CONTROL DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| Revisión | Páginas afectadas | Descripción del cambio | Fecha del cambio |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Elaborado** | **Revisado** | **Aprobado** |
| Nombre: Equipo Técnico  Cargo: Desarrollo  Fecha: 23/12/2019  Firma: | Nombre:  Cargo:  Fecha:  Firma**:** | Nombre:  Cargo:  Fecha:  Firma**:** |

Índice

[1 Documentación Relacionada 3](#_Toc27732190)

[2 Alcance 4](#_Toc27732191)

[3 Supuestos y restricciones 5](#_Toc27732192)

[4 Aprovisionamiento 6](#_Toc27732193)

[4.1 Nuevo 6](#_Toc27732194)

[5 Flujo 7](#_Toc27732195)

[5.1 Flujo / Diagrama procesos 7](#_Toc27732196)

[5.2 Vistas / Pantallas 8](#_Toc27732197)

# Documentación Relacionada

# Alcance

* Se define este documento para explicar y aclarar técnicamente el flujo que se va a llevar a cabo a la hora de hacer la ingesta de productos desde el Tracker.
* En las siguientes páginas se procederá a mostrar los componentes de AWS que se van a utilizar y el diseño del diagrama de flujo completo así como una aclaración de los bloques que lo forman.
* Servicio web que hará de CRUD, (sin borrar), de productos sobre DocumentDB.

El sistema no borrará productos sino que los marcará como cancelado.

* El tracker llamará al servicio web para crear y actualizar productos. Le pasará un producto cada vez, si el producto es nuevo el campo ProductCode vendrá vacío y al darlo de alta en el sistema crearemos el code. Si el producto ya existe vendrá ya con code.
* La API la desarrollaremos con el servicio AWS API-GATEWAY y la lógica estará en Python sobre AWS Lambdas, la orquestación con SQS.
* El sistema también debe admitir la introducción de porductos por S3.
* Habrá un sistema de versionado de productos.

Los estados entrantes solo pueden ser: ‘Borrador’, ‘Activo’ y ‘Cancelado’.

Las versiones pueden ser: 0, 1, 2, 3 …

Lo productos disponen de un campo versión que empezará como valor cero y se irá aumentando con cada nueva versión.

Cuando se dé de alta como borrador empezará con la versión cero, sucesivas actualizaciones del borrador no aumentan versión.

Al actualizarse un producto la versión anterior ira a una colección histórica de productos.

Guardaremos en el campo PreviusStatus, el estado previo a una actualización.

Cuando producto es dado de baja simplemente será la actualización de la fecha fin del producto.

Una vez un producto está en estado cancelado no volverá a ser modificado.

* Los productos antes de insertarlos y actualizarlos habrá que transformarlos, para mejorar su procesamiento en las liquidaciones.

Los campos de fecha from-to pasarán a dos arrays, uno para las fechas y otro para los valores.

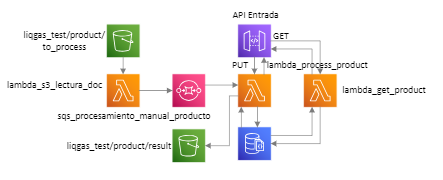
# Supuestos y restricciones

|  |  |  |
| --- | --- | --- |
| **ID** | **Supuesto** | **Descripción** |
|  | El tracker nos mandará los productos de uno en uno. |  |
|  | Versionado de productos: 0, 1, 2… |  |
|  | Status = Borrador, Activo y Cancelado |  |
|  | Haremos los desarrollos en local porque aún faltan temas por concretar. Aún no está cerrados temas de arquitectura, entornos despliegue etc… |  |
|  | Habrá una transformación de los productos, de from-to a arrays. |  |
|  | El sistema debe permitir en esta primera versión una consulta sencilla a la base de datos de productos, devolviendo el resultado como una lista de json de productos con completos. |  |
|  | No están cerrados los json de productos, aunque los campos que nos interesan para este DT si están: ProductCode, ProductVersión, Status, PreviusStatus y Type, el resto de campos se actualizarán todos. |  |

# Aprovisionamiento

## Nuevo

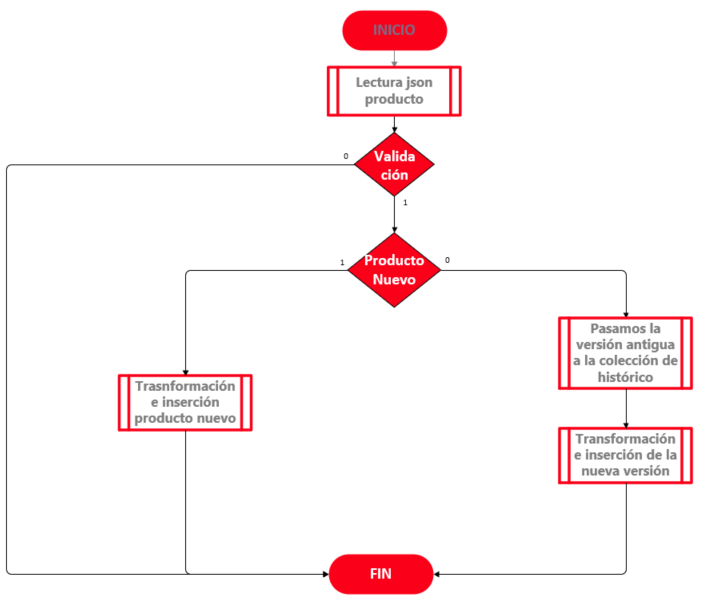
|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | Nombre | **Descripcion** | **Permisos** |
| DocumentDB | db\_test.product | Colección de productos |  |
| DocumentDB | db\_test.product\_old | Colección para el histórico de productos |  |
| S3 | liqgas\_test/product/to\_process | Donde el Tracker depositará los ficheros json de productos |  |
| S3 | liqgas\_test/product/result | Donde el sistema guardará los ficheros json una vez finalizado su procesamiento. |  |
| API Gateway | liqgas\_test\_api\_rest\_product | Componente API Gateway para la interconexión con otros sistemas |  |
| Función Lambda | liqgas\_test\_tracker\_lambda\_process\_product | Procesamiento producto, según diagrama AltaTracker |  |
| Función Lambda | liqgas\_test\_tracker\_lambda\_get\_product | Consulta del producto que se relaiza desde el Tracker |  |
| Función Lambda | liqgas\_test\_tracker\_lambda\_manual\_product\_process | Lectura documento JSON S3 vía TRACKER |  |
| SQS | liqgas\_test\_tracker\_sqs\_manual\_product\_process | Cola para el acceso al procesamiento del producto en la integracion del tracker. Se añade para en caso de estar caido la lambda no haya un error y se mantenga la solicitud de peticion del S3 hasta que pueda ejecutarse |  |



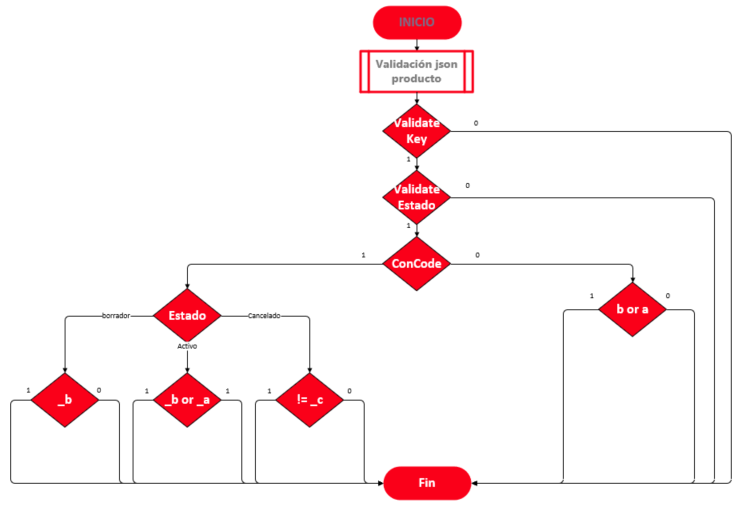
# Flujo

## Flujo / Diagrama procesos

### Proceso general



### Subproceso validación



## Vistas / Pantallas

|  |  |
| --- | --- |
|  | Inicio del proceso de ingesta de un producto |
|  | Lectura del json producto y entrada al sistema |
|  | Subproceso: validamos json producto de entrada. Ver diagrama concreto del subporceso. |
|  | Producto nuevo. Si el producto viene sin código entenderemos que el producto es nuevo. |
|  | En caso que sea un producto nuevo, generaremos un código de producto, lo transformaremos, lo insertaremos en la colección de productos y devolveremos el código generado.  La versión será la cero (¿si es activo tb?). |
|  | Si el producto ya existe, el que hay lo pasamos a la colección de histórico y el nuevo lo transformamos, le aumentamos la versión (si no es borrador), le actualizaremos el campo de la versión anterior y lo insertamos.  Devolveremos el número de la versión. |

|  |  |
| --- | --- |
|  | Subproceso para la validación del json de producto. |
|  | Validamos las keys necesarias para este proceso.  {   "CData":  {  "**ProductCode**": "BloqueFirme261120191130320001",  "**ProductVersion**": 1, 2, 3 ....  "**Status**": Borrador, Activo, Cancelado, \_Old,  "**PreviusStatus**": 'Borrador', 'Activo', 'Cancelado'  "**Type**": 'BloqueFirme', 'ServicioMolécula',...   ...  }      ...} |
|  | Validamos sea un estado correcto. Borrador, Activo o Cancelado. |
|  | Distinguimos si es un porducto nuevo. Viene con código o no. |
|  | Si viene sin código el producto solo puede ser borrador o actvo, nunca cancelado. |
|  | Si viene con código distinguimos entre borrador, activo o cancelado.  Si viene como borrador, en la base de datos debe estar como borrador.  Si viene como activo, en la base de datos debe estar como borrador o activo.  Si viene como cancelado, en la base de datos no puede estar como cancelado. No se puede cancelar un producto que ya está cancelado. |
|  |  |
|  |  |