

APIS do Novo Avanço

- [Fluxo de interação com API para Ecommerce](#)
- [Fluxo de interação para envio de vendas via Json](#)

Fluxo de interação com API para Ecommerce

“ Observação: os nomes exatos de alguns campos podem variar conforme a sua configuração. Use esta referência como guia prático e confira o esquema no Swagger quando precisar do detalhe de cada propriedade. (

api.novo.infovarejo.com.br)

1) Leitura de cadastro de produtos

Rota: **Buscar produtos para PDV** (cadastro básico para e-commerce/PDV) (

api.novo.infovarejo.com.br)

- **Método/URL:** GET /v1/public/produto/pdv
- **Autenticação:** Authorization: Bearer <token>
- **Uso:** obter produtos novos/alterados para sincronização.

Parâmetros (query) comuns

- dataInclusaoInicio, dataInclusaoFim — ISO 8601.
- dataAlteracaoCadastroInicio, dataAlteracaoCadastroFim.
- dataAlteracaoPrecoInicio, dataAlteracaoPrecoFim.
- (Opcional) filtros como ean, departamentoId, ativo (quando aplicável).
- **Paginação** (quando disponível no seu tenant): page, pageSize (ou equivalentes).

Resposta (resumo)

- Lista de produtos com campos como: produtoId, descricao, eanPrincipal, ncm, unidade, ativo, metadados de inclusão/alteração.

Exemplo cURL

```
curl -X GET \  
  'https://api.novo.infovarejo.com.br/v1/public/produto/pdv?dataInclusaoInicio=2022-04-02T07:00:00.000&dataInclusaoFim=2022-08-24T23:59:00.000&dataAlteracaoCadastroInicio=2022-04-02T07:00:00.000&dataAlteracaoCadastroFim=2022-08-24T23:59:00.000&dataAlteracaoPrecoInicio=2022-04-02T07:00:00.000&dataAlteracaoPrecoFim=2022-08-24T23:59:00.000&page=1&pageSize=100' \  
  -H 'Authorization: Bearer <SEU_TOKEN>'
```

2) Leitura de Estoque e Preço

Rota: **Retorna fichas financeiras do produto** (preço/estoque para vitrine) (api.novo.infovarejo.com.br)

- **Método/URL:** GET /v1/public/produto/fichaFinanceira
- **Autenticação:** Authorization: Bearer <token>
- **Uso:** consultar preço, promoções e, em alguns ambientes, saldo/estoque por filial.

Parâmetros (query) comuns

- Identificador do produto (um ou mais): ean, produtoId (ou sku), etc.
- (Opcional) filialId para preço/estoque específico por loja.
- **Paginação** quando retornar múltiplos itens.

Resposta (resumo)

- Campos típicos: produtoId, ean, precoVenda, precoPromocional (se houver), dataVigencia, e possivelmente estoqueAtual / saldo por filial.

Exemplo cURL

```
curl -X GET \  
  'https://api.novo.infovarejo.com.br/v1/public/produto/fichaFinanceira?ean=79075&filialId=123' \  
  -H 'Authorization: Bearer <SEU_TOKEN>'
```

3) Consulta de clientes

Rota: **Retornar lista de clientes por empresa** (api.novo.infovarejo.com.br)

- **Método/URL:** GET /v1/public/pessoa/clientes (nome do path pode variar no Swagger sob Pessoa → Clientes)
- **Autenticação:** Authorization: Bearer <token>
- **Uso:** listar clientes para cadastro/checkout e-commerce.

Parâmetros (query) comuns

- empresaId (obrigatório na maioria dos cenários).
- Filtros: cpfCnpj, nome, email.
- Janelas de atualização: dataAlteracaoInicio, dataAlteracaoFim.
- **Paginação:** page, pageSize.

Resposta (resumo)

- Lista com clienteId, nome, cpfCnpj, email, telefone, endereços, flags (ativo), datas de alteração.

Exemplo cURL

```
curl -X GET \  
  
'https://api.novo.infovarejo.com.br/v1/public/pessoa/clientes?empresaId=1023&page=1&pageSize=100' \  
  
-H 'Authorization: Bearer <SEU_TOKEN>'
```

4) Gravação (upsert) de clientes

Rota: **Cadastra ou atualiza cliente** (api.novo.infovarejo.com.br)

- **Método/URL:** POST /v1/public/pessoa/cliente
- **Autenticação:** Authorization: Bearer <token>
- **Uso:** criar/atualizar cliente (CPF/CNPJ único).

Body (JSON) — exemplo mínimo

```
{  
  "empresaId": 1023,  
  "cpfCnpj": "12345678901",  
  "nome": "Maria Compradora",
```

```
"email": "maria@exemplo.com",
"telefone": "31999990000",
"enderecos": [
  {
    "tipo": "ENTREGA",
    "logradouro": "Rua A, 100",
    "bairro": "Centro",
    "cidade": "Belo Horizonte",
    "uf": "MG",
    "cep": "30130000"
  }
]
```

Exemplo cURL

```
curl -X POST 'https://api.novo.infovarejo.com.br/v1/public/pessoa/cliente' \
-H 'Authorization: Bearer <SEU_TOKEN>' \
-H 'Content-Type: application/json' \
-d @cliente.json
```

5) Consulta de vendedores

Rota: **Retornar lista de vendedores por empresa** (api.novo.infovarejo.com.br)

- **Método/URL:** GET /v1/public/pessoa/vendedores
- **Autenticação:** Authorization: Bearer <token>
- **Uso:** obter IDs/códigos de vendedores para atrelar em pedidos.

Parâmetros (query)

- `empresaId` (geralmente obrigatório).
- (Opcional) filtros por `ativo`, `matricula`, etc.
- **Paginação:** `page`, `pageSize`.

Resposta (resumo)

- `vendedorId`/`codigo`, `nome`, `ativo`, dados de contato.

Exemplo cURL

```
curl -X GET \  
  
'https://api.novo.infovarejo.com.br/v1/public/pessoa/vendedores?empresaId=1023&page=1&pageSize=100' \  
  
-H 'Authorization: Bearer <SEU_TOKEN>'
```

6) Gravação (upsert) de vendedores

Rota: **Upsert dados de vendedores** (api.novo.infovarejo.com.br)

- **Método/URL:** POST /v1/public/pessoa/vendedores
- **Autenticação:** Authorization: Bearer <token>
- **Uso:** criar/atualizar cadastro de vendedores.

Body (JSON) — exemplo

```
{  
  "empresaId": 1023,  
  "vendedores": [  
    {  
      "codigo": "VEN001",  
      "nome": "João Vendas",  
      "email": "joao@exemplo.com",  
      "ativo": true  
    }  
  ]  
}
```

Exemplo cURL

```
curl -X POST 'https://api.novo.infovarejo.com.br/v1/public/pessoa/vendedores' \  
-H 'Authorization: Bearer <SEU_TOKEN>' \  
-H 'Content-Type: application/json' \  
-d @vendedores.json
```

7) Gravação de Pedido de Vendas (CPF/CNPJ)

Rota: **Pedido de venda por CPF/CNPJ** (api.novo.infovarejo.com.br)

- **Método/URL:** POST /v1/public/pedido/venda/cpfcnpj
- **Autenticação:** Authorization: Bearer <token>
- **Uso:** criar pedido de venda para cliente identificado (e-commerce / força de vendas).

Body (JSON) — exemplo prático

```
{
  "empresaId": 1023,
  "filialId": 1901,
  "cliente": { "cpfCnpj": "12345678901" },
  "vendedorCodigo": "VEN001",
  "itens": [
    { "produtoId": 55555, "quantidade": 2, "precoUnitario": 19.9, "ean": "7891234567890" }
  ],
  "pagamentos": [
    { "meio": "CARTAO", "valor": 39.8, "parcelas": 1 }
  ],
  "observacao": "Pedido originado no e-commerce",
  "origem": "ECOMMERCE"
}
```

Exemplo cURL

```
curl -X POST 'https://api.novo.infovarejo.com.br/v1/public/pedido/venda/cpfcnpj' \
-H 'Authorization: Bearer <SEU_TOKEN>' \
-H 'Content-Type: application/json' \
-d @pedido.json
```

Convenções gerais

Autenticação

- Header: `Authorization: Bearer <token>` para todas as rotas públicas autenticadas. (api.novo.infovarejo.com.br)

Paginação

- Quando disponível, use `page` (1-based) e `pageSize` (ex.: 50/100). As respostas costumam trazer metadados de total/última página.

Códigos de resposta (mais comuns)

- `200/201`: sucesso (lista/criação).
- `400`: validação de campos.
- `401/403`: token inválido/sem permissão.
- `404`: recurso não encontrado.
- `409`: conflito (ex.: CPF/CNPJ duplicado em upsert).
- `422`: erro de domínio (quando aplicável).
- `500`: erro interno.

Boas práticas

- Filtrar por janelas de **data de inclusão/alteração** para cargas incrementais.
- Tratar **timeout** e **retries** para rotas que retornam grandes volumes.
- Registrar `empresaId` e `filialId` corretamente para operações sensíveis a contexto (preço, estoque, pedido).
- Logar `requestId`/`correlationId` (se fornecido) para rastreabilidade.

Acesso a documentação Swagger completa em: <https://api.novo.infovarejo.com.br/docs/public/>

Fluxo de interação para envio de vendas via Json

Módulo PV – Integração de Envio de Vendas

1. Objetivo / Descrição

Este módulo documenta como parceiros devem enviar os dados de venda **via JSON** para o sistema **Novo Avanço**, permitindo a integração de vendas realizadas no PDV (loja física) ao sistema central da Avanço.

2. Fluxo Geral

1. PDV gera um JSON com os dados da venda conforme o padrão definido.
2. JSON é enviado para o endpoint do módulo PV (rota exata definida na API da Avanço).
3. O sistema Novo Avanço processa e registra a venda.

- **Destino da chamada:** não informado na página — verifique junto ao time técnico da Avanço ou API correspondente.

3. Estrutura do JSON de Requisição

3.1 Cabeçalho (cabeçalho)

Campo	Descrição	Tamanho máx.	Exemplo
cnpj	CNPJ/CPF do cliente	14	"10542390612"
vendedor	Código do vendedor	3	"000"
desconto	Desconto aplicado no subtotal	9	0
total	Valor total da venda	12	35.10
itens	Número de itens vendidos	3	2
descontoItens	Valor de desconto nos itens	9	0

Campo	Descrição	Tamanho máx.	Exemplo
troco	Valor de troco	9	4.90
dataEmissao	Data de emissão (ISO 8601)	24	"2024-02-21T14:11:00.000Z"
numEcf	Número do CUPOM	2	13
acrescimo	Acréscimo no subtotal	9	0
versao	Versão do PDV	6	62
cartao	Número do cartão	19	0
numeroParcelas	Número de parcelas	2	0
contraVale	Se foi emitido contra-vale com o troco (S/N)	1	N
origem	Código da integração (definir com Avanço)	—	—
operador	Código do Operador	15	13
numeroDAV	Número do DAV	13	0
marcaEcf	Marca da impressora no PDV	15	"NFCE"
modeloEcf	Modelo da impressora no PDV	20	"NFCE"
valorSaque	Valor de saque nas transações TEF	9	0
versaoPdv	Versão do ponto de venda	6	"7.4.66"
nomeOperador	Nome do operador	15	"Carlos"
dataVenda	Data da venda (ISO 8601)	24	"2024-02-21T14:11:00.000Z"
numeroControle	Número de controle da pré-venda	10	"0000000000"
valorAproximadoTributos	Estimativa do valor dos tributos	10	0
pontosDotz	Pontos Dotz (venda + itens)	12	"000000000000"
trocoSolidario	Valor do troco solidário	9	0
descontoScannplus	Se o desconto foi via ScannPlus (S/N)	1	N
ccf	Número do CCF do Cupom Fiscal	6	34
horaCriacaoArquivo	Hora de criação do arquivo (HHMMSS)	6	90903
cnpjLoja	CNPJ da loja	14	71294573000113

Campo	Descrição	Tamanho máx.	Exemplo
cpfSocioTorcedor	CPF do Sócio Torcedor (se aplicável)	11	0
cpfBeBlue	CPF do cliente BeBlue (0 se não informar)	11	0
trocoSimples	Valor do troco simples	9	0
cpfDotz	CPF do cliente Dotz pontuado	11	0
cpfTrocoSolidario	CPF/CNPJ de quem deixou troco solidário	14	0
cpfPromocional	CPF para promoções de parceiros	11	0
cpfCnpjMercafacil	CPF/CNPJ para Clube Mercafacil	14	" "
valorDescontoRateadoSubtotal	Valor total do desconto rateado	10	0
idDescontoSubtotal	ID da transação Mercafacil	50	42d522f4-6b0a-43f9-bd82-721c58609658
cupomVerde	Se foi cupom verde (S/N)	1	N
cpfCupomVerde	Se informou CPF para cupom verde (S/N)	1	N
cpfOperador	CPF do operador	11	"000000000000"

3.2 Itens (itens)

Para cada item vendido:

Campo	Descrição	Exemplo
numeroSerieEcf	Número do PDV (formatação Avanço)	"AV071294573000113025"
coo	Número do cupom (mesmo sem cupom, usar número da nota)	13
sequencialItem	Sequência do item na venda	1
codigoVendido	Código EAN/barras do item	"07896227620014"
quantidade	Quantidade (últimas 3 posições decimais)	5
preço	Preço unitário	3.90
total	Valor total do item (qtde × preço)	19.50
tributacao	Situação tributária (N, F, T, I)	"N"

Campo	Descrição	Exemplo
dataVenda	Data da venda (ISO 8601)	"2024-02-21T03:00:00.000Z"
numEcf	Número do PDV	1
tipoItem	P = Produto ou S = Serviço	"P"
aliquotaProduto	Alíquota real (ex: N=0, T=1800, T=2500)	0
aliquotaSaida	Alíquota efetiva de saída	0
numProduto	Código interno / reduzido do produto	"000857"
codigoAlternativo	Código alternativo do cadastro	"857"
precoVendaAtual	Preço real de venda	3.90
valorDesconto	Desconto aplicado no item	0
Venda no preço promocional (S/N)	Se foi vendido com preço promocional	N
registradorAliquota	Registrador da alíquota no ECF (SPED)	Ex: 01T0700

4. Exemplo de JSON completo (simplificado)

```
{
  "auth": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6Ii..",
  },
  "cabecalho": {
    "cnpj": "71294573000113",
    "cnpjLoja": "71294573000113",
    "operador": "33",
    "nomeOperador": "MARIA OLIVEIRA",
    "cpfOperador": "00000000000",
    "vendedor": "007",
    "numEcf": 12,
    "ccf": 150,
    "dataVenda": "2025-09-10T20:10:00.000Z",
    "dataEmissao": "2025-09-10T20:11:00.000Z",
    "total": 250.00,
    "itens": 1,
    "troco": 0,
    "desconto": 10.00,
    " acrescimo": 0,
  }
}
```

```
"versaoPdv": "7.5.10",
"versao": 63,
"origem": 2
},
"id": 908200,
"itens": [
  {
    "numeroSerieEcf": "AV071294573000113012",
    "coo": 150,
    "sequencialItem": 1,
    "numEcf": 12,
    "numProduto": "002222",
    "codigoVendido": "7892223334445",
    "descricao": "TV LED 50 POL UHD",
    "unidade": "UN",
    "quantidade": 1,
    "preco": 260.00,
    "valorDesconto": 10.00,
    "total": 250.00,
    "ncm": "85287200",
    "cfop": "5102",
    "tributacao": "T",
    "cstPis": "01",
    "cstCofins": "01",
    "aliquotaProduto": 1800,
    "aliquotaSaida": 1800,
    "precoVendaAtual": 260.00,
    "vendaPromocional": "N"
  }
],
"nfce": {
  "chave": "31250971294573000113501250000001501234567890",
  "nfce": "NFCE",
  "serie": "012",
  "underline": "_"
},
"pagamentos": [
  {
    "cdFormaPagto": 3,
    "descForma": "CARTAO CREDITO",
```

```
"nuParcelas": 3,
"valorTotal": 250.00,
"valorParcela": 83.33,
"autorizacaoCreditoCompleto": "A1B2C3D4E5"
}
],
"registro": "51",
"serie": "12",
"uid": "f27f1d22-95bb-41af-8c8f-62dfcb7f12c2",
"xml": "<?xml version=\"1.0\" encoding=\"UTF-8\"?><nfeProc versao=\"4.00\" ...>",
"xmlCancelado": ""
}
```

5. Boas Práticas / Observações

- **Validação:** Certifique-se que campos obrigatórios estejam preenchidos, sobretudo datas no formato ISO 8601 e campos numéricos no formato correto.
 - **Campos condicionais:** Muitos campos (ex: `promocional`, `descontoItens`, `trocoSolidario`) são aplicáveis somente em cenários específicos.
 - **Conversão de dados:** Quantidades e valores devem ser transmitidos numéricos, não strings.
 - **Versão do PDV:** Importante para controle de compatibilidade de payloads.
 - **Origem:** Buscar definição do código de integração junto ao time da Avanço.
 - **Número do DAV / Controle:** Relevante para rastrear pré-vendas e cancelamentos no futuro.
-