Documentación Servicios Vtex

Vital

Documentación Técnica

VERSIÓN 3.0

**Referente del Cliente: Alejandra Perez Weiss**

**Fecha última modificación** 15/02/2022

Autor/es: Jorge Rojas, Lucas Albarez

Revisor/es Nicolás Arroyo

Aprobador/es Nicolás Arroyo

# Historial de Modificaciones

|  |  |  |
| --- | --- | --- |
| Versión | Autor/es | Descripción |
| 1.0 | Jorge Rojas, Lucas Albarez | Generación del documento. |
| 2.0 | Lucas Albarez | Generación de diagrama de secuencia. |
| 3.0 | Lucas Albarez | Modificación de diagrama ‘DireccionPos’ |
|  |  |  |

# Tabla de Contenidos

[Historial de Modificaciones 3](#_Toc92446861)

[Tabla de Contenidos 4](#_Toc92446862)

[Introducción 5](#_Toc92446863)

[Propósito 6](#_Toc92446864)

[Destinatarios 6](#_Toc92446865)

[Documentación de servicios 7](#_Toc92446866)

[Servicio de actualización de Productos y SKU POS VTEX 7](#_Toc92446867)

[Servicio de actualización de Stock por sucursal 10](#_Toc92446868)

[Servicio de actualización de Precios por Sucursal y Canal 12](#_Toc92446869)

[Servicio de actualización de promociones por Sucursal 14](#_Toc92446870)

[Servicio de monitoreo de Pedidos de clientes B2B hechos en VTEX para notificar a POS 16](#_Toc92446871)

[Servicio de recepción de Pedidos de clientes B2B de VTEX a POS 17](#_Toc92446872)

[Servicio de actualización de Clientes de POS a VTEX 19](#_Toc92446873)

[Servicio de actualización de Direcciones de Clientes de POS a VTEX 21](#_Toc92446874)

[Servicio Alta CSV de ofertas a Master DATA de POS a VTEX 23](#_Toc92446875)

# Introducción

# Propósito

El presente documento de tiene como propósito:

* Proporcionar de manera clara información sobre cada clase y método de la solución de servicios.
* Relacionar cada método con la capa se servicio Oracle.
* Especificar que package y procedure se ejecuta en cada invocación de los servicios de Oracle.

## Destinatarios

| Destinatario | Empresa / Sector / Rol |
| --- | --- |
| Alejandra Perez Weiss | Vital |
| Charles Maldonado | Vital |
| Nicolás Arroyo | Huenei |
| Pablo Nuccitelli | Huenei |

# Documentación de servicios

## Servicio de actualización de Productos y SKU POS VTEX

Este servicio Windows es responsable de mantener actualizados los datos de los productos y SKU de VTEX con respecto a los artículos que maneja el sistema POS VITAL.

Funcionalidad

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Caso | En .Net | En PL/SQL |
| Alta o actualización de productos a VTEX desde POS VITAL | Alta o actualización de productos. | ProcesarProductos   * Solicitar a paquete PKG\_GETDATOS\_VTEX.GetProduct la información de los productos a procesar. * Solicita a la API por el GET de refid de productos los datos a procesar. * Si obtiene datos modificar a través del API PUT del producto solo la información de los datos devueltos por POS. El resto de datos deben ser los mismos que devuelve el GET. * Si no obtiene datos insertar a través de API POST del producto todos los datos recuperados del procedimiento. * Almacenar e un arreglo de string la información de alta o actualización de productos con el siguiente formato:   + *del 1-8 el refid equivalente al cdarticulo*   + *el caracter 9 el estado puede ser: 1 procesado sin error, 2 procesado con error*   + *los siguientes 20 caracteres) la secuencia del productoid de VTEX*   + *el resto de caracteres hasta 4000 para el mensaje de error* * El proceso se repite para todos los productos que devuelve el cursor. * Finalmente llamar el procedimiento PKG\_GETDATOS\_VTEX.SETProduct, enviando como parámetro el arreglo construido. | Devuelve un cursor con la información de todos los productos que se deben actualizar o insertar.  GetProduct(  Cur\_Out Out Cursor\_Type)  Actualiza el estado de todos los productos de POS que se intentaron subir a la proforma VTEX. Con el procedimiento:  PROCEDURE SetProduct (p\_refId IN arr\_refId, p\_Ok OUT number, p\_error OUT varchar2) |
| Alta o actualización de SKU a VTEX desde POS VITAL | Alta o actualización de productos. | ProcesarSKU   * Solicitar a paquete PKG\_GETDATOS\_VTEX.GetSKU la información de los SKU a procesar. * Solicita a la API por el GET de SKUID los datos a procesar. * Si obtiene datos modificar a través del API PUT del SKU solo la información de los datos devueltos por POS. El resto de datos deben ser los mismos que devuelve el GET. * Si no obtiene datos insertar a través de API POST del SKU todos los datos recuperados del procedimiento. * Almacenar en un arreglo de string la información de alta o actualización de SKU con el siguiente formato:   + *del 1-8 char del SKUid equivalente al cdarticulo*   + *el caracter 9 el estado puede ser: 1 procesado sin error, 2 procesado con error*   + *los siguientes 20 caracteres) la secuencia del productoid de VTEX*   + *el resto de caracteres hasta 4000 para el mensaje de error* * El proceso se repite para todos los SKU que devuelve el cursor. * Finalmente llamar el procedimiento PKG\_GETDATOS\_VTEX.SetSku, enviando como parámetro el arreglo construido. | Devuelve un cursor con la información de todos los SKU que se deben actualizar o insertar.  GetSKU(  Cur\_Out Out Cursor\_Type)  Actualiza el estado de todos los SKU de POS que se intentaron subir a la proforma VTEX. Con el procedimiento:  PROCEDURE SetSKU (  p\_refId IN arr\_refId, p\_Ok OUT number, p\_error OUT varchar2) |

Especificación Técnica

|  |  |  |  |
| --- | --- | --- | --- |
| API | Service Oracle | PKG\_SP | |
| Nombre:  Vital\_Vtex\_Productos  Método:  BuildSyncProductsThr | **GetProducts**  Parámetros:  *p\_id\_canal IN Varchar2 (Sucursal)*  *Cur\_Out OUT*  Retorna:  *List<VtexProduct>* | | **PKG\_GETDATOS\_VTEX.GetProducts** |
| **GetSku**  Parámetros:  *p\_id\_canal IN Varchar2 (Sucursal)*  *Cur\_Out OUT*  Retorna:  *List<VtexNewSku>* | | **PKG\_GETDATOS\_VTEX.GetSku** |
| Nombre:  Vital\_Vtex\_Productos  Método:  DoMergeProductThr | **SetProducts**  Parámetros:  *p\_refind IN arr\_refId*  *p\_id\_canal IN Varchar2 (Sucursal)*  *p\_Ok OUT number*  *p\_error OUT varchar2*  Retorna:  *SpRet (Ok o Error)* | | **PKG\_GETDATOS\_VTEX.SetProducts** |
| Nombre:  Vital\_Vtex\_Productos  Método:  CreateSku  UpdateSku | **SetSku**  Parámetros:  *p\_refind IN arr\_refId*  *p\_id\_canal IN Varchar2 (Sucursal)*  *p\_Ok OUT number*  *p\_error OUT varchar2*  Retorna:  *SpRet (Ok o Error)* | | **PKG\_GETDATOS\_VTEX.SetSku** |

Diagrama de flujo

Para abrir el diagrama debe darle doble click al archivo DiagramaFlujoActProductYSku.html o DoMergeProductThr (método que se utiliza en el primer archivo mencionado).



## Servicio de actualización de Stock por sucursal

Este servicio Windows es responsable de mantener actualizados los datos del STOCK de VTEX por sucursal o franquicia con respecto a los artículos o SKU que maneja el sistema POS VITAL.

Funcionalidad

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Caso | En .Net | En PL/SQL |
| Alta o actualización de STOCK VTEX desde POS VITAL | Alta o actualización de STOCK. | ProcesarSTOCK   * Solicitar al procedimiento PKG\_GETDATOS\_VTEX.Getsucursales la información de las sucursales o franquicias que serán actualizadas, el paquete devuelve un cursor con los datos necesarios para establecer la conexión con la franquicia VTEX. * Si conexión exitosa solicitar al procedimiento PKG\_GETDATOS\_VTEX.GetSTOCK, el paquete devuelve por sucursal un cursor con los SKUID y la cantidad de stock que se debe actualizar. * modificar a través del API PUT del STOCK SKU la información. * El proceso se repite para todos los SKUs que devuelve el cursor. * Finalmente llamar el procedimiento PKG\_LOG\_GENERAL.WRITE, para indicar la actualización de STOCK finalizo con o sin errores | Devuelve un cursor con la información de todas las sucursales, (enviromentid) key y token  PROCEDURE GetSucursales (  p\_main in integer default 0,  p\_idcanal in integer default 0,  Cur\_Out Out Cursor\_Type)  Devuelve un cursor con la información de SKUID y STOCK para actualizar en VTEX  PROCEDURE GetStock (  p\_cdSucursal In sucursales.cdsucursal%type,  Cur\_Out Out Cursor\_Type)  Inserta en el log si finaliza con errores:  PROCEDURE write (  pi\_codigo tbllog\_general.id\_log\_general%TYPE,  pi\_mensaje tbllog\_general.mensaje%TYPE) |

Especificación Técnica

|  |  |  |
| --- | --- | --- |
| API | Service Oracle | PKG\_SP |
| Nombre:  BranchsStocksSync  Método:  BuildSyncBranchsThr | **GetSucursales**  Parámetros:  *p\_main in integer default 0*  *p\_idcanal in integer default 0*  *Cur\_Out Out*  Retorna:  *List<GetTokens> (lista de sucursales)* | **PKG\_GETDATOS\_VTEX.GetSucursales** |
| **GetStock**  Parámetros:  *p\_cdSucursal IN sucursales.cdsucursal%type,*  *p\_id\_canal IN vtexstock.id\_canal%type,*  *Cur\_Out OUT*  Retorna:  *List<VtexStock>* | **PKG\_GETDATOS\_VTEX.GetStock** |

Diagrama de flujo

Para abrir el diagrama debe darle doble click al archivo DiagramaFlujoActualizaciónStockSucursal.html o DoMergeStockThr (método que se utiliza en el primer archivo mencionado).



## Servicio de actualización de Precios por Sucursal y Canal

Este servicio Windows es responsable de mantener actualizados los datos de precios de VTEX por sucursal o franquicia y por canal o política comercial, con respecto a los artículos o SKU que maneja el sistema POS VITAL.

Funcionalidad

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Caso | En .Net | En PL/SQL |
| Alta o actualización de PRECIOS VTEX desde POS VITAL | Alta o actualización de PRECIOS. | Procesar PRECIOS   * Solicitar al procedimiento PKG\_GETDATOS\_VTEX.GetsucursalesCanal la información de las sucursales o franquicias que serán actualizadas, el paquete devuelve un cursor con los datos necesarios para establecer la conexión con la franquicia VTEX y el canal que corresponda. * Si conexión exitosa solicitar al procedimiento PKG\_GETDATOS\_VTEX.GetPrecio, el paquete devuelve un cursor con los SKUID y los distintos precios que se actualizarán por canal. * modificar a través del API POST de precios SKUs la información de precios por franquicia y política comercial o canal. * En caso de error en API POST SKUID CANAL, con el precio recibido del cursor, llama a la API PUT SKUID por sucursal creando el precio base del producto, posteriormente modifica el precio del canal con el paso anterior. * Almacenar en un arreglo de string la información de alta o actualización de precio del SKU con el siguiente formato:   + *del 1-8 char del SKUid equivalente al cdarticulo*   + *el carácter 9 el estado puede ser: 1 procesado sin error, 2 procesado con error*   + *del 10 al 14 código de la sucursal*   + *el 15 y 16 id\_canal*   + *el resto de caracteres hasta 4000 para el mensaje de error* * El proceso se repite para todos los SKUs que devuelve el cursor por sucursal y canal. * Finalmente llamar el procedimiento PKG\_GETDATOS\_VTEX.Setprecio, enviando como parámetro el arreglo construido. | Devuelve un cursor con la información de todas las sucursales, id política comercial, (enviromentid) key y token.  PROCEDURE GetSucursales (  p\_main in integer default 0,  p\_idcanal in integer default 0,  Cur\_Out Out Cursor\_Type)  Devuelve un cursor con la información de SKUID y PRECIO para actualizar en VTEX.  Por cada artículo de POS devuelve precio lista y precio oferta (si existe), por sucursal y canal.  GetPrecio(  P\_cdsucursal In sucursales.cdsucursal&type,  P\_Id\_canal VtexSellers.Id\_canal%type,  Cur\_Out Out Cursor\_Type  Actualiza el estado de todos los precios de SKU de POS que se intentaron subir a la proforma VTEX. Con el procedimiento:  PROCEDURE SetPrecio(  p\_refId IN arr\_refId, p\_Ok OUT number, p\_error OUT varchar2) |

Especificación Técnica

|  |  |  |
| --- | --- | --- |
| API | Service Oracle | PKG\_SP |
| Nombre:  BranchsPricesSync  Método:  BuildSyncPricesThr | **GetPrice**  Parámetros:  *p\_cdsucursal In sucursales.cdsucursal%type,*  *p\_id\_canal In vtexprice.id\_canal%type,*  *Cur\_Out Out*  Retorna:  *List<VtexPrice>* | **PKG\_GETDATOS\_VTEX.GetPrice** |
| **GetSucursales**  Parámetros:  *p\_main in integer default 0*  *p\_idcanal in integer default 0*  *Cur\_Out Out*  Retorna:  *List<GetTokens> (lista de sucursales)* | **PKG\_GETDATOS\_VTEX.GetSucursales** |

Diagrama de flujo

Para abrir el diagrama debe darle doble click al archivo DiagramaFlujoActPreciosSucYCanal.html o DoMergePriceThr (método que se utiliza en el primer archivo mencionado).



## Servicio de actualización de promociones por Sucursal

Este servicio Windows es responsable de mantener actualizados los datos de promociones de VTEX por sucursal o franquicia y por canal o política comercial, con respecto a las promociones que maneja el sistema POS VITAL.

Funcionalidad

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Caso | En .Net | En PL/SQL |
| Alta o actualización de Promociones VTEX desde POS VITAL | Alta o actualización de Promociones. | Procesar promociones   * Solicitar al procedimiento PKG\_GETDATOS\_VTEX.GetsucursalesCanal la información de las sucursales o franquicias que serán actualizadas, el paquete devuelve un cursor con los datos necesarios para establecer la conexión con la franquicia VTEX y el canal que corresponda. * Si conexión exitosa solicitar al procedimiento PKG\_GETDATOS\_VTEX.GetPromotion, el paquete devuelve un cursor con las promociones que se darán de alta en VTEX * modificar a través del API POST de promociones la información por franquicia. * Almacenar en un arreglo de string la información de alta o actualización de promociones con el siguiente formato:   + arreglo del 1-32 char del idpromo\_pos   + arreglo del 33-73 char del idpromo\_vtex   + el carácter 74 el estado puede ser : 1 procesado sin error, 2 procesado con error   + el carácter 75 al 79 cd sucursal   + el carácter 80 en adelante observación * El proceso se repite para todas las promociones que devuelve el cursor por sucursal. * Finalmente llamar el procedimiento PKG\_GETDATOS\_VTEX.Setpromotion, enviando como parámetro el arreglo construido. | Devuelve un cursor con la información de todas las sucursales, id política comercial, (enviromentid) key y token.  PROCEDURE GetSucursales (  p\_main in integer default 0,  p\_idcanal in integer default 0,  Cur\_Out Out Cursor\_Type)  Devuelve un cursor con la información de las promociones para actualizar en VTEX, por sucursal.  PROCEDURE GetPromotion ( p\_cdsucursal In sucursales.cdsucursal%type,  Cur\_Out Out Cursor\_Type)  Actualiza el estado de todas las promociones de POS que se intentaron subir a la proforma VTEX. Con el procedimiento:  PROCEDURE SetPromotion (p\_refId IN arr\_refId,  p\_Ok OUT number,  p\_error OUT varchar2) |

Especificación Técnica

|  |  |  |
| --- | --- | --- |
| API | Service Oracle | PKG\_SP |
| Nombre:  BranchProm  Método:  BuildSyncPromThr | **GetPromotion**  Parámetros:  *p\_idcanal In vtexsellers.id\_canal%type default 'VE',*  *Cur\_Out Out*  Retorna:  *List<VtexPromotion>* | **PKG\_GETDATOS\_VTEX.GetPromotion** |
| **GetSucursales**  Parámetros:  *p\_main in integer default 0*  *p\_idcanal in integer default 0*  *Cur\_Out Out*  Retorna:  *List<GetTokens> (lista de sucursales)* | **PKG\_GETDATOS\_VTEX.GetSucursales** |

Diagrama de flujo

Para abrir el diagrama debe darle doble click al archivo DiagramaFlujoPromociones.html.



## Servicio de monitoreo de Pedidos de clientes B2B hechos en VTEX para notificar a POS

Este servicio Windows es responsable de monitorear si existe orders o pedidos creados por los clientes B2B (VE y CO) en la proforma VTEX y deben ser enviados al sistema POS VITAL.

Funcionalidad

**PENDIENTE POR DEFINIR**

Especificación Técnica

**PENDIENTE POR DEFINIR**

## Servicio de recepción de Pedidos de clientes B2B de VTEX a POS

Este servicio Windows es responsable de extraer la información de orders o pedidos creados por los clientes B2B (VE y CO) en la proforma VTEX y deben ser armados y gestionados por el sistema POS VITAL.

Funcionalidad

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Caso | En .Net | En PL/SQL |
| Extracción de pedidos desde VTEX a POS VITAL | Entrega de pedios B2B (VE, CO) de VTEX a POS | Procesar pedidos   * Solicitar al procedimiento PKG\_CLD\_DATOS.GetPedidosVtex devuelve la información de los pedidos pendientes (pedidoid\_vtex) en VTEX por enviar a POS. * Si existe pedido por extraer de VTEX con el pedidoid\_vtex se solicita a la API orders de VTEX la información completa del pedido, luego se invoca el procedimiento PKG\_CLD\_DATOS.InsertarPedidoPOS con pedidoid\_vtex y las estructuras de entrada p\_cabecera definida así:   + Texto del 1 + 40 el idCuenta (cliente)   + Texto 41 + 1 marca de Consumidor Final. 1 CF 0 Cliente   + Texto del 42 + 40 idaddress   + Texto del 82 + 30 ammonto pedido   + Texto del 112 + 100 observaciones del pedido * En un arreglo de string la información de todos los ítems del pedido, una línea por ítems en el vector con el siguiente formato en el parámetro p\_detalle:   + Texto del 1 + 8 refId   + Texto del 9 + 20 price   + Texto del 29 + 6 quantity   + Texto del 35 + 40 identifier (promo id VTEX)   + Texto del 75 + 1 marca de promo (1) si la línea es promo (0) si la línea es producto. | Devuelve un cursor con la información de los pedidos que se deben enviar de VTEX a POS  PROCEDURE GetPedidosVtex (Cur\_Out Out Cursor\_Type)  Devuelve un cursor con pedidoid\_vtex del pedido a extraer  PROCEDURE InsertarPedidoPOS (p\_pedidoid\_vtex IN vtexpedidos.pedidoid\_vtex%type,  p\_cabecera IN varchar2,  p\_detalle IN arr\_refId,  p\_Ok OUT number,  p\_error OUT varchar2) |

Especificación Técnica

|  |  |  |
| --- | --- | --- |
| API | Service Oracle | PKG\_SP |
| Nombre:  VtexOrders  Método:  BuildSyncOrderThr | **GetPedidosVtex**  Parámetros:  *p\_id\_canal In vtexorders.id\_canal%type,*  *Cur\_Out OUT*  Retorna:  *List<VtexPedidos>* | **PKG\_TRANSFERIR\_PEDIDOS.GetPedidosVtex** |
| **InsertarPedidoPOS**  Parámetros:  *p\_pedidoid\_vtex IN vtexorders.pedidoid\_vtex%type,*  *p\_id\_canal IN vtexorders.id\_canal%type,*  *p\_cabecera IN OUT varchar2,*  *p\_detalle IN arr\_refId,*  *p\_ok OUT integer*  Retorna:  *SpRet (Ok o Error)* | **PKG\_TRANSFERIR\_PEDIDOS.InsertarPedidoPOS** |

Diagrama de flujo

Para abrir el diagrama debe darle doble click al archivo DiagramaFlujoPedidosDeClientesB2B.html.



## Servicio de actualización de Clientes de POS a VTEX

Este servicio Windows es responsable de mantener actualizados los datos de clientes de VTEX con respecto a los clientes que maneja el sistema POS VITAL.

Funcionalidad

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Caso | En .Net | En PL/SQL |
| Alta o actualización de Clientes VTEX desde POS VITAL | Alta o actualización de Clientes | Procesar Clientes   * Solicitar el listado de clientes al procedimiento PKG\_GETDATOS\_VTEX.GetClients el paquete devuelve un cursor con los datos por procesar en VTEX. * modificar a través del API de clients la información de las altas y modificaciones de clientes registrados en VTEX. * Almacenar en un arreglo de string la información de alta o actualización de clientes con el siguiente formato:   + separa del arreglo del 1 + 40 char del idcuenta   + separa del arreglo del 41 + 40 char del clientsid\_vtex   + el carácter 82 + 1 el estado puede ser : 1 procesado sin error, 2 procesado con error   + separa del arreglo 83 + 3900 observación * El proceso se repite para todos Clientes que devuelve el cursor. * Finalmente llamar el procedimiento PKG\_GETDATOS\_VTEX.Setclients, enviando como parámetro el arreglo construido. | Devuelve un cursor con la información de clientes que se deben procesar en VTEX:  PROCEDURE GetClients (Cur\_Out Out Cursor\_Type)  Actualiza el estado de todos los clientes de POS que se intentaron subir a la proforma VTEX. Con el procedimiento:  PROCEDURE SetClients (p\_refId IN arr\_refId,  p\_Ok OUT number,  p\_error OUT varchar2) |

Especificación Técnica

|  |  |  |
| --- | --- | --- |
| API | Service Oracle | PKG\_SP |
| Nombre:  VtexClients  Método:  BuildSyncClientsThr | **GetClients**  Parámetros:  *p\_id\_canal IN vtexclients.id\_canal%type,*  *Cur\_Out OUT*  Retorna:  *List<VtexClient>* | **PKG\_GETDATOS\_VTEX.GetClients** |
| **SetClients**  parámetros:  *p\_refId IN arr\_refId,*  *p\_Ok OUT number,*  *p\_error OUT varchar2*  Retorna:  *SpRet (OK o Error)* | **PKG\_GETDATOS\_VTEX.SetClients** |

Diagrama de flujo

Para abrir el diagrama debe darle doble click al archivo DiagramaFlujoClientePos.html.



## Servicio de actualización de Direcciones de Clientes de POS a VTEX

Este servicio Windows es responsable de mantener actualizados los datos de las direcciones de los clientes de VTEX con respecto a los clientes que maneja el sistema POS VITAL.

Funcionalidad

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Caso | En .Net | En PL/SQL |
| Alta o actualización de direcciones de Clientes VTEX desde POS VITAL | Alta o actualización de direcciones de Clientes | Procesar direcciones de Clientes   * Solicitar el listado de direcciones de clientes al procedimiento PKG\_GETDATOS\_VTEX.Getadress el paquete devuelve un cursor con los datos por procesar en VTEX. * modificar a través del API de adress clients la información de las altas y modificaciones de direcciones de clientes registrados en VTEX. * Almacenar en un arreglo de string la información de alta o actualización de direcciones de clientes con el siguiente formato:   + arreglo del 1 + 40 char del idcuenta   + arreglo del 41 + 40 char del clientsid\_vtex   + arreglo del 81 + 40 char del iddireccion\_vtex   + arreglo del 121 + 8 char del cdtipodireccion   + arreglo del 129 + 2 char del sqdireccion   + el caracter 131 + 1 el estado puede ser : 1 procesado sin error, 2 procesado con error   + arreglo 132 + 3850 observacion   El proceso se repite para todas las direcciones que devuelve el cursor.   * Finalmente llamar el procedimiento PKG\_GETDATOS\_VTEX.Setadress, enviando como parámetro el arreglo construido. | Devuelve un cursor con la información de direcciones de clientes que se deben procesar en VTEX:  PROCEDURE GetAdress (Cur\_Out Out Cursor\_Type)  Actualiza el estado de todas las direcciones de clientes de POS que se intentaron subir a la proforma VTEX. Con el procedimiento:  PROCEDURE SetAdress (p\_refId IN arr\_refId,  p\_Ok OUT number,  p\_error OUT varchar2) |

Especificación Técnica

|  |  |  |
| --- | --- | --- |
| API | Service Oracle | PKG\_SP |
| Nombre:  VtexClients  Método:  BuildSyncClientsAdThr | **GetAddress**  Parámetros:  *p\_id\_canal IN vtexclients.id\_canal%type,*  *Cur\_Out OUT*  Retorna:  *List<VtexAddress>* | **PKG\_GETDATOS\_VTEX.GetAddress** |
| **SetAddress**  Parámetros:  *p\_refId IN arr\_refId,*  *p\_Ok OUT number,*  *p\_error OUT varchar2*  Retorna:  *SpRet (OK o Error)* | **PKG\_GETDATOS\_VTEX.SetAddress** |

Diagrama de flujo

Para abrir el diagrama debe darle doble click al archivo DiagramaFlujoDireccionPos.html.



## Servicio Alta CSV de ofertas a Master DATA de POS a VTEX

Este servicio Windows es responsable de extraer la información de orders o pedidos creados por los clientes B2B (VE y CO) en la proforma VTEX y deben ser armados y gestionados por el sistema POS VITAL.

Funcionalidad

|  |  |  |  |
| --- | --- | --- | --- |
| Escenario | Caso | En .Net | En PL/SQL |
| Extracción de pedidos desde VTEX a POS VITAL | Entrega de pedios B2B (VE, CO) de VTEX a POS | Procesar pedidos   * Solicitar al procedimiento PKG\_CLD\_DATOS.GetPedidosVtex devuelve la información de los pedidos pendientes (pedidoid\_vtex) en VTEX por enviar a POS. * Si existe pedido por extraer de VTEX con el pedidoid\_vtex se solicita a la API orders de VTEX la información completa del pedido, luego se invoca el procedimiento PKG\_CLD\_DATOS.InsertarPedidoPOS con pedidoid\_vtex y las estructuras de entrada p\_cabecera definida así:   + Texto del 1 + 40 el idCuenta (cliente)   + Texto 41 + 1 marca de Consumidor Final. 1 CF 0 Cliente   + Texto del 42 + 40 idaddress   + Texto del 82 + 30 ammonto pedido   + Texto del 112 + 100 observaciones del pedido * En un arreglo de string la información de todos los ítems del pedido, una línea por ítems en el vector con el siguiente formato en el parámetro p\_detalle:   + Texto del 1 + 8 refId   + Texto del 9 + 20 price   + Texto del 29 + 6 quantity   + Texto del 35 + 40 identifier (promo id VTEX)   + Texto del 75 + 1 marca de promo (1) si la línea es promo (0) si la línea es producto. | Devuelve un cursor con la información de los pedidos que se deben enviar de VTEX a POS  PROCEDURE GetPedidosVtex (Cur\_Out Out Cursor\_Type)  Devuelve un cursor con pedidoid\_vtex del pedido a extraer  PROCEDURE InsertarPedidoPOS (p\_pedidoid\_vtex IN vtexpedidos.pedidoid\_vtex%type,  p\_cabecera IN varchar2,  p\_detalle IN arr\_refId,  p\_Ok OUT number,  p\_error OUT varchar2) |

Especificación Técnica

|  |  |  |
| --- | --- | --- |
| API | Service Oracle | PKG\_SP |
| Nombre:  VtexOrders  Método:  BuildSyncOrderThr | **GetPedidosVtex**  Parámetros:  *p\_id\_canal In vtexorders.id\_canal%type,*  *Cur\_Out OUT*  Retorna:  *List<VtexPedidos>* | **PKG\_TRANSFERIR\_PEDIDOS.GetPedidosVtex** |
| **InsertarPedidoPOS**  Parámetros:  *p\_pedidoid\_vtex IN vtexorders.pedidoid\_vtex%type,*  *p\_id\_canal IN vtexorders.id\_canal%type,*  *p\_cabecera IN OUT varchar2,*  *p\_detalle IN arr\_refId,*  *p\_ok OUT integer*  Retorna:  *SpRet (Ok o Error)* | **PKG\_TRANSFERIR\_PEDIDOS.InsertarPedidoPOS** |

Diagrama de flujo

Para abrir el diagrama debe darle doble click al archivo DiagramaFlujoPedidosDeClientesB2B.html.

