



# BUGS CRÍTICOS ENCONTRADOS - TESTES REAIS DE FUNCIONAMENTO

**Data:** 2025-11-13

**Rodada:** 25 (Testes Reais)

**Objetivo:** Validar funcionamento real do sistema com IAs

## 🔴 BUG CRÍTICO #1: Carregamento de Modelos é SIMULADO

**Severidade:** 🔴 CRÍTICO - Bloqueador de produção

**Descrição:** O endpoint `POST /api/models/:id/load` retorna sucesso mas **NÃO carrega** realmente o modelo no LM Studio.

**Evidência:**

### 1. Resposta da API:

JSON

```
{  
    "success": true,  
    "message": "Model loaded",  
    "data": {  
        "modelId": 2,  
        "modelName": "qwen3-coder-reap-25b-a3b",  
        "status": "loaded",  
        "simulated": true // ✗ SIMULADO!  
    }  
}
```

### 1. Tentativa de usar o modelo:

JSON

```
{  
    "output": "[Erro na execução] LM Studio: No models loaded.  
              Please load a model first using LM Studio UI or CLI command:  
lms load <model-name>,  
    "status": "error"  
}
```

## Comportamento Esperado:

- Endpoint deveria chamar API do LM Studio para carregar modelo
- Modelo deveria ficar disponível para uso
- "simulated": false

## Comportamento Real:

- Endpoint apenas simula o carregamento
- Modelo NÃO é carregado no LM Studio
- "simulated": true
- Execuções de prompts falham

## Impacto:

- ✗ Sistema NÃO pode executar prompts
- ✗ IAs NÃO respondem
- ✗ Orquestração de IAs NÃO funciona
- ✗ **BLOQUEADOR DE PRODUÇÃO**

**Causa Raiz:** Backend não implementou integração real com LM Studio API para carregar/descarregar modelos.

**Solução Necessária:** Implementar chamada real para LM Studio API:

TypeScript

```
// Arquivo: server/src/routes/models.ts (ou similar)

async function loadModel(modelId: string) {
    // ANTES (mock):
    return { success: true, simulated: true };

    // DEPOIS (real):
    const response = await fetch('http://localhost:1234/v1/models/load', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ model: modelId } )
    });

    if (!response.ok) {
        throw new Error('Failed to load model in LM Studio');
    }
}
```

```
    return { success: true, simulated: false };
}
```

## 🟡 BUG #2: Desincronia entre Database e LM Studio

Severidade: 🟡 ALTO

**Descrição:** O campo `isLoading` no banco de dados não reflete o estado real do LM Studio.

**Evidência:**

- Database: `"isLoading": true`
- LM Studio: "No models loaded"

**Impacto:**

- Interface mostra modelos como carregados quando não estão
- Usuário não sabe qual modelo está realmente ativo
- Confusão e frustração do usuário

**Solução:** Sincronizar estado real do LM Studio com database:

TypeScript

```
// Verificar estado real antes de retornar
const lmStudioModels = await fetch('http://localhost:1234/v1/models')
).then(r => r.json());
const isActuallyLoaded = lmStudioModels.data.some(m => m.id === modelId);

// Atualizar database se necessário
if (dbModel.isLoading !== isActuallyLoaded) {
  await updateModel(modelId, { isLoading: isActuallyLoaded });
}
```

## 🔴 TESTES INTERROMPIDOS

Devido ao Bug #1 (bloqueador), os seguintes testes **NÃO puderam ser executados**:

- ✗ Interação 1 com IA (falhou - modelo não carregado)
- ✗ Interação 2 com IA (não executado)
- ✗ Interação 3 com IA (não executado)
- ✗ Descarregar modelo (não testado)

- ✗ Carregar segundo modelo (não testado)
- ✗ Interações com segundo modelo (não executado)
- ✗ Criar IA Especializada (não testado)
- ✗ Encaminhamento automático (não testado)

Status: ● TESTES REAIS FALHARAM - SISTEMA NÃO FUNCIONA

---

## RESUMO

Aspecto	Status
Carregamento de Modelos	✗ SIMULADO (não funciona)
Execução de Prompts	✗ FALHA (modelo não carregado)
Interações com IA	✗ NÃO TESTADO (bloqueado)
Orquestração	✗ NÃO FUNCIONA
Status Geral	● NÃO PRONTO PARA PRODUÇÃO

---

## AÇÕES NECESSÁRIAS

### ● URGENTE (Sprint 19)

#### 1. Implementar carregamento REAL de modelos

- Integrar com LM Studio API
- Remover simulação
- Testar com modelo real

#### 2. Implementar descarregamento REAL de modelos

- Integrar com LM Studio API
- Liberar memória
- Atualizar database

#### 3. Sincronizar estado Database ↔ LM Studio

- Verificar estado real

- Atualizar isLoaded
- Exibir estado correto na interface

## 🟡 IMPORTANTE (Sprint 20)

1. **Validar execução de prompts com modelo real**
2. **Testar interações múltiplas**
3. **Validar orquestração de IAs**

## OBSERVAÇÕES

### Contexto:

- LM Studio está rodando ( `lmStudioAvailable: true` )
- API do LM Studio está acessível
- Backend consegue se comunicar com LM Studio
- **MAS:** Endpoint de carregamento não chama API real

**Recomendação:** Implementar Sprint 19 URGENTEMENTE antes de liberar para produção.

**Relatório gerado durante Rodada 25 - Testes Reais de Funcionamento**