

11ª VALIDAÇÃO - SPRINT 57 - NOVOS PROBLEMAS IDENTIFICADOS

Data: 2025-11-19

Validador: Manus AI








Versão Testada: v3.7.0 (Sprint 57)

Build: Analytics (Sprint 57)

PM2 PID: 380976

RESULTADO DA VALIDAÇÃO:

PROGRESSO PARCIAL - NOVOS PROBLEMAS REVELADOS

Bug	Status Anterior (10ª)	Status Atual (11ª)	Mudança
#1 - Chat	 CORRIGIDO	 NÃO TESTADO	-
#2 - Follow-up	 CORRIGIDO	 NÃO TESTADO	-
#3 - Analytics	 MELHOROU	 NOVOS ERROS	 PROGRESSO + PROBLEMAS

Status:  PROGRESSO PARCIAL (URL corrigida, mas novos erros)

ANÁLISE DETALHADA:

O QUE FOI CORRIGIDO (Sprint 57):

1. URL do tRPC Client:

TypeScript

```
// ANTES (Sprint 56):
url: `${import.meta.env.VITE_API_URL || ''}/api/trpc`
// Resultado: "/api/trpc" ❌

// DEPOIS (Sprint 57):
```

```
url: `${window.location.origin}/api/trpc`  
// Resultado: "http://localhost:3001/api/trpc" ✓
```

2. Logs [SPRINT 57] Funcionando:

Plain Text

```
✓ [SPRINT 57] trPC fetch to: http://localhost:3001/api/trpc/...  
✓ [SPRINT 57] trPC response status: 400
```

3. Queries Sendo Executadas:

- ✓ monitoring.getCurrentMetrics
- ✓ tasks.list
- ✓ projects.list
- ✓ workflows.list
- ✓ templates.list
- ✓ prompts.list
- ✓ teams.list
- ✓ tasks.getStats
- ✓ workflows.getStats
- ✓ templates.getStats

✗ NOVOS PROBLEMAS IDENTIFICADOS:

1. Erro de Validação: "too_big"

JavaScript

```
TRPCClientError: [  
  {  
    "code": "too_big",  
    "maximum": 100,  
    "type": "number",  
    "inclusive": true,  
    "exact": false,  
    "message": "Number must be less than or equal to 100",  
    "path": ["limit"]  
  }  
]
```

Causa: Frontend está enviando `limit: 1000`, mas backend só aceita `limit: 100`

Queries afetadas:

- ❌ tasks.list
- ❌ projects.list
- ❌ workflows.list
- ❌ templates.list
- ❌ prompts.list
- ❌ teams.list

2. Erro de Timeout: "signal timed out"

JavaScript

```
TRPCClientError: signal timed out
```

Causa: Query `monitoring.getCurrentMetrics` demorou mais de 30s

3. Erro HTTP 400 (Bad Request)

Plain Text

```
Failed to load resource: the server responded with a status of 400 (Bad Request)
[SPRINT 57]: # "trpc response status: 400"
```



COMPARAÇÃO ENTRE VALIDAÇÕES:

10ª Validação (Sprint 56):

Plain Text

```
⌚ Carregando analytics... (infinito)
❌ Queries não retornam dados
❌ URL incorreta: "/api/trpc"
```

11ª Validação (Sprint 57):

Plain Text

```
✅ URL correta: "http://localhost:3001/api/trpc"
✅ Queries executam
```

- ✗ Erro de validação: limit 1000 > 100
- ✗ Timeout: monitoring.getCurrentMetrics
- ✗ HTTP 400: Bad Request

Conclusão: Sprint 57 **CORRIGIU a URL** mas **revelou problemas de validação no backend**.



CAUSA RAIZ DOS PROBLEMAS:

Problema 1: Limite de Paginação

Frontend (AnalyticsDashboard.tsx):

TypeScript

```
const tasksQuery = trpc.tasks.list.useQuery({
  limit: 1000, // ✗ Muito alto!
  offset: 0
});
```

Backend (validação Zod):

TypeScript

```
z.object({
  limit: z.number().max(100), // ✔ Máximo 100
  offset: z.number()
})
```

Conflito: Frontend pede 1000, backend aceita no máximo 100.

Problema 2: Timeout de Métricas

Query: monitoring.getCurrentMetrics

Timeout: 30 segundos (configurado na Sprint 57)

Resultado: Query demora mais de 30s e é abortada

Possíveis causas:

- Consulta SQL lenta
- Banco de dados sobrecarregado
- Conexão lenta
- Deadlock no banco



RECOMENDAÇÕES PARA SPRINT 58:

Prioridade 1: Corrigir Limite de Paginação (CRÍTICO)

TypeScript

```
// frontend/src/pages/AnalyticsDashboard.tsx

// ANTES:
const tasksQuery = trpc.tasks.list.useQuery({
  limit: 1000, // ✗
  offset: 0
});

// DEPOIS:
const tasksQuery = trpc.tasks.list.useQuery({
  limit: 100, // ✔ Máximo permitido pelo backend
  offset: 0
});
```

Aplicar em todas as queries:

- tasks.list
- projects.list
- workflows.list
- templates.list
- prompts.list
- teams.list

Prioridade 2: Investigar Timeout de Métricas

Bash

```
# No servidor
cd /var/www/orquestrador-ia-v3

# Verificar logs
pm2 logs orquestrador-v3 | grep "getCurrentMetrics"

# Testar query manualmente
curl http://localhost:3001/api/trpc/monitoring.getCurrentMetrics
```

Verificar tempo de resposta

```
time curl http://localhost:3001/api/trpc/monitoring.getCurrentMetrics
```

Opções:

1. Otimizar query SQL
2. Aumentar timeout (30s → 60s)
3. Implementar cache
4. Remover query se não for crítica

Prioridade 3: Implementar Graceful Degradation

TypeScript

```
// frontend/src/pages/AnalyticsDashboard.tsx

const metricsQuery = trpc.monitoring.getCurrentMetrics.useQuery(
  undefined,
  {
    retry: 2,
    retryDelay: 1000,
    timeout: 30000,
    onError: (error) => {
      console.warn('⚠ Metrics query failed, showing partial data');
    }
  }
);

// Exibir dados parciais mesmo se métricas falharem
if (metricsQuery.error) {
  return (
    <div>
      <h2>📊 Analytics Dashboard</h2>
      <p>⚠ Métricas do sistema indisponíveis</p>
      { /* Exibir outros dados que carregaram */ }
      {tasksQuery.data && <TasksCard data={tasksQuery.data} />}
      {projectsQuery.data && <ProjectsCard data={projectsQuery.data} />}
    </div>
  );
}
```

CHECKLIST PARA SPRINT 58:

Antes de pedir nova validação:

- ☐ Corrigir `limit: 1000` → `limit: 100` em todas as queries
- ☐ Investigar timeout de `monitoring.getCurrentMetrics`
- ☐ Otimizar query SQL ou aumentar timeout
- ☐ Implementar graceful degradation
- ☐ Testar localmente via `npm run dev`
- ☐ Confirmar que Analytics carrega em < 10s
- ☐ Verificar console sem erros "too_big"
- ☐ Verificar console sem erros "signal timed out"
- ☐ Deploy e restart PM2
- ☐ Testar via SSH tunnel

AVALIAÇÃO FINAL:

O QUE DEU CERTO:

- ✓ **URL corrigida:** `window.location.origin` funciona
- ✓ **Logs [SPRINT 57]:** Aparecem no console
- ✓ **Queries executam:** 10 queries sendo feitas
- ✓ **Timeout implementado:** 30 segundos
- ✓ **Error handling funciona:** Erros são capturados

O QUE AINDA PRECISA SER CORRIGIDO:

- ✗ **Limite de paginação:** Frontend pede 1000, backend aceita 100
- ✗ **Timeout de métricas:** Query demora >30s
- ✗ **HTTP 400:** Bad Request devido a validação
- ✗ **Sem graceful degradation:** Página fica em loading infinito
- ✗ **Dados não aparecem:** Usuário não vê nenhum dado

HISTÓRICO DE VALIDAÇÕES:

Validação	Bug #1	Bug #2	Bug #3	Observação
-----------	--------	--------	--------	------------

1ª - 7ª	✗	✗	✗	Tentativas iniciais
8ª	✓	✓	⚠	2/3 corrigidos! ★
9ª	?	?	✗	Regressão (typo)
10ª	?	?	⚠	Typo corrigido, backend lento
11ª	?	?	⚠	URL corrigida, validação falha



CONCLUSÃO:

A Sprint 57 foi um **PROGRESSO PARCIAL**:

Sucesso:

- ✓ Corrigiu a URL do tRPC client
- ✓ Queries executam corretamente
- ✓ Logs [SPRINT 57] funcionam
- ✓ Timeout implementado

Problemas Revelados:

- ✗ Frontend pede `limit: 1000`, backend aceita `limit: 100`
- ✗ Query `monitoring.getCurrentMetrics` demora >30s
- ✗ HTTP 400 (Bad Request) devido a validação
- ✗ Dados não aparecem para o usuário

Impacto:

- ✓ Bug #3 **PROGREDIU** (URL corrigida, queries executam)
- ⚠ **MAS** ainda não está funcional (validação falha)
- ? Bugs #1 e #2 não puderam ser revalidados

Próxima Sprint (58):

- Foco: Corrigir limite de paginação (1000 → 100)
- Prioridade: CRÍTICA

- Tempo estimado: 30 minutos
-

MENSAGEM PARA O DEV:

Parabéns por corrigir a URL! 🎉

A Sprint 57 foi bem-sucedida em fazer as queries executarem. Agora precisamos corrigir um problema simples de validação:

Problema: Frontend pede `limit: 1000` , mas backend só aceita `limit: 100` .

Solução: Mudar `limit: 1000` para `limit: 100` em todas as queries do AnalyticsDashboard.

Arquivos afetados:

- `frontend/src/pages/AnalyticsDashboard.tsx` (6 queries)

Tempo estimado: 15-30 minutos

Estamos MUITO PRÓXIMOS de resolver o Bug #3! 💪

Relatório gerado por: Manus AI

Data: 2025-11-19 09:03:00 GMT-3

Validação: 11ª tentativa

Status: ⚠️ PROGRESSO PARCIAL (URL corrigida, validação falha)