

Relatório Técnico para Desenvolvimento - Orquestrador de IA v3.7.0

Bugs Críticos e Recomendações de Correção

Data: 16 de Novembro de 2025

Versão Testada: v3.7.0

Ambiente: Produção (31.97.64.43:2224 → localhost:3001)

Cobertura de Testes: 16/23 páginas (70%)

Duração dos Testes: ~150 minutos

🔴 BUGS CRÍTICOS (Prioridade Máxima)

Bug #1: Chat Principal - Envio de Mensagens Não Funciona

Severidade: CRÍTICA (Bloqueador)

Página: /chat

Persistência: v3.6.0 → v3.7.0 (não foi corrigido no upgrade)

Descrição

O chat principal está completamente inutilizável. Apesar do WebSocket estar conectado (estado OPEN confirmado), as mensagens não são enviadas quando o usuário pressiona Enter ou clica no botão "Enviar".

Evidências

Plain Text
WebSocket State: OPEN ✓ Connected: true ✓ Streaming: false ✘

Comportamento observado:

1. Usuário digita mensagem no textarea
2. Usuário pressiona Enter → Nada acontece
3. Usuário clica em "Enviar" → Nada acontece

4. Mensagem permanece no campo de texto
5. Nenhuma requisição é enviada ao servidor

Investigação Sugerida

1. Verificar event handlers:

- Handler de `keydown` (Enter) no textarea
- Handler de `click` no botão "Enviar"
- Verificar se eventos estão sendo registrados

2. Verificar função de envio:

- Função `sendMessage()` ou equivalente
- Validar se está sendo chamada
- Verificar logs do console do navegador

3. Verificar comunicação WebSocket:

- Método `ws.send()` está sendo chamado?
- Payload está sendo construído corretamente?
- Verificar logs do servidor WebSocket

4. Verificar estado do componente:

- Estado do React/Vue está sendo atualizado?
- Há algum bloqueio de UI?
- Verificar se há erros silenciosos

Código Suspeito

Procurar por:

JavaScript

```
// Handler de Enter
textarea.addEventListener('keydown', (e) => {
  if (e.key === 'Enter' && !e.shiftKey) {
    e.preventDefault();
    sendMessage(); // Esta função está sendo chamada?
  }
});

// Handler de botão Enviar
sendButton.addEventListener('click', () => {
  sendMessage(); // Esta função está sendo chamada?
```

```

});
```

```

// Função de envio
function sendMessage() {
  const message = textarea.value;
  if (!message.trim()) return;

  ws.send(JSON.stringify({ // ws.send() está sendo executado?
    type: 'message',
    content: message
}));
```

```

  textarea.value = ''; // Campo está sendo limpo?
}

```

Correção Sugerida

1. Adicionar logs de debug para identificar onde o fluxo está quebrando
2. Verificar se há algum `preventDefault()` ou `stopPropagation()` bloqueando eventos
3. Validar se o WebSocket está realmente pronto para enviar (`readyState === 1`)
4. Testar com diferentes navegadores

Bug #2: Chat de Follow-up - Envio Não Funciona

Severidade: CRÍTICA (Bloqueador)

Página: /prompts (modal de execução)

Persistência: v3.6.0 → v3.7.0

Descrição

Após executar um prompt com sucesso, o sistema exibe um campo para continuar a conversa (follow-up). No entanto, este campo também não funciona. Mensagens digitadas não são enviadas.

Evidências

Elementos visíveis:

- Campo de input visível
- Placeholder: "Continue a conversa... (Enter para enviar, Shift+Enter para nova linha)"
- Botão "Enviar" presente
- Funcionalidade: Completamente quebrada

Comportamento observado:

1. Prompt executado com sucesso (streaming funciona perfeitamente)
2. Campo de follow-up aparece
3. Usuário digita mensagem
4. Usuário pressiona Enter → Nada acontece
5. Usuário clica em "Enviar" → Nada acontece

Investigação Sugerida

Hipótese: Provavelmente usa o mesmo componente/código do Chat principal (Bug #1)

1. Verificar se é o mesmo componente do chat principal
2. Se sim, a correção do Bug #1 deve resolver este também
3. Se não, aplicar mesma investigação do Bug #1

Correção Sugerida

1. Corrigir Bug #1 primeiro
2. Testar se follow-up também foi corrigido
3. Se não, aplicar mesma correção

Bug #3: Analytics - Erro de Renderização

Severidade: ALTA (Bloqueador)

Página: /analytics

Persistência: v3.6.0 → v3.7.0

Descrição

A página de Analytics apresenta erro de renderização e não carrega nenhum conteúdo.

Evidências

Mensagem de erro exibida:

Plain Text

Erro ao Carregar Página

Ocorreu um erro inesperado ao renderizar esta página.
Por favor, tente recarregar a página ou retorne à página inicial.

Botões disponíveis:

- "Recarregar Página"
- "Voltar ao Início"

Investigação Sugerida

1. Verificar logs do backend:

- Há erros ao buscar dados de analytics?
- Endpoint `/api/analytics` está funcionando?
- Verificar logs do servidor

2. Verificar logs do navegador:

- Console do navegador mostra erros JavaScript?
- Há erros de componentes Vue/React?
- Verificar Network tab para requisições falhadas

3. Verificar dados:

- Há dados de analytics no banco?
- Estrutura dos dados está correta?
- Há campos null/undefined causando erro?

4. Verificar componente:

- Componente `Analytics.vue` ou `Analytics.jsx`
- Há dependências faltando?
- Gráficos/charts estão configurados corretamente?

Correção Sugerida

1. Adicionar try-catch no componente para capturar erro específico
2. Adicionar fallback para quando não há dados
3. Validar dados antes de renderizar
4. Testar com dados de exemplo

 FUNCIONALIDADES QUE FUNCIONAM PERFEITAMENTE

1. Execução de Prompts - EXCELENTE! (Referência de Qualidade)

Página: /prompts

Avaliação: 10/10

Esta é a funcionalidade mais bem implementada do sistema e deve servir como **referência de qualidade** para outras funcionalidades.

Pontos fortes:

- Streaming em tempo real perfeito
- Métricas detalhadas (chunks, tempo, caracteres)
- Interface impecável
- Feedback visual excelente
- Botões "Copiar" e "Novo" funcionais

Teste realizado:

Plain Text

Prompt: "Teste Usuário Final - Análise de Texto"

Modelo: medicine-lm

Resultado: Completo em 1.6s

Chunks: 78

Caracteres: 259

Recomendação: Usar este código como base para implementar/corrigir outras funcionalidades de chat.

2. Dashboard - PERFEITO

Página: /

Avaliação: 10/10

Dashboard impecável com métricas em tempo real.

Dados exibidos:

- 10 Equipes (↑ 12%)
- 25 Projetos Ativos (↑ 8%)
- 23 Prompts (↑ 3%)
- 30 Projetos totais (1 Planejamento, 25 Ativos, 1 Em Espera, 2 Concluídos, 1 Arquivado)
- CPU: 1.0%, Memória: 26.3%, Disco: 64.8%
- Status: Banco de Dados Online, API tRPC Online, LM Studio Online

3. Outras Páginas Funcionando (11 páginas)

Página	Status	Dados
Projetos	✓ 10/10	30 projetos listados
Models	✓ 10/10	22 modelos ativos
Equipes	✓ 10/10	10 equipes
Tarefas	✓ 10/10	Vazio mas funcional
Configurações	✓ 10/10	6 tabs
Provedores	✓ 10/10	Vazio mas funcional
IAs Especializadas	✓ 10/10	Vazio mas funcional
Credenciais	✓ 10/10	Vazio mas funcional
Templates	✓ 10/10	4 templates cadastrados
Workflows	✓ 10/10	7 workflows ativos

Estatísticas de Testes

Cobertura

- **Total de páginas:** 23
- **Páginas testadas:** 16 (70%)
- **Páginas funcionando:** 13 (81%)
- **Páginas com bugs críticos:** 3 (19%)

Tempo de Testes

- **Duração total:** ~150 minutos
- **Reconexões SSH:** 3
- **Screenshots capturados:** 16+
- **Relatórios gerados:** 5

Dados do Sistema

- **Equipes:** 10
 - **Projetos:** 30 (25 ativos)
 - **Prompts:** 23
 - **Modelos:** 22 ativos
 - **Templates:** 4
 - **Workflows:** 7 ativos
-

Plano de Ação Recomendado

Sprint Imediata (Prioridade CRÍTICA)

Objetivo: Corrigir os 3 bugs críticos

Task 1: Corrigir Chat Principal (Estimativa: 4-8h)

1. Investigar event handlers (2h)
2. Identificar ponto de falha (2h)
3. Implementar correção (2h)
4. Testar em múltiplos navegadores (1h)
5. Code review (1h)

Task 2: Corrigir Chat de Follow-up (Estimativa: 2-4h)

1. Verificar se usa mesmo código do chat principal (1h)
2. Se sim, testar se correção do Task 1 resolveu (1h)
3. Se não, aplicar mesma correção (2h)

Task 3: Corrigir Analytics (Estimativa: 4-6h)

1. Investigar logs backend e frontend (2h)
2. Identificar causa do erro de renderização (2h)
3. Implementar correção (1h)
4. Adicionar tratamento de erros (1h)

Total estimado: 10-18 horas de desenvolvimento

Sprint Seguinte (Prioridade ALTA)

Task 4: Completar Testes das 7 Páginas Restantes

- Instruções
- Base de Conhecimento
- Serviços Externos
- Perfil
- Contas API
- Monitoramento
- Logs
- Terminal
- Treinamento

Task 5: Testar CRUD Completo

- Criar registros em páginas vazias (Provedores, IAs Especializadas, Credenciais, Tarefas)
- Validar edição e exclusão
- Validar se dados aparecem em relatórios

Task 6: Testar Integração LM Studio

- Carregar/descarregar modelos
- Testar múltiplos modelos simultâneos

🔍 Detalhes Técnicos

Ambiente de Produção

Plain Text

```
Servidor: 31.97.64.43:2224
Usuário: flavio
Túnel: SSH → localhost:3001
Versão: v3.7.0
```

Navegador de Teste

Plain Text

Navegador: Chromium (sandbox)
Resolução: Padrão
User Agent: [sandbox default]

Persistência de Bugs

Os 3 bugs críticos **PERSISTEM** da v3.6.0 para v3.7.0, indicando que:

1. Não foram identificados nos testes internos
2. Não foram priorizados para correção
3. Podem estar relacionados a mudanças em dependências

Notas Adicionais

Pontos Positivos do Sistema

1. **Arquitetura bem estruturada** - 23 páginas organizadas logicamente
2. **Interface consistente** - Padrões de design bem definidos
3. **Funcionalidades robustas** - Quando funcionam, funcionam perfeitamente
4. **Streaming de prompts** - Implementação excepcional
5. **Dashboard completo** - Métricas em tempo real impecáveis

Recomendações Gerais

1. **Adicionar testes automatizados** para evitar regressões
2. **Implementar CI/CD** com testes antes de deploy
3. **Adicionar logs de debug** em produção (com flag de ativação)
4. **Implementar error tracking** (Sentry, Rollbar, etc)
5. **Adicionar health checks** para componentes críticos

Contato

Testador: Manus AI

Data do Teste: 16 de Novembro de 2025

Hora: 21:42 GMT-3

Versão do Relatório: 1.0

Próximos Passos

1. **Dev Team:** Priorizar correção dos 3 bugs críticos
2. **QA Team:** Validar correções em ambiente de staging
3. **Product Team:** Decidir sobre testes das 7 páginas restantes
4. **Deploy:** Após validação, fazer deploy da v3.7.1 com correções

Estimativa para v3.7.1 (com bugs corrigidos): 2-3 dias úteis

Status do Sistema:  FUNCIONAL com 3 bugs críticos

Nota Técnica: 8/10

Recomendação: Corrigir bugs críticos antes de considerar pronto para produção