🐛 Fix: Corrigir processamento SSE do evento ‘products’ no frontend

🎯 Problema Identificado

O backend estava enviando corretamente os produtos via SSE usando o evento nomeado `event: products`, mas o frontend **não estava recebendo/processando** esse evento.

Logs do Backend (✅ Funcionando):

```
🛍️ [V2] Sending 7 products via SSE
event: products
data: {"products":[...]}
```

Logs do Frontend (❌ Problema):

- Eventos `delta`, `emotion` e `complete` chegavam normalmente
- Evento `products` **NUNCA** era detectado/processado
- Produtos não apareciam na interface

🔍 Causa Raiz

O código estava resetando `currentEventType` para `'message'` **antes** de processar a linha `data:` correspondente ao evento.

Fluxo problemático:
1. Linha `event: products` → Define `currentEventType = 'products'`
2. Linha `data: {...}` → Processa dados
3. **ANTES** de processar: `currentEventType = 'message'` (linha 461 antiga)
4. Resultado: Evento `products` nunca era detectado

✅ Solução Implementada

1. Nova variável `pendingEventType`

Armazena o tipo de evento entre as linhas `event:` e `data:`:

```
let pendingEventType: string | null = null;
```

2. Captura do evento

Quando `event: X` é detectado, guarda em `pendingEventType`:

```
if (line.startsWith('event: ')) {
  pendingEventType = line.slice(7).trim();
  console.log('📡 [V2] 🎯 Evento SSE detectado:', pendingEventType);
  continue;
}
```

3. Uso do evento correto

Quando `data:` chega, usa `pendingEventType` (se disponível):

```
const activeEventType = pendingEventType || currentEventType;
```

4. Processamento do evento `products`

Com logs detalhados para debug:

```
if (activeEventType === 'products') {
  console.log('🛍️ [V2] ✅ Processando evento PRODUCTS:', {
    hasProducts: !!data.products,
    isArray: Array.isArray(data.products),
    length: data.products?.length,
    firstProduct: data.products?.[0]?.name
  });

  if (data.products && Array.isArray(data.products) && data.products.length > 0) {
    console.log('🛍️ [V2] ✅✅ PRODUTOS RECEBIDOS:', data.products.length);
    // Atualizar estado com produtos
    setFeed(products);
  }
}
```

5. Limpeza após processamento

```
pendingEventType = null; // Limpar após usar
```

📊 Logs Adicionados

Para facilitar debug futuro:

- `📡 [V2] 🎯 Evento SSE detectado:` - Quando `event:` é capturado
- `📨 [V2] SSE data recebido para evento:` - Mostra qual evento está sendo processado
- `🛍️ [V2] ✅ Processando evento PRODUCTS:` - Detalhes dos produtos
- `🛍️ [V2] ✅✅ PRODUTOS RECEBIDOS:` - Confirmação de recebimento
- `🛍️ [V2] Lista de produtos:` - Nomes dos produtos recebidos

📝 Arquivos Modificados

- `client/src/components/GeminiAssistantBar.tsx` (linhas 322-493)
- +65 linhas adicionadas
- -37 linhas removidas
- Total: 102 linhas modificadas

🧪 Como Testar

1. Abrir o console do navegador
2. Buscar por “iPhone 16” no Gemini Assistant

3. Verificar logs:
   - `📡 [V2] 🎯 Evento SSE detectado: products`
   - `🛍️ [V2] ✅✅ PRODUTOS RECEBIDOS via evento nomeado: 7`
4. Confirmar que produtos aparecem na interface

✨ Resultado Esperado

- ✅ Evento `products` é detectado e processado corretamente
- ✅ Produtos aparecem na interface (topBox e feed)
- ✅ Logs detalhados facilitam debug futuro
- ✅ Compatibilidade mantida com formato inline (`data.products`)

🔗 Links

- **Branch**: `fix/sse-products-event-frontend`
- **Base**: `main`
- **Commit**: `58c6423`
- **PR Link**: https://github.com/thiagofregolao-blip/ClickOfertas01/pull/new/fix/sse-products-event-frontend

---

Pronto para merge após revisão e testes! 🚀