

Projeto Pharma — Módulo Fiscal NFC-e (MOCK)

Checklist de Pronto (DoD) + Smoke Tests

Versão: 2026-02-15 21:23 | Ambiente: DEV/MOCK | Base: PostgreSQL local

Escopo coberto (MOCK)

- Prepare (gera XML draft + numeração + chave 44 + DV)
- Assinatura (MOCK) e Envio (MOCK) com simulação de rejeições
- QRCode/URL de consulta (MOCK)
- DANFE NFC-e em PDF (MOCK)
- Cancelamento (MOCK)
- Inutilização de numeração (MOCK)
- Catálogo de erros: GET /fiscal/errors
- Auditoria de envio: FiscalEvent (tentativas, falhas, resultado)
- Padronização de respostas e requestId nos retornos

Definition of Done (DoD)

- 1 Backend sobe sem erro em `npm run dev` e responde em `http://localhost:3000`.
- 2 Endpoints do módulo fiscal respondem com requestId e formato de erro padronizado.
- 3 Prisma migrate ok e Prisma Client gera sem warnings.
- 4 FiscalConfig existe para a loja de teste (storeId) com UF e CNPJ válidos.
- 5 Prepare cria FiscalDocument com status DRAFT, série/número e accessKey (44).
- 6 Send (MOCK) autoriza e preenche protocol + qrCodeUrl (quando MOCK_SEFAZ_REJECT permitir).
- 7 Em rejeição, retorna HTTP 422 com code SEFAZ_REJECT_### e registra FiscalEvent ENVIO_FALHA.
- 8 DANFE PDF abre no browser e contém chave + QRCode.
- 9 Cancelamento muda status para CANCELED e registra FiscalEvent CANCELAMENTO.
- 10 Inutilização registra FiscalEvent INUTILIZACAO e impede intervalo com numeração já usada.
- 11 GET /fiscal/errors retorna lista de códigos para o front.
- 12 Toda tentativa/resultado de envio gera FiscalEvent correspondente.

Variáveis de ambiente relevantes (referência rápida)

- PORT=3000
- DATABASE_URL=postgresql://.../pharma_dev?schema=public
- FISCAL_PROVIDER=MOCK
- FISCAL_SIGNER=MOCK
- MOCK_SEFAZ_REJECT=SEQUENCE:351,999,0 (ou 351 | RANDOM | NONE)

- SEFAZ_RETRY_ATTEMPTS=3
- SEFAZ_RETRY_BASE_DELAY_MS=250

Smoke Tests (passo a passo)

Pré-requisitos

- Backend rodando: `cd C:\pharma\backend && npm run dev`
- PostgreSQL ligado e DATABASE_URL apontando para pharma_dev
- Uma Store existente (storeId) e FiscalConfig preenchido (UF/CNPJ)

1) Prepare

PowerShell:

```
$body = @{"storeId":"SEU_STORE_ID"; saleRef=@{"total=10.50; discount=0; items=@{"sku="123"; description="Produto Teste"; qty=1; unitPrice=10.50 }} } |  
ConvertTo-Json -Depth 10 Invoke-RestMethod -Method Post -Uri  
"http://localhost:3000/fiscal/nfce/prepare" -ContentType "application/json" -Body  
$body
```

- Esperado: ok:true e doc.status=DRAFT com accessKey (44).

2) Send (rejeição simulada)

Defina MOCK_SEFAZ_REJECT=351 no .env e reinicie. Depois:

```
$docId="SEU_DOC_ID" Invoke-RestMethod -Method Post -Uri  
"http://localhost:3000/fiscal/nfce/$docId/send"
```

- Esperado: HTTP 422 com error.code=SEFAZ_REJECT_351.
- FiscalEvent registra ENVIO_TENTATIVA e ENVIO_FALHA.

3) Send (autorização)

Use MOCK_SEFAZ_REJECT=SEQUENCE:351,999,0 e repita o /send até autorizar.

- Esperado: doc.status=AUTHORIZED; protocolo preenchido; qrCodeUrl preenchido.
- FiscalEvent registra ENVIO_RESULTADO.

4) DANFE PDF

Abra no navegador: `http://localhost:3000/fiscal/nfce//danfe.pdf`

- Esperado: PDF 80mm abre e contém chave e QRCode.

5) Cancelamento (MOCK)

```
$docId="" $body=@{"reason":"Cliente desistiu da compra. Cancelamento em ambiente de  
teste." } | ConvertTo-Json Invoke-RestMethod -Method Post -Uri  
"http://localhost:3000/fiscal/nfce/$docId/cancel" -ContentType "application/json"  
-Body $body
```

- Esperado: doc.status=CANCELED e FiscalEvent type=CANCELAMENTO.

6) Inutilização (MOCK)

```
$body=@{"storeId":"SEU_STORE_ID"; series=1; nStart=10; nEnd=12; reason="Correção de  
numeração em ambiente de teste (MOCK)."} | ConvertTo-Json Invoke-RestMethod -Method  
Post -Uri "http://localhost:3000/fiscal/nfce/inutilize" -ContentType  
"application/json" -Body $body
```

- Esperado: ok:true e FiscalEvent type=INUTILIZACAO.

7) Catálogo de erros

```
GET http://localhost:3000/fiscal/errors
```

- Esperado: lista de códigos para o front mapear mensagens.

Checklist final de validação rápida

- Executar 1 ciclo completo: prepare -> send (authorized) -> danfe.pdf -> cancel
- Executar 1 rejeição simulada e confirmar error.code e FiscalEvent ENVIO_FALHA
- Executar 1 inutilização e confirmar FiscalEvent INUTILIZACAO
- Confirmar que tudo retorna requestId