

Correções do Sistema V2 - Produtos Reais




Problema Identificado

O sistema V2 não estava mostrando produtos reais do banco de dados e estava inventando lojas fictícias (“Mega Eletrônicos”, “Cellshop”) com preços inventados. Isso acontecia porque:




1. **Frontend V2 não estava chamando a API V2 do backend**
2. **Sistema V2 do frontend estava usando dados mockados**
3. **Falta de integração entre busca de produtos e SSE (Server-Sent Events)**

Correções Implementadas



1. Frontend - VendorAPI.ts

-  **Adicionada função** `sendMessageToV2()` que faz chamadas reais para `/api/assistant/v2/chat`
-  **Implementado processamento SSE** para receber produtos em tempo real
-  **Melhorado tratamento de eventos** `products`, `delta`, `emotion`, `insights`




2. Frontend - useIntelligentVendor.ts

-  **Substituído sistema mockado** pela chamada real da API V2
-  **Adicionado mapeamento correto** dos produtos do backend para o formato do frontend
-  **Implementado logging** para debug das chamadas V2




3. Frontend - GeminiAssistantBarV2.tsx

-  **Corrigida lógica de ativação do V2** para usar sempre que o modo estiver ativo
-  **Melhorado sistema de detecção** de quando usar V2

4. Backend - intelligent-vendor.ts

-  **Adicionado envio imediato de produtos** via streaming (`__PRODUCTS__`)
-  **Melhorado logging** para debug da busca de produtos
-  **Mantida busca real no banco** com query SQL otimizada

5. Backend - routes.ts

-  **Implementado processamento de eventos** `__PRODUCTS__` no SSE
-  **Adicionado fallback** para produtos via metadata
-  **Melhorado tratamento de eventos SSE**

Como Testar as Correções

1. Ativar o Modo V2

1. Abrir o ClickOfertas
2. Clicar no botão **“V2”** na barra do assistente (deve ficar laranja)
3. Ou digitar palavras como “v2”, “inteligente”, “avançado”

2. Testar Busca de Produtos

Exemplos de consultas para testar:





- "Procuro um iPhone"
- "Quero ver celulares Samsung"
- "Tem perfumes disponíveis?"
- "Mostra notebooks"
- "Preciso de um smartphone barato"

3. Verificar Logs no Console

Abrir DevTools (F12) e verificar logs:

```
[V2] Usando sistema V2 para: "procuro iphone"
[V2] Enviando mensagem para API V2: "procuro iphone"
[V2] Searching products for: "procuro iphone"
[V2] Found X products: [nomes dos produtos]
[V2] Produtos recebidos via SSE: [array de produtos]
[V2] SSE processado - Resposta: X chars, Produtos: Y
```

4. Verificar Produtos Reais

-  **Nomes reais** dos produtos (não “iPhone Mega Eletrônicos”)
-  **Lojas reais** cadastradas no sistema
-  **Preços reais** do banco de dados
-  **Imagens reais** dos produtos

Arquivos Modificados

```
client/src/components/intelligent-vendor-v2/services/VendorAPI.ts
client/src/components/intelligent-vendor-v2/hooks/useIntelligentVendor.ts
client/src/components/GeminiAssistantBarV2.tsx
server/assistant/v2/intelligent-vendor.ts
server/routes.ts
```

Próximos Passos

1. **Testar em produção** com dados reais
2. **Verificar performance** das consultas SQL
3. **Adicionar cache** para buscas frequentes
4. **Implementar analytics** para tracking de uso V2
5. **Otimizar SSE** para conexões lentas

Debugging

Se ainda houver problemas:

1. **Verificar se há produtos no banco:**

```
sql
```

```
SELECT COUNT(*) FROM products WHERE is_active = true;
```

2. Verificar logs do servidor:

```
bash
# Procurar por logs V2
grep "V2" logs/server.log
```

3. Testar API diretamente:

```
bash
curl -X POST http://localhost:3000/api/assistant/v2/chat \
-H "Content-Type: application/json" \
-d '{"userId":"test","message":"procuro iphone"}
```

Status das Correções

- ☒ Integração Frontend ↔ Backend V2
- ☒ Busca real de produtos no banco
- ☒ SSE para produtos em tempo real
- ☒ Mapeamento correto de dados
- ☒ Logging para debugging
- ☒ Ativação correta do modo V2

Resultado esperado: A IA V2 agora deve mostrar produtos reais do banco de dados em vez de inventar lojas e preços fictícios.