baixo-Médio	Baseado em código, não em métricas
Modelo de pricing (estrutura)	● Alto	Padrão de mercado
Modelo de pricing (valores)	● Médio	Depende de validação de custos

Confiança Geral da Análise: ● **MÉDIO (65-75%)**

O que aumentaria a confiança para ● Alto (>90%): 1. Load tests validando sizing 2. KubeCost com dados reais de 30+ dias 3. Benchmark script executado com resultados documentados

10. Próximos Passos Sugeridos

Curto Prazo (1-2 semanas)

1. **Revisão deste documento** com Jefferson e Fred
2. **Executar benchmark script** quando Docker estiver disponível
3. **Instalar KubeCost** no cluster K8s existente

Médio Prazo (2-4 semanas)

1. **Load test** do Flowker com 25K-100K workflows
2. **Ajustar sizing** baseado nos resultados
3. **Criar Terraform** do Flowker para produção

Validação Final

1. **Recalcular pricing** com dados reais
 2. **Validar com prospects** (pesquisa de willingness-to-pay)
 3. **Definir pricing público** para lançamento
-

11. Conclusão

Esta análise apresenta um modelo de pricing para o Flowker baseado em:

- **Metodologia conservadora** (standalone-first) que garante sustentabilidade
- **Dados disponíveis** (código, docker-compose, pricing público)
- **Transparência sobre limitações** e grau de confiança

Os valores recomendados (Growth: R\$ 2.990 + R\$ 0.08/wf) geram margem positiva mesmo no pior cenário, com upside significativo quando a infraestrutura for compartilhada.

A principal incerteza está no sizing de produção, que pode ser resolvida com load tests e KubeCost. Recomendo priorizar essas validações antes de definir o pricing público final.

Documento gerado em Janeiro 2025 Análise preliminar - Aguardando validação técnica

Anexo A: Referências de Código

Workflow Principal

- `internal/bounded_contexts/execution_management/infrastructure/temporal/workflows/validation_workflow.go`
- Função: `ValidationWorkflow(ctx workflow.Context, input ValidationWorkflowInput)`

Activities

- `internal/bounded_contexts/execution_management/infrastructure/temporal/workflows/activities.go`
- Operações: Lock, Validate, Audit, Release

Configuração de Infra

- `docker-compose.yml` (raiz do projeto)
- `.env.example` (configurações padrão)

Anexo B: Pricing de Referência (Concorrentes)

Concorrente	Modelo	Preço Base	Preço Variável
Spreadly	Subscription + per- txn	~\$500/mês	\$0.10-0.30/txn
Primer	Per-transaction	-	\$0.05-0.15/txn
Gr4vy	Subscription	~\$1000/ mês	Incluído
Temporal Cloud	Per-action	\$25/mês	\$0.000025/ action

Anexo C: Cálculos Detalhados

Custo AWS (Janeiro 2025)

RDS PostgreSQL db.r6g.large - On-demand: $\$0.146/\text{hora} \times 730 \text{ horas} = \$106.58 \approx \$108/\text{mês}$ - Região: us-east-1

ElastiCache Valkey cache.r6g.large - On-demand: $\$0.128/\text{hora} \times 730 \text{ horas} = \$93.44 \approx \$94/\text{mês}$ - Região: us-east-1

Amazon MQ RabbitMQ mq.m5.large - On-demand: $\$0.246/\text{hora} \times 730 \text{ horas} = \$179.58 \approx \$180/\text{mês}$ - Região: us-east-1

EKS c6g.large (2 nodes) - On-demand: $\$0.0836/\text{hora} \times 730 \text{ horas} \times 2 = \$122.06 \approx \$122/\text{mês}$ - Região: us-east-1

MongoDB Atlas M30 - Pricing público: $\$0.54/\text{hora} \times 720 \text{ horas} = \$388.80 \approx \$389/\text{mês}$ - Região: AWS us-east-1