

AVISO IMPORTANTE: Este relatório foi interrompido. A análise subsequente revelou que a causa raiz do problema era um **build incorreto**, onde o código-fonte atualizado com `useMemo` não foi compilado para o arquivo de build final, apesar dos timestamps parecerem corretos. O arquivo `Analytics-CNXQ1dWw.js` continha o código antigo da Sprint 65, explicando por que o erro persistiu. A recomendação é forçar um novo build completo (`npm run build`) e reiniciar o PM2. O restante deste relatório reflete a análise feita *antes* desta descoberta final.

✖ 19ª VALIDAÇÃO - SPRINT 66 - FALHA CRÍTICA

Data: 20 de novembro de 2025

Horário: 14:50 (GMT-3)

Validador: Sistema de Testes Automatizado

Sprint: 66 - Implementação de `useMemo` para corrigir React Error #310

SUMÁRIO EXECUTIVO

Status: ✖ FALHA CRÍTICA - BUG #3 PERSISTE

Apesar da implementação da correção com `useMemo` no código-fonte, o **React Error #310 (infinite re-render loop)** ainda persiste na página de Analytics. A investigação revelou uma inconsistência crítica: o build `Analytics-CNXQ1dWw.js` que está sendo servido, embora tenha o nome e timestamp corretos, **não contém o código atualizado da Sprint 66**, continuando a executar a lógica da Sprint 55.



Problema Crítico Identificado:

- BUILD INCONSISTENTE:** O arquivo de build (`Analytics-CNXQ1dWw.js`) não corresponde ao código-fonte (`AnalyticsDashboard.tsx`). O build está servindo código antigo que não contém a lógica de `useMemo` nem os logs da Sprint 66.

ANÁLISE DETALHADA DA FALHA

1. Confirmação do Código-Fonte (CORRETO)

O código-fonte no servidor **está correto** e inclui a implementação de `useMemo`.

Bash

```
# Verificação do código-fonte no servidor
$ grep -n 'useMemo'
/home/flavio/webapp/client/src/components/AnalyticsDashboard.tsx

6:import React, { useState, useEffect, useMemo } from 'react';
373:const health = useMemo(() => {
406:const stats = useMemo(() => {
```

2. Análise do Build (INCORRETO)

O console do navegador mostra que o novo build `Analytics-CNXQ1dWw.js` está sendo carregado. No entanto, os logs de console e o erro persistente provam que este build foi gerado a partir de um código-fonte antigo.

JavaScript

```
// Console do browser - Logs da Sprint 55, não da 66!
[SPRINT 55]: # "Analytics queries starting..."
[SPRINT 55]: # "Calling tasks.getStats with empty object..."

// Erro persiste, originado do novo arquivo de build
Error: Minified React error #310
    at o (http://localhost:3001/assets/Analytics-CNXQ1dWw.js:1:9264 )
```

3. Análise dos Timestamps

Os timestamps indicavam que o build foi gerado *após* a modificação do código-fonte, o que tornou o diagnóstico inicial confuso. Isso sugere um problema no processo de build que não está pegando a última versão do arquivo, mesmo que os timestamps sugiram o contrário.

Plain Text

```
Código fonte: 16:23:59 (AnalyticsDashboard.tsx)
Build:        16:24:38 (Analytics-CNXQ1dWw.js)
```

Conclusão: O processo de build (`npm run build`) falhou em compilar a versão mais recente do código-fonte, resultando em um deploy de uma versão antiga e não corrigida.



STATUS DOS BUGS

| Bug | Descrição | Status Sprint 66 | Tentativas |
|-----|---------------------------------|----------------------------|------------|
| #1 | Chat - Send button | ✓ RESOLVIDO (Sprint 54) | 4 |
| #2 | Follow-up - Chat | ✓ RESOLVIDO (Sprint 54) | 4 |
| #3 | Analytics - React Error #310 | ✗ FALHOU | 7 |

RECOMENDAÇÕES PARA SPRINT 67

Prioridade CRÍTICA

1. Forçar Rebuild Completo do Projeto

O problema não está no código, mas no **processo de build**. É essencial garantir que o ambiente de build esteja limpo e que o código mais recente seja compilado.

Ações Recomendadas:

1. **Limpar Cache do Build:** Remover o diretório `node_modules/.vite` ou qualquer cache que o `npm` ou `vite` possa usar.
2. **Reinstalar Dependências (Opcional, mas recomendado):**
3. **Executar o Build Novamente:**
4. **Verificar o Novo Build:** Após o build, verificar se o conteúdo do novo arquivo JavaScript contém os logs `[SPRINT 66]`.
5. **Reiniciar o PM2:**

2. Validação Pós-Deploy

Após o deploy, realizar os seguintes passos de validação:

1. Limpar cache do navegador e fazer hard refresh.
2. Verificar o console do navegador para confirmar que os logs `[SPRINT 66]` estão aparecendo.
3. Confirmar que a página de Analytics carrega os 10 cards de dados sem o React Error #310.



NOTAS FINAIS

A implementação de `useMemo` na Sprint 66 está **correta no código-fonte**, mas uma falha no processo de build impediu que a correção fosse efetivamente deployada. O foco da Sprint 67 deve ser exclusivamente em **garantir a integridade do processo de build e deploy**.

Status: ⌚ Aguardando novo build e deploy para iniciar a 20ª rodada de validação.