

API SGF

API de integracao com o Trier Drograrias

2026-09-23

API-SGF

Overview

Este documento apresenta as especificações da API de integração com o sistema Trier Drogarias, desenvolvida para atender clientes e parceiros da Trier Sistemas. A API tem como objetivo facilitar a comunicação entre sistemas externos e o SGF, promovendo interoperabilidade e automação de processos.

A interface pode ser acessada por meio do endereço do servidor, conforme o exemplo:
http://localhost:4647/sgfpod1/rest/*

Para utilizar os métodos disponíveis, é necessário realizar autenticação via token do tipo Bearer, fornecido pela Trier Sistemas.

Informações técnicas:

- BASE_URL: <http://localhost:4647/sgfpod1> (localhost deve ser substituído pelo IP do servidor no ambiente do cliente)
- Acesso externo: requer IP fixo ou DDNS e redirecionamento da porta 4647 no modem ou roteador da farmácia (NAT ou Virtual Server)
- Acesso interno: não é necessário redirecionamento de porta

Em caso de dúvidas durante o processo de homologação ou desenvolvimento, entre em contato com a equipe de parcerias da Trier Sistemas pelo e-mail: parcerias@grupotrier.com.br.

Version

1.5.14

Contact

Suporte Trier Sistemas

<https://triersistemas.com.br/suporte/>

POST /integracao/agilego/atualizar-status-v1

Atualiza o status da venda conforme notificação do Agilego.

Para fazer uma requisição para esse endpoint, é necessário que o Agilego esteja configurado no sistema.

Request Body:

Objeto contendo as informações do webhook do Agilego.

Content: [application/json](#) | [WebhookDto](#)

```
{
  ActionBy: string; // Quem realizou a ação.
  CurrentOrderValue: number; // Valor atual do pedido.
  CustomerOrderCode: string; // Código interno do pedido.
  DeliveredTo: string; // Nome do recebedor.
  DeliveryOrderId: integer; // Identificador da ordem de entrega.
  DeliveryOrderPointId: integer; // Identificador do ponto da ordem de entrega.
  Distance: number; // Distância percorrida em quilômetros.
  DriverName: string; // Nome do entregador.
  Lat: string; // Latitude da entrega.
```

```

    Lng: string; // Longitude da entrega.
    OldOrderValue: number; // Valor antigo do pedido.
    Status: string; // Enumeração que representa os possíveis status de um pedido no Agilego.
    StoreGroupId: string; // Identificador do grupo da loja.
    StoreGroupName: string; // Nome do grupo da loja.
    StoreId: string; // Identificador da loja.
    SubStatus: string; // Enumeração que representa os substatus de um pedido no AgileGO.
    Type: string; // Identificador do status.
    UpdateAt: string; // Data e hora da atualização do status.
}

```

Response 200:

Status da venda atualizado com sucesso.

Content: `application/json` | `string`

Values: `OK`, `INTERNAL_SERVER_ERROR`, `BAD_REQUEST`

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na atualização do status da venda.

GET /integracao/cartao/obter-alterados-v1

Lista cartões alterados.

Retorna os cartões cadastrados que foram alterados dentro do intervalo de datas informado.

Request Parameters:

dataInicial: `string`; // Informar data em formato ISO 8601.

dataFinal: `string`; // Informar data em formato ISO 8601.

primeiroRegistro: `integer`; // Índice do primeiro registro a ser retornado.

quantidadeRegistros: `integer`; // Quantidade máxima de registros a serem retornados.

Response 200:

Lista de cartões alterados retornada com sucesso.

Content: `application/json` | `Array<CartaoIntegracaoRetornoDto>`

```

{
  administradora: FornecedorDto;
  ativo: boolean; // Define se o cartão está ativo ou inativo.
  carteiraDigital: string; // Carteira digital do cartão.
  codigoCartao: integer; // Código do cartão.
  descontoCartao: number;
  descontoMaximoCartao: number;
  gerenciadorTEF: BandeiraDto;
  modalidadeVenda: string; // Modalidade de venda do cartão.
  nomeCartao: string; // Nome do cartão.
  qtdMaxParcelas: integer; // Quantidade máxima de parcelas permitidas.
  taxaAdministrativa: number; // Taxa administrativa do cartão.
}

```

```

    taxaAdministrativaParcela: Array<ListaTaxaParcelaIntegracaoDto>; // Lista de taxas
    administrativas por parcela.
    vencimentoParcelas: Array<string>; // Dias de vencimento por parcela.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos cartões alterados.

GET /integracao/cartao/obter-todos-v1

Lista todos os cartões.

Retorna todos os cartões cadastrados.

Request Parameters:

primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

Response 200:

Lista de cartões retornada com sucesso.

Content: application/json | Array<CartaoIntegracaoRetornoDto>

```

{
  administradora: FornecedorDto;
  ativo: boolean; // Define se o cartão está ativo ou inativo.
  carteiraDigital: string; // Carteira digital do cartão.
  codigoCartao: integer; // Código do cartão.
  descontoCartao: number;
  descontoMaximoCartao: number;
  gerenciadorTEF: BandeiraDto;
  modalidadeVenda: string; // Modalidade de venda do cartão.
  nomeCartao: string; // Nome do cartão.
  qtdMaxParcelas: integer; // Quantidade máxima de parcelas permitidas.
  taxaAdministrativa: number; // Taxa administrativa do cartão.
  taxaAdministrativaParcela: Array<ListaTaxaParcelaIntegracaoDto>; // Lista de taxas
  administrativas por parcela.
  vencimentoParcelas: Array<string>; // Dias de vencimento por parcela.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos cartões.

GET /integracao/cartao/obter-v1

Lista cartões.

Retorna os cartões cadastrados conforme os filtros informados.

Request Parameters:

codigoCartao: **string**; // Código do cartão.
 ativo: **boolean**; // Indica se o cartão está ativo ou inativo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Lista de cartões retornada com sucesso.

Content: **application/json** | Array<CartaoIntegracaoRetornoDto>

```
{
  administradora: FornecedorDto;
  ativo: boolean; // Define se o cartão está ativo ou inativo.
  carteiraDigital: string; // Carteira digital do cartão.
  codigoCartao: integer; // Código do cartão.
  descontoCartao: number;
  descontoMaximoCartao: number;
  gerenciadorTEF: BandeiraDto;
  modalidadeVenda: string; // Modalidade de venda do cartão.
  nomeCartao: string; // Nome do cartão.
  qtdMaxParcelas: integer; // Quantidade máxima de parcelas permitidas.
  taxaAdministrativa: number; // Taxa administrativa do cartão.
  taxaAdministrativaParcela: Array<ListaTaxaParcelaIntegracaoDto>; // Lista de taxas
    administrativas por parcela.
  vencimentoParcelas: Array<string>; // Dias de vencimento por parcela.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos cartões.

GET /integracao/categoria/obter-alterados-v1

Lista categorias alteradas no sistema.

Retorna uma lista com as categorias alteradas entre as datas informadas.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Categorias alteradas retornadas com sucesso.

Content: **application/json** | **Array<CategoriaIntegracaoDto>**

```
{
  codigo: integer; // Código da categoria.
  nome: string; // Nome da categoria.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de categorias alteradas.

GET /integracao/categoria/obter-todos-v1

Lista categorias do sistema.

Retorna uma lista com todas as categorias.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Categorias retornadas com sucesso.

Content: **application/json** | **Array<CategoriaIntegracaoDto>**

```
{
  codigo: integer; // Código da categoria.
  nome: string; // Nome da categoria.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de categorias.

GET /integracao/cliente/obter-alterados-v1

Lista clientes alterados em um período.

Retorna lista de clientes que tiveram seus cadastros alterados no período informado, respeitando a paginação.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Clientes alterados retornados com sucesso.

Content: **application/json** | [Array<ClienteIntegracaoDto>](#)

```
{
  ativo?: boolean; // Indica se o cliente está ativo.
  bairro?: string; // Bairro do cliente.
  cep?: string; // CEP do cliente.
  codigo?: integer; // Código do cliente, se não informado, será gerado automaticamente pelo PDV.
  codigoCidade?: string; // Código da cidade do cliente.
  email?: string; // E-mail do cliente.
  empresaConvenio?: EmpresaConvenio;
  estado?: string; // Estado do cliente.
  fone?: string; // Número de telefone do cliente.
  grupo?: Grupo;
  logradouro?: string; // Logradouro do cliente.
  nome: string; // Nome do cliente.
  numeroCpfCnpj: string; // Número do CPF ou CNPJ do cliente.
  numeroEndereco?: string; // Número do endereço do cliente.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de clientes alterados.

GET /integracao/cliente/obter-todos-v1

Lista todos os clientes.

Retorna lista com todos os clientes cadastrados, respeitando a paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Clientes retornados com sucesso.

Content: `application/json` | `Array<ClienteIntegracaoDto>`

```
{
  ativo?: boolean; // Indica se o cliente está ativo.
  bairro?: string; // Bairro do cliente.
  cep?: string; // CEP do cliente.
  codigo?: integer; // Código do cliente, se não informado, será gerado automaticamente pelo PDV.
  codigoCidade?: string; // Código da cidade do cliente.
  email?: string; // E-mail do cliente.
  empresaConvenio?: EmpresaConvenio;
  estado?: string; // Estado do cliente.
  fone?: string; // Número de telefone do cliente.
  grupo?: Grupo;
  logradouro?: string; // Logradouro do cliente.
  nome: string; // Nome do cliente.
  numeroCpfCnpj: string; // Número do CPF ou CNPJ do cliente.
  numeroEndereco?: string; // Número do endereço do cliente.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de clientes.

GET /integracao/cliente/obter-v1

Lista clientes com filtros.

Retorna lista de clientes cadastrados conforme os filtros informados, respeitando a paginação.

Request Parameters:

```
codigo: string; // Códigos dos clientes a serem filtrados.
ativo: boolean; // Filtra clientes ativos ou inativos.
numeroCpfCnpj: string; // CPF ou CNPJ do cliente a ser filtrado.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.
```

Response 200:

Clientes retornados com sucesso.

Content: `application/json` | `Array<ClienteIntegracaoDto>`

```
{
  ativo?: boolean; // Indica se o cliente está ativo.
  bairro?: string; // Bairro do cliente.
  cep?: string; // CEP do cliente.
  codigo?: integer; // Código do cliente, se não informado, será gerado automaticamente pelo PDV.
  codigoCidade?: string; // Código da cidade do cliente.
  email?: string; // E-mail do cliente.
  empresaConvenio?: EmpresaConvenio;
  estado?: string; // Estado do cliente.
}
```

```

fone?: string; // Número de telefone do cliente.
grupo?: Grupo;
logradouro?: string; // Logradouro do cliente.
nome: string; // Nome do cliente.
numeroCpfCnpj: string; // Número do CPF ou CNPJ do cliente.
numeroEndereco?: string; // Número do endereço do cliente.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de clientes.

GET /integracao/compra/obter-alterados-v1

Lista compras alteradas.

Obtém todas as compras que foram alteradas entre as datas informadas.

Request Parameters:

```

dataInicial: string; // Informar data em formato ISO 8601.
dataFinal: string; // Informar data em formato ISO 8601.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Compras alteradas retornadas com sucesso.

Content: application/json | [Array<CompraIntegracaoDto>](#)

```

{
  chaveAcessoNfe: string; // Chave de acesso NFE.
  codigoFornecedor: integer; // Código do fornecedor.
  datEmissao: string;
  dataEntrada: string; // Data de entrada.
  itens: Array<ComprasItemIntegracaoDto>; // Itens da compra.
  numeroNotaFiscal: integer; // Número da nota fiscal.
  quantidadeItens: integer; // Quantidade de itens da nota.
  valorTotalNota: number; // Valor total da nota.
  valorTotalProdutos: number; // Valor total dos produtos.
  xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de compras alteradas.

GET /integracao/compra/obter-todos-v1

Lista todas as compras.

Obtém todas as compras cadastradas.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Compras retornadas com sucesso.

Content: **application/json** | Array<CompraIntegracaoDto>

```
{
  chaveAcessoNfe: string; // Chave de acesso NFE.
  codigoFornecedor: integer; // Código do fornecedor.
  datEmissao: string;
  dataEntrada: string; // Data de entrada.
  itens: Array<ComprasItemIntegracaoDto>; // Itens da compra.
  numeroNotaFiscal: integer; // Número da nota fiscal.
  quantidadeItens: integer; // Quantidade de itens da nota.
  valorTotalNota: number; // Valor total da nota.
  valorTotalProdutos: number; // Valor total dos produtos.
  xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de compras.

GET /integracao/compra/obter-v1

Lista compras com filtros.

Consulta a lista de compras com filtros opcionais e paginação.

Request Parameters:

dataEntradaInicial: **string**; // Data de entrada inicial em formato ISO 8601.
dataEntradaFinal: **string**; // Data de entrada final em formato ISO 8601.
chavenfe: **string**; // Chave de acesso da NF-e.
xmlNfe: **boolean**; // Quando true, retorna informações com referência ao XML da NF-e.
primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Compras retornadas com sucesso.

Content: `application/json` | `Array<CompraIntegracaoDto>`

```
{
  chaveAcessoNfe: string; // Chave de acesso NFE.
  codigoFornecedor: integer; // Código do fornecedor.
  datEmissao: string;
  dataEntrada: string; // Data de entrada.
  itens: Array<ComprasItemIntegracaoDto>; // Itens da compra.
  numeroNotaFiscal: integer; // Número da nota fiscal.
  quantidadeItens: integer; // Quantidade de itens da nota.
  valorTotalNota: number; // Valor total da nota.
  valorTotalProdutos: number; // Valor total dos produtos.
  xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de compras.

GET /integracao/estoque/obter-alterados-v1

Lista estoques de produtos que possuem alteração.

Retorna lista de estoques de produtos alterados no período especificado.

Request Parameters:

```
integracaoEcommerce: boolean; // Filtra produtos disponíveis ou não disponíveis para integração com E-Commerce.
dataInicial: string; // Informar data em formato ISO 8601.
dataFinal: string; // Informar data em formato ISO 8601.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.
```

Response 200:

Estoques alterados retornados com sucesso.

Content: `application/json` | `Array<EstoqueIntegracaoDto>`

```
{
  alterCurva: string;
  codigoProduto: integer; // Código reduzido de um produto.
  dataUltimaEntrada: string; // Data da última entrada do produto.
  quantidadeEstoque: integer; // Quantidade em estoque.
  valorCustoMedio: number; // Custo médio do produto.
  valorUltimaEntrada: number; // Valor da última entrada do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de estoques alterados.

GET /integracao/estoque/obter-movimentados-v1

Lista estoques de produtos que possuem movimentações.

Retorna lista de estoques de produtos movimentados no período especificado.

Request Parameters:

integracaoEcommerce: **boolean**; // Filtra produtos disponíveis ou não disponíveis para integração com E-Commerce.
 dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Estoques movimentados retornados com sucesso.

Content: **application/json** | **Array<EstoqueIntegracaoDto>**

```
{
  alterCurva: string;
  codigoProduto: integer; // Código reduzido de um produto.
  dataUltimaEntrada: string; // Data da última entrada do produto.
  quantidadeEstoque: integer; // Quantidade em estoque.
  valorCustoMedio: number; // Custo médio do produto.
  valorUltimaEntrada: number; // Valor da última entrada do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de estoques movimentados.

GET /integracao/estoque/obter-todos-v1

Lista todos os estoques de produtos.

Retorna o estoque de produtos com filtros opcionais e paginação.

Request Parameters:

integracaoEcommerce: **boolean**; // Filtra produtos disponíveis ou não disponíveis para integração com E-Commerce.

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Estoques retornados com sucesso.

Content: **application/json** | **Array<EstoqueIntegracaoDto>**

```
{
  alterCurva: string;
  codigoProduto: integer; // Código reduzido de um produto.
  dataUltimaEntrada: string; // Data da última entrada do produto.
  quantidadeEstoque: integer; // Quantidade em estoque.
  valorCustoMedio: number; // Custo médio do produto.
  valorUltimaEntrada: number; // Valor da última entrada do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de estoques.

GET /integracao/estoque/obter-v1

Consulta estoques de produtos com filtros avançados.

Consulta o estoque de produtos com filtros avançados.

Request Parameters:

codigoProduto: **integer**; // Código reduzido do produto.
 quantidadeEstoqueInicial: **integer**; // Exibe os produtos com estoque maior ou igual ao valor informado.
 quantidadeEstoqueFinal: **integer**; // Filtra produtos com estoque negativo. Pode ser combinado com quantidadeEstoqueInicial para definir uma faixa.
 ativo: **boolean**; // Filtra por estoque de produtos ativos ou inativos.
 integracaoEcommerce: **boolean**; // Filtra produtos disponíveis ou não disponíveis para integração com E-Commerce.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Estoques retornados com sucesso.

Content: **application/json** | **Array<EstoqueIntegracaoDto>**

```
{
  alterCurva: string;
  codigoProduto: integer; // Código reduzido de um produto.
  dataUltimaEntrada: string; // Data da última entrada do produto.
  quantidadeEstoque: integer; // Quantidade em estoque.
  valorCustoMedio: number; // Custo médio do produto.
  valorUltimaEntrada: number; // Valor da última entrada do produto.
}
```

```
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de estoques.

GET /integracao/financeiro/conta/obter-alterados-v1

Lista contas que possuem alteração.

Retorna lista de contas alteradas no período especificado.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Contas alteradas retornadas com sucesso.

Content: **application/json** | [Array<FinanceiroContaIntegracaoDto>](#)

```
{
  dataAtualizacao: string; // Data de atualização.
  digito: string; // Dígito da conta.
  digitoAgencia: string; // Dígito da agência.
  nome: string; // Nome da conta.
  numero: string; // Número da conta.
  numeroAgencia: string; // Número da agência.
  saldo: number; // Saldo da conta.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de contas alteradas.

GET /integracao/financeiro/conta/obter-movimentados-v1

Lista contas que possuem movimentações.

Retorna lista de contas movimentadas no período especificado.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Contas movimentadas retornadas com sucesso.

Content: **application/json** | **Array<FinanceiroContaIntegracaoDto>**

```
{
  dataAtualizacao: string; // Data de atualização.
  digito: string; // Dígito da conta.
  digitoAgencia: string; // Dígito da agência.
  nome: string; // Nome da conta.
  numero: string; // Número da conta.
  numeroAgencia: string; // Número da agência.
  saldo: number; // Saldo da conta.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de contas movimentadas.

GET /integracao/financeiro/conta/obter-todos-v1

Lista todas as contas financeiras.

Consulta contas com filtros opcionais e paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Contas retornadas com sucesso.

Content: **application/json** | **Array<FinanceiroContaIntegracaoDto>**

```
{
  dataAtualizacao: string; // Data de atualização.
  digito: string; // Dígito da conta.
  digitoAgencia: string; // Dígito da agência.
  nome: string; // Nome da conta.
  numero: string; // Número da conta.
  numeroAgencia: string; // Número da agência.
  saldo: number; // Saldo da conta.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de contas.

GET /integracao/financeiro/departamento/obter-alterados-v2

Lista departamentos que possuem alteração.

Retorna lista de departamentos alterados no período especificado.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Departamentos alterados retornados com sucesso.

Content: **application/json** | **Array<DepartamentoIntegracaoDto>**

```
{
  ativo: boolean; // Define se o departamento está ativo ou inativo.
  codigo: integer; // Código do departamento.
  nome: string; // Nome do departamento.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de departamentos alterados.

GET /integracao/financeiro/departamento/obter-movimentados-v2

Lista departamentos que possuem movimentações.

Retorna lista de departamentos movimentados no período especificado.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.

quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Departamentos movimentados retornados com sucesso.

Content: **application/json** | **Array<DepartamentoIntegracaoDto>**

```
{
  ativo: boolean; // Define se o departamento está ativo ou inativo.
  codigo: integer; // Código do departamento.
  nome: string; // Nome do departamento.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de departamentos movimentados.

GET /integracao/financeiro/departamento/obter-todos-v2

Lista todos os departamentos.

Consulta departamentos com filtros opcionais e paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Departamentos retornados com sucesso.

Content: **application/json** | **Array<DepartamentoIntegracaoDto>**

```
{
  ativo: boolean; // Define se o departamento está ativo ou inativo.
  codigo: integer; // Código do departamento.
  nome: string; // Nome do departamento.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de departamentos.

GET /integracao/financeiro/departamento/obter-v2

Consulta departamentos com filtros avançados.

Consulta departamentos com filtros avançados.

Request Parameters:

codigoDepartamento: **string**; // Código do departamento.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Departamentos retornados com sucesso.

Content: **application/json** | Array<DepartamentoIntegracaoDto>

```
{
  ativo: boolean; // Define se o departamento está ativo ou inativo.
  codigo: integer; // Código do departamento.
  nome: string; // Nome do departamento.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de departamentos.

GET /integracao/financeiro/obter-classificacao-financeira-v1

Lista centros de custos.

Retorna lista de classificações financeiras no período especificado.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Classificações financeiras retornadas com sucesso.

Content: **application/json** | Array<FinanceiroClassificacaoFinanceiraIntegracaoDto>

```
{
  ativo: boolean; // Ativo ou inativo.
  centroCusto: string; // Código do centro de custo que pertence.
  creditoDebito: string; // D = Débito / C = Crédito.
  fixoVariavel: string; // F = Fixo / V = Variável.
  nomeDescricao: string; // Nome do centro de custo que pertence.
  tipoClassificacao: string; // D = Despesa / R = Receita.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta das classificações financeiras.

GET /integracao/financeiro/obter-resultado-exercicio-v1

Lista resultados do exercício.

Retorna lista de resultado do exercício no período especificado.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Resultado do exercício retornado com sucesso.

Content: **application/json** | **Array<FinanceiroResultadoExercicioIntegracaoDto>**

```
{
  centroCusto: string; // Centro de custo que ele pertence.
  codigoDepartamento: integer; // Código departamento.
  data: string; // Data de lançamento do movimento.
  valorBruto: number; // Valor bruto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta do resultado do exercício.

GET /integracao/financeiro/obter-resultado-financeiro-v1

Lista resultados financeiros.

Retorna lista de resultado financeiro no período especificado.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Resultado financeiro retornado com sucesso.

Content: **application/json** | [Array<FinanceiroResultadoFinanceiroIntegracaoDto>](#)

```
{
  baixadoAberto: string; // Movimento baixado ou aberto.
  centroCusto: string; // Centro de custo que pertence.
  codigoDepartamento: integer; // Código do departamento.
  data: string; // Data de baixa do movimento.
  valorBruto: number; // Valor bruto do movimento.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta do resultado financeiro.

GET /integracao/fornecedor/obter-alterados-v1

Lista fornecedores alterados.

Obtém os fornecedores que foram alterados dentro de um intervalo de datas.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Fornecedores alterados retornados com sucesso.

Content: **application/json** | [FornecedorIntegracaoDto](#)

```
{
  ativo: boolean; // Ativo ou inativo.
  codigo: integer; // Código do fornecedor.
  email: string; // E-mail do fornecedor.
  nomeCidade: string; // Cidade do fornecedor.
  nomeFantasia: string; // Nome fantasia do fornecedor.
  numeroCnpj: string; // CNPJ do fornecedor.
  razaoSocial: string; // Razão social do fornecedor.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de fornecedores alterados.

GET /integracao/fornecedor/obter-movimentados-v1

Lista fornecedores movimentados.

Obtém os fornecedores que foram movimentados dentro de um intervalo de datas.

Request Parameters:

dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Fornecedores movimentados retornados com sucesso.

Content: **application/json** | FornecedorIntegracaoDto

```
{
  ativo: boolean; // Ativo ou inativo.
  codigo: integer; // Código do fornecedor.
  email: string; // E-mail do fornecedor.
  nomeCidade: string; // Cidade do fornecedor.
  nomeFantasia: string; // Nome fantasia do fornecedor.
  numeroCnpj: string; // CNPJ do fornecedor.
  razaoSocial: string; // Razão social do fornecedor.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de fornecedores movimentados.

GET /integracao/fornecedor/obter-todos-v1

Lista todos os fornecedores.

Obtém todos os fornecedores cadastrados no sistema.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Fornecedores retornados com sucesso.

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

```
{
  ativo: boolean; // Ativo ou inativo.
  codigo: integer; // Código do fornecedor.
  email: string; // E-mail do fornecedor.
  nomeCidade: string; // Cidade do fornecedor.
  nomeFantasia: string; // Nome fantasia do fornecedor.
  numeroCnpj: string; // CNPJ do fornecedor.
  razaoSocial: string; // Razão social do fornecedor.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de fornecedores.

GET `/integracao/laboratorio/obter-alterados-v1`

Obtém laboratórios alterados.

Obtém os laboratórios que foram alterados dentro de um intervalo de datas.

Request Parameters:

dataInicial: `string`; // Data inicial em formato ISO 8601.
 dataFinal: `string`; // Data final em formato ISO 8601.
 primeiroRegistro: `integer`; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: `integer`; // Quantidade máxima de registros a serem retornados.

Response 200:

Laboratórios alterados retornados com sucesso.

Content: `application/json` | [Array<LaboratorioIntegracaoDto>](#)

```
{
  ativo: boolean; // Define se o laboratório está ativo ou inativo.
  cnpj: string; // CNPJ do laboratório.
  codigo: integer; // Código do laboratório.
  nome: string; // Nome do laboratório.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos laboratórios alterados.

GET /integracao/laboratorio/obter-movimentados-v1

Obtém laboratórios movimentados.

Obtém os laboratórios que foram movimentados dentro de um intervalo de datas.

Request Parameters:

dataInicial: **string**; // Data inicial em formato ISO 8601.
dataFinal: **string**; // Data final em formato ISO 8601.
primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Laboratórios movimentados retornados com sucesso.

Content: **application/json** | [Array<LaboratorioIntegracaoDto>](#)

```
{
  ativo: boolean; // Define se o laboratório está ativo ou inativo.
  cnpj: string; // CNPJ do laboratório.
  codigo: integer; // Código do laboratório.
  nome: string; // Nome do laboratório.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos laboratórios movimentados.

GET /integracao/laboratorio/obter-todos-v1

Obtém todos os laboratórios.

Obtém todos os laboratórios cadastrados no sistema.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Laboratórios retornados com sucesso.

Content: **application/json** | [Array<LaboratorioIntegracaoDto>](#)

```
{
  ativo: boolean; // Define se o laboratório está ativo ou inativo.
}
```

```

    cnpj: string; // CNPJ do laboratório.
    codigo: integer; // Código do laboratório.
    nome: string; // Nome do laboratório.
  }

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos laboratórios.

GET /integracao/parcelas-cartao/estorno-v1

Obtém parcelas de cartão de estorno.

Obtém as parcelas de cartão de estorno cadastradas no sistema, com filtros de data de emissão.

Request Parameters:

```

dataEmissaoInicial: string; // Data inicial de emissão em formato ISO 8601.
dataEmissaoFinal: string; // Data final de emissão em formato ISO 8601.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Parcelas de cartão de estorno retornadas com sucesso.

Content: application/json | [ParcelaCartaoEstornoIntegracaoDto](#)

```

{
  cnpjLoja: string; // Número do CNPJ da filial.
  codigoLoja: integer; // Código da filial.
  estornos: Array<ParcelaCartaoEstornoTransacaoIntegracaoDto>; // Lista de estornos de
    parcelas de cartão.
  inscricaoEstadualLoja: string; // Inscrição estadual da filial.
  razaoSocial: string; // Razão social da filial.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta das parcelas de cartão de estorno.

GET /integracao/parcelas-cartao/obter-alterados-v1

Obtém parcelas de cartão alteradas.

Obtém as parcelas de cartão cadastradas no sistema que foram alteradas em um determinado período.

Request Parameters:

dataInicial: **string**; // Data inicial em formato ISO 8601.
 dataFinal: **string**; // Data final em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Parcelas de cartão alteradas retornadas com sucesso.

Content: **application/json** | ParcelaCartaoIntegracaoDto

```
{
  cnpjLoja: string; // Número do CNPJ da filial.
  codigoLoja: integer; // Código da filial.
  inscricaoEstadualLoja: string; // Inscrição estadual da filial.
  razaoSocial: string; // Razão social da filial.
  transacoes: Array<ParcelaCartaoTransacaoIntegracaoDto>; // Lista de parcelas de cartão.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta das parcelas de cartão alteradas.

GET /integracao/parcelas-cartao/obter-todos-v1

Obtém todas as parcelas de cartão.

Obtém todas as parcelas de cartão cadastradas no sistema.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Parcelas de cartão retornadas com sucesso.

Content: **application/json** | ParcelaCartaoIntegracaoDto

```
{
  cnpjLoja: string; // Número do CNPJ da filial.
  codigoLoja: integer; // Código da filial.
  inscricaoEstadualLoja: string; // Inscrição estadual da filial.
  razaoSocial: string; // Razão social da filial.
  transacoes: Array<ParcelaCartaoTransacaoIntegracaoDto>; // Lista de parcelas de cartão.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta das parcelas de cartão.

GET /integracao/parcelas-cartao/obter-v1

Obtém parcelas de cartão.

Obtém as parcelas de cartão cadastradas no sistema, com filtros de data de emissão.

Request Parameters:

dataEmissaoInicial: **string**; // Data inicial de emissão em formato ISO 8601.
dataEmissaoFinal: **string**; // Data final de emissão em formato ISO 8601.
primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Parcelas de cartão retornadas com sucesso.

Content: **application/json** | ParcelaCartaoIntegracaoDto

```
{  
  cnpjLoja: string; // Número do CNPJ da filial.  
  codigoLoja: integer; // Código da filial.  
  inscricaoEstadualLoja: string; // Inscrição estadual da filial.  
  razaoSocial: string; // Razão social da filial.  
  transacoes: Array<ParcelaCartaoTransacaoIntegracaoDto>; // Lista de parcelas de cartão.  
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta das parcelas de cartão.

GET /integracao/pedido/itens/resumido/obter-alterados-v1

Obtém itens de pedidos resumidos alterados.

Obtém itens de pedidos resumidos cadastrados no sistema que foram alterados dentro do intervalo de datas fornecido.

Request Parameters:

dataInicial: **string**; // Data inicial em formato ISO 8601.
dataFinal: **string**; // Data final em formato ISO 8601.

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Itens de pedidos resumidos alterados retornados com sucesso.

Content: **application/json** | [Array<ItemPedidoResumidoIntegracaoDto>](#)

```
{
  cnpjFornecedorUltimaCompra: string; // CNPJ do fornecedor da última compra do produto.
  codigoBarras: integer; // Código de barras do produto.
  codigoComprador: integer; // Código do usuário que efetuou o pedido no sistema.
  codigoFornecedor: integer; // Código do fornecedor do pedido.
  codigoFornecedorUltimaCompra: integer; // Código do fornecedor da última compra do produto.
  codigoProduto: integer; // Código do produto.
  dataEmissao: string; // Data de emissão do pedido no formato ISO 8601.
  fatorCompra: integer; // Fator de compra do produto.
  fatorCompraUltimaCompra: integer; // Fator de compra da última compra do produto.
  horaEmissao: string; // Hora de emissão do pedido no formato ISO 8601.
  nomeComprador: string; // Nome do comprador.
  nomeFornecedor: string; // Nome do fornecedor do pedido.
  nomeFornecedorUltimaCompra: string; // Nome do fornecedor da última compra do produto.
  nomeProduto: string; // Nome do produto.
  numeroPedido: integer; // Número do pedido.
  numeroSequencialItem: integer; // Número sequencial do item no pedido.
  quantidadeEstoque: integer; // Quantidade do produto em estoque.
  quantidadeProdutos: integer; // Quantidade de produtos no item do pedido.
  transmitido: boolean; // Indica se o pedido foi transmitido.
  valorUnitario: number; // Valor unitário do produto no item do pedido.
  valorUnitarioUltimaCompra: number; // Valor unitário da última compra do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos itens de pedidos resumidos alterados.

GET /integracao/pedido/itens/resumido/obter-todos-v1

Obtém todos os itens de pedidos resumidos.

Obtém todos os itens de pedidos resumidos cadastrados no sistema.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Itens de pedidos resumidos retornados com sucesso.

Content: **application/json** | [Array<ItemPedidoResumidoIntegracaoDto>](#)

```

{
  cnpjFornecedorUltimaCompra: string; // CNPJ do fornecedor da última compra do produto.
  codigoBarras: integer; // Código de barras do produto.
  codigoComprador: integer; // Código do usuário que efetuou o pedido no sistema.
  codigoFornecedor: integer; // Código do fornecedor do pedido.
  codigoFornecedorUltimaCompra: integer; // Código do fornecedor da última compra do produto.
  codigoProduto: integer; // Código do produto.
  dataEmissao: string; // Data de emissão do pedido no formato ISO 8601.
  fatorCompra: integer; // Fator de compra do produto.
  fatorCompraUltimaCompra: integer; // Fator de compra da última compra do produto.
  horaEmissao: string; // Hora de emissão do pedido no formato ISO 8601.
  nomeComprador: string; // Nome do comprador.
  nomeFornecedor: string; // Nome do fornecedor do pedido.
  nomeFornecedorUltimaCompra: string; // Nome do fornecedor da última compra do produto.
  nomeProduto: string; // Nome do produto.
  numeroPedido: integer; // Número do pedido.
  numeroSequencialItem: integer; // Número sequencial do item no pedido.
  quantidadeEstoque: integer; // Quantidade do produto em estoque.
  quantidadeProdutos: integer; // Quantidade de produtos no item do pedido.
  transmitido: boolean; // Indica se o pedido foi transmitido.
  valorUnitario: number; // Valor unitário do produto no item do pedido.
  valorUnitarioUltimaCompra: number; // Valor unitário da última compra do produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos itens de pedidos resumidos.

GET /integracao/pedido/itens/resumido/obter-v1

Obtém itens de pedidos resumidos.

Obtém itens de pedidos resumidos cadastrados no sistema com base em filtros opcionais.

Request Parameters:

```

dataEmissaoInicial: string; // Data de emissão inicial para filtro dos pedidos.
dataEmissaoFinal: string; // Data de emissão final para filtro dos pedidos.
numeroPedido: string; // Número do pedido para filtro.
codigoComprador: string; // Código do comprador para filtro.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Itens de pedidos resumidos retornados com sucesso.

Content: application/json | Array<ItemPedidoResumidoIntegracaoDto>

```

{
  cnpjFornecedorUltimaCompra: string; // CNPJ do fornecedor da última compra do produto.
  codigoBarras: integer; // Código de barras do produto.
  codigoComprador: integer; // Código do usuário que efetuou o pedido no sistema.

```

```

codigoFornecedor: integer; // Código do fornecedor do pedido.
codigoFornecedorUltimaCompra: integer; // Código do fornecedor da última compra do produto.
codigoProduto: integer; // Código do produto.
dataEmissao: string; // Data de emissão do pedido no formato ISO 8601.
fatorCompra: integer; // Fator de compra do produto.
fatorCompraUltimaCompra: integer; // Fator de compra da última compra do produto.
horaEmissao: string; // Hora de emissão do pedido no formato ISO 8601.
nomeComprador: string; // Nome do comprador.
nomeFornecedor: string; // Nome do fornecedor do pedido.
nomeFornecedorUltimaCompra: string; // Nome do fornecedor da última compra do produto.
nomeProduto: string; // Nome do produto.
numeroPedido: integer; // Número do pedido.
numeroSequencialItem: integer; // Número sequencial do item no pedido.
quantidadeEstoque: integer; // Quantidade do produto em estoque.
quantidadeProdutos: integer; // Quantidade de produtos no item do pedido.
transmitido: boolean; // Indica se o pedido foi transmitido.
valorUnitario: number; // Valor unitário do produto no item do pedido.
valorUnitarioUltimaCompra: number; // Valor unitário da última compra do produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos itens de pedidos resumidos.

GET /integracao/pedido/resumido/obter-alterados-v1

Obtém pedidos resumidos alterados.

Obtém pedidos resumidos cadastrados no sistema que foram alterados dentro do intervalo de datas fornecido.

Request Parameters:

```

dataInicial: string; // Data inicial em formato ISO 8601.
dataFinal: string; // Data final em formato ISO 8601.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Pedidos resumidos alterados retornados com sucesso.

Content: application/json | [Array<PedidoResumidoIntegracaoDto>](#)

```

{
  codigoComprador: integer; // Código do usuário que efetuou o pedido no sistema.
  dataEmissao: string; // Data de emissão do pedido no formato ISO 8601.
  horaEmissao: string; // Hora de emissão do pedido no formato ISO 8601.
  nomeComprador: string; // Nome do comprador.
  numeroPedido: integer; // Número do pedido.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos pedidos resumidos alterados.

GET /integracao/pedido/resumido/obter-todos-v1

Obtém todos os pedidos resumidos.

Obtém todos os pedidos resumidos cadastrados no sistema.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Pedidos resumidos retornados com sucesso.

Content: **application/json** | **Array<PedidoResumidoIntegracaoDto>**

```
{
  codigoComprador: integer; // Código do usuário que efetuou o pedido no sistema.
  dataEmissao: string; // Data de emissão do pedido no formato ISO 8601.
  horaEmissao: string; // Hora de emissão do pedido no formato ISO 8601.
  nomeComprador: string; // Nome do comprador.
  numeroPedido: integer; // Número do pedido.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos pedidos resumidos.

GET /integracao/pedido/resumido/obter-v1

Obtém pedidos resumidos.

Obtém pedidos resumidos cadastrados no sistema com base em filtros opcionais.

Request Parameters:

dataEmissaoInicial: **string**; // Data de emissão inicial para filtro dos pedidos.
dataEmissaoFinal: **string**; // Data de emissão final para filtro dos pedidos.
numeroPedido: **string**; // Número do pedido para filtro.

codigoComprador: **string**; // Código do comprador para filtro.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Pedidos resumidos retornados com sucesso.

Content: **application/json** | **Array<PedidoResumidoIntegracaoDto>**

```
{
  codigoComprador: integer; // Código do usuário que efetuou o pedido no sistema.
  dataEmissao: string; // Data de emissão do pedido no formato ISO 8601.
  horaEmissao: string; // Hora de emissão do pedido no formato ISO 8601.
  nomeComprador: string; // Nome do comprador.
  numeroPedido: integer; // Número do pedido.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos pedidos resumidos.

GET /integracao/produto/desconto/condicao-pagamento/obter-alterados-v1

Obter descontos por produto e condição de pagamento alterados no período.

Consulta descontos alterados por produto e condição de pagamento em um período específico, com paginação e filtros opcionais.

Request Parameters:

codigoProduto: **string**; // Código interno do produto para filtro exato.
 codigoBarras: **string**; // Código de barras do produto para filtro exato.
 nomeProduto: **string**; // Nome ou parte do nome do produto para filtro.
 codigoCondicaoPagamento: **string**; // Código da condição de pagamento para filtro.
 somenteAtivos: **boolean**; // Filtrar somente produtos ativos.
 removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 dataInicial: **string**; // Data inicial para filtro de alteração dos descontos (formato: yyyy-MM-dd).
 dataFinal: **string**; // Data final para filtro de alteração dos descontos (formato: yyyy-MM-dd).
 primeiroRegistro: **integer**; // Índice do primeiro registro (offset) para paginação. Inicia em 0.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros por página.

Response 200:

Consulta realizada com sucesso.

Content: **application/json** | **Array<ProdutoDescontoCondicaoPagamentoIntegracaoRetornoDto>**

```
{
  codigoFiscal: integer;
  condicoesPagamento: Array<CondicaoPagamentoDto>;
  dataFim: string;
  dataInicio: string;
}
```

```

diasSemana: Array<string>;
nomeCampanha: string;
percentualDesconto: number;
percentualDescontoMax: number;
produto: ProdutoCondicaoPagamentoDto;
valorPromocao: number;
valorVenda: number;
}

```

Response 400:

Requisição inválida ou parâmetros de data inválidos.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por produto e condição de pagamento.

GET /integracao/produto/desconto/condicao-pagamento/obter-todos-v1

Obter todos os descontos por produto e condição de pagamento.

Consulta todos os descontos vigentes por produto e condição de pagamento, com paginação.

Request Parameters:

```

removerRestricaoEstoque: boolean; // Remove a filtragem de somente produtos com estoque
positivo.
primeiroRegistro: integer; // Índice do primeiro registro (offset) para paginação. Inicia em 0.
quantidadeRegistros: integer; // Quantidade máxima de registros por página.

```

Response 200:

Consulta realizada com sucesso.

Content: application/json | Array<ProdutoDescontoCondicaoPagamentoIntegracaoRetornoDto>

```

{
  codigoFilial: integer;
  condicoesPagamento: Array<CondicaoPagamentoDto>;
  dataFim: string;
  dataInicio: string;
  diasSemana: Array<string>;
  nomeCampanha: string;
  percentualDesconto: number;
  percentualDescontoMax: number;
  produto: ProdutoCondicaoPagamentoDto;
  valorPromocao: number;
  valorVenda: number;
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por produto e condição de pagamento.

GET **/integracao/produto/desconto/condicao-pagamento/obter-v1**

Obter descontos por produto e condição de pagamento.

Consulta descontos vigentes por produto e condição de pagamento, com paginação e filtros opcionais.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro (offset) para paginação. Inicia em 0.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros por página.
 dataReferencia: **string**; // Data de referência para verificar a vigência dos descontos. Se não informada, utiliza a data atual.
 codigoProduto: **integer**; // Código interno do produto para filtro exato.
 codigoBarras: **integer**; // Código de barras do produto para filtro exato.
 nomeProduto: **string**; // Nome ou parte do nome do produto para filtro.
 codigoCondicaoPagamento: **integer**; // Código da condição de pagamento para filtro.
 somenteAtivos: **boolean**; // Filtrar somente produtos ativos.
 removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.

Response 200:

Consulta realizada com sucesso.

Content: **application/json** | Array<ProdutoDescontoCondicaoPagamentoIntegracaoRetornoDto>

```
{
  codigoFilial: integer;
  condicoesPagamento: Array<CondicaoPagamentoDto>;
  dataFim: string;
  dataInicio: string;
  diasSemana: Array<string>;
  nomeCampanha: string;
  percentualDesconto: number;
  percentualDescontoMax: number;
  produto: ProdutoCondicaoPagamentoDto;
  valorPromocao: number;
  valorVenda: number;
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por produto e condição de pagamento.

GET **/integracao/produto/desconto/empresa-grupo-produto/obter-alterados-v1**

Obtém descontos por empresa e grupo de produto alterados no período.

Retorna descontos por empresa e grupo de produto que foram alterados no período informado. Caso seja utilizado o token do central, serão retornados os descontos de todas as lojas. Caso seja utilizado o token de uma filial, retorna somente os descontos da filial em questão.

Request Parameters:

nomeEmpresa: **string**; // Nome (ou parte do nome) da empresa convênio.
 codigoEmpresa: **string**; // Código da empresa convênio.
 nomeGrupoProduto: **string**; // Nome (ou parte do nome) do grupo de produtos.
 codigoGrupoProduto: **string**; // Código do grupo de produtos.
 dataInicial: **string**; // Data/hora inicial da última alteração a ser considerada, em formato ISO 8601.
 dataFinal: **string**; // Data/hora final da última alteração a ser considerada, em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos por empresa e grupo de produto alterados retornados com sucesso.

Content: **application/json** | **Array<DescontoEmpresaGrupoProdutoIntegracaoDto>**

```
{
  codigoEmpresa: integer; // Código da empresa (convênio).
  codigoFilial: integer; // Código da filial.
  codigoGrupoProduto: integer; // Código do grupo de produtos.
  condicoesPagamento: Array<CondicaoPagamentoDto>; // Lista de condições de pagamento
    vinculadas ao desconto.
  dataFim: string; // Data final de vigência da promoção.
  dataInicio: string; // Data inicial de vigência da promoção.
  nomeEmpresa: string; // Nome da empresa (convênio).
  nomeGrupoProduto: string; // Nome do grupo de produtos.
  percentualDesconto: number; // Percentual de desconto cadastrado para o grupo de produto.
  percentualDescontoMax: number; // Percentual de desconto máximo para o grupo de produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos por empresa e grupo de produto alterados.

GET /integracao/produto/desconto/empresa-grupo-produto/obter-todos-v1

Obtém todos os descontos por empresa e grupo de produto.

Retorna todos os descontos cadastrados por empresa e grupo de produto. Caso seja utilizado o token do central, serão retornados os descontos de todas as lojas. Caso seja utilizado o token de uma filial, retorna somente os descontos da filial em questão.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos por empresa e grupo de produto retornados com sucesso.

Content: `application/json` | `Array<DescontoEmpresaGrupoProdutoIntegracaoDto>`

```
{
  codigoEmpresa: integer; // Código da empresa (convênio).
  codigoFilial: integer; // Código da filial.
  codigoGrupoProduto: integer; // Código do grupo de produtos.
  condicoesPagamento: Array<CondicaoPagamentoDto>; // Lista de condições de pagamento
    vinculadas ao desconto.
  dataFim: string; // Data final de vigência da promoção.
  dataInicio: string; // Data inicial de vigência da promoção.
  nomeEmpresa: string; // Nome da empresa (convênio).
  nomeGrupoProduto: string; // Nome do grupo de produtos.
  percentualDesconto: number; // Percentual de desconto cadastrado para o grupo de produto.
  percentualDescontoMax: number; // Percentual de desconto máximo para o grupo de produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos por empresa e grupo de produto.

GET /integracao/produto/desconto/empresa-grupo-produto/obter-v1

Obtém descontos por empresa e grupo de produto com filtros.

Retorna descontos cadastrados por empresa e grupo de produto, permitindo filtros por nome da empresa, nome do grupo de produtos, código do grupo de produtos e período de vigência. Caso seja utilizado o token do central, serão retornados os descontos de todas as lojas. Caso seja utilizado o token de uma filial, retorna somente os descontos da filial em questão.

Request Parameters:

```
nomeEmpresa: string; // Nome (ou parte do nome) da empresa convênio.
codigoEmpresa: string; // Código da empresa convênio.
nomeGrupoProduto: string; // Nome (ou parte do nome) do grupo de produtos.
codigoGrupoProduto: string; // Código do grupo de produtos.
dataReferencia: string; // Data de vigência em formato ISO 8601.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.
```

Response 200:

Descontos por empresa e grupo de produto retornados com sucesso.

Content: `application/json` | `Array<DescontoEmpresaGrupoProdutoIntegracaoDto>`

```
{
  codigoEmpresa: integer; // Código da empresa (convênio).
  codigoFilial: integer; // Código da filial.
  codigoGrupoProduto: integer; // Código do grupo de produtos.
  condicoesPagamento: Array<CondicaoPagamentoDto>; // Lista de condições de pagamento
    vinculadas ao desconto.
  dataFim: string; // Data final de vigência da promoção.
```

```

dataInicio: string; // Data inicial de vigência da promoção.
nomeEmpresa: string; // Nome da empresa (convênio).
nomeGrupoProduto: string; // Nome do grupo de produtos.
percentualDesconto: number; // Percentual de desconto cadastrado para o grupo de produto.
percentualDescontoMax: number; // Percentual de desconto máximo para o grupo de produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos por empresa e grupo de produto.

GET **/integracao/produto/desconto/encarte/obter-alterados-v1**

Obter descontos de encarte alterados.

Retorna os produtos com descontos em encartes que sofreram alterações dentro do intervalo de datas informado.

Request Parameters:

```

primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.
dataInicial: string; // Data inicial em formato ISO 8601.
dataFinal: string; // Data final em formato ISO 8601.

```

Response 200:

Lista de descontos de encarte alterados retornada com sucesso.

Content: **application/json** | [Array<ProdutoDescontoEncarteIntegracaoRetorno>](#)

```

{
  codigoEncarte: integer; // Código identificador do encarte.
  dataFim: string; // Data de término da vigência do encarte.
  dataInicio: string; // Data de início da vigência do encarte.
  diasSemana: Array<string>; // Dias da semana em que o encarte é válido.
  nomeCampanha: string; // Nome da campanha/encarte.
  produtosEncarte: ProdutoEncarteDto;
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por grupo e condição de pagamento.

GET /integracao/produto/desconto/encarte/obter-todos-v1

Obter todos os descontos de encarte.

Retorna todos os produtos com descontos em encartes ativos.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Lista completa de descontos de encarte retornada com sucesso.

Content: **application/json** | Array<ProdutoDescontoEncarteIntegracaoRetorno>

```
{
  codigoEncarte: integer; // Código identificador do encarte.
  dataFim: string; // Data de término da vigência do encarte.
  dataInicio: string; // Data de início da vigência do encarte.
  diasSemana: Array<string>; // Dias da semana em que o encarte é válido.
  nomeCampanha: string; // Nome da campanha/encarte.
  produtosEncarte: ProdutoEncarteDto;
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por grupo e condição de pagamento.

GET /integracao/produto/desconto/encarte/obter-v1

Obter descontos de encarte.

Retorna uma lista de produtos com descontos ativos em encartes, filtrando por parâmetros opcionais de consulta.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
dataReferencia: **string**; // Data de referência para considerar os descontos (formato: ISO 8601). Se não informada, utiliza a data atual.
codigoProduto: **integer**; // Código interno do produto.
codigoBarras: **integer**; // Código de barras do produto.
nomeProduto: **string**; // Nome (ou parte do nome) do produto para filtro.
nomeEncarte: **string**; // Nome (ou parte do nome) do encarte para filtro.

Response 200:

Lista de descontos de encarte retornada com sucesso.

Content: **application/json** | Array<ProdutoDescontoEncarteIntegracaoRetorno>

```
{
```

```

codigoEncarte: integer; // Código identificador do encarte.
dataFim: string; // Data de término da vigência do encarte.
dataInicio: string; // Data de início da vigência do encarte.
diasSemana: Array<string>; // Dias da semana em que o encarte é válido.
nomeCampanha: string; // Nome da campanha/encarte.
produtosEncarte: ProdutoEncarteDto;
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por grupo e condição de pagamento.

GET /integracao/produto/desconto/grupo/obter-alterados-v1

Obtém descontos por grupo e condição de pagamento alterados.

Obtém os descontos por grupo e condição de pagamento que foram alterados entre as datas informadas.

Request Parameters:

```

grupoProduto: string; // Filtro pelo código do grupo de produto.
dataInicial: string; // Data inicial em formato ISO 8601.
dataFinal: string; // Data final em formato ISO 8601.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Descontos por grupo e condição de pagamento alterados retornados com sucesso.

Content: application/json | Array<ProdutoDescontoGrupoIntegracaoRetornoDto>

```

{
  codigoFilial: integer; // Código da filial.
  condicaoPagamento: CondicaoPagamentoDto;
  dataFim: string; // Data de fim da campanha.
  dataInicio: string; // Data de início da campanha.
  diasSemana: Array<string>; // Dias da semana em que o desconto é aplicável. Caso não possua
    nenhum dia específico, deverá ser retornado todos os dias da semana, de DOMINGO a
    SÁBADO.
  grupoProduto: integer; // Código do grupo de produto.
  nomeCampanha: string; // Nome da campanha de desconto.
  percentualDesconto: number; // Percentual de desconto.
  percentualDescontoMax: number; // Percentual máximo de desconto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por grupo e condição de pagamento alterados.

GET /integracao/produto/desconto/grupo/obter-todos-v1

Obtém todos os descontos por grupo e condição de pagamento.

Obtém todos os descontos por grupo e condição de pagamento cadastrados no sistema.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos por grupo e condição de pagamento retornados com sucesso.

Content: **application/json** | [Array<ProdutoDescontoGrupoIntegracaoRetornoDto>](#)

```
{
  codigoFilial: integer; // Código da filial.
  condicaoPagamento: CondicaoPagamentoDto;
  dataFim: string; // Data de fim da campanha.
  dataInicio: string; // Data de início da campanha.
  diasSemana: Array<string>; // Dias da semana em que o desconto é aplicável. Caso não possua
    nenhum dia específico, deverá ser retornado todos os dias da semana, de DOMINGO a
    SÁBADO.
  grupoProduto: integer; // Código do grupo de produto.
  nomeCampanha: string; // Nome da campanha de desconto.
  percentualDesconto: number; // Percentual de desconto.
  percentualDescontoMax: number; // Percentual máximo de desconto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por grupo e condição de pagamento.

GET /integracao/produto/desconto/grupo/obter-v1

Obtém descontos por grupo e condição de pagamento.

Obtém os descontos por grupo e condição de pagamento cadastrados no sistema.

Request Parameters:

grupoProduto: **string**; // Filtro pelo código do grupo de produto.
primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos por grupo e condição de pagamento retornados com sucesso.

Content: `application/json` | `Array<ProdutoDescontoGrupoIntegracaoRetornoDto>`

```
{
  codigoFilial: integer; // Código da filial.
  condicaoPagamento: CondicaoPagamentoDto;
  dataFim: string; // Data de fim da campanha.
  dataInicio: string; // Data de início da campanha.
  diasSemana: Array<string>; // Dias da semana em que o desconto é aplicável. Caso não possua
    nenhum dia específico, deverá ser retornado todos os dias da semana, de DOMINGO a
    SÁBADO.
  grupoProduto: integer; // Código do grupo de produto.
  nomeCampanha: string; // Nome da campanha de desconto.
  percentualDesconto: number; // Percentual de desconto.
  percentualDescontoMax: number; // Percentual máximo de desconto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por grupo e condição de pagamento.

GET `/integracao/produto/desconto/melhor/obter-alterados-v1`

Obtém melhores descontos alterados.

Obtém os melhores descontos cadastrados no sistema que foram alterados dentro do intervalo de datas fornecido.

Request Parameters:

removerRestricaoEstoque: `boolean`; // Remove a filtragem de somente produtos com estoque positivo.
 dataInicial: `string`; // Data inicial em formato ISO 8601.
 dataFinal: `string`; // Data final em formato ISO 8601.
 primeiroRegistro: `integer`; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: `integer`; // Quantidade máxima de registros a serem retornados.

Response 200:

Melhores descontos alterados retornados com sucesso.

Content: `application/json` | `Array<DescontoMelhorIntegracaoDto>`

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoCondicaoPagamento: integer; // Código da condição de pagamento que oferece o melhor
    desconto.
  codigoLaboratorio: integer; // Código do laboratório.
  codigoProduto: integer; // Código do produto.
  nomeCondicaoPagamento: string; // Nome da condição de pagamento que oferece o melhor
    desconto.
}
```

```

nomeLaboratorio: string; // Nome do laboratório.
nomeProduto: string; // Nome do produto.
percentualDesconto: number; // Melhor percentual de desconto encontrado para o produto.
quantidadeEstoque: integer; // Quantidade em estoque.
valorPromocao: number; // Valor de promoção com base no melhor percentual de desconto
encontrado para o produto.
valorVenda: number; // Valor de venda cadastrado no produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos melhores descontos alterados.

GET /integracao/produto/desconto/melhor/obter-todos-v1

Obtém todos os melhores descontos.

Obtém todos os melhores descontos cadastrados no sistema.

Request Parameters:

```

removerRestricaoEstoque: boolean; // Remove a filtragem de somente produtos com estoque
positivo.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Melhores descontos retornados com sucesso.

Content: application/json | Array<DescontoMelhorIntegracaoDto>

```

{
  codigoBarras: integer; // Código de barras do produto.
  codigoCondicaoPagamento: integer; // Código da condição de pagamento que oferece o melhor
  desconto.
  codigoLaboratorio: integer; // Código do laboratório.
  codigoProduto: integer; // Código do produto.
  nomeCondicaoPagamento: string; // Nome da condição de pagamento que oferece o melhor
  desconto.
  nomeLaboratorio: string; // Nome do laboratório.
  nomeProduto: string; // Nome do produto.
  percentualDesconto: number; // Melhor percentual de desconto encontrado para o produto.
  quantidadeEstoque: integer; // Quantidade em estoque.
  valorPromocao: number; // Valor de promoção com base no melhor percentual de desconto
  encontrado para o produto.
  valorVenda: number; // Valor de venda cadastrado no produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos melhores descontos.

GET /integracao/produto/desconto/melhor/obter-v1

Obtém melhores descontos.

Obtém os melhores descontos cadastrados no sistema com base nos parâmetros fornecidos.

Request Parameters:

codigoBarras: **string**; // Código de barras do produto.
 codigoProduto: **string**; // Código do produto.
 considerarTodasCondicoesPagamento: **boolean**; // Indica se todas as condições de pagamento devem ser consideradas.
 removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Melhores descontos retornados com sucesso.

Content: **application/json** | [Array<DescontoMelhorIntegracaoDto>](#)

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoCondicaoPagamento: integer; // Código da condição de pagamento que oferece o melhor desconto.
  codigoLaboratorio: integer; // Código do laboratório.
  codigoProduto: integer; // Código do produto.
  nomeCondicaoPagamento: string; // Nome da condição de pagamento que oferece o melhor desconto.
  nomeLaboratorio: string; // Nome do laboratório.
  nomeProduto: string; // Nome do produto.
  percentualDesconto: number; // Melhor percentual de desconto encontrado para o produto.
  quantidadeEstoque: integer; // Quantidade em estoque.
  valorPromocao: number; // Valor de promoção com base no melhor percentual de desconto encontrado para o produto.
  valorVenda: number; // Valor de venda cadastrado no produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos melhores descontos.

GET /integracao/produto/desconto/parceiro/obter-alterados-v1

Obtém descontos por parceiro e produto alterados no período.

Retorna descontos/promoções de parceiro e produto que foram alterados no período informado. Caso seja utilizado o token do central, serão retornados os descontos de todas as lojas. Caso seja utilizado o token de uma filial, retorna somente os descontos da filial em questão.

Request Parameters:

codigoProduto: **string**; // Código do produto para filtro.
 codigoBarras: **string**; // Código de barras do produto para filtro.
 nomeProduto: **string**; // Nome ou parte do nome do produto para filtro.
 removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 dataInicial: **string**; // Data/hora inicial da última alteração em formato ISO 8601.
 dataFinal: **string**; // Data/hora final da última alteração em formato ISO 8601.
 primeiroRegistro: **integer**; // Índice do primeiro registro (offset) para paginação. Inicia em 0.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros por página.

Response 200:

Descontos por parceiro e produto alterados retornados com sucesso.

Content: **application/json** | **Array<DescontoParceiroIntegracaoDto>**

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoFilial: integer; // Código da filial.
  codigoParceiro: integer; // Código do parceiro.
  codigoProduto: integer; // Código do produto.
  dataFim: string; // Data final de vigência da promoção.
  dataInicio: string; // Data inicial de vigência da promoção.
  diasSemana: Array<string>; // Lista de dias da semana em que o desconto é vigente. Caso vazia, todos os dias são vigentes.
  nomeParceiro: string; // Nome do parceiro.
  nomeProduto: string; // Nome do produto.
  percentualDesconto: number; // Percentual de desconto cadastrado.
  valorPromocao: number; // Valor de promoção cadastrado.
  valorVenda: number; // Valor de venda do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos por parceiro e produto alterados.

GET /integracao/produto/desconto/parceiro/obter-todos-v1

Obtém todos os descontos por parceiro e produto.

Retorna todos os descontos/promoções cadastrados em parceiro e produto. Caso seja utilizado o token do central, serão retornados os descontos de todas as lojas. Caso seja utilizado o token de uma filial, retorna somente os descontos da filial em questão.

Request Parameters:

removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro (offset) para paginação. Inicia em 0.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros por página.

Response 200:

Descontos por parceiro e produto retornados com sucesso.

Content: **application/json** | **Array<DescontoParceiroIntegracaoDto>**

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoFilial: integer; // Código da filial.
  codigoParceiro: integer; // Código do parceiro.
  codigoProduto: integer; // Código do produto.
  dataFim: string; // Data final de vigência da promoção.
  dataInicio: string; // Data inicial de vigência da promoção.
  diasSemana: Array<string>; // Lista de dias da semana em que o desconto é vigente. Caso vazia, todos os dias são vigentes.
  nomeParceiro: string; // Nome do parceiro.
  nomeProduto: string; // Nome do produto.
  percentualDesconto: number; // Percentual de desconto cadastrado.
  valorPromocao: number; // Valor de promoção cadastrado.
  valorVenda: number; // Valor de venda do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos por parceiro e produto.

GET **/integracao/produto/desconto/parceiro/obter-v1**

Obtém descontos por parceiro e produto com filtros.

Retorna descontos/promoções cadastrados em parceiro e produto, permitindo filtros por produto, código de barras, nome, data de vigência e somente produtos ativos. Caso seja utilizado o token do central, serão retornados os descontos de todas as lojas. Caso seja utilizado o token de uma filial, retorna somente os descontos da filial em questão.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro (offset) para paginação. Inicia em 0.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros por página.
 dataReferencia: **string**; // Data de vigência em formato ISO 8601. Se não informada, utiliza a data atual.
 codigoProduto: **integer**; // Código do produto para filtro.
 codigoBarras: **integer**; // Código de barras do produto para filtro.
 nomeProduto: **string**; // Nome ou parte do nome do produto para filtro.
 somenteAtivos: **boolean**; // Filtrar somente produtos ativos.
 removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque

positivo.

Response 200:

Descontos por parceiro e produto retornados com sucesso.

Content: `application/json` | [Array<DescontoParceiroIntegracaoDto>](#)

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoFilial: integer; // Código da filial.
  codigoParceiro: integer; // Código do parceiro.
  codigoProduto: integer; // Código do produto.
  dataFim: string; // Data final de vigência da promoção.
  dataInicio: string; // Data inicial de vigência da promoção.
  diasSemana: Array<string>; // Lista de dias da semana em que o desconto é vigente. Caso
    vazia, todos os dias são vigentes.
  nomeParceiro: string; // Nome do parceiro.
  nomeProduto: string; // Nome do produto.
  percentualDesconto: number; // Percentual de desconto cadastrado.
  valorPromocao: number; // Valor de promoção cadastrado.
  valorVenda: number; // Valor de venda do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos por parceiro e produto.

GET `/integracao/produto/desconto/progressivo/obter-alterados-v1`

Obtém descontos progressivos alterados.

Obtém os descontos progressivos cadastrados no sistema que foram alterados dentro do intervalo de datas fornecido.

Request Parameters:

removerRestricaoEstoque: `boolean`; // Remove a filtragem de somente produtos com estoque positivo.
 dataInicial: `string`; // Data inicial em formato ISO 8601.
 dataFinal: `string`; // Data final em formato ISO 8601.
 primeiroRegistro: `integer`; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: `integer`; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos progressivos alterados retornados com sucesso.

Content: `application/json` | [Array<DescontoProgressivoIntegracaoDto>](#)

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoLaboratorio: integer; // Código do laboratório.
  codigoProduto: integer; // Código do produto.
}
```

```

condicaoPagamento: integer;
dataFim: string; // Data final da campanha.
dataInicio: string; // Data de início da campanha.
nomeLaboratorio: string; // Nome do laboratório.
nomeProduto: string; // Nome do produto.
origemDesconto: string;
percentualDesconto: number; // Percentual de desconto para quantidades vendidas maior ou
    igual a quantidadeIens.
quantidadeEstoque: integer; // Quantidade em estoque.
quantidadeIens: integer; // Quantidade de itens para obter o desconto progressivo.
valorPromocao: number; // Preço de venda para quantidades vendidas maior ou igual a
    quantidadeIens.
valorVenda: number; // Preço de venda cadastrado no produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos progressivos alterados.

GET /integracao/produto/desconto/progressivo/obter-alterados-v2

Obtém os descontos progressivos alterados.

Obtém os descontos progressivos que tiveram alterações no período informado, retornando apenas os registros modificados.

Request Parameters:

```

removerRestricaoEstoque: boolean; // Remove a filtragem de somente produtos com estoque
    positivo.
dataInicial: string; // Data/hora inicial da última alteração a ser considerada, em formato ISO 8601.
dataFinal: string; // Data/hora final da última alteração a ser considerada, em formato ISO 8601.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Descontos progressivos alterados retornados com sucesso.

Content: application/json | Array<DescontoProgressivoRetornoIntegracaoDto>

```

{
    codigoBarras: integer;
    codigoLaboratorio: integer;
    codigoProduto: integer;
    condicoesPagamento: Array<string>; // Lista de condições de pagamento aplicadas no desconto.
    dataFim: string;
    dataInicio: string;
    nomeLaboratorio: string;
    nomeProduto: string;
    origemDesconto: string; // Origem do desconto aplicado.
    percentualDesconto: number;
    quantidadeEstoque: integer;
}

```

```

    quantidadeItens: integer;
    valorPromocao: number;
    valorVenda: number;
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos progressivos alterados.

GET /integracao/produto/desconto/progressivo/obter-todos-v1

Obtém todos os descontos progressivos.

Obtém todos os descontos progressivos cadastrados no sistema.

Request Parameters:

removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos progressivos retornados com sucesso.

Content: **application/json** | [Array<DescontoProgressivoIntegracaoDto>](#)

```

{
  codigoBarras: integer; // Código de barras do produto.
  codigoLaboratorio: integer; // Código do laboratório.
  codigoProduto: integer; // Código do produto.
  condicaoPagamento: integer;
  dataFim: string; // Data final da campanha.
  dataInicio: string; // Data de início da campanha.
  nomeLaboratorio: string; // Nome do laboratório.
  nomeProduto: string; // Nome do produto.
  origemDesconto: string;
  percentualDesconto: number; // Percentual de desconto para quantidades vendidas maior ou
    igual a quantidadeItens.
  quantidadeEstoque: integer; // Quantidade em estoque.
  quantidadeItens: integer; // Quantidade de itens para obter o desconto progressivo.
  valorPromocao: number; // Preço de venda para quantidades vendidas maior ou igual a
    quantidadeItens.
  valorVenda: number; // Preço de venda cadastrado no produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos progressivos.

GET /integracao/produto/desconto/progressivo/obter-todos-v2

Obtém todos os descontos progressivos.

Obtém todos os descontos progressivos cadastrados no sistema.

Request Parameters:

removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos progressivos retornados com sucesso.

Content: **application/json** | **Array<DescontoProgressivoRetornoIntegracaoDto>**

```
{
  codigoBarras: integer;
  codigoLaboratorio: integer;
  codigoProduto: integer;
  condicoesPagamento: Array<string>; // Lista de condições de pagamento aplicadas no desconto.
  dataFim: string;
  dataInicio: string;
  nomeLaboratorio: string;
  nomeProduto: string;
  origemDesconto: string; // Origem do desconto aplicado.
  percentualDesconto: number;
  quantidadeEstoque: integer;
  quantidadeItens: integer;
  valorPromocao: number;
  valorVenda: number;
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos progressivos.

GET /integracao/produto/desconto/progressivo/obter-v1

Obtém descontos progressivos.

Obtém os descontos progressivos cadastrados no sistema com base nos parâmetros fornecidos.

Request Parameters:

codigoBarras: **string**; // Código de barras do produto.
 codigoProduto: **string**; // Código do produto.
 considerarTodasCondicoesPagamento: **boolean**; // Indica se todas as condições de pagamento devem ser consideradas.
 removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos progressivos retornados com sucesso.

Content: **application/json** | **Array<DescontoProgressivoIntegracaoDto>**

```

{
  codigoBarras: integer; // Código de barras do produto.
  codigoLaboratorio: integer; // Código do laboratório.
  codigoProduto: integer; // Código do produto.
  condicaoPagamento: integer;
  dataFim: string; // Data final da campanha.
  dataInicio: string; // Data de início da campanha.
  nomeLaboratorio: string; // Nome do laboratório.
  nomeProduto: string; // Nome do produto.
  origemDesconto: string;
  percentualDesconto: number; // Percentual de desconto para quantidades vendidas maior ou igual a quantidadeItens.
  quantidadeEstoque: integer; // Quantidade em estoque.
  quantidadeItens: integer; // Quantidade de itens para obter o desconto progressivo.
  valorPromocao: number; // Preço de venda para quantidades vendidas maior ou igual a quantidadeItens.
  valorVenda: number; // Preço de venda cadastrado no produto.
}
  
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos progressivos.

GET **/integracao/produto/desconto/progressivo/obter-v2**

Obtém os descontos progressivos.

Obtém os descontos progressivos cadastrados no sistema com base nos parâmetros fornecidos.

Request Parameters:

codigoBarras: **string**; // Código de barras do produto.
 codigoProduto: **string**; // Código do produto.
 considerarTodasCondicoesPagamento: **boolean**; // Indica se todas as condições de pagamento devem ser consideradas.
 dataReferencia: **string**; // Informar data em formato ISO 8601.

removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos progressivos retornados com sucesso.

Content: **application/json** | **Array<DescontoProgressivoRetornoIntegracaoDto>**

```
{
  codigoBarras: integer;
  codigoLaboratorio: integer;
  codigoProduto: integer;
  condicoesPagamento: Array<string>; // Lista de condições de pagamento aplicadas no desconto.
  dataFim: string;
  dataInicio: string;
  nomeLaboratorio: string;
  nomeProduto: string;
  origemDesconto: string; // Origem do desconto aplicado.
  percentualDesconto: number;
  quantidadeEstoque: integer;
  quantidadeItens: integer;
  valorPromocao: number;
  valorVenda: number;
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta dos descontos progressivos.

GET /integracao/produto/desconto/vigencia/obter-alterados-v1

Obtém descontos de produto por vigência alterados.

Retorna descontos de produto por vigência alterados no período especificado.

Request Parameters:

dataInicial: **string**; // Data inicial em formato ISO 8601.
 dataFinal: **string**; // Data final em formato ISO 8601.
 removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos de produto por vigência alterados retornados com sucesso.

Content: **application/json** | **Array<DescontoProdutoVigenciaIntegracaoDto>**

```
{
```

```

codigoDesconto: integer; // Código interno do desconto.
codigoFilial: integer; // Código da filial.
codigoProduto: integer; // Código do produto (SKU).
dataFim: string; // Data de término da vigência.
dataInicio: string; // Data de início da vigência.
percentualDesconto: number; // Percentual de desconto aplicado (0 a 100).
valorPromocao: number; // Valor promocional final do produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos de produto por vigência alterados.

GET /integracao/produto/desconto/vigencia/obter-movimentados-v1

Obtém descontos de produto por vigência movimentados.

Retorna descontos de produto por vigência movimentados no período especificado.

Request Parameters:

```

dataInicial: string; // Data inicial em formato ISO 8601.
dataFinal: string; // Data final em formato ISO 8601.
removerRestricaoEstoque: boolean; // Remove a filtragem de somente produtos com estoque
positivo.
dataInicial: integer; // Data inicial em formato ISO 8601.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Descontos de produto por vigência movimentados retornados com sucesso.

Content: application/json | Array<DescontoProdutoVigenciaIntegracaoDto>

```

{
  codigoDesconto: integer; // Código interno do desconto.
  codigoFilial: integer; // Código da filial.
  codigoProduto: integer; // Código do produto (SKU).
  dataFim: string; // Data de término da vigência.
  dataInicio: string; // Data de início da vigência.
  percentualDesconto: number; // Percentual de desconto aplicado (0 a 100).
  valorPromocao: number; // Valor promocional final do produto.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos de produto por vigência movimentados.

GET /integracao/produto/desconto/vigencia/obter-todos-v1

Obtém todos os descontos de produto por vigência.

Retorna todos os descontos de produto por vigência com paginação.

Request Parameters:

removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos de produto por vigência retornados com sucesso.

Content: **application/json** | **Array<DescontoProdutoVigenciaIntegracaoDto>**

```
{
  codigoDesconto: integer; // Código interno do desconto.
  codigoFilial: integer; // Código da filial.
  codigoProduto: integer; // Código do produto (SKU).
  dataFim: string; // Data de término da vigência.
  dataInicio: string; // Data de início da vigência.
  percentualDesconto: number; // Percentual de desconto aplicado (0 a 100).
  valorPromocao: number; // Valor promocional final do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos de produto por vigência.

GET /integracao/produto/grupo/obter-alterados-v1

Obtém grupos de produtos alterados.

Retorna grupos de produtos alterados no período especificado.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
 dataInicial: **string**; // Data inicial em formato ISO 8601.
 dataFinal: **string**; // Data final em formato ISO 8601.

Response 200:

Grupos de produtos alterados retornados com sucesso.

Content: `application/json` | `Array<GrupoProdutoIntegracaoDto>`

```
{
  ativo: boolean; // Indica se o grupo está ativo.
  codigo: integer; // Código do grupo de produtos.
  nome: string; // Nome do grupo de produtos.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de grupos de produtos alterados.

GET `/integracao/produto/grupo/obter-movimentados-v1`

Obtém grupos de produtos movimentados.

Retorna grupos de produtos movimentados no período especificado.

Request Parameters:

primeiroRegistro: `integer`; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: `integer`; // Quantidade máxima de registros a serem retornados.
 dataInicial: `string`; // Data inicial em formato ISO 8601.
 dataFinal: `string`; // Data final em formato ISO 8601.

Response 200:

Grupos de produtos movimentados retornados com sucesso.

Content: `application/json` | `Array<GrupoProdutoIntegracaoDto>`

```
{
  ativo: boolean; // Indica se o grupo está ativo.
  codigo: integer; // Código do grupo de produtos.
  nome: string; // Nome do grupo de produtos.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de grupos de produtos movimentados.

GET `/integracao/produto/grupo/obter-todos-v1`

Obtém todos os grupos de produtos.

Retorna grupos de produtos com paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Grupos de produtos retornados com sucesso.

Content: **application/json** | **Array<GrupoProdutoIntegracaoDto>**

```
{
  ativo: boolean; // Indica se o grupo está ativo.
  codigo: integer; // Código do grupo de produtos.
  nome: string; // Nome do grupo de produtos.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de grupos de produtos.

GET /integracao/produto/identificador/obter-alterados-v1

Lista identificadores de produtos alterados.

Retorna identificadores de produtos alterados no período especificado.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
dataInicial: **string**; // Informar data em formato ISO 8601.
dataFinal: **string**; // Informar data em formato ISO 8601.

Response 200:

Identificadores de produtos alterados retornados com sucesso.

Content: **application/json** | **Array<IdentificadorProdutoIntegracaoDto>**

```
{
  ativo: boolean; // Indica se o identificador esta ativo.
  codigo: integer; // Codigo reduzido do produto.
  identificador: integer; // Identificador do produto (codigo de barras ou codigo reduzido).
  principal: boolean; // Indica se o identificador e principal.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de identificadores de produtos alterados.

GET /integracao/produto/identificador/obter-movimentados-v1

Lista identificadores de produtos movimentados.

Retorna identificadores de produtos movimentados no período especificado.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
dataInicial: **string**; // Informar data em formato ISO 8601.
dataFinal: **string**; // Informar data em formato ISO 8601.

Response 200:

Identificadores de produtos movimentados retornados com sucesso.

Content: **application/json** | Array<IdentificadorProdutoIntegracaoDto>

```
{
  ativo: boolean; // Indica se o identificador esta ativo.
  codigo: integer; // Codigo reduzido do produto.
  identificador: integer; // Identificador do produto (codigo de barras ou codigo reduzido).
  principal: boolean; // Indica se o identificador e principal.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de identificadores de produtos movimentados.

GET /integracao/produto/identificador/obter-todos-v1

Lista identificadores de produtos.

Retorna identificadores de produtos com paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Identificadores de produtos retornados com sucesso.

Content: **application/json** | Array<IdentificadorProdutoIntegracaoDto>

```
{
  ativo: boolean; // Indica se o identificador esta ativo.
  codigo: integer; // Codigo reduzido do produto.
  identificador: integer; // Identificador do produto (codigo de barras ou codigo reduzido).
  principal: boolean; // Indica se o identificador e principal.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de identificadores de produtos.

GET /integracao/produto/obter-alterados-v1

Lista produtos alterados.

Retorna produtos alterados no período especificado com filtros opcionais e paginação.

Request Parameters:

```
integracaoEcommerce: boolean; // Filtra por integração e-commerce.
processaCustoMedio: boolean; // Processa custo médio.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.
dataInicial: string; // Informar data em formato ISO 8601.
dataFinal: string; // Informar data em formato ISO 8601.
```

Response 200:

Produtos alterados retornados com sucesso.

Content: `application/json` | [Array<ProdutoIntegracaoDto>](#)

```
{
  alturaEcommerce: integer; // Altura em milímetros para e-commerce.
  ativo: boolean; // Indica se o produto esta ativo.
  codClaFiscal: string; // Codigo da classificacao fiscal.
  codDepto: integer; // Codigo do departamento.
  codTipo: integer; // Codigo do tipo.
  codigo: integer; // Codigo do produto.
  codigoBarras: integer; // Codigo de barras.
  codigoCategoria: integer; // Codigo da categoria.
  codigoClassificacao: integer; // Codigo da classificacao.
  codigoDepartamento: integer; // Codigo do departamento.
  codigoGrupo: integer; // Codigo do grupo.
  codigoLaboratorio: integer; // Codigo do laboratorio.
  codigoPrincipioAtivo: string; // Codigo do principio ativo.
  corEcommerce: CorEcommerce;
  descricaoEcommerce: string; // Descricao do produto para e-commerce.
  idProduto: string; // Identificador externo do produto.
  integracaoEcommerce: boolean; // Indica integracao com e-commerce.
  larguraEcommerce: integer; // Largura em milímetros para e-commerce.
  nomDepto: string; // Nome do departamento.
  nomTipo: string; // Nome do tipo.
}
```

```

nome: string; // Nome do produto.
nomeCategoria: string; // Nome da categoria.
nomeClassificacao: string; // Nome da classificacao.
nomeDepartamento: string; // Nome do departamento.
nomeEcommerce: string; // Nome do produto para e-commerce.
nomeGrupo: string; // Nome do grupo.
nomeLaboratorio: string; // Nome do laboratorio.
nomePrincipioAtivo: string; // Nome do principio ativo.
observacaoVenda: string; // Observacao de venda.
percentualDesconto: number; // Percentual de desconto.
percentualDescontoMax: number; // Percentual maximo de desconto.
pesoEcommerce: integer; // Peso em gramas para e-commerce.
produtoManipulado: string; // Indica se o produto e manipulado.
profundidadeEcommerce: integer; // Profundidade em milímetros para e-commerce.
qtdLimiteCarrinhoEcommerce: integer; // Limite de itens por carrinho no e-commerce.
quantidadeEstoque: integer; // Quantidade em estoque.
quantidadeEstoqueEcommerce: integer; // Quantidade de estoque para e-commerce.
tags: Array<string>; // Tags do produto.
tamanhoEcommerce: TamanhoEcommerce;
tipo: Tipo;
tipoLista: string; // Tipo de lista de precos.
unidade: string; // Unidade de medida.
valorCusto: number; // Valor de custo.
valorCustoMedio: number; // Valor de custo medio.
valorVenda: number; // Valor de venda.
valorVendaEcommerce: number; // Valor de venda para e-commerce.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de produtos alterados.

GET /integracao/produto/obter-movimentados-v1

Lista produtos movimentados.

Retorna produtos movimentados no período especificado e com paginação.

Request Parameters:

```

primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.
dataInicial: string; // Informar data em formato ISO 8601.
dataFinal: string; // Informar data em formato ISO 8601.

```

Response 200:

Produtos movimentados retornados com sucesso.

Content: application/json | Array<ProdutoIntegracaoDto>

```

{
  alturaEcommerce: integer; // Altura em milímetros para e-commerce.

```

```

ativo: boolean; // Indica se o produto esta ativo.
codClaFiscal: string; // Codigo da classificacao fiscal.
codDepto: integer; // Codigo do departamento.
codTipo: integer; // Codigo do tipo.
codigo: integer; // Codigo do produto.
codigoBarras: integer; // Codigo de barras.
codigoCategoria: integer; // Codigo da categoria.
codigoClassificacao: integer; // Codigo da classificacao.
codigoDepartamento: integer; // Codigo do departamento.
codigoGrupo: integer; // Codigo do grupo.
codigoLaboratorio: integer; // Codigo do laboratorio.
codigoPrincipioAtivo: string; // Codigo do principio ativo.
corEcommerce: CorEcommerce;
descricaoEcommerce: string; // Descricao do produto para e-commerce.
idProduto: string; // Identificador externo do produto.
integracaoEcommerce: boolean; // Indica integracao com e-commerce.
larguraEcommerce: integer; // Largura em milímetros para e-commerce.
nomDepto: string; // Nome do departamento.
nomTipo: string; // Nome do tipo.
nome: string; // Nome do produto.
nomeCategoria: string; // Nome da categoria.
nomeClassificacao: string; // Nome da classificacao.
nomeDepartamento: string; // Nome do departamento.
nomeEcommerce: string; // Nome do produto para e-commerce.
nomeGrupo: string; // Nome do grupo.
nomeLaboratorio: string; // Nome do laboratorio.
nomePrincipioAtivo: string; // Nome do principio ativo.
observacaoVenda: string; // Observacao de venda.
percentualDesconto: number; // Percentual de desconto.
percentualDescontoMax: number; // Percentual maximo de desconto.
pesoEcommerce: integer; // Peso em gramas para e-commerce.
produtoManipulado: string; // Indica se o produto e manipulado.
profundidadeEcommerce: integer; // Profundidade em milímetros para e-commerce.
qtdLimiteCarrinhoEcommerce: integer; // Limite de itens por carrinho no e-commerce.
quantidadeEstoque: integer; // Quantidade em estoque.
quantidadeEstoqueEcommerce: integer; // Quantidade de estoque para e-commerce.
tags: Array<string>; // Tags do produto.
tamanhoEcommerce: TamanhoEcommerce;
tipo: Tipo;
tipoLista: string; // Tipo de lista de precos.
unidade: string; // Unidade de medida.
valorCusto: number; // Valor de custo.
valorCustoMedio: number; // Valor de custo medio.
valorVenda: number; // Valor de venda.
valorVendaEcommerce: number; // Valor de venda para e-commerce.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de produtos movimentados.

GET /integracao/produto/obter-todos-v1

Lista todos os produtos.

Retorna produtos com paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.

quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Produtos retornados com sucesso.

Content: **application/json** | **Array<ProdutoIntegracaoDto>**

```
{
  alturaEcommerce: integer; // Altura em milímetros para e-commerce.
  ativo: boolean; // Indica se o produto esta ativo.
  codClaFiscal: string; // Codigo da classificacao fiscal.
  codDepto: integer; // Codigo do departamento.
  codTipo: integer; // Codigo do tipo.
  codigo: integer; // Codigo do produto.
  codigoBarras: integer; // Codigo de barras.
  codigoCategoria: integer; // Codigo da categoria.
  codigoClassificacao: integer; // Codigo da classificacao.
  codigoDepartamento: integer; // Codigo do departamento.
  codigoGrupo: integer; // Codigo do grupo.
  codigoLaboratorio: integer; // Codigo do laboratorio.
  codigoPrincipioAtivo: string; // Codigo do principio ativo.
  corEcommerce: CorEcommerce;
  descricaoEcommerce: string; // Descricao do produto para e-commerce.
  idProduto: string; // Identificador externo do produto.
  integracaoEcommerce: boolean; // Indica integracao com e-commerce.
  larguraEcommerce: integer; // Largura em milímetros para e-commerce.
  nomDepto: string; // Nome do departamento.
  nomTipo: string; // Nome do tipo.
  nome: string; // Nome do produto.
  nomeCategoria: string; // Nome da categoria.
  nomeClassificacao: string; // Nome da classificacao.
  nomeDepartamento: string; // Nome do departamento.
  nomeEcommerce: string; // Nome do produto para e-commerce.
  nomeGrupo: string; // Nome do grupo.
  nomeLaboratorio: string; // Nome do laboratorio.
  nomePrincipioAtivo: string; // Nome do principio ativo.
  observacaoVenda: string; // Observacao de venda.
  percentualDesconto: number; // Percentual de desconto.
  percentualDescontoMax: number; // Percentual maximo de desconto.
  pesoEcommerce: integer; // Peso em gramas para e-commerce.
  produtoManipulado: string; // Indica se o produto e manipulado.
  profundidadeEcommerce: integer; // Profundidade em milímetros para e-commerce.
  qtdLimiteCarrinhoEcommerce: integer; // Limite de itens por carrinho no e-commerce.
  quantidadeEstoque: integer; // Quantidade em estoque.
  quantidadeEstoqueEcommerce: integer; // Quantidade de estoque para e-commerce.
  tags: Array<string>; // Tags do produto.
  tamanhoEcommerce: TamanhoEcommerce;
  tipo: Tipo;
  tipoLista: string; // Tipo de lista de precos.
  unidade: string; // Unidade de medida.
  valorCusto: number; // Valor de custo.
  valorCustoMedio: number; // Valor de custo medio.
}
```

```

    valorVenda: number; // Valor de venda.
    valorVendaEcommerce: number; // Valor de venda para e-commerce.
  }

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de produtos.

GET /integracao/produto/obter-v1

Lista produtos.

Consulta a lista de produtos com filtros opcionais e paginação.

Request Parameters:

```

codigo: Array<integer>; // Código reduzido do produto. Aceita múltiplos valores.
tipo.codigo: Array<integer>; // Código do tipo do produto. Aceita múltiplos valores.
codigoBarras: Array<integer>; // Código de barras. Aceita múltiplos valores.
tipoLista: Array<string>; // Tipo de lista.
observacaoVenda: Array<string>; // Observação de venda.
ativo: boolean; // Filtra por ativo.
integracaoEcommerce: boolean; // Filtra por integração e-commerce.
processaCustoMedio: boolean; // Processa custo médio.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Produtos retornados com sucesso.

Content: application/json | Array<ProdutoIntegracaoDto>

```

{
  alturaEcommerce: integer; // Altura em milímetros para e-commerce.
  ativo: boolean; // Indica se o produto esta ativo.
  codClaFiscal: string; // Codigo da classificacao fiscal.
  codDepto: integer; // Codigo do departamento.
  codTipo: integer; // Codigo do tipo.
  codigo: integer; // Codigo do produto.
  codigoBarras: integer; // Codigo de barras.
  codigoCategoria: integer; // Codigo da categoria.
  codigoClassificacao: integer; // Codigo da classificacao.
  codigoDepartamento: integer; // Codigo do departamento.
  codigoGrupo: integer; // Codigo do grupo.
  codigoLaboratorio: integer; // Codigo do laboratorio.
  codigoPrincipioAtivo: string; // Codigo do principio ativo.
  corEcommerce: CorEcommerce;
  descricaoEcommerce: string; // Descricao do produto para e-commerce.
  idProduto: string; // Identificador externo do produto.
  integracaoEcommerce: boolean; // Indica integracao com e-commerce.
  larguraEcommerce: integer; // Largura em milímetros para e-commerce.
  nomDepto: string; // Nome do departamento.

```

```

nomTipo: string; // Nome do tipo.
nome: string; // Nome do produto.
nomeCategoria: string; // Nome da categoria.
nomeClassificacao: string; // Nome da classificacao.
nomeDepartamento: string; // Nome do departamento.
nomeEcommerce: string; // Nome do produto para e-commerce.
nomeGrupo: string; // Nome do grupo.
nomeLaboratorio: string; // Nome do laboratorio.
nomePrincipioAtivo: string; // Nome do principio ativo.
observacaoVenda: string; // Observacao de venda.
percentualDesconto: number; // Percentual de desconto.
percentualDescontoMax: number; // Percentual maximo de desconto.
pesoEcommerce: integer; // Peso em gramas para e-commerce.
produtoManipulado: string; // Indica se o produto e manipulado.
profundidadeEcommerce: integer; // Profundidade em milímetros para e-commerce.
qtdLimiteCarrinhoEcommerce: integer; // Limite de itens por carrinho no e-commerce.
quantidadeEstoque: integer; // Quantidade em estoque.
quantidadeEstoqueEcommerce: integer; // Quantidade de estoque para e-commerce.
tags: Array<string>; // Tags do produto.
tamanhoEcommerce: TamanhoEcommerce;
tipo: Tipo;
tipoLista: string; // Tipo de lista de precos.
unidade: string; // Unidade de medida.
valorCusto: number; // Valor de custo.
valorCustoMedio: number; // Valor de custo medio.
valorVenda: number; // Valor de venda.
valorVendaEcommerce: number; // Valor de venda para e-commerce.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de produtos.

GET /integracao/produto/precificacao/obter-alterados-v1

Lista descontos por precificação alterados.

Retorna descontos por precificação alterados no período especificado.

Request Parameters:

```

dataInicial: string; // Data inicial em formato ISO 8601.
dataFinal: string; // Data final em formato ISO 8601.
removerRestricaoEstoque: boolean; // Remove a filtragem de somente produtos com estoque
positivo.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Descontos alterados retornados com sucesso.

Content: `application/json` | `Array<DescontoPrecificacaoIntegracaoDto>`

```
{
  codigoFilial: integer; // Código da filial.
  codigoProduto: integer; // Código do produto.
  valorVenda: number; // Valor de venda do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por precificação alterados.

GET `/integracao/produto/precificacao/obter-movimentados-v1`

Lista descontos por precificação movimentados.

Retorna descontos por precificação movimentados no período especificado.

Request Parameters:

`dataInicial`: `string`; // Data inicial em formato ISO 8601.
`dataFinal`: `string`; // Data final em formato ISO 8601.
`removerRestricaoEstoque`: `boolean`; // Remove a filtragem de somente produtos com estoque positivo.
`primeiroRegistro`: `integer`; // Índice do primeiro registro a ser retornado.
`quantidadeRegistros`: `integer`; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos movimentados retornados com sucesso.

Content: `application/json` | `Array<DescontoPrecificacaoIntegracaoDto>`

```
{
  codigoFilial: integer; // Código da filial.
  codigoProduto: integer; // Código do produto.
  valorVenda: number; // Valor de venda do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por precificação movimentados.

GET `/integracao/produto/precificacao/obter-todos-v1`

Lista todos os descontos por precificação.

Retorna todos os descontos por precificação com paginação.

Request Parameters:

removerRestricaoEstoque: **boolean**; // Remove a filtragem de somente produtos com estoque positivo.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Descontos retornados com sucesso.

Content: **application/json** | [Array<DescontoPrecificacaoIntegracaoDto>](#)

```
{
  codigoFilial: integer; // Código da filial.
  codigoProduto: integer; // Código do produto.
  valorVenda: number; // Valor de venda do produto.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de descontos por precificação.

GET /integracao/produto/principio-ativo/obter-alterados-v1

Lista princípios ativos de produtos alterados.

Retorna princípios ativos de produtos alterados no período especificado.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
 dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.

Response 200:

Princípios ativos de produtos alterados retornados com sucesso.

Content: **application/json** | [Array<PrincipioAtivoProdutoIntegracaoDto>](#)

```
{
  ativo: boolean; // Indica se o principio ativo esta ativo.
  codigo: string; // Codigo do principio ativo.
  nome: string; // Nome do principio ativo.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de princípios ativos de produtos alterados.

GET /integracao/produto/principio-ativo/obter-movimentados-v1

Lista princípios ativos de produtos movimentados.

Retorna princípios ativos de produtos movimentados no período especificado.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
 dataInicial: **string**; // Informar data em formato ISO 8601.
 dataFinal: **string**; // Informar data em formato ISO 8601.

Response 200:

Princípios ativos de produtos movimentados retornados com sucesso.

Content: **application/json** | Array<PrincipioAtivoProdutoIntegracaoDto>

```
{
  ativo: boolean; // Indica se o principio ativo esta ativo.
  codigo: string; // Codigo do principio ativo.
  nome: string; // Nome do principio ativo.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de princípios ativos de produtos movimentados.

GET /integracao/produto/principio-ativo/obter-todos-v1

Lista princípios ativos de produtos.

Retorna princípios ativos de produtos com paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Princípios ativos de produtos retornados com sucesso.

Content: **application/json** | Array<PrincipioAtivoProdutoIntegracaoDto>

```
{
  ativo: boolean; // Indica se o principio ativo esta ativo.
  codigo: string; // Codigo do principio ativo.
  nome: string; // Nome do principio ativo.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de princípios ativos de produtos.

GET **/integracao/smartped/obter-todos-v1**

Lista todas as movimentações do Smartped.

Retorna a lista de movimentações do Smartped com paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Movimentações retornadas com sucesso.

Content: **application/json** | **Array<RetornoConsultaSmartpedDTO>**

```
{
  cnpjLoja: string; // CNPJ da loja, somente números.
  codigoLoja: integer; // Código da loja.
  datCotacao: string; // Data da cotação no formato ISO 8601.
  razaoSocial: string; // Razão social da loja.
  tipCotacao: string; // Tipo da cotação.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de movimentações do Smartped.

GET **/integracao/smartped/obter-v1**

Obtém movimentações do Smartped.

Consulta movimentações do Smartped com filtros opcionais e paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Movimentações retornadas com sucesso.

Content: **application/json** | **Array<RetornoConsultaSmartpedDTO>**

```
{
  cnpjLoja: string; // CNPJ da loja, somente números.
  codigoLoja: integer; // Código da loja.
  datCotacao: string; // Data da cotação no formato ISO 8601.
  razaoSocial: string; // Razão social da loja.
  tipCotacao: string; // Tipo da cotação.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de movimentações do Smartped.

GET **/integracao/venda/cancelamento/obter-alterados-v1**

Lista cancelamentos de venda alterados.

Retorna lista de cancelamentos de venda alterados no período especificado.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
 dataInicial: **string**; // Data inicial em formato ISO 8601.
 dataFinal: **string**; // Data final em formato ISO 8601.

Response 200:

Cancelamentos de venda alterados retornados com sucesso.

Content: **application/json** | **Array<VendaIntegracaoDto>**

```
{
  codEcommerce: integer; // Código da venda no e-commerce.
  codFilial: integer; // Código da filial.
  codParceiro: integer; // Código do parceiro (quando aplicável).
  codigoCliente: integer; // Código do cliente.
  codigoVendedor: integer; // Código do vendedor.
  condicaoPagamento: CondicaoPagamento;
  dadosEntrega: DadosEntregaDto;
  dataEmissao: string; // Data de emissão em formato ISO 8601.
  entrega: boolean; // Indica se possui entrega vinculada.
  horaEmissao: string; // Hora de emissão em formato ISO 8601.
  itens: Array<VendaItemIntegracaoDto>; // Itens da venda.
}
```

```

modeloVenda: string; // Modelo da venda.
numeroCupomFiscal: integer; // Número do cupom fiscal.
numeroNota: integer; // Número da venda/nota.
numeroNotaFiscal: integer; // Número da nota fiscal (NF).
numeroNotaOrigem: integer; // Número da nota de origem (para devoluções/cancelamentos).
pagamentoNaEntrega: boolean; // Indica se o pagamento é na entrega.
serNotaFiscal: string; // Série da nota fiscal.
tipoCancelamento: string; // Tipo de cancelamento: D=Devolução, E=Estorno.
vendaEcommerce: string; // Indicador de venda no e-commerce.
vendafood: boolean; // Indica se a venda é do iFood.
vlrTroco: number; // Valor do troco da venda.
xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de cancelamentos de venda alterados.

GET /integracao/venda/cancelamento/obter-todos-v1

Lista todos os cancelamentos de venda.

Retorna a lista de cancelamentos de venda com paginação.

Request Parameters:

primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

Response 200:

Cancelamentos de venda retornados com sucesso.

Content: application/json | Array<VendaIntegracaoDto>

```

{
  codEcommerce: integer; // Código da venda no e-commerce.
  codFilial: integer; // Código da filial.
  codParceiro: integer; // Código do parceiro (quando aplicável).
  codigoCliente: integer; // Código do cliente.
  codigoVendedor: integer; // Código do vendedor.
  condicaoPagamento: CondicaoPagamento;
  dadosEntrega: DadosEntregaDto;
  dataEmissao: string; // Data de emissão em formato ISO 8601.
  entrega: boolean; // Indica se possui entrega vinculada.
  horaEmissao: string; // Hora de emissão em formato ISO 8601.
  itens: Array<VendaItemIntegracaoDto>; // Itens da venda.
  modeloVenda: string; // Modelo da venda.
  numeroCupomFiscal: integer; // Número do cupom fiscal.
  numeroNota: integer; // Número da venda/nota.
  numeroNotaFiscal: integer; // Número da nota fiscal (NF).
  numeroNotaOrigem: integer; // Número da nota de origem (para devoluções/cancelamentos).
  pagamentoNaEntrega: boolean; // Indica se o pagamento é na entrega.
}

```

```

serNotaFiscal: string; // Série da nota fiscal.
tipoCancelamento: string; // Tipo de cancelamento: D=Devolução, E=Estorno.
vendaEcommerce: string; // Indicador de venda no e-commerce.
vendalfood: boolean; // Indica se a venda é do iFood.
vlrTroco: number; // Valor do troco da venda.
xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de cancelamentos de venda.

GET /integracao/venda/cancelamento/obter-v1

Lista cancelamentos de venda com filtros.

Consulta a lista de cancelamentos de venda com filtros opcionais e paginação.

Request Parameters:

```

numeroNota: Array<integer>; // Número da nota fiscal. Aceita múltiplos valores.
codigoCliente: Array<integer>; // Código do cliente. Aceita múltiplos valores.
dataEmissaoInicial: string; // Data de emissão inicial em formato ISO 8601.
dataEmissaoFinal: string; // Data de emissão final em formato ISO 8601.
xmlNfe: boolean; // Quando true, retorna informações com referência ao XML da NF-e.
primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.

```

Response 200:

Cancelamentos de venda retornados com sucesso.

Content: application/json | Array<VendaIntegracaoDto>

```

{
  codEcommerce: integer; // Código da venda no e-commerce.
  codFilial: integer; // Código da filial.
  codParceiro: integer; // Código do parceiro (quando aplicável).
  codigoCliente: integer; // Código do cliente.
  codigoVendedor: integer; // Código do vendedor.
  condicaoPagamento: CondicaoPagamento;
  dadosEntrega: DadosEntregaDto;
  dataEmissao: string; // Data de emissão em formato ISO 8601.
  entrega: boolean; // Indica se possui entrega vinculada.
  horaEmissao: string; // Hora de emissão em formato ISO 8601.
  itens: Array<VendaItemIntegracaoDto>; // Itens da venda.
  modeloVenda: string; // Modelo da venda.
  numeroCupomFiscal: integer; // Número do cupom fiscal.
  numeroNota: integer; // Número da venda/nota.
  numeroNotaFiscal: integer; // Número da nota fiscal (NF).
  numeroNotaOrigem: integer; // Número da nota de origem (para devoluções/cancelamentos).
  pagamentoNaEntrega: boolean; // Indica se o pagamento é na entrega.
  serNotaFiscal: string; // Série da nota fiscal.
}

```

```

tipoCancelamento: string; // Tipo de cancelamento: D=Devolução, E=Estorno.
vendaEcommerce: string; // Indicador de venda no e-commerce.
vendalfood: boolean; // Indica se a venda é do iFood.
vlrTroco: number; // Valor do troco da venda.
xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de cancelamentos de venda.

POST /integracao/venda/ecommerce/atualizar-status-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 /integracao/venda/ecommerce/consultar-venda-v1

Consultar venda e-commerce.

Consulta vendas do e-commerce por parâmetros.

Request Parameters:

numeroNota: **string**; // Número da venda gerada no PDV.
 numeroPedido: **string**; // Número do pedido no e-commerce vinculado à venda.
 numerosPedidos: **string**; // Números dos pedidos 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: string; // Status do pedido no e-commerce.
  xmlNfe: string; // XML da nota fiscal eletrônica.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendas.

POST /integracao/venda/ecommerce/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.
  enviarMaquinaPOS?: boolean; // Indica se deve enviar para a máquina POS.
  numeroPedido: string; // Número do pedido no e-commerce que será vinculado à venda.
  pagamento?: PagamentoIntegracaoDto;
  pagamentoMultiplo?: PagamentoMultiploIntegracaoDto;
  produtos?: Array<VendaItemEcommerceIntegracaoDto>; // Lista de produtos da venda.
  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.

GET [/integracao/venda/obter-alterados-v1](#)

Lista vendas alteradas.

Retorna lista de vendas alteradas no período especificado.

Request Parameters:

primeiroRegistro: `integer`; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: `integer`; // Quantidade máxima de registros a serem retornados.
 dataInicial: `string`; // Data inicial em formato ISO 8601.
 dataFinal: `string`; // Data final em formato ISO 8601.

Response 200:

Vendas alteradas retornadas com sucesso.

Content: `application/json` | [Array<VendaIntegracaoDto>](#)

```
{
  codEcommerce: integer; // Código da venda no e-commerce.
  codFilial: integer; // Código da filial.
  codParceiro: integer; // Código do parceiro (quando aplicável).
  codigoCliente: integer; // Código do cliente.
  codigoVendedor: integer; // Código do vendedor.
  condicaoPagamento: CondicaoPagamento;
  dadosEntrega: DadosEntregaDto;
  dataEmissao: string; // Data de emissão em formato ISO 8601.
  entrega: boolean; // Indica se possui entrega vinculada.
  horaEmissao: string; // Hora de emissão em formato ISO 8601.
  itens: Array<VendaItemIntegracaoDto>; // Itens da venda.
  modeloVenda: string; // Modelo da venda.
  numeroCupomFiscal: integer; // Número do cupom fiscal.
  numeroNota: integer; // Número da venda/nota.
  numeroNotaFiscal: integer; // Número da nota fiscal (NF).
  numeroNotaOrigem: integer; // Número da nota de origem (para devoluções/cancelamentos).
```

```

pagamentoNaEntrega: boolean; // Indica se o pagamento é na entrega.
serNotaFiscal: string; // Série da nota fiscal.
tipoCancelamento: string; // Tipo de cancelamento: D=Devolução, E=Estorno.
vendaEcommerce: string; // Indicador de venda no e-commerce.
vendalfood: boolean; // Indica se a venda é do iFood.
vlrTroco: number; // Valor do troco da venda.
xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendas alteradas.

GET /integracao/venda/obter-atendimentos-diario-vendedor-v1

Lista atendimentos diários por vendedor.

Retorna a lista de atendimentos diários por vendedor no período especificado.

Request Parameters:

```

primeiroRegistro: integer; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: integer; // Quantidade máxima de registros a serem retornados.
dataInicial: string; // Data inicial em formato ISO 8601.
dataFinal: string; // Data final em formato ISO 8601.

```

Response 200:

Atendimentos retornados com sucesso.

Content: **application/json** | [Array<VendasVendedorIntegracaoDto>](#)

```

{
  codigoVendedor: integer; // Código do vendedor.
  dataEmissao: string; // Data de emissão em formato ISO 8601.
  quantidadeAtendimentos: integer; // Quantidade de atendimentos realizados.
  quantidadeItens: integer; // Quantidade de itens vendidos.
}

```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de atendimentos por vendedor.

GET /integracao/venda/obter-todos-v1

Lista todas as vendas.

Retorna a lista de vendas com paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.

quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Vendas retornadas com sucesso.

Content: **application/json** | **Array<VendaIntegracaoDto>**

```
{
  codEcommerce: integer; // Código da venda no e-commerce.
  codFilial: integer; // Código da filial.
  codParceiro: integer; // Código do parceiro (quando aplicável).
  codigoCliente: integer; // Código do cliente.
  codigoVendedor: integer; // Código do vendedor.
  condicaoPagamento: CondicaoPagamento;
  dadosEntrega: DadosEntregaDto;
  dataEmissao: string; // Data de emissão em formato ISO 8601.
  entrega: boolean; // Indica se possui entrega vinculada.
  horaEmissao: string; // Hora de emissão em formato ISO 8601.
  itens: Array<VendaItemIntegracaoDto>; // Itens da venda.
  modeloVenda: string; // Modelo da venda.
  numeroCupomFiscal: integer; // Número do cupom fiscal.
  numeroNota: integer; // Número da venda/nota.
  numeroNotaFiscal: integer; // Número da nota fiscal (NF).
  numeroNotaOrigem: integer; // Número da nota de origem (para devoluções/cancelamentos).
  pagamentoNaEntrega: boolean; // Indica se o pagamento é na entrega.
  serNotaFiscal: string; // Série da nota fiscal.
  tipoCancelamento: string; // Tipo de cancelamento: D=Devolução, E=Estorno.
  vendaEcommerce: string; // Indicador de venda no e-commerce.
  vendaFood: boolean; // Indica se a venda é do iFood.
  vlrTroco: number; // Valor do troco da venda.
  xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendas.

GET /integracao/venda/obter-v1

Lista vendas com filtros.

Consulta a lista de vendas com filtros opcionais e paginação.

Request Parameters:

numeroNota: **Array<integer>**; // Número da nota fiscal. Aceita múltiplos valores.
 codigoCliente: **Array<integer>**; // Código do cliente. Aceita múltiplos valores.
 dataEmissaoInicial: **string**; // Data de emissão inicial em formato ISO 8601.
 dataEmissaoFinal: **string**; // Data de emissão final em formato ISO 8601.
 xmlNfe: **boolean**; // Quando true, retorna informações com referência ao XML da NF-e.
 entrega: **boolean**; // Quando true, filtra vendas de tele entrega.
 primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Vendas retornadas com sucesso.

Content: **application/json** | **Array<VendaIntegracaoDto>**

```

{
  codEcommerce: integer; // Código da venda no e-commerce.
  codFilial: integer; // Código da filial.
  codParceiro: integer; // Código do parceiro (quando aplicável).
  codigoCliente: integer; // Código do cliente.
  codigoVendedor: integer; // Código do vendedor.
  condicaoPagamento: CondicaoPagamento;
  dadosEntrega: DadosEntregaDto;
  dataEmissao: string; // Data de emissão em formato ISO 8601.
  entrega: boolean; // Indica se possui entrega vinculada.
  horaEmissao: string; // Hora de emissão em formato ISO 8601.
  itens: Array<VendaItemIntegracaoDto>; // Itens da venda.
  modeloVenda: string; // Modelo da venda.
  numeroCupomFiscal: integer; // Número do cupom fiscal.
  numeroNota: integer; // Número da venda/nota.
  numeroNotaFiscal: integer; // Número da nota fiscal (NF).
  numeroNotaOrigem: integer; // Número da nota de origem (para devoluções/cancelamentos).
  pagamentoNaEntrega: boolean; // Indica se o pagamento é na entrega.
  serNotaFiscal: string; // Série da nota fiscal.
  tipoCancelamento: string; // Tipo de cancelamento: D=Devolução, E=Estorno.
  vendaEcommerce: string; // Indicador de venda no e-commerce.
  vendalfood: boolean; // Indica se a venda é do iFood.
  vlrTroco: number; // Valor do troco da venda.
  xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}
  
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendas.

GET /integracao/vendedor/obter-alterados-v1

Lista vendedores alterados.

Retorna lista de vendedores alterados no período especificado.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
 dataInicial: **string**; // Data inicial em formato ISO 8601.
 dataFinal: **string**; // Data final em formato ISO 8601.

Response 200:

Vendedores alterados retornados com sucesso.

Content: **application/json** | **Array<VendedorIntegracaoDto>**

```
{
  ativo: boolean; // Indica se o vendedor está ativo.
  cep: string; // CEP do endereço do vendedor.
  codigo: integer; // Código do vendedor.
  email: string; // E-mail do vendedor.
  nome: string; // Nome completo do vendedor.
  numeroCpf: string; // CPF do vendedor, somente números.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendedores alterados.

GET **/integracao/vendedor/obter-movimentados-v1**

Lista vendedores movimentados.

Retorna lista de vendedores movimentados no período especificado.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
 quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.
 dataInicial: **string**; // Data inicial em formato ISO 8601.
 dataFinal: **string**; // Data final em formato ISO 8601.

Response 200:

Vendedores movimentados retornados com sucesso.

Content: **application/json** | **Array<VendedorIntegracaoDto>**

```
{
  ativo: boolean; // Indica se o vendedor está ativo.
  cep: string; // CEP do endereço do vendedor.
  codigo: integer; // Código do vendedor.
  email: string; // E-mail do vendedor.
  nome: string; // Nome completo do vendedor.
  numeroCpf: string; // CPF do vendedor, somente números.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendedores movimentados.

GET /integracao/vendedor/obter-todos-v1

Lista todos os vendedores.

Retorna a lista de vendedores com paginação.

Request Parameters:

primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.
quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Vendedores retornados com sucesso.

Content: **application/json** | **Array<VendedorIntegracaoDto>**

```
{
  ativo: boolean; // Indica se o vendedor está ativo.
  cep: string; // CEP do endereço do vendedor.
  codigo: integer; // Código do vendedor.
  email: string; // E-mail do vendedor.
  nome: string; // Nome completo do vendedor.
  numeroCpf: string; // CPF do vendedor, somente números.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendedores.

GET /integracao/vendedor/obter-v1

Lista vendedores com filtros.

Consulta a lista de vendedores com filtros opcionais e paginação.

Request Parameters:

codigo: **Array<integer>**; // Código do vendedor. Aceita múltiplos valores.
primeiroRegistro: **integer**; // Índice do primeiro registro a ser retornado.

quantidadeRegistros: **integer**; // Quantidade máxima de registros a serem retornados.

Response 200:

Vendedores retornados com sucesso.

Content: **application/json** | [Array<VendedorIntegracaoDto>](#)

```
{
  ativo: boolean; // Indica se o vendedor está ativo.
  cep: string; // CEP do endereço do vendedor.
  codigo: integer; // Código do vendedor.
  email: string; // E-mail do vendedor.
  nome: string; // Nome completo do vendedor.
  numeroCpf: string; // CPF do vendedor, somente números.
}
```

Response 400:

Requisição inválida.

Response 401:

Não autorizado.

Response 500:

Erro na consulta de vendedores.

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.
}
```

BandeiraDto

```
{
  codigoBandeira: integer; // Código da bandeira.
  nomeBandeira: string; // Nome da bandeira.
}
```

CartaoIntegracaoRetornoDto

```
{
  administradora: FornecedorDto;
  ativo: boolean; // Define se o cartão está ativo ou inativo.
  carteiraDigital: string; // Carteira digital do cartão.
  codigoCartao: integer; // Código do cartão.
  descontoCartao: number;
  descontoMaximoCartao: number;
  gerenciadorTEF: BandeiraDto;
  modalidadeVenda: string; // Modalidade de venda do cartão.
  nomeCartao: string; // Nome do cartão.
  qtdMaxParcelas: integer; // Quantidade máxima de parcelas permitidas.
  taxaAdministrativa: number; // Taxa administrativa do cartão.
  taxaAdministrativaParcela: Array<ListaTaxaParcelaIntegracaoDto>; // Lista de taxas administrativas por parcela.
  vencimentoParcelas: Array<string>; // Dias de vencimento por parcela.
}
```

CartaoVendaIntegracaoDTO

```
{
  codigo: integer; // Código do cartão (Trier).
  numeroAutorizacao?: integer; // Número de autorização da transação no cartão.
  pagamentoRealizado: boolean; // Indica se o pagamento foi realizado.
  qtdParcela?: integer; // Número de parcelas no cartão.
  valor: number; // Valor pago no cartão.
}
```

CategoriaIntegracaoDto

```
{
  codigo: integer; // Código da categoria.
  nome: string; // Nome da categoria.
}
```

ClienteEcommerceIntegracaoDto

```
{
  ativo?: boolean; // Indica se o cliente está ativo.
  bairro?: string; // Bairro do cliente.
  celular?: string; // Número de celular do cliente.
  cep?: string; // CEP do cliente.
  codigo?: integer; // Código do cliente, se não informado, será gerado automaticamente pelo PDV.
  codigoCidade?: string; // Código da cidade do cliente.
  dataNascimento?: string; // Data de nascimento do cliente.
  email?: string; // E-mail do cliente.
  empresaConvenio?: EmpresaConvenio;
  estado?: string; // Estado do cliente.
  fone?: string; // Número de telefone do cliente.
  grupo?: Grupo;
  logradouro?: string; // Logradouro do cliente.
  nome: string; // Nome do cliente.
  numeroCpfCnpj: string; // Número do CPF ou CNPJ do cliente.
  numeroEndereco?: string; // Número do endereço do cliente.
  numeroRGIE?: string; // Número do RG ou IE do cliente.
  sexo?: string; // Sexo do cliente.
}
```

ClienteIntegracaoDto

```
{
  ativo?: boolean; // Indica se o cliente está ativo.
  bairro?: string; // Bairro do cliente.
  cep?: string; // CEP do cliente.
  codigo?: integer; // Código do cliente, se não informado, será gerado automaticamente pelo PDV.
  codigoCidade?: string; // Código da cidade do cliente.
  email?: string; // E-mail do cliente.
  empresaConvenio?: EmpresaConvenio;
  estado?: string; // Estado do cliente.
  fone?: string; // Número de telefone do cliente.
  grupo?: Grupo;
  logradouro?: string; // Logradouro do cliente.
  nome: string; // Nome do cliente.
  numeroCpfCnpj: string; // Número do CPF ou CNPJ do cliente.
  numeroEndereco?: string; // Número do endereço do cliente.
}
```

CompraIntegracaoDto

```
{
  chaveAcessoNfe: string; // Chave de acesso NFE.
  codigoFornecedor: integer; // Código do fornecedor.
  datEmissao: string;
  dataEntrada: string; // Data de entrada.
  itens: Array<ComprasiItemIntegracaoDto>; // Itens da compra.
  numeroNotaFiscal: integer; // Número da nota fiscal.
  quantidadeItens: integer; // Quantidade de itens da nota.
  valorTotalNota: number; // Valor total da nota.
  valorTotalProdutos: number; // Valor total dos produtos.
  xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}
```

ComprasItemIntegracaoDto

```
{
  codigoProduto: integer; // Código do produto.
  fatorCompra: integer; // Fator de compra do produto.
  quantidadeProdutos: integer; // Quantidade do produto.
  valorCusto: number; // Valor de custo do produto.
  valorST: number; // Valor ST do produto.
  valorUnitario: number; // Valor unitário do produto.
  valorUnitarioLiquido: number; // Valor unitário líquido do produto (valor unitário - desconto).
}
```

CondicaoPagamento

```
{
  codigo: integer; // Código da condição de pagamento.
  nome: string; // Nome da condição de pagamento.
}
```

CondicaoPagamentoDto

```
{
  codigo: integer; // Código da condição de pagamento.
  nome: string; // Nome da condição de pagamento.
}
```

CorEcommerce

```
{
  codigo: integer; // Código da cor.
  nome: string; // Nome da cor.
}
```

DadosEntregaDto

```
{
  bairro: string;
  cep: string;
  cidade: string;
  complemento: string;
  endereco: string;
  fone: string;
  identificacao: string;
  itinerario: string;
  nome: string;
  numero: string;
  uf: string;
}
```

DepartamentoIntegracaoDto

```
{
  ativo: boolean; // Define se o departamento está ativo ou inativo.
  codigo: integer; // Código do departamento.
  nome: string; // Nome do departamento.
}
```

DescontoEmpresaGrupoProdutoIntegracaoDto

```
{
  codigoEmpresa: integer; // Código da empresa (convênio).
  codigoFilial: integer; // Código da filial.
  codigoGrupoProduto: integer; // Código do grupo de produtos.
  condicoesPagamento: Array<CondicaoPagamentoDto>; // Lista de condições de pagamento
    vinculadas ao desconto.
  dataFim: string; // Data final de vigência da promoção.
  dataInicio: string; // Data inicial de vigência da promoção.
  nomeEmpresa: string; // Nome da empresa (convênio).
  nomeGrupoProduto: string; // Nome do grupo de produtos.
  percentualDesconto: number; // Percentual de desconto cadastrado para o grupo de produto.
  percentualDescontoMax: number; // Percentual de desconto máximo para o grupo de produto.
}
```

DescontoMelhorIntegracaoDto

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoCondicaoPagamento: integer; // Código da condição de pagamento que oferece o melhor
    desconto.
  codigoLaboratorio: integer; // Código do laboratório.
  codigoProduto: integer; // Código do produto.
  nomeCondicaoPagamento: string; // Nome da condição de pagamento que oferece o melhor desconto.
  nomeLaboratorio: string; // Nome do laboratório.
  nomeProduto: string; // Nome do produto.
  percentualDesconto: number; // Melhor percentual de desconto encontrado para o produto.
  quantidadeEstoque: integer; // Quantidade em estoque.
  valorPromocao: number; // Valor de promoção com base no melhor percentual de desconto
    encontrado para o produto.
  valorVenda: number; // Valor de venda cadastrado no produto.
}
```

DescontoParceiroIntegracaoDto

```
{
  codigoBarras: integer; // Código de barras do produto.
  codigoFilial: integer; // Código da filial.
  codigoParceiro: integer; // Código do parceiro.
  codigoProduto: integer; // Código do produto.
  dataFim: string; // Data final de vigência da promoção.
  dataInicio: string; // Data inicial de vigência da promoção.
  diasSemana: Array<string>; // Lista de dias da semana em que o desconto é vigente. Caso vazia,
    todos os dias são vigentes.
  nomeParceiro: string; // Nome do parceiro.
  nomeProduto: string; // Nome do produto.
  percentualDesconto: number; // Percentual de desconto cadastrado.
  valorPromocao: number; // Valor de promoção cadastrado.
}
```

```

    valorVenda: number; // Valor de venda do produto.
}

```

DescontoPrecificacaoIntegracaoDto

```

{
  codigoFilial: integer; // Código da filial.
  codigoProduto: integer; // Código do produto.
  valorVenda: number; // Valor de venda do produto.
}

```

DescontoProdutoVigencialIntegracaoDto

```

{
  codigoDesconto: integer; // Código interno do desconto.
  codigoFilial: integer; // Código da filial.
  codigoProduto: integer; // Código do produto (SKU).
  dataFim: string; // Data de término da vigência.
  dataInicio: string; // Data de início da vigência.
  percentualDesconto: number; // Percentual de desconto aplicado (0 a 100).
  valorPromocao: number; // Valor promocional final do produto.
}

```

DescontoProgressivoIntegracaoDto

```

{
  codigoBarras: integer; // Código de barras do produto.
  codigoLaboratorio: integer; // Código do laboratório.
  codigoProduto: integer; // Código do produto.
  condicaoPagamento: integer;
  dataFim: string; // Data final da campanha.
  dataInicio: string; // Data de início da campanha.
  nomeLaboratorio: string; // Nome do laboratório.
  nomeProduto: string; // Nome do produto.
  origemDesconto: string;
  percentualDesconto: number; // Percentual de desconto para quantidades vendidas maior ou igual a
    quantidadeItens.
  quantidadeEstoque: integer; // Quantidade em estoque.
  quantidadeItens: integer; // Quantidade de itens para obter o desconto progressivo.
  valorPromocao: number; // Preço de venda para quantidades vendidas maior ou igual a
    quantidadeItens.
  valorVenda: number; // Preço de venda cadastrado no produto.
}

```

DescontoProgressivoRetornoIntegracaoDto

```

{
  codigoBarras: integer;
  codigoLaboratorio: integer;
  codigoProduto: integer;
  condicoesPagamento: Array<string>; // Lista de condições de pagamento aplicadas no desconto.
  dataFim: string;
  dataInicio: string;
}

```

```

nomeLaboratorio: string;
nomeProduto: string;
origemDesconto: string; // Origem do desconto aplicado.
percentualDesconto: number;
quantidadeEstoque: integer;
quantidadeItens: integer;
valorPromocao: number;
valorVenda: number;
}

```

DinheiroIntegracaoDTO

```

{
  pagamentoRealizado: boolean; // Indica se o pagamento foi realizado.
  valor: number; // Valor pago.
  valorTroco?: number; // Valor do troco.
}

```

EmpresaConvenio

```

{
  codigo: integer;
  nome: string;
}

```

EnderecoEntregaIntegracaoDto

```

{
  bairro: string; // Bairro.
  cep?: string; // CEP.
  cidade: string; // Cidade.
  complemento?: string; // Complemento.
  estado?: string; // Estado.
  itinerario1?: string; // Itinerário 01.
  itinerario2?: string; // Itinerário 02.
  itinerario3?: string; // Itinerário 03.
  itinerario4?: string; // Itinerário 04.
  logradouro: string; // Logradouro.
  numero?: string; // Número.
  referencia?: string; // Referência.
}

```

EstoqueIntegracaoDto

```

{
  alterCurva: string;
  codigoProduto: integer; // Código reduzido de um produto.
  dataUltimaEntrada: string; // Data da última entrada do produto.
  quantidadeEstoque: integer; // Quantidade em estoque.
  valorCustoMedio: number; // Custo médio do produto.
  valorUltimaEntrada: number; // Valor da última entrada do produto.
}

```

FinanceiroClassificacaoFinanceiraIntegracaoDto

```
{
  ativo: boolean; // Ativo ou inativo.
  centroCusto: string; // Código do centro de custo que pertence.
  creditoDebito: string; // D = Débito / C = Crédito.
  fixoVariavel: string; // F = Fixo / V = Variável.
  nomeDescricao: string; // Nome do centro de custo que pertence.
  tipoClassificacao: string; // D = Despesa / R = Receita.
}
```

FinanceiroContaIntegracaoDto

```
{
  dataAtualizacao: string; // Data de atualização.
  digito: string; // Dígito da conta.
  digitoAgencia: string; // Dígito da agência.
  nome: string; // Nome da conta.
  numero: string; // Número da conta.
  numeroAgencia: string; // Número da agência.
  saldo: number; // Saldo da conta.
}
```

FinanceiroResultadoExercicioIntegracaoDto

```
{
  centroCusto: string; // Centro de custo que ele pertence.
  codigoDepartamento: integer; // Código departamento.
  data: string; // Data de lançamento do movimento.
  valorBruto: number; // Valor bruto.
}
```

FinanceiroResultadoFinanceiroIntegracaoDto

```
{
  baixadoAberto: string; // Movimento baixado ou aberto.
  centroCusto: string; // Centro de custo que pertence.
  codigoDepartamento: integer; // Código do departamento.
  data: string; // Data de baixa do movimento.
  valorBruto: number; // Valor bruto do movimento.
}
```

FornecedorDto

```
{
  codigoFornecedor: integer; // Código do fornecedor.
  nomeFornecedor: string; // Nome do fornecedor.
}
```

FornecedorIntegracaoDto

```
{
  ativo: boolean; // Ativo ou inativo.
  codigo: integer; // Código do fornecedor.
  email: string; // E-mail do fornecedor.
  nomeCidade: string; // Cidade do fornecedor.
  nomeFantasia: string; // Nome fantasia do fornecedor.
  numeroCnpj: string; // CNPJ do fornecedor.
  razaoSocial: string; // Razão social do fornecedor.
}
```

Grupo

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

GrupoProdutoIntegracaoDto

```
{
  ativo: boolean; // Indica se o grupo está ativo.
  codigo: integer; // Código do grupo de produtos.
  nome: string; // Nome do grupo de produtos.
}
```

IdentificadorProdutoIntegracaoDto

```
{
  ativo: boolean; // Indica se o identificador esta ativo.
  codigo: integer; // Codigo reduzido do produto.
  identificador: integer; // Identificador do produto (codigo de barras ou codigo reduzido).
  principal: boolean; // Indica se o identificador e principal.
}
```

ItemPedidoResumidoIntegracaoDto

```
{
  cnpjFornecedorUltimaCompra: string; // CNPJ do fornecedor da última compra do produto.
  codigoBarras: integer; // Código de barras do produto.
  codigoComprador: integer; // Código do usuário que efetuou o pedido no sistema.
  codigoFornecedor: integer; // Código do fornecedor do pedido.
  codigoFornecedorUltimaCompra: integer; // Código do fornecedor da última compra do produto.
  codigoProduto: integer; // Código do produto.
  dataEmissao: string; // Data de emissão do pedido no formato ISO 8601.
  fatorCompra: integer; // Fator de compra do produto.
  fatorCompraUltimaCompra: integer; // Fator de compra da última compra do produto.
  horaEmissao: string; // Hora de emissão do pedido no formato ISO 8601.
  nomeComprador: string; // Nome do comprador.
  nomeFornecedor: string; // Nome do fornecedor do pedido.
  nomeFornecedorUltimaCompra: string; // Nome do fornecedor da última compra do produto.
  nomeProduto: string; // Nome do produto.
  numeroPedido: integer; // Número do pedido.
  numeroSequencialItem: integer; // Número sequencial do item no pedido.
}
```

```

quantidadeEstoque: integer; // Quantidade do produto em estoque.
quantidadeProdutos: integer; // Quantidade de produtos no item do pedido.
transmitido: boolean; // Indica se o pedido foi transmitido.
valorUnitario: number; // Valor unitário do produto no item do pedido.
valorUnitarioUltimaCompra: number; // Valor unitário da última compra do produto.
}

```

LaboratorioIntegracaoDto

```

{
  ativo: boolean; // Define se o laboratório está ativo ou inativo.
  cnpj: string; // CNPJ do laboratório.
  codigo: integer; // Código do laboratório.
  nome: string; // Nome do laboratório.
}

```

ListaTaxaParcelaIntegracaoDto

```

{
  parcelaFinal: string; // Parcela final.
  parcelaInicial: string; // Parcela inicial.
  taxaAdministrativa: string; // Taxa administrativa da parcela.
}

```

PagamentoIntegracaoDto

```

{
  dataVencimento?: string; // Data de vencimento do pagamento.
  numeroAutorizacao?: string; // Número de autorização da transação.
  pagamentoRealizado: boolean; // Indica se o pagamento foi realizado.
  valorDinheiro?: number; // Valor em dinheiro.
  valorParcela: number; // Valor da parcela.
  valorTroco?: number; // Valor do troco.
}

```

PagamentoMultiplIntegracaoDto

```

{
  cartao: Array<CartaoVendaIntegracaoDTO>; // Lista de pagamentos em cartão.
  dinheiro: DinheiroIntegracaoDTO;
  pix: PixIntegracaoDTO;
}

```

ParcelaCartaoEstornoIntegracaoDto

```

{
  cnpjLoja: string; // Número do CNPJ da filial.
  codigoLoja: integer; // Código da filial.
  estornos: Array<ParcelaCartaoEstornoTransacaoIntegracaoDTO>; // Lista de estornos de parcelas de cartão.
  inscricaoEstadualLoja: string; // Inscrição estadual da filial.
}

```

```

    razaoSocial: string; // Razão social da filial.
}

```

ParcelaCartaoEstornoTransacaoIntegracaoDto

```

{
    dataEmissaoDevolucao: string; // Data de emissão da nota de devolução.
    dataEmissaoOrigem: string; // Data de emissão da nota original da venda.
    numeroNotaDevolucao: integer; // Número da nota de devolução.
    numeroNotaOrigem: integer; // Número da nota original da venda.
    totalNotaDevolucao: number; // Total da nota de devolução (apenas considerado total do cartão).
    totalNotaOrigem: number; // Total da nota original da venda (apenas considerado total do cartão).
}

```

ParcelaCartaoIntegracaoDto

```

{
    cnpjLoja: string; // Número do CNPJ da filial.
    codigoLoja: integer; // Código da filial.
    inscricaoEstadualLoja: string; // Inscrição estadual da filial.
    razaoSocial: string; // Razão social da filial.
    transacoes: Array<ParcelaCartaoTransacaoIntegracaoDto>; // Lista de parcelas de cartão.
}

```

ParcelaCartaoTransacaoIntegracaoDto

```

{
    administradoraCartao: string; // Administradora do cartão.
    codigoAutorizacao: string; // Código de autorização.
    codigoBandeira: string; // Código nacional da bandeira (opcional).
    codigoCartao: integer; // Código do cartão.
    codigoEstabelecimentoAdquirente: integer; // Código do estabelecimento na adquirente (opcional).
    dataEmissao: string; // Data de emissão.
    dataPagamento: string; // Data do pagamento.
    dataVencimento: string; // Data do vencimento.
    documentoFiscal: integer; // Número do documento fiscal.
    formaCaptura: string; // Forma de captura (TEF / POS).
    idParcela: integer; // Identificação da parcela.
    idTransacao: string; // Identificação da transação (Venda + ID / Recebimento + ID).
    idTransacaoPIX: string; // Identificação da transação Pix (Skytef).
    modalidadeVenda: string; // Modalidade da venda (CREDITO / DEBITO).
    nomeAdquirente: string; // Nome da adquirente (opcional).
    nomeCartao: string; // Nome do cartão.
    nsuAdministradora: string; // NSU da administradora.
    nsuOrigem: integer; // NSU de origem.
    numeroParcela: integer; // Número da parcela.
    prazoVenda: string; // Prazo da venda (A VISTA / PARCELADO).
    totalParcelas: number; // Quantidade total de parcelas.
    valorParcela: number; // Valor líquido a receber da parcela.
    valorTaxas: number; // Valor total das taxas da transação.
    valorTotal: number; // Valor total pago em cartão.
}

```

PedidoResumidoIntegracaoDto

```
{
  codigoComprador: integer; // Código do usuário que efetuou o pedido no sistema.
  dataEmissao: string; // Data de emissão do pedido no formato ISO 8601.
  horaEmissao: string; // Hora de emissão do pedido no formato ISO 8601.
  nomeComprador: string; // Nome do comprador.
  numeroPedido: integer; // Número do pedido.
}
```

PixIntegracaoDTO

```
{
  codigo: integer; // Código do tipo de pagamento PIX (Cartão).
  idTransacaoPIX?: string; // ID da transação PIX.
  numeroAutorizacao?: integer; // Número da autorização.
  pagamentoRealizado: boolean; // Indica se o pagamento foi realizado.
  valor: number; // Valor pago no PIX.
}
```

PrincipioAtivoProdutoIntegracaoDto

```
{
  ativo: boolean; // Indica se o principio ativo esta ativo.
  codigo: string; // Codigo do principio ativo.
  nome: string; // Nome do principio ativo.
}
```

ProdutoCondicaoPagamentoDto

```
{
  codigoBarras: string;
  codigoProduto: integer;
  nomeProduto: string;
}
```

ProdutoDescontoCondicaoPagamentoIntegracaoRetornoDto

```
{
  codigoFilial: integer;
  condicoesPagamento: Array<CondicaoPagamentoDto>;
  dataFim: string;
  dataInicio: string;
  diasSemana: Array<string>;
  nomeCampanha: string;
  percentualDesconto: number;
  percentualDescontoMax: number;
  produto: ProdutoCondicaoPagamentoDto;
  valorPromocao: number;
  valorVenda: number;
}
```

ProdutoDescontoEncarteIntegracaoRetorno

```
{
  codigoEncarte: integer; // Código identificador do encarte.
  dataFim: string; // Data de término da vigência do encarte.
  dataInicio: string; // Data de início da vigência do encarte.
  diasSemana: Array<string>; // Dias da semana em que o encarte é válido.
  nomeCampanha: string; // Nome da campanha/encarte.
  produtosEncarte: ProdutoEncarteDto;
}
```

ProdutoDescontoGrupoIntegracaoRetornoDto

```
{
  codigoFilial: integer; // Código da filial.
  condicaoPagamento: CondicaoPagamentoDto;
  dataFim: string; // Data de fim da campanha.
  dataInicio: string; // Data de início da campanha.
  diasSemana: Array<string>; // Dias da semana em que o desconto é aplicável. Caso não possua
    nenhum dia específico, deverá ser retornado todos os dias da semana, de DOMINGO a SÁBADO.
  grupoProduto: integer; // Código do grupo de produto.
  nomeCampanha: string; // Nome da campanha de desconto.
  percentualDesconto: number; // Percentual de desconto.
  percentualDescontoMax: number; // Percentual máximo de desconto.
}
```

ProdutoEncarteDto

```
{
  codigoProduto: integer; // Código interno do produto.
  nomeProduto: string; // Nome do produto.
  quantidadeProduto: integer; // Quantidade de unidades do produto necessária para que o desconto
    seja aplicado.
  valorPromocao: number; // Valor promocional aplicado ao produto.
}
```

ProdutoIntegracaoDto

```
{
  alturaEcommerce: integer; // Altura em milímetros para e-commerce.
  ativo: boolean; // Indica se o produto esta ativo.
  codClaFiscal: string; // Codigo da classificacao fiscal.
  codDepto: integer; // Codigo do departamento.
  codTipo: integer; // Codigo do tipo.
  codigo: integer; // Codigo do produto.
  codigoBarras: integer; // Codigo de barras.
  codigoCategoria: integer; // Codigo da categoria.
  codigoClassificacao: integer; // Codigo da classificacao.
  codigoDepartamento: integer; // Codigo do departamento.
  codigoGrupo: integer; // Codigo do grupo.
  codigoLaboratorio: integer; // Codigo do laboratorio.
  codigoPrincipioAtivo: string; // Codigo do principio ativo.
  corEcommerce: CorEcommerce;
  descricaoEcommerce: string; // Descricao do produto para e-commerce.
}
```

```

idProduto: string; // Identificador externo do produto.
integracaoEcommerce: boolean; // Indica integracao com e-commerce.
larguraEcommerce: integer; // Largura em milímetros para e-commerce.
nomDepto: string; // Nome do departamento.
nomTipo: string; // Nome do tipo.
nome: string; // Nome do produto.
nomeCategoria: string; // Nome da categoria.
nomeClassificacao: string; // Nome da classificacao.
nomeDepartamento: string; // Nome do departamento.
nomeEcommerce: string; // Nome do produto para e-commerce.
nomeGrupo: string; // Nome do grupo.
nomeLaboratorio: string; // Nome do laboratorio.
nomePrincipioAtivo: string; // Nome do principio ativo.
observacaoVenda: string; // Observacao de venda.
percentualDesconto: number; // Percentual de desconto.
percentualDescontoMax: number; // Percentual maximo de desconto.
pesoEcommerce: integer; // Peso em gramas para e-commerce.
produtoManipulado: string; // Indica se o produto e manipulado.
profundidadeEcommerce: integer; // Profundidade em milímetros para e-commerce.
qtdLimiteCarrinhoEcommerce: integer; // Limite de itens por carrinho no e-commerce.
quantidadeEstoque: integer; // Quantidade em estoque.
quantidadeEstoqueEcommerce: integer; // Quantidade de estoque para e-commerce.
tags: Array<string>; // Tags do produto.
tamanhoEcommerce: TamanhoEcommerce;
tipo: Tipo;
tipoLista: string; // Tipo de lista de precos.
unidade: string; // Unidade de medida.
valorCusto: number; // Valor de custo.
valorCustoMedio: number; // Valor de custo medio.
valorVenda: number; // Valor de venda.
valorVendaEcommerce: number; // Valor de venda para e-commerce.
}

```

RetornoConsultaSmartpedDTO

```

{
  cnpjLoja: string; // CNPJ da loja, somente números.
  codigoLoja: integer; // Código da loja.
  datCotacao: string; // Data da cotação no formato ISO 8601.
  razaoSocial: string; // Razão social da loja.
  tipCotacao: string; // Tipo da cotação.
}

```

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: string; // Status do pedido no e-commerce.
xmlNfe: string; // XML da nota fiscal eletrônica.
}

```

TamanhoEcommerce

```

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

```

Tipo

```

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

```

VendaEcommerceIntegracaoDto

```

{
  cliente?: ClienteEcommerceIntegracaoDto;
  dataPedido: string; // Data do pedido no e-commerce.
  enderecoEntrega?: EnderecoEntregaIntegracaoDto;
  entrega?: boolean; // Indica se é uma venda Tele-Entrega ou não.
  enviarMaquinaPOS?: boolean; // Indica se deve enviar para a máquina POS.
  numeroPedido: string; // Número do pedido no e-commerce que será vinculado à venda.
  pagamento?: PagamentoIntegracaoDto;
  pagamentoMultiplo?: PagamentoMultiploIntegracaoDto;
  produtos?: Array<VendaItemEcommerceIntegracaoDto>; // Lista de produtos da venda.
  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;
}

```

VendaIntegracaoDto

```

{
  codEcommerce: integer; // Código da venda no e-commerce.
  codFilial: integer; // Código da filial.
}

```

```

codParceiro: integer; // Código do parceiro (quando aplicável).
codigoCliente: integer; // Código do cliente.
codigoVendedor: integer; // Código do vendedor.
condicaoPagamento: CondicaoPagamento;
dadosEntrega: DadosEntregaDto;
dataEmissao: string; // Data de emissão em formato ISO 8601.
entrega: boolean; // Indica se possui entrega vinculada.
horaEmissao: string; // Hora de emissão em formato ISO 8601.
itens: Array<VendaltemIntegracaoDto>; // Itens da venda.
modeloVenda: string; // Modelo da venda.
numeroCupomFiscal: integer; // Número do cupom fiscal.
numeroNota: integer; // Número da venda/nota.
numeroNotaFiscal: integer; // Número da nota fiscal (NF).
numeroNotaOrigem: integer; // Número da nota de origem (para devoluções/cancelamentos).
pagamentoNaEntrega: boolean; // Indica se o pagamento é na entrega.
serNotaFiscal: string; // Série da nota fiscal.
tipoCancelamento: string; // Tipo de cancelamento: D=Devolução, E=Estorno.
vendaEcommerce: string; // Indicador de venda no e-commerce.
vendalfood: boolean; // Indica se a venda é do iFood.
vlrTroco: number; // Valor do troco da venda.
xmlNfe: string; // Conteúdo do XML da NF-e (quando disponível).
}

```

VendaltemEcommerceIntegracaoDto

```

{
  codigoProduto: integer; // Código do produto.
  nomeProduto?: string; // Nome do produto.
  quantidade: integer; // Quantidade do produto.
  valorDesconto?: number; // Valor unitário do desconto aplicado ao produto.
  valorUnitario: number; // Valor unitário do produto.
}

```

VendaltemIntegracaoDto

```

{
  codBarras: integer; // Código de barras do produto.
  codigoMedico: integer; // Código do médico vinculado ao item, quando aplicável.
  codigoProduto: integer; // Código reduzido do produto.
  codigoVendedor: integer; // Código do vendedor associado ao item.
  idVenda: string; // Identificador calculado da venda (hash MD5).
  numSequencial: integer; // Número sequencial do item na venda.
  parceiro: boolean; // Indica se o item é de parceiro.
  prcComissao: number; // Percentual de comissão aplicado ao item.
  prcDesconto: number; // Percentual de desconto aplicado.
  prcDescontoMax: number; // Percentual máximo de desconto permitido.
  quantidadeProdutos: integer; // Quantidade de produtos no item.
  tabelaDesconto: string; // Nome da tabela de desconto aplicada.
  valorTotalBruto: number; // Valor total bruto do item (2 casas decimais).
  valorTotalCusto: number; // Valor total de custo do item (2 casas decimais).
  valorTotalLiquido: number; // Valor total líquido do item (2 casas decimais).
  vendaComDesconto: number; // Valor da venda considerando o desconto (4 casas decimais).
  vlrcustoAquisicao: number; // Custo de aquisição do produto (4 casas decimais).
  vlrcustoProduto: number; // Custo cadastrado do produto.
  vlrDesconto: number; // Valor de desconto em moeda.
  vlrUnitario: number; // Valor unitário do produto.
}

```

```
}
```

VendasVendedorIntegracaoDto

```
{
  codigoVendedor: integer; // Código do vendedor.
  dataEmissao: string; // Data de emissão em formato ISO 8601.
  quantidadeAtendimentos: integer; // Quantidade de atendimentos realizados.
  quantidadeItens: integer; // Quantidade de itens vendidos.
}
```

VendedorEcommerceIntegracaoDto

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

VendedorIntegracaoDto

```
{
  ativo: boolean; // Indica se o vendedor está ativo.
  cep: string; // CEP do endereço do vendedor.
  codigo: integer; // Código do vendedor.
  email: string; // E-mail do vendedor.
  nome: string; // Nome completo do vendedor.
  numeroCpf: string; // CPF do vendedor, somente números.
}
```

WebhookDto

```
{
  ActionBy: string; // Quem realizou a ação.
  CurrentOrderValue: number; // Valor atual do pedido.
  CustomerOrderCode: string; // Código interno do pedido.
  DeliveredTo: string; // Nome do recebedor.
  DeliveryOrderId: integer; // Identificador da ordem de entrega.
  DeliveryOrderPointId: integer; // Identificador do ponto da ordem de entrega.
  Distance: number; // Distância percorrida em quilômetros.
  DriverName: string; // Nome do entregador.
  Lat: string; // Latitude da entrega.
  Lng: string; // Longitude da entrega.
  OldOrderValue: number; // Valor antigo do pedido.
  Status: string; // Enumeração que representa os possíveis status de um pedido no Agilego.
  StoreGroupId: string; // Identificador do grupo da loja.
  StoreGroupName: string; // Nome do grupo da loja.
  StoreId: string; // Identificador da loja.
  SubStatus: string; // Enumeração que representa os substatus de um pedido no AgileGO.
  Type: string; // Identificador do status.
  UpdateAt: string; // Data e hora da atualização do status.
}
```