

API PDV

API de integracao com o PDV

2026-09-23

API para integração de pedidos e-commerce no PDV

Overview

API para geração e manutenção de vendas e notas fiscais do e-commerce no PDV.

Version

1.0

POST /api/desconto/melhor/obter-v1

Consultar melhor desconto por produto.

Consulta melhor desconto do produto consultando no próprio sistema e no PEC.

Request Body:

Content: `application/json` | [DescontoMelhorIntegracaoReqDto](#)

```
{
  considerarTodasCondicoesPagamento?: string; // Considerar todas as condições de pagamento.
  identificadorCliente?: string; // Identificador do Cliente (cartão, CPF ou CNPJ).
  produtos: Array<ProdutoDescontoMelhorIntegracaoDto>;
}
```

Response 200:

Consulta realizada com sucesso.

Content: `application/json` | [DescontoMelhorIntegracaoDto](#)

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoCondicaoPagamento: string; // Código da condição de pagamento do desconto aplicado.
  codigoLaboratorio: integer; // Código identificador do laboratório.
  codigoProduto: integer; // Código identificador do produto.
  nomeCondicaoPagamento: string; // Nome da condição de pagamento do desconto aplicado.
  nomeDesconto: string; // Nome do desconto aplicado.
  nomeLaboratorio: string; // Nome do laboratório fabricante.
  nomeProduto: string; // Nome do produto.
  percentualDesconto: number; // Percentual de desconto aplicado.
  quantidade: integer; // Quantidade do produto.
  quantidadeEstoque: integer; // Quantidade disponível em estoque.
  valorVenda: number; // Valor de venda do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro interno na consulta.

GET /api/nota-fiscal/download-xml-nota-v1

Download do XML da nota fiscal.

Faz o download do XML da nota fiscal emitida.

Request Parameters:

numeroNota: **integer**; // Número da venda gerada no PDV.
numeroPedido: **string**; // Número do pedido no e-commerce vinculado à venda.

Response 200:

XML da nota fiscal.

Content: **application/xml** | string

Response 401:

Não autorizado.

Response 404:

Nota fiscal não localizada.

POST /api/nota-fiscal/emitir-nota-v1

Emitir nota fiscal.

Emite uma nota fiscal para um pedido de e-commerce.

Request Body:

Content: **application/json** | EmissaoNotaFiscalIntegracaoDto

```
{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce.
}
```

Response 200:

Nota(s) emitida(s) com sucesso.

Content: **application/json** | EmissaoNotaFiscalIntegracaoRetDto

```
{
  chaveNotaFiscal: string; // Chave da nota fiscal eletrônica.
  codigoRetorno: string; // Código do retorno.
  descricaoRetorno: string; // Descrição detalhada do retorno.
  modeloNotaFiscal: string; // Modelo da nota fiscal.
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroNotaFiscal: integer; // Número da nota fiscal.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  numeroSequencial: integer; // Número sequencial da emissão.
  serieNotaFiscal: string; // Série da nota fiscal.
  tipoRetorno: string; // Tipo do retorno.
}
```

Response 400:

Erro na emissão da nota.

Response 401:

Não autorizado.

Response 500:

Erro na emissão da nota.

POST /api/venda/atualizar-status-venda-v1

Atualizar status da venda e-commerce.

Atualiza o status de uma venda e-commerce.

Request Body:

Content: `application/json` | AtualizaStatusPedidoEcommerceDto

```
{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  status: string; // Status do pedido no e-commerce.
}
```

Response 200:

Status atualizado com sucesso.

Content: `application/json` | VendaEcommerceIntegracaoRetDto

```
{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  statusPedido: StatusPedido;
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na atualização do status da venda.

GET /api/venda/consultar-venda-numero-nota-v1

Consultar venda pelo número da venda.

Consulta vendas pelo número da venda gerada no PDV.

Request Parameters:

numeroNota: `integer`; // Número da venda gerada no PDV.

Response 200:

Consulta realizada com sucesso.

Content: `application/json` | [VendaEcommerceIntegracaoDto](#)

```
{
  cliente?: ClienteEcommerceIntegracaoDto;
  dataPedido: string; // Data do pedido no e-commerce.
  enderecoEntrega?: EnderecoEntregaIntegracaoDto;
  entrega?: boolean; // Indica se é uma venda Tele-Entrega ou não.
  numeroNota?: integer; // Número da venda gerada no PDV (somente quando originado de um DAV).
  numeroPedido: string; // Número do pedido no e-commerce que será vinculado à venda.
  pagamentoMultiplo?: PagamentoMultiploIntegracaoDto;
  produtos?: Array<VendaItemEcommerceIntegracaoDto>;
  valorFrete?: number; // Valor do frete do pedido.
  valorTotalVenda: number; // Valor total do pedido, que será utilizado como total da venda.
  vendedor?: VendedorEcommerceIntegracaoDto;
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro interno na consulta.

GET /api/venda/consultar-venda-v1

Consultar venda e-commerce.

Consulta vendas do e-commerce por parâmetros.

Request Parameters:

numeroNota: `integer`; // Número da venda gerada no PDV.
 numeroPedido: `string`; // Número do pedido no e-commerce vinculado à venda.
 limiteMensal: `boolean`; // Se true, limita a consulta às vendas dos últimos 30 dias.

Response 200:

Consulta realizada com sucesso.

Content: `application/json` | [StatusVendaEcommerceIntegracaoRetDto](#)

```
{
  dataPedido: string; // Data do pedido no e-commerce.
  numeroCupomFiscal: integer; // Número do cupom fiscal.
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroNotaFiscal: integer; // Número da nota fiscal.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  statusPedido: StatusPedido;
  xmlNfe: string; // XML da nota fiscal eletrônica.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendas.

POST /api/venda/efetuar-venda-v1

Efetuar venda e-commerce.

Efetua uma venda no e-commerce.

Request Body:

Content: application/json | VendaEcommerceIntegracaoDto

```
{
  cliente?: ClienteEcommerceIntegracaoDto;
  dataPedido: string; // Data do pedido no e-commerce.
  enderecoEntrega?: EnderecoEntregaIntegracaoDto;
  entrega?: boolean; // Indica se é uma venda Tele-Entrega ou não.
  numeroNota?: integer; // Número da venda gerada no PDV (somente quando originado de um DAV).
  numeroPedido: string; // Número do pedido no e-commerce que será vinculado à venda.
  pagamentoMultiplo?: PagamentoMultiploIntegracaoDto;
  produtos?: Array<VendaItemEcommerceIntegracaoDto>;
  valorFrete?: number; // Valor do frete do pedido.
  valorTotalVenda: number; // Valor total do pedido, que será utilizado como total da venda.
  vendedor?: VendedorEcommerceIntegracaoDto;
}
```

Response 200:

Venda efetuada com sucesso.

Content: application/json | VendaEcommerceIntegracaoRetDto

```
{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  statusPedido: StatusPedido;
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na geração da venda.

POST /api/venda/finalizar-venda-v1

Finalizar venda e-commerce.

Finaliza uma venda no e-commerce.

Request Body:

Content: `application/json` | [VendaEcommerceFinalizacaoIntegracaoDto](#)

```
{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  pagamentoMultiplo?: PagamentoMultiploIntegracaoDto;
}
```

Response 200:

Venda finalizada com sucesso.

Content: `application/json` | [VendaEcommerceIntegracaoRetDto](#)

```
{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  statusPedido: StatusPedido;
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na finalização da venda.

GET /nfe/devolucao-rs

Response default:

Default response.

Content: `text/plain` | string

POST /nfe/devolucao-rs

Request Body:

Content: `application/json` | [DevolucaoFornecedorDTO](#)

```
{
  ambiente: string;
  codAplicacao: integer;
  deveDestacarIcmsStEmCampoProprio: boolean;
  deveDestacarIpiEmCampoProprio: boolean;
  notaEmitida: NotaFiscalDevolucao;
  nt2025002Vigente: boolean;
  numeroLote: integer;
}
```

```
numeroNSU: integer;  
numeroSerie: string;  
numeroUltimaNota: integer;  
proximoLote: integer;  
tipoRegime: string;  
tipoRegimeFornec: string;  
venda: Venda;  
}
```

Content: text/plain | DevolucaoFornecedorDTO

```
{  
  ambiente: string;  
  codAplicacao: integer;  
  deveDestacarIcmsStEmCampoProprio: boolean;  
  deveDestacarIpiEmCampoProprio: boolean;  
  notaEmitida: NotaFiscalDevolucao;  
  nt2025002Vigente: boolean;  
  numeroLote: integer;  
  numeroNSU: integer;  
  numeroSerie: string;  
  numeroUltimaNota: integer;  
  proximoLote: integer;  
  tipoRegime: string;  
  tipoRegimeFornec: string;  
  venda: Venda;  
}
```

Response default:

Default response.

Schemas

AtualizaStatusPedidoEcommerceDto

```
{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  status: string; // Status do pedido no e-commerce.
}
```

CartaoVendaIntegracaoDTO

```
{
  codigo: integer; // Código do cartão (Trier).
  numeroAutorizacao?: integer; // Número de autorização da transação no cartão.
  qtdParcela?: integer; // Número de parcelas no cartão.
  valor: number; // Valor pago no cartão.
}
```

ClienteEcommerceIntegracaoDto

```
{
  celular?: string; // Número de celular do cliente.
  codigo?: integer; // Código do cliente, se não informado, será gerado automaticamente pelo PDV.
  dataNascimento?: string; // Data de nascimento do cliente.
  email?: string; // E-mail do cliente.
  fone?: string; // Número de telefone do cliente.
  nome: string; // Nome do cliente.
  numeroCpfCnpj: string; // Número do CPF ou CNPJ do cliente.
  numeroRGIE?: string; // Número do RG ou IE do cliente.
  sexo?: string; // Sexo do cliente.
}
```

CrediarioIntegracaoDTO

```
{
  pagamentoRealizado: boolean; // Indica se o pagamento foi realizado.
  qtdParcela: integer; // Número de parcelas do crediario.
  valor: number; // Valor total pago no crediario.
  vencimentos?: Array<string>; // Vencimentos manuais por parcela (yyyy-MM-dd), na ordem das parcelas. Se omitido, calculado pela configuração do cliente.
}
```

DadosDestinatario

```
{
  bairro: string;
  cep: string;
  cidade: string;
  cnae: string;
  cnpj: string;
  codigoMunicipio: string;
```

```

codigoPais: string;
complemento: string;
email: string;
indicadorIE: string;
inscricaoEstadual: string;
inscricaoEstadualST: string;
inscricaoMunicipal: string;
inscricaoSuframa: string;
logradouro: string;
numeroEndereco: string;
pais: string;
razaoSocial: string;
telefone: string;
uf: string;
}

```

DadosEmitente

```

{
  bairro: string;
  cep: string;
  cidade: string;
  cnae: string;
  cnpj: string;
  codigoMunicipio: string;
  codigoPais: string;
  complemento: string;
  inscricaoEstadual: string;
  inscricaoEstadualST: string;
  inscricaoMunicipal: string;
  logradouro: string;
  numeroEndereco: string;
  pais: string;
  razaoSocial: string;
  telefone: string;
  uf: string;
}

```

DadosTransporte

```

{
  cpfCnpjTransportador: string;
  enderecoTransportador: string;
  ieTransportador: string;
  nomeMunicipioTransportador: string;
  nomeTransportador: string;
  ufTransportador: string;
}

```

DescontoMelhorIntegracaoDto

```

{
  codigoBarras: integer; // Código de barras do produto.
  codigoCondicaoPagamento: string; // Código da condição de pagamento do desconto aplicado.
  codigoLaboratorio: integer; // Código identificador do laboratório.
}

```

```

codigoProduto: integer; // Código identificador do produto.
nomeCondicaoPagamento: string; // Nome da condição de pagamento do desconto aplicado.
nomeDesconto: string; // Nome do desconto aplicado.
nomeLaboratorio: string; // Nome do laboratório fabricante.
nomeProduto: string; // Nome do produto.
percentualDesconto: number; // Percentual de desconto aplicado.
quantidade: integer; // Quantidade do produto.
quantidadeEstoque: integer; // Quantidade disponível em estoque.
valorVenda: number; // Valor de venda do produto.
}

```

DescontoMelhorIntegracaoReqDto

```

{
  considerarTodasCondicoesPagamento?: string; // Considerar todas as condições de pagamento.
  identificadorCliente?: string; // Identificador do Cliente (cartão, CPF ou CNPJ).
  produtos: Array<ProdutoDescontoMelhorIntegracaoDto>;
}

```

DevolucaoFornecedorDTO

```

{
  ambiente: string;
  codAplicacao: integer;
  deveDestacarIcmsStEmCampoProprio: boolean;
  deveDestacarIpiEmCampoProprio: boolean;
  notaEmitida: NotaFiscalDevolucao;
  nt2025002Vigente: boolean;
  numeroLote: integer;
  numeroNSU: integer;
  numeroSerie: string;
  numeroUltimaNota: integer;
  proximoLote: integer;
  tipoRegime: string;
  tipoRegimeFornec: string;
  venda: Venda;
}

```

DfeReferenciado

```

{
  chave: string;
  nitem: integer;
}

```

DinheiroIntegracaoDTO

```

{
  valor: number; // Valor pago em dinheiro.
}

```

EmissaoNotaFiscalIntegracaoDto

```
{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce.
}
```

EmissaoNotaFiscalIntegracaoRetDto

```
{
  chaveNotaFiscal: string; // Chave da nota fiscal eletrônica.
  codigoRetorno: string; // Código do retorno.
  descricaoRetorno: string; // Descrição detalhada do retorno.
  modeloNotaFiscal: string; // Modelo da nota fiscal.
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroNotaFiscal: integer; // Número da nota fiscal.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  numeroSequencial: integer; // Número sequencial da emissão.
  serieNotaFiscal: string; // Série da nota fiscal.
  tipoRetorno: string; // Tipo do retorno.
}
```

EnderecoEntregaIntegracaoDto

```
{
  bairro: string; // Bairro.
  cep?: string; // CEP.
  cidade: string; // Cidade.
  complemento?: string; // Complemento.
  estado: string; // Estado.
  logradouro: string; // Logradouro.
  numero?: string; // Número.
  referencia?: string; // Referência.
}
```

IbsCbs

```
{
  aliquota: number;
  aliquotaEfetiva: number;
  percentualDiferimento: number;
  percentualReducaoAliquota: number;
  valor: number;
  valorDiferimento: number;
}
```

IbsCbsProduto

```
{
  aliquotaEfetivaRegularCbs: number;
  aliquotaEfetivaRegularIbsMun: number;
  aliquotaEfetivaRegularIbsUf: number;
  baseCalculo: number;
}
```

```

cbs: IbsCbs;
classificacaoTributaria: string;
classificacaoTributariaRegular: string;
cst: string;
cstRegular: string;
doacao: boolean;
ibsMun: IbsCbs;
ibsUf: IbsCbs;
valorItem: number;
valorRegularCbs: number;
valorRegularIbsMun: number;
valorRegularIbsUf: number;
}

```

IntegracaoJsonItemNotaFiscal

```

{
  dados: Array<ItemNotaFiscal>;
  tabela: string;
}

```

ItemNotaFiscal

```

{
  bas_cofins: number;
  bas_fcp: number;
  bas_fcpst: number;
  bas_icms: number;
  bas_ipi: number;
  bas_pis: number;
  bas_substrib: number;
  cod_barra: string;
  cod_filial: integer;
  cod_modelo: string;
  cod_natureza: string;
  cod_origem: string;
  cod_produto: string;
  cod_rede: integer;
  cod_reduzido: integer;
  cod_tribut: string;
  deveSubtrairIcmsDesonerado: boolean;
  deveSubtrairIcmsStDesonerado: boolean;
  ficticio: boolean;
  indDeduzDeson: string;
  medicamento: Medicamentos;
  nom_produto: string;
  num_nota: integer;
  num_sequencial: integer;
  obs_livre_contribuinte_conteudo: string;
  obs_livre_contribuinte_id: string;
  obs_livre_fisco_conteudo: string;
  obs_livre_fisco_id: string;
  prc_cofins: number;
  prc_descto: number;
  prc_diferimento: number;
  prc_fcp: number;
}

```

```

prc_fcpst: number;
prc_icms: number;
prc_icms_st_retido: number;
prc_icmsefet: number;
prc_icmsst: number;
prc_ipi: number;
prc_margemst: number;
prc_pis: number;
prc_red_base_icms_efetivo: number;
prc_redicms: number;
prc_redicmsst: number;
qtd_produto: number;
rastros: Array<Rastros>;
ser_nota: string;
sit_cofins: string;
sit_ipi: string;
sit_pis: string;
vlr_base_icms_efetivo: number;
vlr_base_icms_st_retido: number;
vlr_cofins: number;
vlr_descsto: number;
vlr_despaccess: number;
vlr_diferimento: number;
vlr_fcp: number;
vlr_fcpst: number;
vlr_frete: number;
vlr_icms: number;
vlr_icms_efetivo: number;
vlr_icms_op: number;
vlr_icms_st_deson: number;
vlr_icms_st_retido: number;
vlr_icms_substituto: number;
vlr_icmsdeson: number;
vlr_ipi: number;
vlr_pis: number;
vlr_racrearred: number;
vlr_rtdescarred: number;
vlr_seguro: number;
vlr_substrib: number;
vlr_total: number;
vlr_unitario: number;
}

```

ItemVenda

```

{
  baseCalculoCofins: number;
  baseCalculoFcp: number;
  baseCalculoFcpST: number;
  baseCalculoIcms: number;
  baseCalculoIcmsST: number;
  baseCalculoIpi: number;
  baseCalculoPis: number;
  codBenef: string;
  codCest: string;
  codigoBarra: integer;
  codigoFilial: integer;
  codigoNatureza: string;
}

```

```

codigoOrigem: string;
codigoProduto: integer;
codigoRede: integer;
codigoTributario: string;
descricaoProduto: string;
deveSubtrairIcmsDesonerado: boolean;
deveSubtrairIcmsStDesonerado: boolean;
dfeReferenciado: DfeReferenciado;
ibsCbsProduto: IbsCbsProduto;
ncm: string;
nt2023004Vigente: boolean;
numeroSequencial: integer;
obsLivreContribuinteConteudo: string;
obsLivreContribuinteId: string;
obsLivreFiscoConteudo: string;
obsLivreFiscold: string;
percentualCalculoIpi: number;
percentualCofins: number;
percentualDesconto: number;
percentualDiferimento: number;
percentualFcp: number;
percentualFcpST: number;
percentualIcms: number;
percentualIcmsEfetivo: number;
percentualIcmsST: number;
percentualMargemSt: number;
percentualPis: number;
percentualReducaoIcms: number;
percentualReducaoIcmsSt: number;
prcIcmsStRetido: number;
quantidade: number;
situacaoIpi: string;
unidade: string;
valorCofins: number;
valorDesconto: number;
valorDespesaAcessoria: number;
valorDiferimento: number;
valorFcp: number;
valorFcpSt: number;
valorFrete: number;
valorIcms: number;
valorIcmsDesonerado: number;
valorIcmsOperacao: number;
valorIcmsSt: number;
valorIcmsStDesonerado: number;
valorIpi: number;
valorItem: number;
valorPis: number;
valorRateioAcrescimoArredondamento: number;
valorRateioDescontoArredondamento: number;
valorSeguro: number;
valorTotal: number;
valorUnitario: number;
vlrBaseIcmsStRetido: number;
vlrICMSSubstituto: number;
vlrIcmsStRetido: number;
}

```

Medicamentos

```
{
  getvPMC: string;
  numRegms: string;
}
```

NotaFiscalDevolucao

```
{
  cod_filial: integer;
  cod_fornec: integer;
  cod_modelo: integer;
  cod_rede: integer;
  cod_statusnfe: string;
  cod_transp: integer;
  dat_emissao: string;
  dat_notentra: string;
  dat_saiant: string;
  deveDestacarIcmsStEmCampoProprio: boolean;
  flg_cancelado: boolean;
  flg_devol: boolean;
  hor_emissao: string;
  itens: IntegracaoJsonItemNotaFiscal;
  min_emissao: string;
  nom_chavenfe: string;
  nom_obsnf: string;
  num_controleent: integer;
  num_nf: integer;
  num_nfficticia: integer;
  num_notentra: integer;
  num_protocoloenv: integer;
  num_seqcadinfis: integer;
  num_seqnfe: integer;
  num_ultlotenfe: integer;
  qtd_parcelas: integer;
  seg_emissao: string;
  ser_notas: string;
  tip_ambiente: string;
  tip_condpagto: string;
  tip_frete: string;
  tip_operacao: string;
  tip_origem: string;
  tip_saida: integer;
  tip_venda: string;
  tot_bruto: number;
  tot_notas: number;
  vlr_acrescarred: number;
  vlr_desconto: number;
  vlr_icmsdeson: number;
}
```

NotaFiscalOrigem

```
{
  chaveNfe: string;
}
```

```

cnpj: string;
cpf: string;
dataEmissao: string;
devolucaoTotal: boolean;
estado: string;
inscricaoEstadual: string;
modelo: string;
numero: string;
serie: string;
tipoNota: string;
}

```

PagamentoMultiploIntegracaoDto

```

{
  cartao: Array<CartaoVendaIntegracaoDTO>;
  crediario: CrediarioIntegracaoDTO;
  dinheiro: DinheiroIntegracaoDTO;
  pix: PixIntegracaoDTO;
}

```

PixIntegracaoDTO

```

{
  codigo: integer; // Código do tipo de pagamento PIX (Cartão).
  idTransacaoPIX?: string; // ID da transação PIX.
  numeroAutorizacao?: integer; // Número da autorização.
  valor: number; // Valor pago no PIX.
}

```

ProdutoDescontoMelhorIntegracaoDto

```

{
  codigoProduto: integer; // Código do produto.
  quantidade: integer; // Quantidade do produto.
}

```

Rastros

```

{
  getCAgreg: string;
  getdFab: string;
  getdVal: string;
  getnLote: string;
  getqLote: string;
}

```

StatusPedido

```

{
  codigo: string; // Código do status do pedido.
}

```

```

    descricao: string; // Descrição do status do pedido.
}

```

StatusVendaEcommerceIntegracaoRetDto

```

{
  dataPedido: string; // Data do pedido no e-commerce.
  numeroCupomFiscal: integer; // Número do cupom fiscal.
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroNotaFiscal: integer; // Número da nota fiscal.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  statusPedido: StatusPedido;
  xmlNfe: string; // XML da nota fiscal eletrônica.
}

```

Venda

```

{
  aliquotaFcpst: number;
  baseCalculoFcp: number;
  baseCalculoFcpst: number;
  codigoControle: integer;
  codigoFilial: integer;
  codigoFornecedor: integer;
  codigoRede: integer;
  codigoTransportadora: integer;
  dadosDestinatario: DadosDestinatario;
  dadosDestinatarioCreditoProprio: DadosDestinatario;
  dadosEmitente: DadosEmitente;
  dadosTransporte: DadosTransporte;
  dataSaida: string;
  informacoesAdicionais: string;
  itens: Array<ItemVenda>;
  notaOrigem: NotaFiscalOrigem;
  quantidadeParcelas: integer;
  somarSTOutrasDespesasAcessorias: boolean;
  tipoCondicaoPagamento: string;
  tipoFrete: string;
  tipoImpressao: string;
  valorAcrescimoArredondamento: number;
  valorBruto: number;
  valorFcp: number;
  valorFcpst: number;
  valorMercadoria: number;
  valorTotal: number;
  valorTotalICMS: number;
  valorTotalICMSDesonerado: number;
  valorTotalICMSST: number;
  valorTotalICMSStDesonerado: number;
}

```

VendaEcommerceFinalizacaoIntegracaoDto

```

{
  numeroNota: integer; // Número da venda gerada no PDV.
}

```

```

numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
pagamentoMultiplo?: PagamentoMultiploIntegracaoDto;
}

```

VendaEcommerceIntegracaoDto

```

{
  cliente?: ClienteEcommerceIntegracaoDto;
  dataPedido: string; // Data do pedido no e-commerce.
  enderecoEntrega?: EnderecoEntregaIntegracaoDto;
  entrega?: boolean; // Indica se é uma venda Tele-Entrega ou não.
  numeroNota?: integer; // Número da venda gerada no PDV (somente quando originado de um DAV).
  numeroPedido: string; // Número do pedido no e-commerce que será vinculado à venda.
  pagamentoMultiplo?: PagamentoMultiploIntegracaoDto;
  produtos?: Array<VendaItemEcommerceIntegracaoDto>;
  valorFrete?: number; // Valor do frete do pedido.
  valorTotalVenda: number; // Valor total do pedido, que será utilizado como total da venda.
  vendedor?: VendedorEcommerceIntegracaoDto;
}

```

VendaEcommerceIntegracaoRetDto

```

{
  numeroNota: integer; // Número da venda gerada no PDV.
  numeroPedido: string; // Número do pedido no e-commerce vinculado à venda.
  statusPedido: StatusPedido;
}

```

VendaItemEcommerceIntegracaoDto

```

{
  codigoProduto: integer; // Código do produto.
  nomeDesconto?: string; // Origem do desconto.
  nomeProduto?: string; // Nome do produto.
  numeroSequencial?: integer; // Número sequencial (somente quando originado de um DAV).
  quantidade: integer; // Quantidade do produto.
  valorDesconto?: number; // Valor unitário do desconto aplicado ao produto.
  valorUnitario: number; // Valor unitário do produto.
}

```

VendedorEcommerceIntegracaoDto

```

{
  codigo: integer; // Código do vendedor.
  nome: string; // Nome do vendedor.
}

```