Objetos:

- Product2 (Salesforce estándar)
    - Campos utilizados:
        - **Name**: Guardo el nombre del producto que viene informado en el JSON como **descripcion**.
        - **Código Producto (PT_codProducto__c)**: Campo custom acorde a la necesidad del desarrollo. El código del artículo que se recibe tiene que ser único. Se ha creado como Unique Case Insensitive además de External ID.
          El motivo principal para crear un campo custom ha sido la imposibilidad de reutilizar el campo estándar, además se podría utilizar este campo para cualquier actualización que el producto requiera.
        - **Description**: campo estándar de Salesforce. Se guarda la descripción del articulo que viene informado en el JSON como **descripcion.**

- Producto Provisional (PT_productoProvisional__c)
    - Objeto custom que hace de intermediario entre la recepción en salesforce de un artículo y su posterior aprobación o rechazo.
        - Campos:
            - **Almacen Destino (PT_almacenDestino__c):** Campo que guarda el almacén de destino de un producto. Es de tipo picklist además está restringida a los valores marcados. BCN, MAD, SEV, VLN. No admite otros valores. (Picklist,Restricted)
            - **Cantidad (PT_cantidad__c)**: Campo que guarda la cantidad que se va a recibir de cada artículo.Este campo es de tipo Number(18,0).
            - **Código Producto (PT_codProducto__c)**: Campo que almacena el código del producto.(Text 20).
            - **Descripcion (PT_descripcion__c)**: Campo que guarda la descripción del producto. Text(255)
            - **Estado (PT_estadoAprobacion__c)**: Campo que guarda el estado en el que se encuentra el artículo: aprobado, pendiente o rechazado. Es un campo que está restringido a los siguientes valores: APR, PEND, RECH. (Picklist, Restricted)
            - **Fecha Envio (PT_fechaEnvio__c)**: Campo tipo date que guarda la fecha de envío de los artículos que se reciben.
            - **Producto (PT_producto__c)**: Campo relacionado con Product2. (Lookup(Product))

- Almacen (PT_almacenes__c)
    - Objeto custom que guarda los artículos que han sido aprobados por el Director de Logistica.
        - Campos:
            - **Almacen Destino (PT_almacenDestino__c):** Campo que guarda el almacén de destino de un producto. Es de tipo picklist además está restringida a los valores marcados. BCN, MAD, SEV, VLN. No admite otros valores. (Picklist,Restricted)
            - **Cantidad (PT_cantidad__c)**: Campo que guarda la cantidad que se va a recibir de cada artículo.Este campo es de tipo Number(18,0).
            - **Código Producto (PT_codProducto__c)**: Campo que almacena el código del producto.(Text 20).
            - **Descripcion (PT_descripcion__c)**: Campo que guarda la descripción del producto. Text(255)
            - **Fecha Envio (PT_fechaEnvio__c)**: Campo tipo date que guarda la fecha de envío de los artículos que se reciben.
            - **Producto (PT_producto__c)**: Campo relacionado con Product2. (Lookup(Product))

Clases:

PT_constantes.cls

Clase que guarda las constantes necesarias para evitar código hardcodeado y poder mantener el código de una forma más eficaz.

PT_insertarProductosAlmacen.cls

Clase que inserta los productos en objeto Almacen (PT_almacenes__c) una vez que se hayan aprobado por parte del Director de Logística.

- ➔ Métodos:
    - ◆ insertProductosAlmacen()
        - Método que recibe mapa por parámetro.
          Se realiza una consulta al objeto Product2 para obtener los datos correspondiente al producto.
          Después de realizar la consulta, se rellena un mapa con el código de producto como clave principal y el producto como valor.
          Para insertar los productos aprobados se realiza una iteración sobre los valores del mapa que se recibe por parámetro.
          Se utilizan mapas para coger en todo momento los datos que corresponden al artículo.

PT_postEnvioProductos.cls

Clase que ejecuta la lógica de recepción de productos mediante Apex Rest. Está declarada como @RestResource. El nombre del servicio es *envioProductos*.

Esta clase tiene dos “subclases” declaradas que guardan la estructura del JSON que se recibe.

- ➔ Métodos:
  - ◆ recepecionProductos()
    - Método declarado como @HttpPost. Es el encargado de ejecutar la lógica de recibir el JSON y lanzar las diferentes clases en el orden de ejecución correcto.

      Primero se hace un deserialize del JSON recibido, el cual se recoge con una llamada RestContext donde se realiza la request del body y se pasa a String.

      Después se realiza la deserialización al objeto o clase que tiene la estructura exacta del JSON. Una vez que se haya realizado la deserialización se realiza un bucle for sobre la lista de productos recibidos vía JSON y estos se añaden a una lista.

      Adicionalmente, para mantener la legibilidad del código y mantenimiento del mismo, se ha hecho uso de una clase auxiliar.

PT_postEnvioProductosAux.cls

Clase auxiliar que ejecuta la lógica de la clase “PT_postEnvioProductos”. Recibe por parámetro una lista con el detalle de los artículos: *id, código de producto, descripcion, almacén, fecha de envío, etc.*

- ➔ Métodos:
  - ◆ crearMapaProductosRecibidos()
    - Método que crea un mapa con los artículos recibidos. Tiene de clave principal el código de artículo dado que es un identificador único.Este método devuelve un mapa mapa relleno.
  - ◆ comprobarProductos()
    - Método que comprueba si un producto está insertado en Salesforce. Se realiza instancia un Set con los valores clave del mapa, obteniendo como resultado los registros únicos.
      Después, se realiza una consulta al objeto Product para comprobar si el producto está insertado en Salesforce.En caso de estar insertado, el producto se elimina del Set obteniendo un listado de productos no insertados.
      Por último, se comprueba que el listado Set esté relleno y se realiza un bucle for sobre el mismo para insertar los productos. Se devuelve una lista con los productos insertados.
  - ◆ insertProductosProvisionales()
    - Método que recibe mapa por parámetro. Se realiza una consulta al objeto Product2 para obtener los datos correspondiente al producto.Después de realizar la consulta, se rellena un mapa con el código de producto como clave principal y el producto como valor. Para insertar los productos pendientes de aprobación aprobados se realiza una iteración sobre los valores del mapa que se recibe por parámetro. Se utilizan mapas para coger en todo momento los datos que corresponden al artículo. Estos artículos se insertan en el objeto Producto Provisional a la espera de ser aprobados o rechazados.

PT_insertarProductosAlmacen.trigger

Trigger que se ejecuta después de que un registro en el objeto PT_productoProvisional__c se haya actualizado.

Se añade el ID de los productos actualizados en un Set que posteriormente se utiliza para hacer una consulta para recoger los datos relativos al producto actualizado.

En la misma consulta se comprueba que el estado del artículo actualizado esté en Aprobado (APR). Los artículos aprobados se añaden a un mapa que posteriormente se enviarán a la clase PT_insertarProductosAlmacen para ser procesados correctamente y dar el alta únicamente del producto válido.

Se ha desarrollado esta lógica pensando en la limpieza de la base de datos.

PT_lanzarProcesoAprobacion.trigger

Trigger que se ejecuta después de que un registro en el objeto PT_productoProvisional__c se haya insertado.

Se realiza una comprobación inicial de que el estado del producto sea Pendiente (PEND), y posteriormente se añade a una lista.

Después, se comprueba que la lista esté rellena para proceder a lanzar el proceso de aprobación.

El proceso de aprobación se lanzará N veces en función de los registros que se hayan insertado.

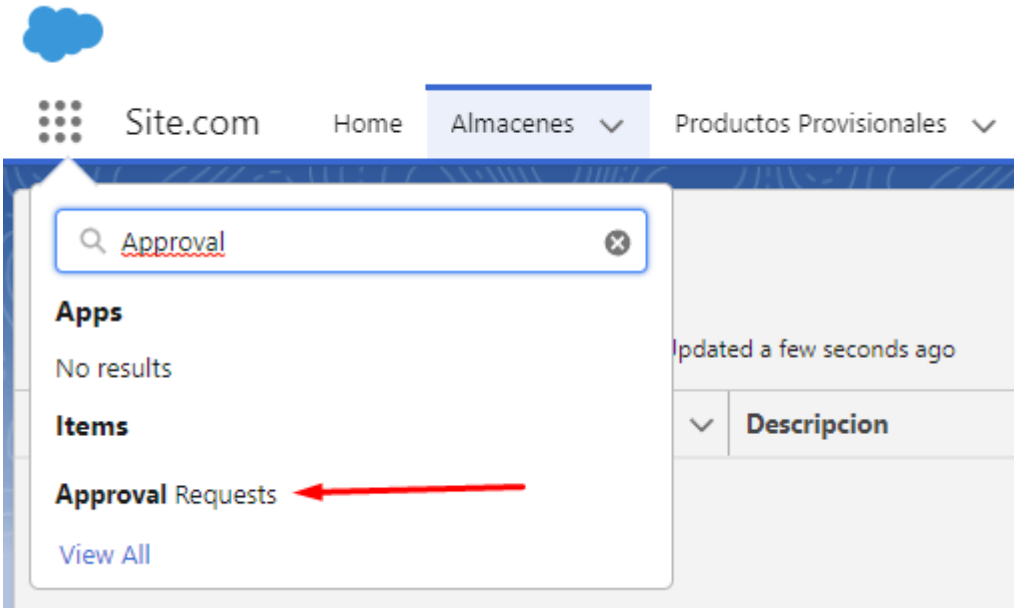
Usuarios:

Se ha habilitado el siguiente usuario en la sandbox para realizar pruebas.

URL: https://empathetic-shark-8im5y9-dev-ed.trailblaze.lightning.force.com/
Usuario: ddlogistic@yopmail.com
Contraseña: Ddl0#gistic1
Token: BpExMkp0158BB58GLdwm8v3qK

Para acceder al buzón de correos hay que dirigirse a yopmail.com e insertar ddlogistic.

Para ver los productos pendientes de Aprobación hay que darle a los nueve (9) y buscar Approval. Es una vista que no me ha dejado anclar en ningún momento.



Repositorio GIT: https://github.com/pviulian/KenosPT