

Correção: Gráfico de Vendas não exibido no Dashboard

Data: 13 de outubro de 2025


Branch: `fix/sales-chart-api-endpoint`

Status:  Corrigido



Problema Identificado

O gráfico de vendas no dashboard do DELIVEREI estava exibindo uma mensagem de erro:

 Erro ao carregar dados de vendas
Tente novamente mais tarde


Sintoma

- O componente `SalesChart` estava recebendo um erro ao tentar buscar os dados da API
- Frontend não conseguia se comunicar com o endpoint `/api/dashboard/vendas`
- Erro HTTP 404 (Not Found)




Investigação


1. Frontend (SalesChart)

- **Arquivo:** `src/components/dashboard/SalesChart.tsx`
- **Status:**  Componente funcionando corretamente
- O componente aceita props `data`, `loading` e `error` e exibe as informações adequadamente


2. Serviço de API (Frontend)

- **Arquivo:** `src/services/dashboardApi.ts`
- **Status:**  Serviço configurado corretamente
- Método `getGraficoVendasCustom()` faz chamada para `/dashboard/vendas`

3. Cliente HTTP (Frontend)

- **Arquivo:** `src/services/apiClient.ts`
- **Base URL:** `https://delivere-backend.onrender.com/api`
- **Status:**  Configuração correta

4. Backend - Controller

- **Arquivo:** `backend/src/dashboard/dashboard.controller.ts`
- **Status:**  PROBLEMA ENCONTRADO
- Decorator: `@Controller('api/dashboard')`

5. Backend - Configuração Global

- **Arquivo:** `backend/src/main.ts`
- **Prefixo Global:** `app.setGlobalPrefix('api')`



Causa Raiz

O problema estava na **duplicação do prefixo** `/api/` nas rotas do backend:

Fluxo da Requisição:

```
Frontend → apiClient.get('/dashboard/vendas')
      ↓
Base URL: https://delivere-backend.onrender.com/api
      ↓
URL Final: https://delivere-backend.onrender.com/api/dashboard/vendas
```

Configuração do Backend (INCORRETA):

```
// main.ts
app.setGlobalPrefix('api'); // Prefixo global

// dashboard.controller.ts
@Controller('api/dashboard') // ✗ Prefixo duplicado
```

Resultado: Rota registrada como `/api/api/dashboard/vendas` ✗

Rota Esperada vs Rota Atual:

Componente	Esperado	Atual (Incorreto)
Frontend chama	<code>/dashboard/vendas</code>	<code>/dashboard/vendas</code>
Base URL adiciona	<code>https://...com/api</code>	<code>https://...com/api</code>
URL Final	<code>https://...com/api/dashboard/vendas</code>	<code>https://...com/api/dashboard/vendas</code>
Backend registra	<code>/api/dashboard/vendas</code>	<code>/api/api/dashboard/vendas</code> ✗
Resultado	✅ 200 OK	✗ 404 Not Found



Solução Implementada

Correção Aplicada

Removido o prefixo `api/` de todos os controllers afetados, mantendo apenas o prefixo global configurado em `main.ts`.

Arquivos Corrigidos (9 arquivos)

1. backend/src/dashboard/dashboard.controller.ts

```
``typescript
// Antes
@Controller('api/dashboard')

// Depois
@Controller('dashboard')
``
```

1. backend/src/cupons/cupons.controller.ts

```
typescript
@Controller('api/cupons') → @Controller('cupons')
```

2. backend/src/avaliacoes/avaliacoes.controller.ts

```
typescript
@Controller('api/avaliacoes') → @Controller('avaliacoes')
```

3. backend/src/pedidos/pedidos.controller.ts

```
typescript
@Controller('api/pedidos') → @Controller('pedidos')
```

4. backend/src/notificacoes/notificacoes.controller.ts

```
typescript
@Controller('api/notificacoes') → @Controller('notificacoes')
```

5. backend/src/modules/whatsapp/whatsapp.controller.ts

```
typescript
@Controller('api/whatsapp') → @Controller('whatsapp')
```

6. backend/src/modules/webhooks/webhooks.controller.ts

```
typescript
@Controller('api/webhooks') → @Controller('webhooks')
```

7. backend/src/modules/assinaturas/assinaturas.controller.ts

```
typescript
@Controller('api/assinaturas') → @Controller('assinaturas')
```

8. backend/src/modules/pagamentos/pagamentos.controller.ts

```
typescript
@Controller('api/pagamentos') → @Controller('pagamentos')
```

Resultado

Após a Correção:

```
// main.ts
app.setGlobalPrefix('api'); // Prefixo global

// dashboard.controller.ts
@Controller('dashboard') //  Sem duplicação
```

Rota registrada: `/api/dashboard/vendas` 

Endpoints Corrigidos:

Módulo	Endpoint Correto
Dashboard	<code>/api/dashboard/*</code>
Cupons	<code>/api/cupons/*</code>
Avaliações	<code>/api/avaliacoes/*</code>
Pedidos	<code>/api/pedidos/*</code>
Notificações	<code>/api/notificacoes/*</code>
WhatsApp	<code>/api/whatsapp/*</code>
Webhooks	<code>/api/webhooks/*</code>
Assinaturas	<code>/api/assinaturas/*</code>
Pagamentos	<code>/api/pagamentos/*</code>



Próximos Passos

Para Deploy:

1. Fazer merge da branch `fix/sales-chart-api-endpoint` para `main`
2. Render detectará as mudanças e fará o deploy automático
3. Aguardar deploy finalizar (~5-10 minutos)
4. Testar o dashboard em produção: <https://delivereit.netlify.app>




Validação:

1. Fazer login no sistema
2. Acessar o Dashboard
3. Verificar se o gráfico de vendas está sendo exibido corretamente
4. Testar diferentes filtros de período
5. Confirmar que os dados estão sendo carregados



Impacto

Funcionalidades Corrigidas:

-  Gráfico de vendas no dashboard
-  Estatísticas do dashboard
-  Produtos populares

- ✓ Endpoints de cupons, avaliações, pedidos, notificações
- ✓ Webhooks e integrações
- ✓ Sistema de assinaturas
- ✓ Pagamentos

Riscos:

- ⚠ **Baixo risco:** Correção apenas remove duplicação de prefixo
- ✓ **Sem breaking changes:** Rotas passam a funcionar conforme esperado
- ✓ **Sem migração de dados necessária**

Links Importantes

- **Pull Request:** <https://github.com/nerdrico2025/deliverei-v1/pull/new/fix/sales-chart-api-endpoint>
- **Commit:** 67ab143 - fix: corrigir duplicação de prefixo 'api' nos controllers
- **Frontend (Netlify):** <https://deliverei.netlify.app>
- **Backend (Render):** <https://deliverei-backend.onrender.com>

Autor

DeepAgent (Abacus.AI)

Data: 13 de outubro de 2025

Notas Adicionais

Padrão Recomendado para Novos Controllers:

```
// ✓ CORRETO
@Controller('nome-do-recurso')
export class RecursoController {
  // O prefixo 'api' será adicionado automaticamente pelo main.ts
}

// ✗ INCORRETO
@Controller('api/nome-do-recurso')
export class RecursoController {
  // Causa duplicação: /api/api/nome-do-recurso
}
```

Configuração Atual do Backend:

```
// backend/src/main.ts
app.setGlobalPrefix('api'); // ← Todos os controllers recebem este prefixo
```

Importante: Ao criar novos controllers, **não** incluir `api/` no decorator `@Controller()`.