

MILLENNIUM NETWORK

---

# Millennium ECO 5.44

---

## Documentação Técnica

05/2017

Este documento contém as instruções para a utilização da biblioteca Millenium\_Eco que se presta à comunicação de aplicativos de terceiros com o servidor de aplicações Millennium.

## Sumário

Sumário.....	2
1 Introdução.....	3
2 Millenium_Eco.....	3
2.1 Leitura de alterações.....	3
2.2 Leitura de alterações por paginação.....	4
2.3 Fluxo básico de integração.....	4
2.4 Conceito de Vitrine.....	5
2.5 Integração de Classificações ou Categorias .....	6
2.6 Integração de Produtos.....	6
2.7 Integração de Preços.....	7
2.8 Integração de Estoque .....	7
2.9 Integração de Pedidos.....	7
2.9.1 Integração de Status.....	7
2.9.2 Inclusão e Alteração de Clientes .....	10
2.9.3 Inclusão de Pedidos.....	10
3 Referência dos métodos .....	10

# 1 Introdução

O Servidor de aplicação Millennium (wtsBroker) é um serviço que permite a invocação de procedimentos de negócios (chamados de transações) executando em um computador servidor por um programa *client*. A plataforma Millennium é totalmente construída com base neste servidor, ou seja, não se trata de um recurso feito apenas para permitir integrações. Por isto, o servidor é escalável e provado, provendo um grande número de funções disponíveis que podem ser chamadas pelas APIs disponíveis.

Para permitir a chamada das funções expostas pelo servidor Millennium por qualquer linguagem de programação do Windows, a Millennium desenvolveu um componente ActiveX capaz de manipular e invocar as transações no servidor de aplicações por meio de TCP/IP utilizando um protocolo extremamente compacto e eficiente. A versão 2.0 do SDK possui também um gateway que permite a comunicação via OData/REST com o servidor. Este protocolo utiliza apenas padrões abertos da internet, permitindo a qualquer linguagem capaz de fazer chamadas HTTP acesso às funções do servidor. Qualquer nova integração deve utilizar a API OData/REST.

Devido ao grande volume de chamadas disponíveis na aplicação Millennium, foram desenvolvidas algumas bibliotecas específicas, com chamadas mais concisas, especializadas em certas áreas. Neste documento trataremos do MILLENIUM\_ECO, que foi desenvolvido para integrações onde o pedido de venda é capturado externamente ao MILLENIUM. Este cenário é típico de sites de e-commerce ou m-commerce, mas pode também ser utilizados em outros cenários como integração com coletores de dados ou qualquer outro dispositivo.

## 2 Millenium\_Eco

A biblioteca Millenium\_Eco foi desenvolvida especialmente para facilitar o uso por integradores com a plataforma Millennium. Esta biblioteca expõe apenas métodos especializados em tarefas comuns de integração como:

- Leitura da árvore de classificações do produto
- Leitura de produtos com suas características
- Leitura de estoque
- Leitura de preço
- Leitura de cadastros auxiliares
- Inserção e alteração combinada de clientes e consulta
- Inserção e consulta de pedidos

### 2.1 Leitura de alterações

Para melhor desempenho as APIs de leitura foram implementadas com recurso de "trans\_id". O "trans\_id" corresponde a um campo auto incrementável dentro do Millennium, este campo é incrementado sempre que o registro é adicionado ou atualizado, permitindo assim que o integrador consiga filtrar apenas os registros novos ou que foram atualizados desde a última listagem.

## ***2.2 Leitura de alterações por paginação***

Para melhor desempenho as APIs de leitura foram implementadas com recurso de “top”. O parâmetro “top” restringe a quantidade de registros retornados conforme condição determinada pelo integrador. Este recurso deve ser utilizado em conjunto com o parâmetro “trans\_id”. Os métodos que suportam controle de mudança são:

- **Vitrine.Lista\_Classificacoes**
- **Produtos.Lista** ou **Produtos.ListaVitrine**
- **Produtos.PrecoDeTabela**
- **Produtos.SaldoDeEstoque**
- **Pedido\_Venda.ProcessarStatus**

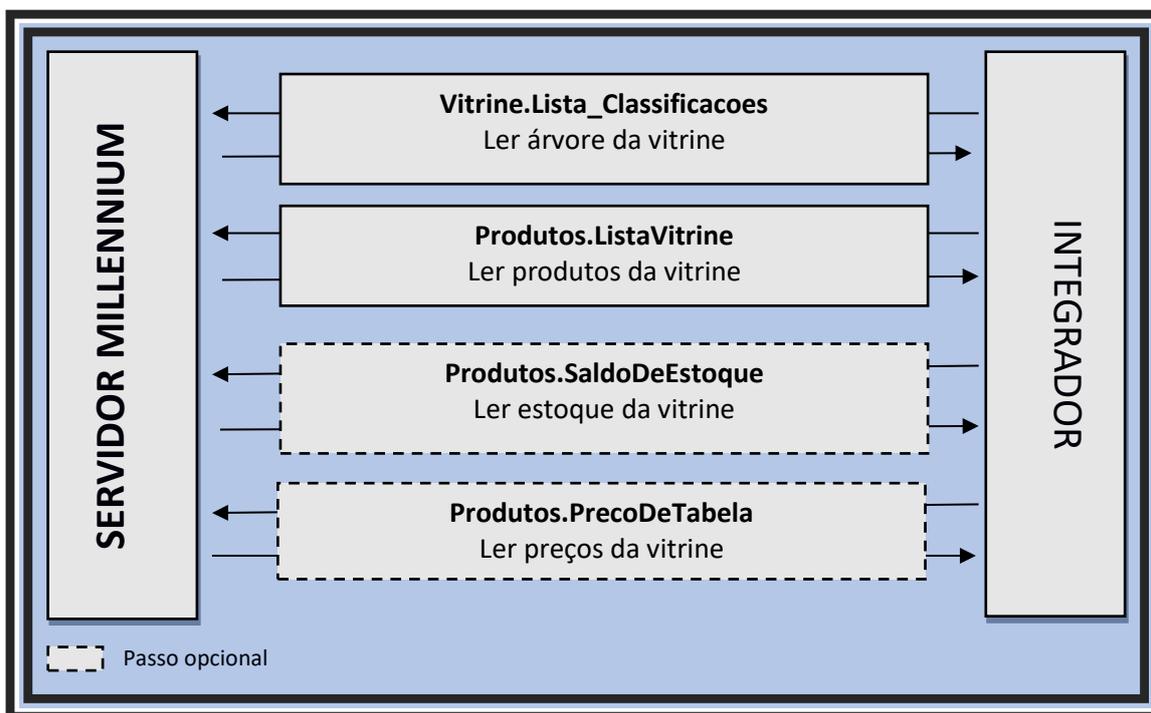
## ***2.3 Fluxo básico de integração***

O fluxo básico para o qual o millenium\_eco foi desenvolvido se baseia na exportação das informações relacionadas aos produtos (incluindo categorias, preços e estoque) e importação dos clientes e pedidos.

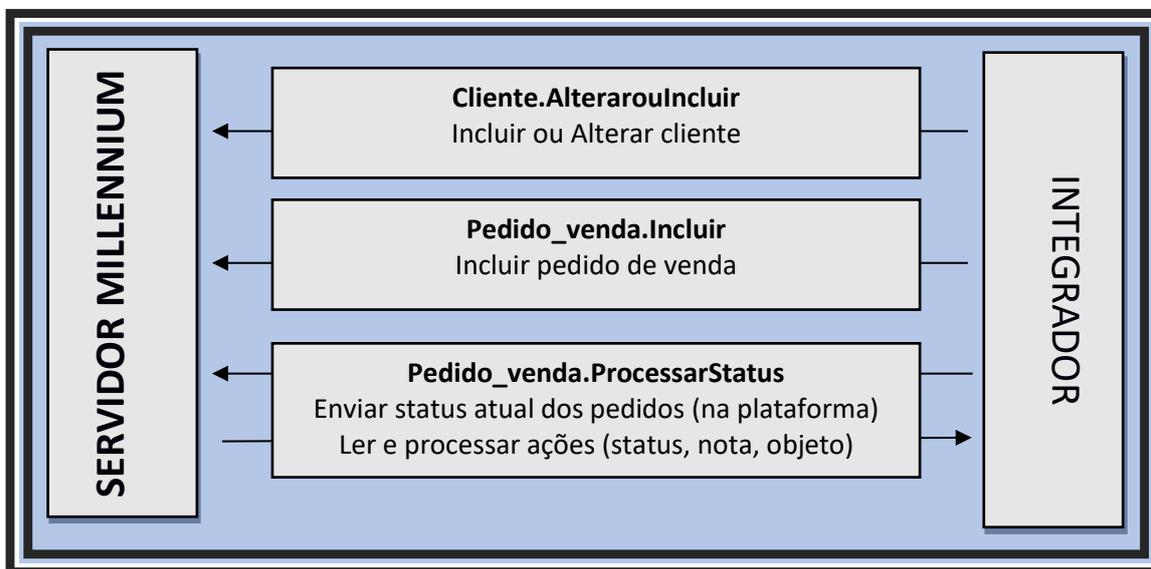
Os produtos possuem informações sensíveis como preços e estoque que podem ser sincronizadas com maior frequência e informações cadastrais que mudam menos e podem ter maior frequência de atualização. As informações de preços e estoques são obtidas por métodos especializados ao invés de serem listados como mudanças de produtos porque a lista de produtos é muito mais complexa, por outro lado, com métodos especializados é possível retornar milhares de registros mais simples com grande desempenho.

É importante enviar os pedidos ao Millennium imediatamente após serem lançados para que o estoque seja reservado e reflita a baixa. Se isto não for feito, uma chamada que busca o estoque fará a leitura incorreta do saldo, podendo levar a vendas sem estoque disponível. Se a sua plataforma possuir o conceito de reserva, esta chance é bastante minimizada.

## Fluxo de Produtos



## Fluxo de Pedido de Venda



## 2.4 Conceito de Vitrine

O conceito de vitrine foi desenvolvido na plataforma Millennium para permitir diversas apresentações diferentes dos produtos cadastrados no sistema, levando o conceito *omnichannel* ao extremo. Com este conceito, é possível organizar a árvore de produtos, determinar preços e estabelecer regras para cada canal, mesmo que estes acessem o mesmo servidor Millennium. Cada vitrine pode ser configurada, por exemplo, para atender a um site

B2B, outra para B2C, outra para uma loja no facebook e ainda outra para uma loja com parceiro todas compartilhando ou não os mesmos produtos. As possibilidades são ilimitadas.

Várias chamadas à API recebem como parâmetro um código de vitrine, por isto você deve solicitar ao administrador do seu cliente Millennium criar uma vitrine e informar seu id para que sua plataforma possa acessar.

Nesta vitrine estarão todas as configurações, produtos e pedidos para a sua plataforma.

## 2.5 Integração de Classificações ou Categorias

A integração de produtos se inicia pela integração de categorias. A chamada **Vitrine.Lista\_Classificacoes** recebe uma entrada e retorna uma lista com uma árvore de classificação no formato pai/filho. Sempre que existir qualquer mudança em um nó da categoria, o sistema irá retornar toda sua hierarquia até o último pai para facilitar a verificação dos nós.

## 2.6 Integração de Produtos

O segundo passo da integração de produtos é a leitura dos cadastros. Para isto é utilizada a chamada **Produtos.ListaVitrine**. Esta chamada recebe uma entrada e retorna uma lista com os dados básicos do produto. O `trans_id` deve ser utilizada conforme o padrão demonstrado no tópico 2.1. Além dos dados básicos de produtos, esta chamada retorna algumas listas importantes:

- **SKU:** Esta lista possui os dados das variações de um produto como cor e tamanho, além de suas características como dimensões e tempo de entrega. O campo “inativo” também deve ser verificado, já que o usuário pode desativar certos SKUs para que não apareçam para venda. As mudanças de preço e estoque devem ser lidas por meio dos métodos **PRODUTOS.PrecoDeTabela** e **PRODUTOS.SaldoDeEstoque** que também possuem controle de mudança.
- **CLASSIFICACOES:** Retorna uma lista com os identificadores das classificações da vitrine associadas ao produto. Um produto pode estar associado a mais de uma classificação (isto pode ser controlado por vitrine). Se um produto é adicionado a uma classificação e depois removido, a lista ainda assim retornará tal classificação com o campo “EXCLUIDO” marcado como *true*.
- **ESPECIFICACOES:** A lista de especificações pode ser usada para adicionar classificações que variam produto a produto. Geralmente estes valores são usados para formar a área de filtro dos produtos, ou a aba de especificações técnicas. É possível identificar o uso da especificação por seu código. Cada especificação possui um tipo, um nome e uma descrição que podem ser utilizadas para compor o conteúdo dependendo da necessidade da plataforma. O uso pela plataforma (filtro, aba etc.) pode ser controlado pelo tipo da especificação.

## 2.7 Integração de Preços

Os preços são integrados por meio do método **produtos.PrecoDeTabela** que recebe um identificador de vitrine, por onde o sistema já identifica as tabelas de preço e as promoções ativas. Este método também recebe o `trans_id` que deve ser utilizada conforme o padrão demonstrado no tópico 2.1.

São retornados os dados do SKU (produto, cor, estampa e tamanho e id SKU) o `PRECO2` (preço “de”) e `PRECO1` (preço “por”). Quando o cliente utiliza apenas uma tabela de preço o preço “de” corresponderá ao último preço cadastrado para o sku, caso não exista, será utilizado o mesmo preço listado para o preço “por”. Quando o cliente utilizar duas tabelas de preços, uma para os preços “de” e outra para o preço “por” os preços listados para “de” e “por” corresponderam respectivamente as tabelas cadastradas.

Para ambos os casos, o retorno do preço “de” e “por” sempre seguira a premissa de que o preço “por” é menor ou igual ao preço “de”, caso isso não ocorra a API irá igualar o preço “de” ao mesmo valor do preço “por”

## 2.8 Integração de Estoque

Os saldos de estoque são integrados por meio do método **produtos.SaldoDeEstoque** que recebe um identificador de vitrine. Este método também recebe o `trans_id` que deve ser utilizada conforme o padrão demonstrado no tópico 2.1.

São retornados os dados do SKU (produto, cor, estampa e tamanho e id SKU) e os saldos separados que devem ser utilizados da seguinte forma:

Se sua plataforma possui o conceito de reserva deve ser utilizado como saldo:

$$((\text{SALDO} + \text{RESERVA\_VITRINE}) - (\text{SALDO\_NAOVITRINE} - \text{RESERVA\_NAOVITRINE})) - \text{ESTOQUE\_MIN}$$

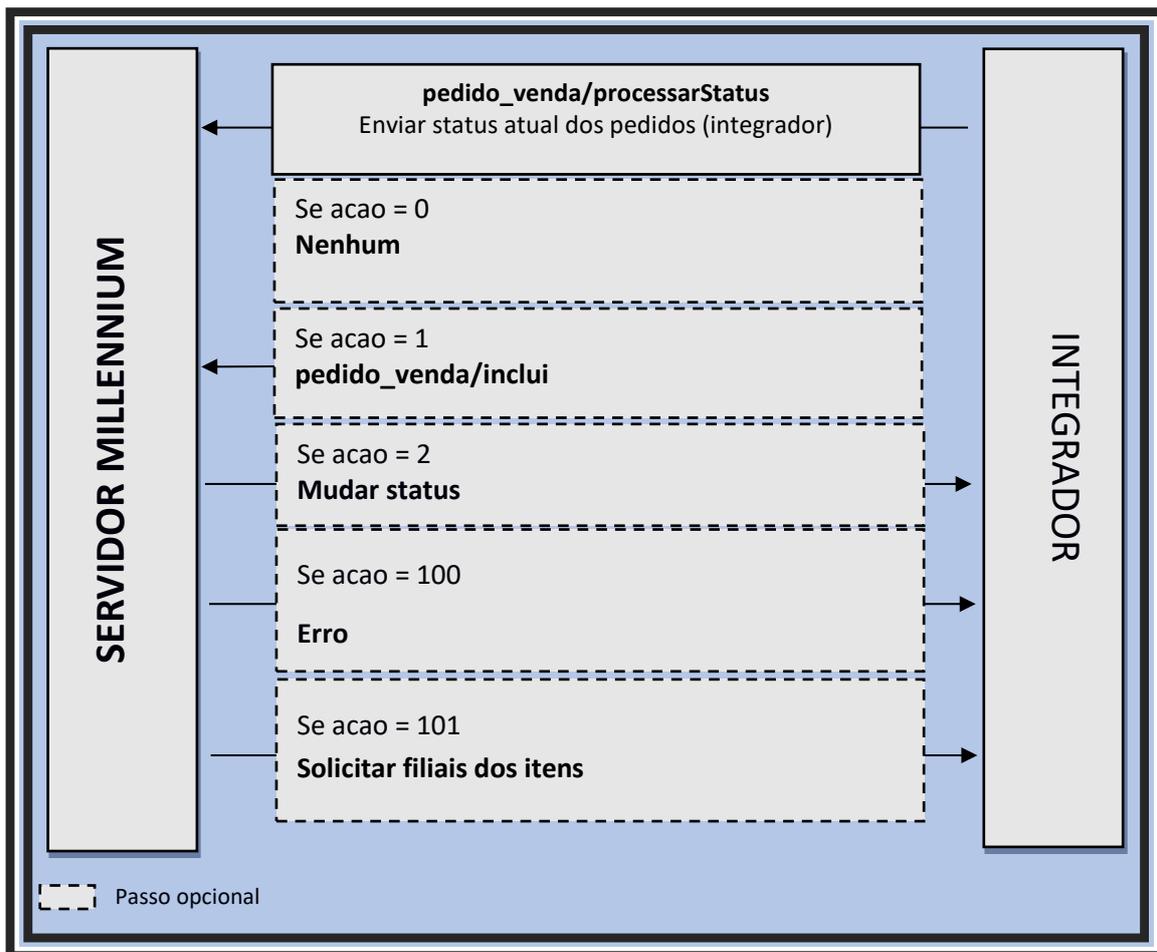
Se sua plataforma não possui o conceito de reserva, deve ser utilizado como saldo:

$$(\text{SALDO}) - (\text{SALDO\_VITRINE} - \text{RESERVA\_VITRINE}) - (\text{SALDO\_NAOVITRINE} - \text{RESERVA\_NAOVITRINE}) - \text{ESTOQUE\_MIN}$$

## 2.9 Integração de Pedidos

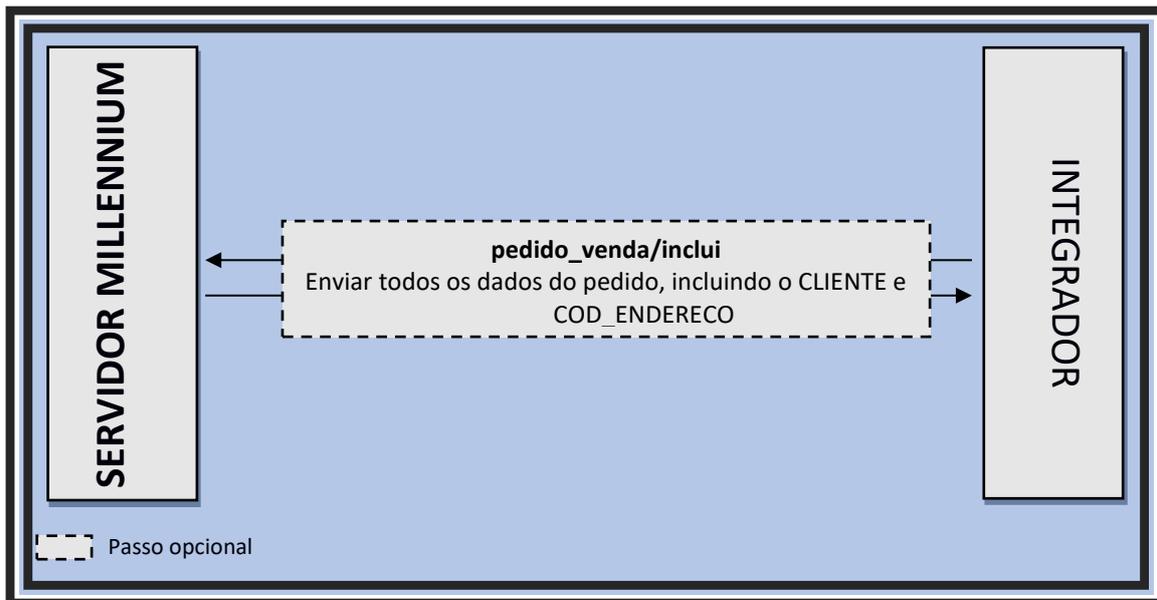
### 2.9.1 Integração de Status

Toda a integração de pedidos é orquestrada pelo método **pedido\_venda/processaStatus**. Este método bidirecional (tanto grava quanto lê status) recebe uma lista de registros contendo o código do pedido e o status atual no seu ponto de vista. Esta lista é processada no Millennium e como resposta, é retornada uma lista de códigos de pedido e ações para que você processe, como no diagrama abaixo:



Quando o Millennium retornar à ação 0, significa que os pedidos tanto pela parte da plataforma quanto pelo Millennium não sofreram alterações.

Quando o Millennium retornar à ação 1, o pedido de venda não foi localizado no Millennium e deverá ser incluído através do **pedido\_venda/inclui** conforme sequencia abaixo:



Quando o Millennium retorna à ação 2 (mudar status), o campo **status** associado a ele retornará para qual status você deve alterar seu pedido na plataforma. Caso a mudança tenha que ser feita pela parte do Millennium, o pedido andará um status para frente a cada chamada executada. É importante lembrar que existem status definidos no Millennium e estes são seguidos de forma progressiva e não retrograda, ou seja, quando um pedido estiver com status 4 (Entregue), ele não poderá retornar para status 2 (Em preparação).

Deve ser feito um mapeamento pela parte da plataforma para sincronizar os status com o Millennium. Os status definidos atualmente são:

- Status 0 = Aguardando Pagamento
- Status 1 = Pagamento Confirmado
- Status 2 = Em Preparação
- Status 3 = despachado
- Status 4 = entregue
- Status 5 = cancelado
- Status 6 = Problemas

Quando o Millennium retorna à ação 100, ocorreu um erro que será gravado no campo **“OBS”**.

Quando o Millennium retorna à ação 101, é necessário informar as filiais dos itens do pedido. Esse processo pode ser feito pelo próprio **pedido\_venda/processarStatus** informando o objeto **“itens\_filiais”**. Essa situação ocorrerá quando for informado mais de uma filial no cadastro a vitrine.

## 2.9.2 Inclusão e Alteração de Clientes

Os clientes são integrados pelo método `clientes/incluiroualterar` que recebe um identificador do cliente para o sistema realiza uma busca. Caso o cliente seja identificado pelo Millennium, os dados fornecidos vão alterar os dados do cliente já existentes. Se nenhum cliente for identificado, um novo será cadastrado

## 2.9.3 Inclusão de Pedidos

Os pedidos de venda são integrados pelo método `pedido_venda/inclui`. A flexibilidade de obrigatoriedade dos inputs deste método será de acordo com a necessidade de uso. Como qualquer outro método, existem alguns campos default obrigatórios, outros não, que podem se tornar conforme a regra de negócio da plataforma.

## 3 Referência dos métodos

Uma referência de todos os métodos disponíveis pode ser obtida on-line através da própria instância utilizada na integração. Para nossa instância de testes deve ser utilizada a seguinte url: [http://millennium.iwise.com.br:888/api/millennium\\_eco/\\$help](http://millennium.iwise.com.br:888/api/millennium_eco/$help)