

Manual de usuario para subida de archivos

version 5.0

Fecha:

Abril 14, 2020

Entregado a:

Empresas de la Zona Libre

Presentado por:

CrimsonLogic Panama Inc.

Ciudad del Saber, Bldg. 239

Fort Clayton, Panamá

|  |  |  |  |
| --- | --- | --- | --- |
| **Rol** | **Nombre** | **Firma** | **Fecha (dd.mm.yyyy)** |
| **Programador** | DANIEL PHILLIPS |  | 27.10.2019 |
| **Programador** | Juan torres |  | 02.04.2020 |
| **Programador Senior** | LUIS MUñOz |  | 07.04.2020 |

**Revisado por**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rol** | **Nombre** | **Version** | **Firma** | **Fecha (dd.mm.yyyy)** |
| **Ingeniero de software principal** | NIRMALYA BASU | 1.0 |  | 27.10.2019 |
| **Programador Senior** | Luis muñoz | 2.0 |  | 01.03.2020 |
| **Programador Senior** | LUIS MUñOz | 3.0 |  | 07.04.2020 |
| **Programador Senior** | LUIS MUñOz | 4.0 |  | 13.04.2020 |
| **Programador Senior** | LUIS MUñOz | 5.0 |  | 14.04.2020 |

**Aprobaciones por**

|  |  |  |  |
| --- | --- | --- | --- |
| **Designación** | **Nombre** | **Firma** | **Fecha (dd.mm.yyyy)** |
| **Ingeniero de software principal** | NIRMALYA BASU |  | 27.10.2019 |
| **Gerente del Proyecto** | ANDREVIS BERROA |  | 27.10.2019 |
| **Ingeniero de software principal** | NIRMALYA BASU |  | 14.04.2020 |

**Lista de distribución (Aplicable solo si se distribuyen copias impresas)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de control de documento** | **número de versión** | **Nombre del Recipiente** | **Fecha**  **(dd.mm.yyyy)** |
|  |  |  |  |
|  |  |  |  |

# Tabla de contenido

[Tabla de contenido 4](#_Toc37788489)

[1. Introducción 6](#_Toc37788490)

[1.1 PROPÓSITO Y ALCANCE 6](#_Toc37788491)

[1.2 DEFINICIÓN Y ABREVIATURAS 7](#_Toc37788492)

[2. Configurar el cliente “File Upload Helper” 8](#_Toc37788493)

[2.1 REQUISITOS PARA EL USO DEL CLIENTE 8](#_Toc37788494)

[2.2 ESTRUCTURA DE CARPETAS 9](#_Toc37788495)

[2.3 FORMATO DEL NOMBRE DE ARCHIVOS Y ARCHIVOS SOPORTADOS 10](#_Toc37788496)

[2.4 EL ARCHIVO DE PROPIEDADES 11](#_Toc37788497)

[2.5 PARAMETROS QUE NECESITARA EL CLIENTE AL EJECUTARSE 13](#_Toc37788498)

[2.6 QUE HACER SI SE HIZO UN CAMBIO DE USUARIO O CONTRASEÑA 14](#_Toc37788499)

[2.7 Conocer si los archivos fueron enviados con éxito 14](#_Toc37788500)

[2.8 ¿QUÉ OCURRE CON LOS ARCHIVOS DUPLICADOS? 15](#_Toc37788501)

[2.9 EJECUTAR EL CLIENTE 15](#_Toc37788502)

[2.10 ESTADO DE PROCESAMIENTO DE ARCHIVOS EN EL SISTEMA DMCE 17](#_Toc37788503)

[3. Consumir Servicio Web “File Upload” 21](#_Toc37788504)

[3.1 REQUISITOS PARA CONSUMIR EL SERVICIO WEB DESDE OTRO SISTEMA 21](#_Toc37788505)

[3.2 ESTRUCTURA DE LA CLASE “ListContainer” 22](#_Toc37788506)

[3.3 ESTRUCTURA DE LA CLASE “ServiceResponse” 24](#_Toc37788507)

[3.4 PROGRAMANDO EL CLIENTE 26](#_Toc37788508)

[3.5 CONVIRTIENDO LOS ARCHIVOS EN ARRAYS DE BYTES 28](#_Toc37788509)

[3.6 AUTENTICACIÓN EN EL SERVICIO WEB 31](#_Toc37788510)

[3.7 CIFRADO DE CREDENCIALES 33](#_Toc37788511)

[4. Formato de archivos 35](#_Toc37788512)

[4.1 ARCHIVOS DE TEXTO PLANO 35](#_Toc37788513)

[4.1.1 Orden que se deben seguir con este formato 35](#_Toc37788514)

[4.1.2 Reglas que se deben seguir con este formato 36](#_Toc37788515)

[4.1.3 EJEMPLO PARA ENTORNO DE PRUEBA 37](#_Toc37788516)

[4.1.4 EJEMPLO PARA ENTORNO DE PRODUCCIÓN 39](#_Toc37788517)

[4.2 ARCHIVOS XML 41](#_Toc37788518)

[4.2.1 Orden que se deben seguir con este formato 41](#_Toc37788519)

[4.2.2 Reglas que se deben seguir con este formato 42](#_Toc37788520)

[4.2.3 EJEMPLO PARA ENTORNO DE PRUEBA 43](#_Toc37788521)

[4.2.4 EJEMPLO PARA ENTORNO DE PRODUCCIÓN 45](#_Toc37788522)

[4.3 TABLA DE VALORES 47](#_Toc37788523)

[4.4 Ver valores de codigos dentro del sistema 72](#_Toc37788524)

[4.5 Ver estado de procesamiento de los archivos subidos 75](#_Toc37788525)

# Introducción

## PROPÓSITO Y ALCANCE

El objetivo de este documento es detallar las configuraciones para el servicio de la subida de archivos en el nuevo sistema del DMCE. Este documento se dividirá en dos partes como la carga de archivos se puede hacer de dos maneras:

1. **Configuración para utilizar el cliente creado por CrimsonLogic**:

La empresa a desarrollado un pequeño cliente para la subida de archivos. Las configuraciones son bastantes sencillas y no se requiere de muchos conocimientos técnicos.  
Las empresas simplemente pueden usar este cliente para cargar archivos sin la necesidad de ningún esfuerzo de desarrollo.

1. **Configuración y programación necesaria para consumir el servicio web (Web Service) desde otro sistema**: Esta sección abarca todo lo que se necesita para poder crear su propio cliente del File Upload. Esta sección es ideal para aquellas empresas que desean anexar este servicio web a sus sistemas / aplicaciones web.

Usted puede elegir cualquiera de los dos caminos si desea empezar a consumir el servicio. Con esta funcionalidad las empresas (a quienes va dirigido este documento), podrán subir archivos como declaraciones, inventarios o lista de materiales.

## DEFINICIÓN Y ABREVIATURAS

|  |  |
| --- | --- |
| **Termino** | **Descripción** |
| DMCE | Declaración de Movimiento Comercial. |
| CFZ | Zona Libre de Colón. |
| Número CR | Clave de operación de la empresa (Customs Registration Number). |
| BOMLIST | Archivos de tipo lista de materiales (Bill of Materials). |
| DMCFORM | Archivos de tipo Declaraciones. |
| INVENTORY | Archivos de tipo Inventario. |
| INITIAL CONTAINER | Archivos de tipo Contenedor Inicial. |
| CONTAINER | Archivos de tipo Contenedor. |
| File Upload | Subida de archivos. |
| File Transfer Helper | Nombre del ejecutable del cliente para la subida de archivos. |
| . batch | Archivos de procesamiento por lotes. |
| Shell Script | Fichero en donde se introduce comandos que son ejecutados en el Shell. |
| . jar | Archivos ejecutables JAVA. |
| FUSERV | Rol del sistema para otorgar la subida de archivos desde el cliente. |
| . properties | Archivos de propiedades. |
| WSDL | Protocolo XML (Web Services Description Language) |
| SOAP | Simple Object Access Protocol |

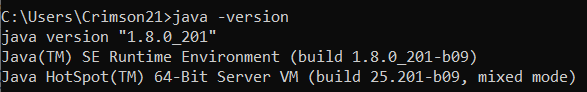
***TABLA 1: Cliente File Upload***

# Configurar el cliente “File Upload Helper”

## REQUISITOS PARA EL USO DEL CLIENTE

* **Tener instalado JAVA en el sistema operativo**

Para conocer si la maquina ya tiene instalado JAVA, abra la consola y ejecute el comando *java -*version



***FIGURA 2: Revisión de version de JAVA por consola***

Si al ejecutar este comando no obtiene un resultado similar al de la imagen presentada, entonces necesitara instalar JAVA.

Sistemas Operativos Windows**:**

* Puede descargar el instalador de JAVA en la siguiente dirección: <https://www.java.com/es/download/>.
* Ejecutar el archivo ejecutable de JAVA.
* **Manejo del sistema de archivos (File System) en el sistema operativo**

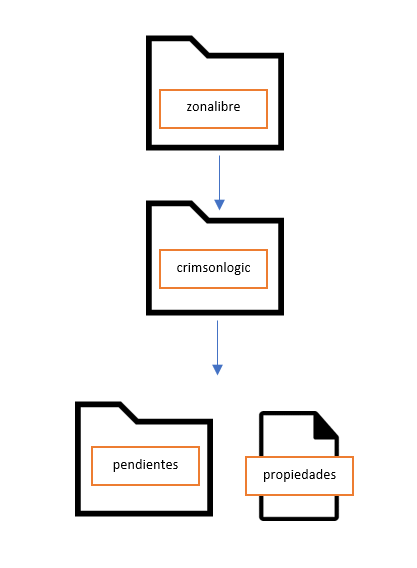
La persona debe conocer y saber manejar el sistema de archivos del computador. Esto abarca desde la creación y manejo de carpetas dentro del sistema operativo hasta el control de las rutas de estos.

* **Conocimientos de ordenes en procesamientos por lotes**

Esto se requiere en caso tal de que se necesiten personalizaciones por parte de la empresa al momento de la ejecución del cliente (por ejemplo, si se desea ejecutar el cliente File Transfer Helper en cada intervalo de tiempo que se desea).

## ESTRUCTURA DE CARPETAS

Se necesita tener la siguiente estructura de archivos para que el cliente funcione. Esta estructura se debe crear en el directorio raíz del sistema (‘C:/’ en Windows o ‘/’ en Linux):



***FIGURA 2: Estructura de archivos necesaria para la configuración del cliente***

\* Los dos archivos que se manipularan frecuentemente son la carpeta “pendientes” y el archivo . properties “propiedades”.

\* En la carpeta (“pendientes”) se van a depositar todos los archivos que se desean subir al servidor y el archivo de propiedades (“propiedades.properties”) almacenara información sobre toda la configuración que necesita tener para poder acceder al servicio web.

\* También el archivo de propiedades almacenara las credenciales de usuario que se utilizaran para la subida de archivos en caso tal que se desee hacerse automáticamente por medio de archivos. batch o Shell script.

## FORMATO DEL NOMBRE DE ARCHIVOS Y ARCHIVOS SOPORTADOS

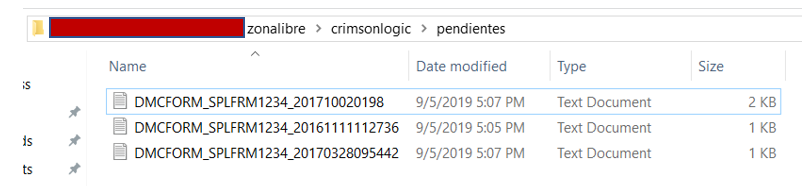
Los archivos que se deseen subir al servidor deben cumplir con un formato específico para poder ser procesados posteriormente. Si no se cumple dicho formato, entonces como consecuencia, esto causara errores de procesamiento dentro del servidor. El formato utilizado por el sistema es el siguiente:

**[Tipo de Documento]\_[Número CR]\_[Fecha]**

**La parte resaltada en rojo es importante**, ya que la primera parte especifica que tipo de archivo se está subiendo al servidor. La segunda parte es clave de operación. Y la tercera parte es la fecha y la hora, pero en **milisegundos**.

Los tipos de archivos actualmente soportados por el servidor son los siguientes:

/ Archivos de Declaraciones (**DMCFORM**).



***FIGURA 3: Formato de nombre requerido para archivos a subir***

## EL ARCHIVO DE PROPIEDADES

El archivo de propiedades contiene la información de credenciales de usuario. Estas propiedades son requeridas si usted desea que el cliente automáticamente empiece a enviar archivos al ejecutarse.



***FIGURA 4: Valores de configuración de Usuario***

Si usted no crea un archivo de propiedades con estos valores, cuando ejecute el cliente, este le pedirá las credenciales por consola.

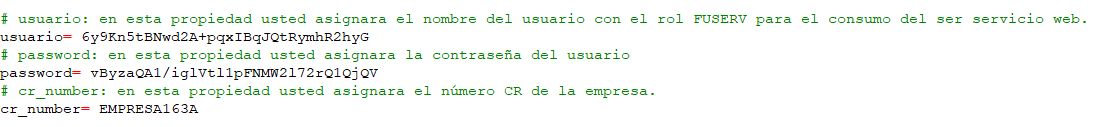
Como medida de seguridad y lograr ofuscación en las credenciales, el cliente encriptara automáticamente las credenciales de usuario cuando se ejecute. Este es un comportamiento por defecto del cliente.



***FIGURA 5: Valores de usuario cifradas***

**Nota:**

1. **Las credenciales solo se proporcionarán a pedido. Póngase en contacto con la Administración de ZLC para que se proporcionen las credenciales para este sistema.**

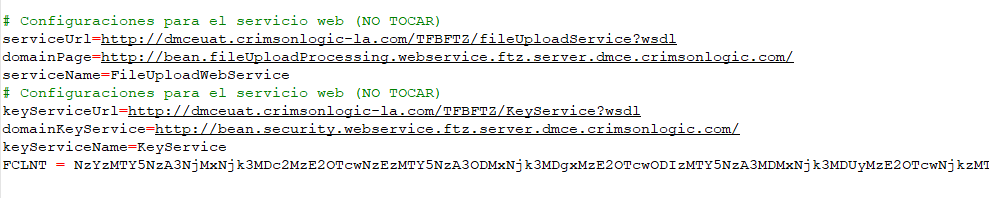
1. **Las credenciales son para cargar archivos de una empresa individual. Si es una empresa que representa a varias compañías y carga archivos para varias compañías, debe mantener las credenciales adecuadas de la compañía respectiva en el archivo de propiedades antes de ejecutar el cliente para enviar el archivo.  
     
   Además, cada vez que realice un cambio en el archivo de propiedades para cambiar la información de la compañía, se deben cambiar los siguientes campos usuario, contraseña y número\_cr. Y por favor elimine la propiedad auto-generada "FLCLNT".  
   Imagen abajo:**



## PARAMETROS QUE NECESITARA EL CLIENTE AL EJECUTARSE

El cliente para poder funcionar requiere de algunos parámetros. Los únicos parámetros que se deberán modificar dentro del archivo de propiedades son los parámetros de **usuario**, **password** y **cr\_number**.

1. **La siguiente información es para el entorno de prueba**: Propiedades mostradas en la siguiente imagen son utilizadas por el cliente y estos no se deben manipular.



***FIGURA 6: Propiedades que utilizara el cliente***

1. **La siguiente información es para el entorno de producción**: El entorno de producción DMCE 2.0 aún no está listo. Cuando esté listo, se compartirá la información necesaria que deberá actualizarse en el archivo de propiedades

## QUE HACER SI SE HIZO UN CAMBIO DE USUARIO O CONTRASEÑA

Si por algún motivo usted realizo un cambio de contraseña al usuario utilizado para este cliente o simplemente desea cambiar el usuario, usted deberá volver a actualizar las propiedades “usuario” y “password” con los valores requeridos y, por último, eliminar la propiedad auto generada por el cliente llamado “**CLINT**” del archivo de propiedades para que este vuelva a funcionar con normalidad.

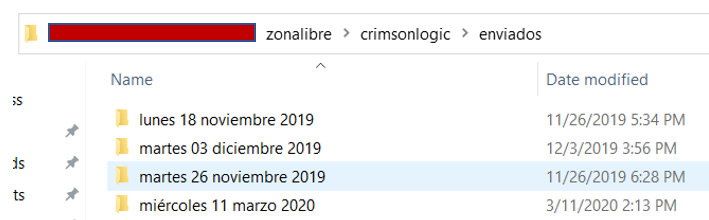
## Conocer si los archivos fueron enviados con éxito

\* Localmente usted puede comprobar si los archivos fueron enviados con éxito si dentro de la ruta ***\zonalibre\crimsonlogic***existe una carpeta con el nombre de “**enviados**”.

\* De igual manera, para saber los archivos que no pudieron ser enviados al servidor, usted puede acceder a la carpeta “**fallados**” en la misma ubicación.

\* Cada vez que se finaliza una transferencia, el cliente de acuerdo con la respuesta recibida por el servidor, ubicara el archivo en la carpeta correspondiente. Se crean carpetas con fecha para almacenar los archivos para ubicarlos fácilmente.

Los archivos se irán guardando **por fecha** dentro de cada carpeta.

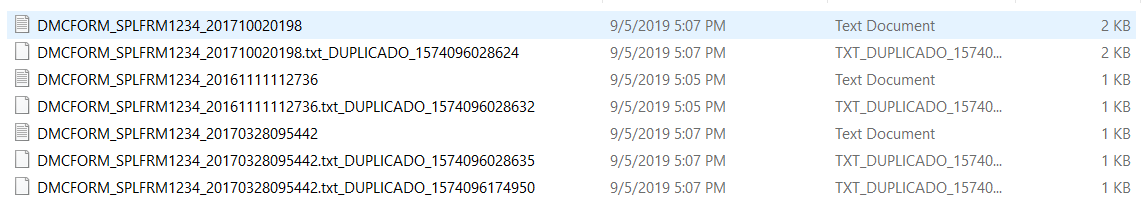


***FIGURA 7: Carpetas generadas por el cliente***

Las carpetas mencionadas se crearán **automáticamente** por el cliente.

## ¿QUÉ OCURRE CON LOS ARCHIVOS DUPLICADOS?

Si por algún motivo, se volvió a subir un archivo que posee el mismo nombre que otro archivo dentro del servidor, el cliente marcara este archivo como duplicado, seguido de un valor en milisegundos. Hay que resaltar que, si ocurre esto, significa que del lado del servidor el archivo no se ha procesado aún.El comportamiento predeterminado dentro del servidor para estos casos es **reemplazar el archivo**, por lo que se pide cuidado con el manejo de los archivos.

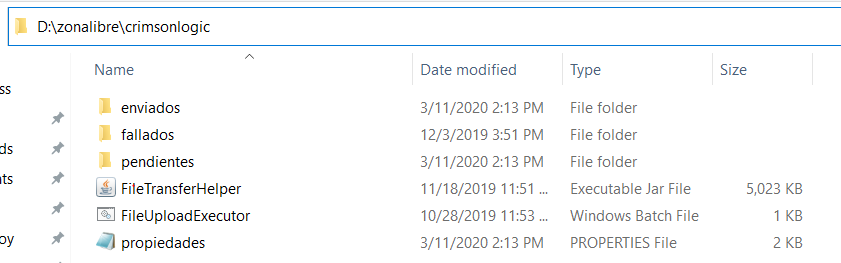


***FIGURA 8: Archivos duplicados***

## EJECUTAR EL CLIENTE

Para ejecutar el cliente, usted deberá ejecutar el archivo **.jar** desde un archivo **.bat** (puede utilizar que la empresa le proporciono). Dentro del archivo .bat la empresa podrá ingresar personalizaciones propias, como la de configurar el intervalo de tiempo el cual el ejecutable .jar será llamado. Para que el archivo .jar se ejecute sin problemas, usted deberá realizar la configuración mencionada en los puntos anteriores, además de que los archivos se encuentren **bajo la misma unidad de disco**.

**Por ejemplo**:   
  
si usted realizo la configuración dentro de la unidad **C:/** (para Windows) y los archivos **.bat** tanto como el **.jar** se encuentran en otra unidad (por ejemplo, **D:/**) el cliente no será capaz de encontrar la configuración hecha.   
Como sugerencia, usted puede conservar ambos archivos bajo la carpeta **crimsonlogic**.



***FIGURA 9: Ubicación sugerida de .jar y .bat***

Actualmente para que él **.jar** se ejecute, este tiene que estar al mismo nivel que el archivo **.bat** (ósea, misma carpeta).

Ejemplo de la estructura se muestra en la imagen de arriba. Consulte la **sección 2.2** para cualquier duda relacionada con la estructura de las carpetas.

## ESTADO DE PROCESAMIENTO DE ARCHIVOS EN EL SISTEMA DMCE

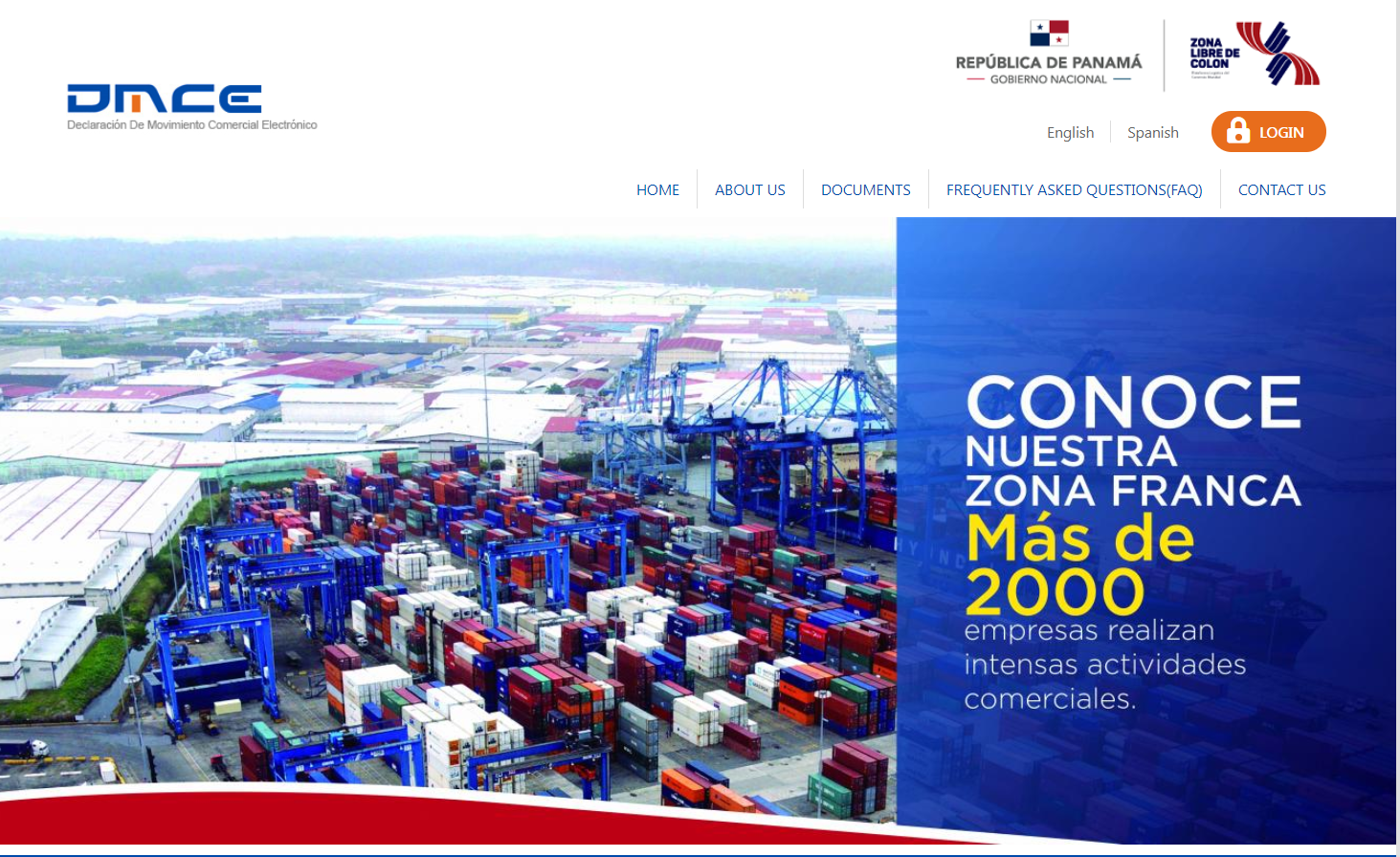
Una vez que los archivos se han enviado al servidor DMCE con éxito, la aplicación DMCE en un intervalo de 5 minutos procesa los archivos.

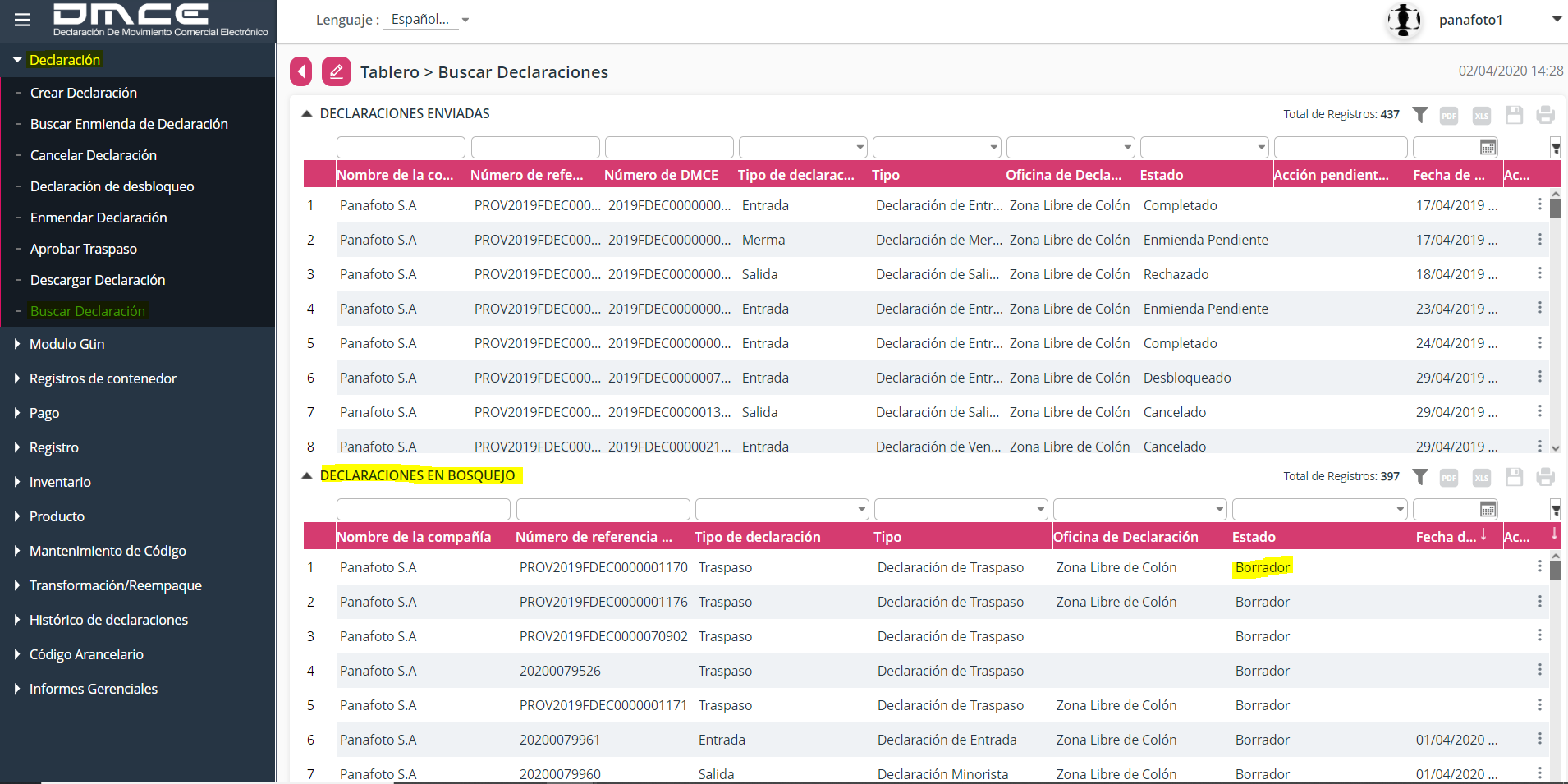
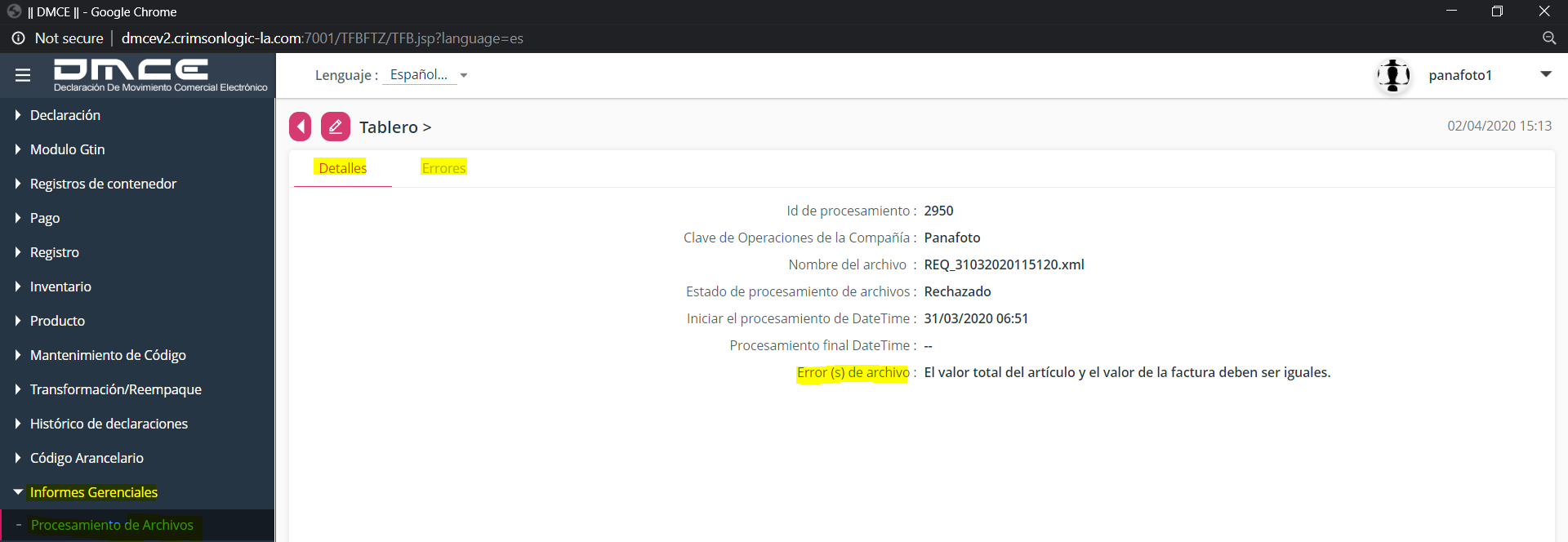
El usuario de la empresa puede iniciar sesión en la aplicación DMCE y revisar si los archivos enviados se han procesado correctamente o si hay errores de validación en los datos enviados.

A continuación, los pasos para ver el estado del procesamiento de archivos en la aplicación:

1. Inicie sesión en la aplicación DMCE:   
     
   **URL Prueba:** [http://dmceuat.crimsonlogic-la.com/TFBFTZ/cusLogin/login.cl](https://dmceuat.crimsonlogic-la.com/TFBFTZ/cusLogin/login.cl)

**URL Producción:** *No disponible en este momento.*



1. Vaya al menú, Informes Gerenciales. Y haga clic en el submenú Procesamiento de Archivos.  
     
   
2. Cuando el procesamiento sea exitoso, la declaración se guardará en el borrador de estado. Este DMCE en borrador de estado se puede encontrar en el menú buscar declaración.  
     
   
3. Cuando falla el procesamiento, el usuario puede ver el motivo detallado.  
     
   

# Consumir Servicio Web “File Upload”

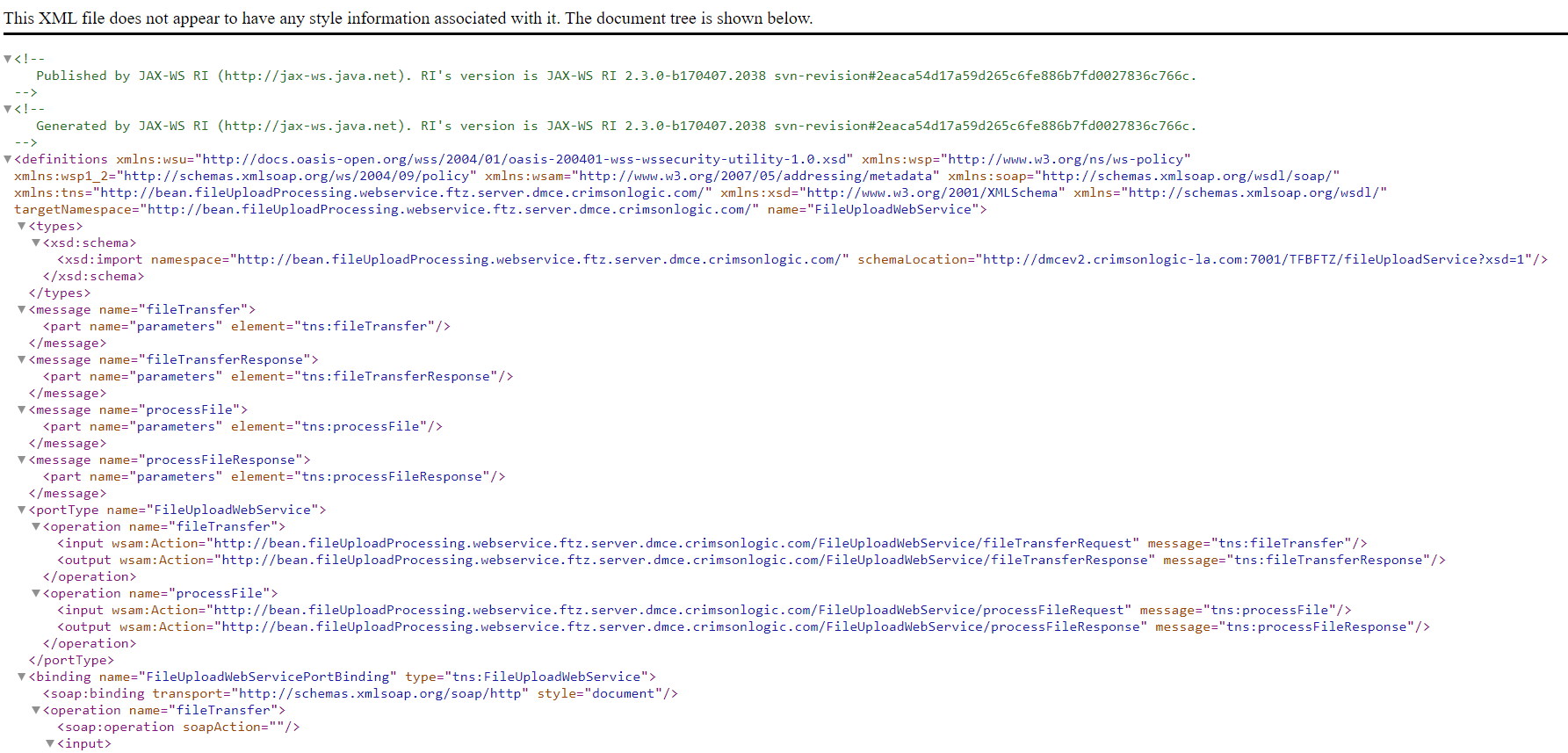
## REQUISITOS PARA CONSUMIR EL SERVICIO WEB DESDE OTRO SISTEMA

Si opta por esta opción, se está suponiendo que usted sea un programador. La dirección WSDL del servicio web donde usted puede observar la estructura de este es el siguiente:

**Dirección para el Entorno de Prueba 🡪** <http://dmceuat.crimsonlogic-la.com/TFBFTZ/fileUploadService?wsdl>

**Dirección para el Entorno de Producción 🡪** *El usuario debe solicitar la dirección a la administración de ZLC. Esto será cuando el entorno de producción esté listo.*

Actualmente el servicio web cuenta solo con dos servicios; El servicio para la transferencia de archivos al servidor y el servicio para el procesamiento de archivos. El segundo servicio es únicamente consumido por el servidor.



***FIGURA 10: WSDL del servicio File Upload***

El servidor para la subida de archivos cuenta con dos parámetros:

🡪 El primer parámetro es un Objeto del tipo **ListContainer**.

🡪 El segundo parámetro es un Objeto String que es el Clave de Operación.

También se debe mencionar que el servicio web devuelve una instancia del tipo **ServiceResponse**.

El servicio web fue desarrollado bajo el API **JAX**-**WS** (**Java** API for XML Web Services). Mas adelante mostraremos un ejemplo de cómo se puede desarrollar el cliente utilizando JAVA.

**Nota**: El cliente del servicio web se puede desarrollar en otros lenguajes de programación, así como en C #. Como es imposible poner un ejemplo de codificación para diferentes lenguajes de programación, por lo tanto, solo mostramos Java.

Primero veamos la estructura de las dos clases utilizados por el servicio.

## ESTRUCTURA DE LA CLASE “ListContainer”

Esta clase desempeña el rol de ser un contenedor (como su nombre lo indica). Esta clase internamente tiene como atributo, una lista de instancias del tipo **BatchFile**.



***FIGURA 11: Estructura de la clase ListContainer***

La siguiente imagen contiene la estructura de la clase “**BatchFile”**:



***FIGURA 12: Estructura de la clase BatchFile***

El tipo BatchFile, son objetos que se encargaran de almacenar los “pedacitos” de los archivos. Cuando el servicio recibe un objeto batchFile, este extrae los bytes que el objeto almacena (**byte [] fileContent**) para poder volver a construir nuevamente el archivo.

Con el atributo de clase **sequence** velamos por la integridad de los datos y nos aseguramos de que el archivo no haya sufrido modificaciones durante la transferencia o no se envié como corrupto al servidor.

## ESTRUCTURA DE LA CLASE “ServiceResponse”

La clase ServiceResponse se encarga de comunicarnos el estado de la transferencia. Esta clase almacena un resumen general del proceso:



***FIGURA 13: Estructura de la clase ServiceResponse***

Podemos observar que los atributos de la clase son los siguientes:

🡪 **completed:** Este es un Set que almacena todos los archivos que fueron transferidos exitosamente.

🡪 **failed:** Al igual que el atributo anterior, tambien se trata de un Set pero que esta vez almacena todos los archivos que han fallado durante la transferencia.

🡪 **status:** Este objeto de tipo string almacenara el resultado final de la transferencia, es decir, si la transferencia fue exitosa o la misma fue fallida.

🡪 **message:** La funcion de este objeto de tipo string es almacenar los mensajes que el servicio envia al cliente, ya sea por ejemplo errores de validacion, errores dentro del servidor, mensaje exitoso de transferencia, etc.

## PROGRAMANDO EL CLIENTE

Como primer paso para programar el cliente, necesitamos crear una interface del servicio.



***FIGURA 14: Estructura de la interfaz del servicio***

Como observamos en la imagen, debemos importar unas cuantas librerías de JAX-WS. Nótese que las clases expuestas anteriormente deben crearse ya que como se mencionó, estas clases son parámetros del servicio web. La información del **targetNamespace** puede ser obtenida de la dirección WSDL expuesta anteriormente en el punto **3.1.**

Vamos a desarrollar un ejemplo de como podriamos invocar el servicio. Supongase que en nuestra clase tenemos un metodo llamado “***callFileUploadService***”. Este metodo va a tener un parametro, el cual sera una Lista de objetos BatchFile, identico como el parametro del servicio web. Tenemos que recordar tambien que el servicio de la subida de archivos devuelve una respuesta. Es por eso que nuestro metodo sera tipo funcion y devolvera un objeto **ServiceResponse**:



***FIGURA 15: Ejemplo de método para llamar el servicio***

Vamos a ver este método de ejemplo por partes (similar al método del cliente que se desarrolló). En la primera parte del método, tenemos declarado nuestro objeto que contendrá la respuesta por parte del servidor o por parte del mismo método en caso tal de que una excepción haya sucedido durante la ejecución. Como siguiente parte se abre una sentencia try-catch en donde se empezará a asignar todos los valores de configuración necesarios. Empezamos asignándole el ***service URI*** al objeto “url” y luego crearemos un ‘qualified name’ (**QName**) que contendrá igualmente el ‘namespace’ URI y el nombre del servicio que se quiere consumir.

El servicio finalmente se creará cuando se ejecute la línea *Service.create(url, qname)*. Nuestro método como paso siguiente después de crearse el servicio, va a crear una instancia de nuestra interfaz **FileUploadWebService** en donde obtendrá el puerto de parte del servicio que se acabó de crear anteriormente.

Ahora viene una parte importante. Notaremos que el método creara una instancia de la clase ListContainer para almacenar la lista de objetos BatchFile. Antes de llamar a este método, primero se debió convertir los archivos a subir en bytes. Aquí se presenta un ejemplo de cómo se puede hacer esto.

## CONVIRTIENDO LOS ARCHIVOS EN ARRAYS DE BYTES

Este es un ejemplo de cómo podemos convertir nuestros archivos en bytes y almacenarlos en los objetos BatchFile.

Se recomienda seguir este metodo para generar una lista de objetos BatchFiles. Este metodo itera en cada archivo dentro del array de la clase File.



***FIGURA 16: Método para procesar nuestros archivos***

En cada iteracion que se ejecuta, se realiza el siguiente procedimiento:

🡪 Crea un array nuevo de bytes.

🡪 Lee los bytes gracias al objeto FileInputStream.

🡪 Crea un Checksum. Este objeto tiene la funcion de asegurarse de que el archivo no haya sufrido modificaciones durante el envio, gracias a su algorithmo de suma de checkeo.

🡪 Creamos un objeto BatchFile nuevo.

🡪 Asignamos los bytes al atributo de clase FileContent.

🡪 Creamos un StringBuilder para construir el valor para el atributo Sequence. Este atributo esta conformado por dos partes:

**Nombre\_del\_archivo**@Checksum

🡪 Por ultimo se agrega el objeto BatchFile a la lista.

## AUTENTICACIÓN EN EL SERVICIO WEB

Volviendo al metodo que estabamos exponiendo anteriormente (***callFileUploadService***), al almacenar la lista de archivos BatchFile, llegamos a una linea en donde se ha declarado un Mapa de tipo <String,Object>:



***FIGURA 17: Asignando parámetros para Autenticación***

**Nota**: La URL en la imagen de arriba apunta al entorno de prueba. Cuando el entorno de producción esté disponible, deberá realizar cambios para apuntar a la URL correcta. La URL de producción se proporcionará cuando el entorno de producción esté listo.

Como medida de seguridad que se ha implementado, el servicio web requiere autenticación para evitar que personal no autorizado la consuma. Usted necesitara ingresar credenciales de usuario validas y con los permisos requeridos. El servicio web ejecuta la autenticación por cabecera, esto quiere decir, que nuestras credenciales se enviaran dentro de la cabecera de la petición. La primera instancia de la clase Map construirá nuestra petición. Por ahora, como primer paso se especifica la dirección de destino (**Endpoint Address**).

Mas abajo veremos que se creara otro Map. Este segundo Map será el que contendrá las credenciales de usuario. Antes de que se almacene cualquier credencial dentro del Map, primero se deberá encriptar dicha información. El servicio mantiene en ejecución un algoritmo de cifrado asíncrono; esto quiere decir que usted deberá obtener una llave publica para poder efectuar el cifrado de sus credenciales.

Para finalizar la ejecución de este método, se encriptará las credenciales de usuario y se agregaran en el Mapa de la cabecera. Luego el primer mapa que creamos que contiene nuestro punto de destino, almacenara dicho mapa de la cabecera.

Por último, llamaremos al servicio, pasándole como parámetros la lista de archivos y el numero CR, y capturaremos la respuesta dentro de un objeto ServiceResponse. Veamos las configuraciones que usted necesitara hacer para poder cifrar sus credenciales.

## CIFRADO DE CREDENCIALES

Deberá crear otra interfaz para poder acceder al servicio de generación de llaves.



***FIGURA 18: Servicio para obtener una llave publica***

Este servicio nos proporciona una llave publica codificada en bytes[]. Para poder llamar al servicio, podemos seguir los mismos pasos expuestos anteriormente.



***FIGURA 19: Obteniendo una llave publica***

**Nota**: La URL en la imagen de arriba apunta al entorno de prueba. Cuando el entorno de producción esté disponible, deberá realizar cambios para apuntar a la URL correcta. La URL de producción se proporcionará cuando el entorno de producción esté listo.

Con la llave que hemos obtenido, vamos a poder cifrar nuestras credenciales para luego enviarlas al servidor. Esta es la clase sugerida para poder ejecutar nuestros cifrados. Tenga en cuenta que usted debera configurar su proveedor de algorithmos. El JDK de Java posee uno predeterminado, por lo que no necesariamente usted tendra que importar el API del BouncyCastle en su proyecto.



***FIGURA 20: Clase para cifrar datos***

Con estas configuraciones, ya usted podrá comenzar a consumir el servicio web.

# Formato de archivos

Los archivos que usted subira en el web service debe cumplir con el mismo formato del sistema anterior. Solo se permiten dos tipos de formato de archivo, **Txt** y **Xml**.

A continuacion se especificaran algunos ejemplos.

## ARCHIVOS DE TEXTO PLANO

## Orden que se deben seguir con este formato

**Se debe respetar el siguiente orden:**

1) Encabezado (DMCFORM).

2) Detalles (DTLS).

3) Articulo(s) (ITEM).

4) Factura(s) (INVOICE).

5) Vagon.

6) Cierre (DMCEFORM más la cantidad de lineas).



***FIGURA 21: Estructura del archivo .TXT***

**Nota:** La imagen de arriba muestra cómo debería verse el archivo txt.

## Reglas que se deben seguir con este formato

🡪 Nombre del archivo debe cumplir con el formato especificado anteriormente en el punto **2.3** de este documento:

**[Tipo de Documento]\_[Número CR]\_[Fecha]**

🡪 Los archivos no deben tener el mismo nombre.

🡪 No se puede eliminar espacios, aunque su sistema no tenga el campo.

🡪 Los campos deben ir en la secuencia indicada.

🡪 No se pueden dejar espacios en blanco delante o detrás del dato.

🡪 No es necesario cerrar con el pipe (|) el ultimo campo de las líneas.

## EJEMPLO PARA ENTORNO DE PRUEBA

1. **DMCE Entrada**:
   1. Archivo de texto (\*.txt) que contiene valor para *todos los campos*.



* 1. Archivo de texto (\*.txt) que contiene valor para *solo campos obligatorios.*



1. **DMCE Salida**:
   1. Archivo de texto (\*.txt) que contiene valor para *todos los campos*.



* 1. Archivo de texto (\*.txt) que contiene valor para *solo campos obligatorios.*

**

1. **DMCE Traspaso**:
   1. Archivo de texto (\*.txt) que contiene valor para *todos los campos*.



* 1. Archivo de texto (\*.txt) que contiene valor para *solo campos obligatorios.*

**

1. **DMCE Merma**:
   1. Archivo de texto (\*.txt) que contiene valor para *todos los campos*.



* 1. Archivo de texto (\*.txt) que contiene valor para *solo campos obligatorios.*

**

## EJEMPLO PARA ENTORNO DE PRODUCCIÓN

**Nota**: El entorno de producción no está listo. Cuando esté listo, se proporcionarán los archivos.

1. **DMCE Entrada**:
   1. Archivo de texto (\*.txt) que contiene valor para *todos los campos*.

* 1. Archivo de texto (\*.txt) que contiene valor para *solo campos obligatorios.*

1. **DMCE Salida**:
   1. Archivo de texto (\*.txt) que contiene valor para *todos los campos*.

* 1. Archivo de texto (\*.txt) que contiene valor para *solo campos obligatorios.*

1. **DMCE Traspaso**:
   1. Archivo de texto (\*.txt) que contiene valor para *todos los campos*.

* 1. Archivo de texto (\*.txt) que contiene valor para *solo campos obligatorios.*

1. **DMCE Merma**:
   1. Archivo de texto (\*.txt) que contiene valor para *todos los campos*.

* 1. Archivo de texto (\*.txt) que contiene valor para *solo campos obligatorios.*

## ARCHIVOS XML

## Orden que se deben seguir con este formato

**Se debe respetar el siguiente orden:**

1) Listado de formularios DMC  **<listadeformulariodmc total="1"  ( igual o mayor a “1”/ depende de la cantidad de formularios)**

2) Formulario DMC (Valor repetitivo)

3) Detalles del Formulario DMC **(<detallesdmc>/ abre y cierra)**

4) Listado de Articulos

**<listadearticulo total="1"> ( igual o mayor a “1” / depende de la cantidad de articulos)**

5) Articulos  **(<articulos>/ abre y cierra)**

6) Lista de Articulos  **(</listadearticulo> cierra lista)**

7) Listado de Facturas **<listadefacturas total="1"> ( igual o mayor a “1” / depende de la cantidad de Facturas)**

8) Listado de Vagones

**<listadevagon total=”1”> ( igual o mayor a “1”/ depende de la cantidad de Vagones)**

9) Contenedores **(<contenedor>/ abre y cierra)**

10) Lista de Contenedor **(</listadevagon> cierra lista)**

11) Formulario DMC **(Valor repetitivo / Cierra formulario)**

12) Lista de formulario **(</listadeformulariodmc> Cierra formulario)**

## Reglas que se deben seguir con este formato

🡪 Nombre del archivo debe cumplir con el formato especificado anteriormente en el punto **2.3** de este documento:

**[Tipo de Documento]\_[Número CR]\_[Fecha]**

🡪 Los archivos no deben tener el mismo nombre.

🡪 No se puede eliminar espacios, aunque su sistema no tenga el campo.

🡪 Los campos deben ir en la secuencia indicada.

🡪 No se pueden dejar espacios en blanco delante o detrás del dato

Para los archivos **XML** se puede tomar este schema en formato **XSD** comoguía para generar los xml:  


## EJEMPLO PARA ENTORNO DE PRUEBA

1. **DMCE Entrada**:
   1. Archivo de XML (\*.xml) que contiene valor para *todos los campos*.



* 1. Archivo de XML (\*.xml) que contiene valor para *solo campos obligatorios.*



1. **DMCE Salida**:
   1. Archivo de XML (\*.xml) que contiene valor para *todos los campos*.



* 1. Archivo de XML (\*.xml) que contiene valor para *solo campos obligatorios.*

**

1. **DMCE Traspaso**:
   1. Archivo de XML (\*.xml) que contiene valor para *todos los campos*.



* 1. Archivo de XML (\*.xml) que contiene valor para *solo campos obligatorios.*

**

1. **DMCE Merma**:
   1. Archivo de XML (\*.xml) que contiene valor para *todos los campos*.



* 1. Archivo de XML (\*.xml) que contiene valor para *solo campos obligatorios.*



## EJEMPLO PARA ENTORNO DE PRODUCCIÓN

**Nota**: El entorno de producción no está listo. Cuando esté listo, se proporcionarán los archivos.

1. **DMCE Entrada**:
   1. Archivo de XML (\*.xml) que contiene valor para *todos los campos*.

* 1. Archivo de XML (\*.xml) que contiene valor para *solo campos obligatorios.*

1. **DMCE Salida**:
   1. Archivo de XML (\*.xml) que contiene valor para *todos los campos*.

* 1. Archivo de XML (\*.xml) que contiene valor para *solo campos obligatorios.*

1. **DMCE Trespaso**:
   1. Archivo de XML (\*.xml) que contiene valor para *todos los campos*.

* 1. Archivo de XML (\*.xml) que contiene valor para *solo campos obligatorios.*

1. **DMCE Merma**:
   1. Archivo de XML (\*.xml) que contiene valor para *todos los campos*.

* 1. Archivo de XML (\*.xml) que contiene valor para *solo campos obligatorios.*

## TABLA DE VALORES

A continuacion se compartidra informacion de los valores que se estaran manipulando durante el proceso de la subida de archivos. Por favor, se recomienda respetarlas ya que el sistema durante el procesamiento de los archivos **validara los campos**. Por lo tanto, todos los valores que contendra el archivo **no deben ser valores invalidos ni campos vacios**.

**Nota:**

1. Para los archivos **TXT** se debe respetar la secuencia en la tabla de abajo indicada en la columna **No**.
2. Para los archivos **TXT** la secuencia en la columna **No** se reinicia para cada sección, para el **encabezado** la secuencia de la columna **No** comienza en **1,** para la **factura** la secuencia de la columna **No** comienza en **1,** para la **articulos** la secuencia de la columna **No** comienza en **1** y para la **sección** la secuencia de la columna **No** comienza en **1.**
3. Para los archivos **XML** se debe seguir el elemento la columna **Campo en XML**
4. Para los campos opcionales, si no desea enviar un valor en el archivo de **texto (\* .txt)**, mantenga el pipe “|” con valor blanco.
5. Para los campos opcionales, si no desea enviar un valor en el archivo **xml (\*.xml)**, mantenga en blanco con la apertura y cierre del elemento xml. **Por ejemplo**: <nombre></nombre>

**DEFINICION DE CAMPOS PARA ENCABEZADO**



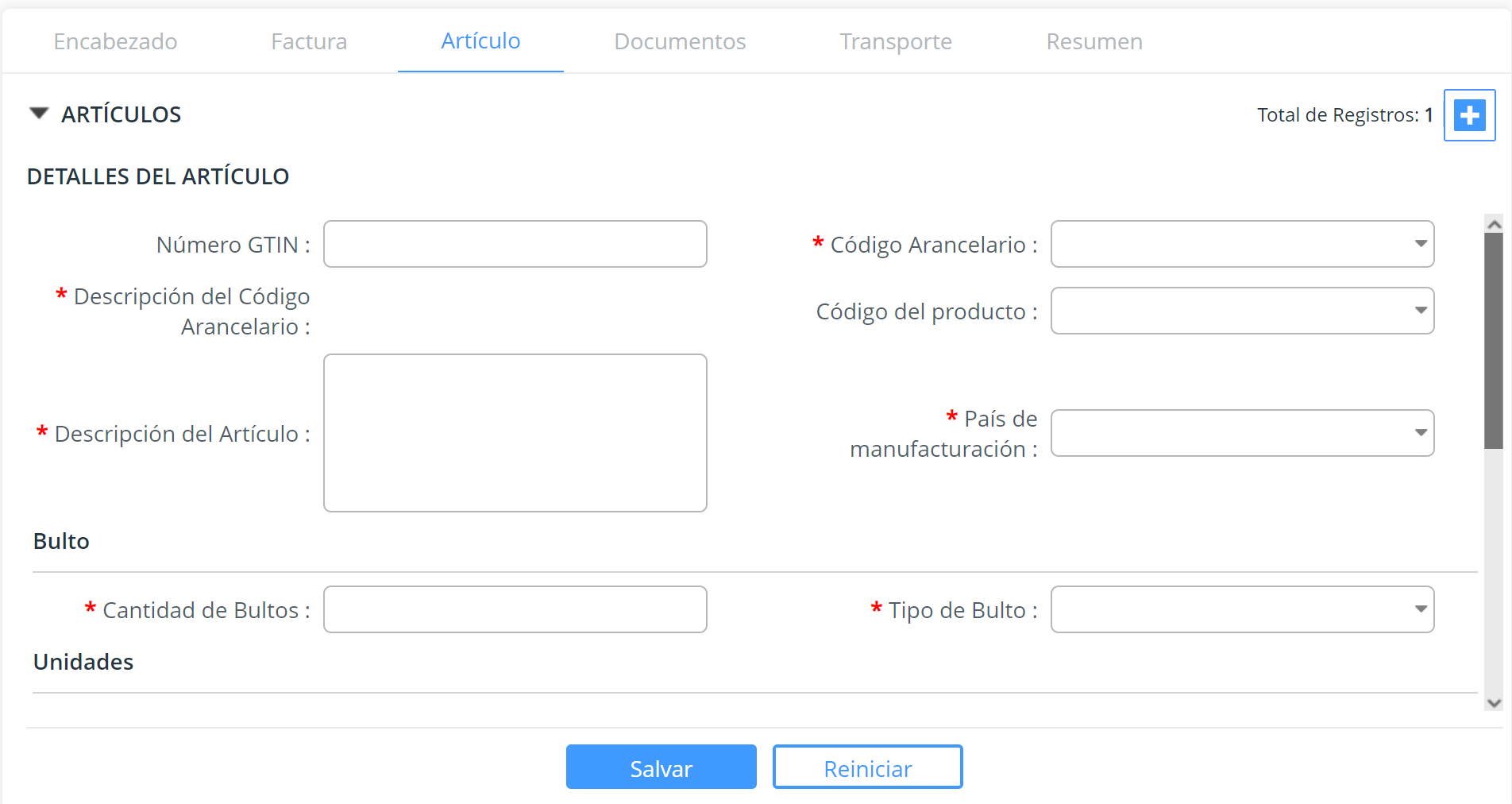
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Campo | Tipo de  Data | Maximo de Dato | Valor de Dato | Obligatorio | Valor en XML | Descripción |
|  | **Identificador de Formulario** | integer | 13 |  | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | Para el formato texto es un identificador de a que formulario pertenece esa línea.  Puede haber más de 1 formulario en 1 archivo de carga de interface. |
|  | **Record Type** | string | - | DTLS | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | RESPETAR CONSTANTE |
|  | **Tipo de Formulario** | string | - | • entry  • exit  • transfer  • writeoff | Sí | tipodeformulario | Se selecciona el tipo de declaración que se va a tramitar.  **Los valores en constantes son:**  entry = entrada; exit= salida; transfer = traspaso; writeoff =merma.  **FAVOR USAR LAS CONSTANTES SOLO EN INGLES EN LOS ARCHIVOS.** |
|  | **Declaración Anterior** | string | 20 |  | No | nodeformulariodereferencia | Puede ser usado en caso de ajuste/cancelación/reentrada para referirse al formulario original. Este campo es opcional en caso de no enviarlo se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar un contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Numero de Contacto de la Compañía** | string | 20 |  | Sí | nodecontactodelacompania | Número de teléfono de contacto de la compañía |
|  | **Medio de Transporte** | string | - | • Air  • Sea  • Land | Sí | mediodetransporte | Se selecciona cual es medio de transporte que utilizará la empresa.  **Los valores en constantes son**:  Air = Aire;  Sea = Mar; Land = tierra.  Para un DMCE Merma siempre será Land = Tierra.  **FAVOR USAR LAS CONSTANTES SOLO EN INGLES EN LOS ARCHIVOS.** |
|  | **Nombre de Transporte** | string | 100 |  | Sí | nombredelmediodetransporte | Campo obligatorio que debe ser utilizado  para agregar el nombre de la  embarcación o aerolínea. También se  puede agregar número de viaje o vuelo. |
|  | **Número del Documento de**  **Transporte** | string | 30 |  | Sí | nodeldocumentodetransporte | El usuario registra el número del conocimiento de embarque (BL, Guía Aérea y Carta de Porte), si desea agregar mas de un registro por favor separelos por coma, e.j: **2345,23333,14346**  Este campo solo es válido para: **Entrada** y **Salida.** No se debe de enviar en declaraciones de tipo **Traspaso** y **Descarte** para estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Compañía de Transporte** | string | 20 |  | No | transportefrio | Se Registra el codigo de la empresa de transporte.. e.j: **SP7-53** Este campo es opcional en caso de no enviarlo se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar un contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Nombre del Embarcante** | string | 128 |  | Sí | nombredelembarcante | Se registra el nombre del proveedor de la mercancía de origen.  Este campo solo es válido para: **Entrada.** No se debe de enviar en declaraciones de tipo **Salida,** **Traspaso** y **Descarte** para estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **País del Embarcante** | string | 100 |  | Sí | paisdelembarcante | Se selecciona el país donde se realiza el embarque de la mercancía.  **EL FORMATO DEBE SER DE 2 LETRAS TAL**  **CUAL APARECE EN LA SECCION “CODIGO**  **DE DESCRIPCION”** EJEMPLO: BRASIL = PR;  PANAMA = PA  Este campo solo es válido para: **Entrada.** No se debe de enviar en declaraciones de tipo **Salida,** **Traspaso** y **Descarte** para estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Tipo de Vagón** | string | **-** | • cntr  • bulkcargo  • luggage  • offshore  • sales | Sí | tipodevagon | Se selecciona el tipo de carga que se usará para la declaración según la via a utilizar.  **Los valores en constantes son:**  cntr = contenedor; bulkcargo = Carga en  bulto/sin contenedor; luggage = equipaje  de mano; offshore = ventas fuera del país;  **FAVOR USAR LAS CONSTANTES SOLO EN**  **INGLES EN LOS ARCHIVOS.** |
|  | **Puerto de Origen / ENTRADA**  **Puerto de Salida/ SALIDA Y MERMA** | string | **5** |  | Sí | puertodeinicio | **Declaracion de Entrada**: Se selecciona el Puerto de Origen de donde proviene la carga.  EL FORMATO DEBE IGUAL COMO LA SECCION “CODIGO DE DESCRIPCION”  Para una Entrada “**Puerto de Origen**”  **Declaracion de Salida**: Se selecciona el Puerto de Salida a dónde va la carga, de acuerdo a la lista configurada de puertos panameños.  EL FORMATO DEBE IGUAL COMO LA SECCION “CODIGO DE DESCRIPCION” Para una Salida “**Puerto de Salida**”  Este campo solo es válido para: **Entrada, Salida, Merma.** No se debe de enviar en declaraciones de tipo **Traspaso**, **Para de DMCE Merma** valor siempre será **CFZ**.  Para estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Puerto de Transbordo** / ENTRADA,SALIDA | string | **5** |  | No | puertodetransferencia | Se selecciona el puerto de transbordo siempre que la carga se descargue en un puerto antes de su llegada al destino final.  Debe ser igual a “Código de descripción”.  Este campo solo es válido y opcional para declaraciones de tipo: **Entrada, Salida.** No se debe de enviar en declaraciones de tipo **Traspaso y Merma**, para estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Puerto de Entrada** / ENTRADA **Puerto de Destino** / SALIDA | string | **5** |  | Sí | puertofinal | **Declaración de Entrada:** Se selecciona el Puerto de Entrada, de acuerdo a la lista configurada de puertos panameños.  EL FORMATO DEBE SER TAL CUAL APARECE  EN LA SECCION “CODIGO DE  DESCRIPCION”  Para una Entrada “**Puerto de Entrada**”  **Declaración de Salida:** Se selecciona el Puerto de Destino a dónde va la carga.  EL FORMATO DEBE SER TAL CUAL APARECE  EN LA SECCION “CODIGO DE DESCRIPCION”  Para una Salida “**Puerto de Destino**”  Este campo solo es válido para declaraciones de tipo: **Entrada, Salida.** No se debe de enviar en declaraciones de tipo **Traspaso y Merma**, para estos casos se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **País de Origen** /ENTRADA | string | 2 |  | Sí | paisdeorigen | Se registra el Pais de Origen de donde proviene la carga basado en el puerto de origen.  Este campo solo es válido para declaraciones de tipo: **Entrada** No se debe de enviar en declaraciones de tipo **Salida, Traspaso y Merma**, para estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **País de Destino/ SALIDA** | string | 2 |  | Sí | paisdedestino | Se registra el Pais de Destino a donde va la carga basado en el puerto de destino.  Este campo solo es válido para declaraciones de tipo: **Salida** No se debe de enviar en declaraciones de tipo **Entrada, Traspaso y Merma**, para estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Nombre del Consignado** | string | 200 |  | Sí/No | nombredelconsignado | Se registra el nombre de la empresa o persona a quién esta consignada la mercancía  **Solo es mandatario en DMCE Salida y opcional en DMCE Merma y Traspaso.**  PARA SALIDA, TRASPASO y MERMA.  DEJAR EL ESPACIO (PIPE) PARA ENTRADAS  Este campo es obligatorio para declaraciones de tipo: **Salida y Traspaso,** es opcional **Entrada** y **Merma**.  Para los casos donde es opcional se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **País del Consignado /** SALIDA y  MERMA | string | 2 |  | No | paisdelconsignado | Se selecciona el país del consignatario.  -EL FORMATO DEBE SER TAL CUAL APARECE  EN LA SECCION “CODIGO DE  DESCRIPCION”  Este campo solo es válido para declaraciones de tipo: **Salida** y **Merma,** para declaraciones de tipo **Merma** no debe de enviarse ya que el valor siempre sera **PA.**  No se debe de enviar en declaraciones de tipo**, Entrada y Traspaso**, para estos casos donde no debe enviarse se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Número del Documento de**  **Control /** SALIDA Y MERMA | string | 20 |  | No | nodeldocumentodecontrol | Se registra los números de documentos de control. Este campo solo es válido y opcional para declaraciones de tipo: **Salida, Merma.** No se debe de enviar en declaraciones de tipo **Entrada, Traspaso y** para estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Número de Registro de la**  **Compañía Receptora /** Clave de operación. | string | 20 |  | Sí | noderegistrodeltransferidor | Se registra la **clave de operación** de la compañía a transferir **‐ Requerido solo para formulario de**  **Traspaso.**  ‐ Debe ser una compañía registrada  en el sistema DMCE.  Este campo solo es válido para declaraciones de tipo: **Traspaso** No se debe de enviar en declaracioneslos **otros tipos** estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Destino de la Merma** | string | 128 | • landfill  • incinerator | Sí | destinaciondeldescarte | Se selecciona el destino de la merma ya sea el incinerador o el vertedero.  Las constantes para este campo son: **landfill** = vertedero, **incinerator** = incinerador.  Este campo solo es válido para declaraciones de tipo: **Merma** No se debe de enviar en declaracioneslos **otros tipos** estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Representante de la Compania** | string | 20 |  | No | representantedelacompania | ‐ Requerido solo para formulario de  Merma.  Este campo solo es válido para declaraciones de tipo: **Merma** No se debe de enviar en declaracioneslos **otros tipos** estos casos debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Consumo Propio** | string | 1 | • Y  • N | Sí | industriadeconsumo | Se utiliza para la entrada de mercancía para uso interno de la empresa  Y= SI  N=NO (usar constante tal cual, esto es solo  traducción)  Este campo solo es válido para declaraciones de tipo: **Entrada** Para los otros tipos **otros tipos** se debe de enviar siempreel valor **N** |
|  | **Re‐Importación** | string | 1 | • Y  • N | Sí | industriadereimportacion | Indicador de reimportacion  Y= SI  N=NO (usar constante tal cual, esto es solo  traducción)  Este campo solo es válido para declaraciones de tipo: **Entrada** Para los otros tipos **otros tipos** se debe de enviar siempreel valor **N** |
|  | **Recinto Publico** | string | 12 | - cfzph  - cfzpharen | No | areadeesperapublica | cfzph = CFZ PH Area - C  cfzpharen = CFZ PH Area – N  Este campo solo es válido para declaraciones de tipo: **Entrada** No se debe de enviar en declaracioneslos **otros tipos,** en estos casos se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Seguro / Entrada** | decimal | 10,2 |  | No | cargoporseguro | Se registra Seguro Total de la mercancía que debe indicar el usuario en las entradas.  Este campo solo es válido y opcional para declaraciones de tipo: **Entrada** No se debe de enviar en declaraciones delos **otros tipos,** en estos casos se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Flete / Entrada** | decimal | 10,2 |  | Sí | cargoporflete | Se registra el Flete Total de la carga que debe indicar el usuario en las entradas.  Este campo solo es válido para declaraciones de tipo: **Entrada** No se debe de enviar en declaracioneslos **otros tipos,** en estos casos se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Descuento / Entrada, Salida** | decimal | 10,2 |  | No | descuento | Se registra el descuento aplicado por la empresa sobre los artículo en caso que lo haya.  Este campo solo es válido y opcional para declaraciones de tipo: **Entrada, Salida.** No se debe de enviar en declaraciones de tipo **Traspaso y Merma**, para estos casos se debe cumplir con las siguientes reglas **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Otros Gastos / Entrada, Traspaso, Salida** | decimal | 10,2 |  | No | otrosgastos | Se registra de existir los otros gastos que indica el usuario.  Este campo solo es válido y opcional para declaraciones de tipo: **Entrada, Salida y Traspaso** No se debe de enviar en declaraciones de tipo **Merma**, para estos casos se debe cumplir con las siguientes reglas: **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Comentarios del Formulario** | string | 128 |  | No | comentariosdelformulario | Información adicional de la declaración que detalla la empresa. Este campo es opcional en este caso se debe cumplir con las siguientes reglas: **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Total Bulto** | decimal | 10,2 |  | Sí | totalbulto | valor del bulto total de los artículos |
|  | **Nombre de contacto de la compañía** | string | 20 |  | Sí | personadecontactodelacompania | Nombre de contacto de la compañía |
|  | **Área de Declaración** | string | 2 | * 0 * 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 * 11 * 12 * 13 | No | areadedeclaracion | Se selecciona el área en donde está ubicada la empresa dentro de la Zona Libre de Colón.  Los valores en constantes son:  0 = campo Vacío,  1 = France Field,  2 = Coco Solito,  3 = Zoliacion, consultoría y Capacitacion,  4 = Mit Logistic Park,  5 = Crystal Park,  6 = Logistic Services,  7 = Davis,  8 = Best Tires Corp,  9 = Hamzi S.A,  10 = Koyo Latin America S.A,  11 = Neptune Lines Panama Inc,  12 = Popular Zona Libre S.A,  13 = Wallenius Wilhelmsen,  Este campo es opcional en este caso se debe cumplir con las siguientes reglas: **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Indicador de liquidación** | string | 1 | • Y  • N | Sí | indicatordeliquidacion | Si este DMCE pasará por el proceso de liquidación de aduanas |
|  | **Razón de la Merma** | string | 128 |  | Sí | razon | Se registra el porque de la merma Este campo solo es obligatorio para las declaraciones de **Merma,** para **los otros** tipos de declaración este campo no se debe llenar, en estos casos se debn de seguir las siguientes reglas:  **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |

**DEFINICION DE CAMPOS PARA FACTURA**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Campo | Tipo de  Data | Maximo de Dato | Valor de Dato | Obligatorio | Valor en XML | Descripción |
|  | **Identificador de Formulario** | integer | 13 |  | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | Para el formato texto es un identificador de a que formulario pertenece esa línea.  Puede haber más de 1 formulario en 1 archivo de carga de interface. |
|  | **Record Type** | string | - | INVOICE | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | RESPETAR CONSTANTE |
|  | **Fecha** | date | 10 | yyyy-MM-dd e.g. 2008-08-17 | Sí | fechadelafactura | Se registra la fecha que aparece en la factura  Por favor seguir el formato: **yyyy-MM-dd** en caso de no seguirlo el sistema no tomara la fecha de manera correcta. |
|  | **Numero** | string | 20 |  | Sí | nodefactura | Se registra el numero de la transacción comercial realizada entre el comprador y el vendedor. |
|  | **Nombre de la compañía** | string | 100 |  | Sí | companiaquefactura | El sistema coloca en automático el nombre de la empresa consignante.  Este campo es **Nombre de Proveedor** en pantalla de aplicación. |
|  | **Nombre del Consignado** | string | 100 |  | Sí | consignado | Se registra el nombre de la empresa consignante. |
|  | **Tipo de Recibo** | string | 3 | • 1  • 2  • 3  • 5  • OTH | Sí | tipoderecibo | Forma de pago pactada entre el comprador y el vendedor.  Las constantes son:  1 = efectivo,  2 = cheque,  3 = tarjeta de crédito,  5 = transferencia  OTH = Otros. |
|  | **Moneda** | string | 3 |  | Sí | monedadelafactura | USAR CODIGO DE MONIEDA DE 3  LETRAS. EJEMPLO: DOLARES = USD O  VERIFICAR EN “CODIGO DE DESCRIPCION” |
|  | **Cantidad** | decimal | 16,2 |  | Sí | cantidaddelafactura | Se registra el valor de venta de la mercancía en la moneda de la factura |
|  | **Cantidad de la Factura en Dólares Americanos** | decimal | 12,2 |  | Sí | cantidadendolaresamericanos | Se registra el valor de venta de la mercancía en la moneda de la factura en USD(Dolares Americanos) |
|  | **Incoterms** | string | - | 1  2  3  4  5  6  7  8  9  10  11 | No | incoterms | El usuario seleccione la forma de pago pactada. Las constantes son:  1 EXW – En Fábrica  10 CFR – Costo y Flete  11 CIF – Costo, seguro y flete  2 FCA – Franco Porteador  3 CPT – Transporte pagado hasta  4 CIP – Transporte y seguro pagados hasta  5 DAP – Entregada en lugar  6 DPU – Entregada en lugar descargada  7 DDP – Entregada derechos pagados  8 FAS – Franco al costado del buque  9 FOB – Franco a bordo  Este campo es opcional en este caso se debe cumplir con las siguientes reglas: **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |

**DEFINICION DE CAMPOS PARA ARTÍCULOS**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Campo | Tipo de  Data | Maximo de Dato | Valor de Dato | Obligatorio | Valor en XML | Descripción |
|  | **Identificador de Formulario** | integer | 13 |  | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | Para el formato texto es un identificador de a que formulario pertenece esa línea.  Puede haber más de 1 formulario en 1 archivo de carga de interface. |
|  | **Record Type** | string | - | ITEM | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | RESPETAR CONSTANTE |
|  | **Código Arancelario** | string | 20 |  | Sí | codigoarancelario | Dígitos de código arancelario sin puntos.  Estos códigos pueden ser consultados en  “código de descripción” o libro de aranceles de la autoridad de aduanas actualizado. |
|  | **Código de producto** | string | 15 |  | No | codigodeproducto | En este campo se debe colocar el codigo de producto que se ha paremetrizado en el sistema.  Este campo es opcional en este caso se debe cumplir con las siguientes reglas: **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Descripción del Artículo** | string | 128 |  | Sí | descripciondelarticulo | Descripción detallada del producto que tiene que ser registrada por el usuario. |
|  | **País de Manufactura** | string | 2 |  | Sí | paisdemanufactura | Se registra el código del pais de fabricación de el articulo este codigo debe ser de 2 caracteres por ejemplo:: BRASIL = BR |
|  | **Bulto / Cantidad** | decimal | 10 |  | Sí | cantidadebulto | El usuario registra la cantidad de bultos que corresponden al artículo |
|  | **Bulto / Tipo** | string | 5 |  | Sí | tipodebulto | Se registra el tipo de bulto del artículo  El tipo debe de existir en el mantenimiendo de codigos en el sistema. |
|  | **UDM (Cantidad)** | decimal | 10 |  | Sí | unidaddecantidaddelaunidaddemedida | CANTIDAD DE UNIDAD DE MEDIDA,  EJEMPLO SI LOS LICORES EL ARANCEL LO  DETERMINA EN “LITROS” SE DEBE PONER LA  CANTIDAD DE LITROS QUE VIENE EN EL  BULTO (ESTO ES DISPOCISION DE  CONTRALORIA, NO DE ZLC O EL  CONSORCIO) |
|  | **Unidades Sueltas** | decimal | 10 |  | No | unidadsueltaunidaddemedida | El usuario registra la cantidad de unidades sueltas que corresponde a la carga.  Este campo es opcional en este caso se debe cumplir con las siguientes reglas: **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Artículo de Muestra** | string | 1 | • Y  • N | No | industriademuestra | El usuario en el caso que sea muestra debe seleccionar:  Y = SI  N = NO  **RESPETAR LAS CONSTANTES**  Este campo es opcional en este caso se debe cumplir con las siguientes reglas: **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Peso Bruto** | decimal | 10.2 |  | Sí | unidadbruta | El usuario debe indicar el peso del artículo en kilogramos |
|  | **Valor en Dólares Americanos** | decimal | 12.2 |  | Sí | valordelarticulo | Se registra el valor del artículo en dólares americanos USD, por línea del producto. |
|  | **Descripción del Código Arancelario** | string | 128 |  | Si |  |  |

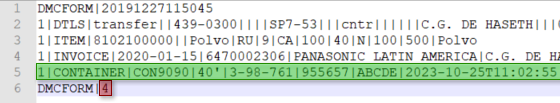
**Sección de Transporte**

**Es importante aclarar que los campos de la ventana de TRANSPORTE cambian dependiendo del tipo de carga de ventana encabezado.**

**En seguida va encontrar 5 secciones con campos y valores para los mismos.**

**Esta seccion es opcional solo para DMCE Trespaso. Es mandatario para otra tipo de DMCE. Si usted no desea enviar esta seccion debe seguir las siguientes reglas:  
Archivo de texto (\* .txt):**

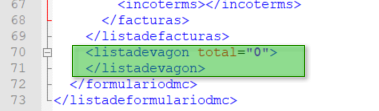
**Elimine la fila que indica los datos de transporte** y **reduzca el numero que indica la cantidad de filas que se estan mandando**

  
**Ejemplo con los datos de transporte:**  
  
**Ejemplo sin los datos de transporte:**



**Archivo de xml (\* .xml):**

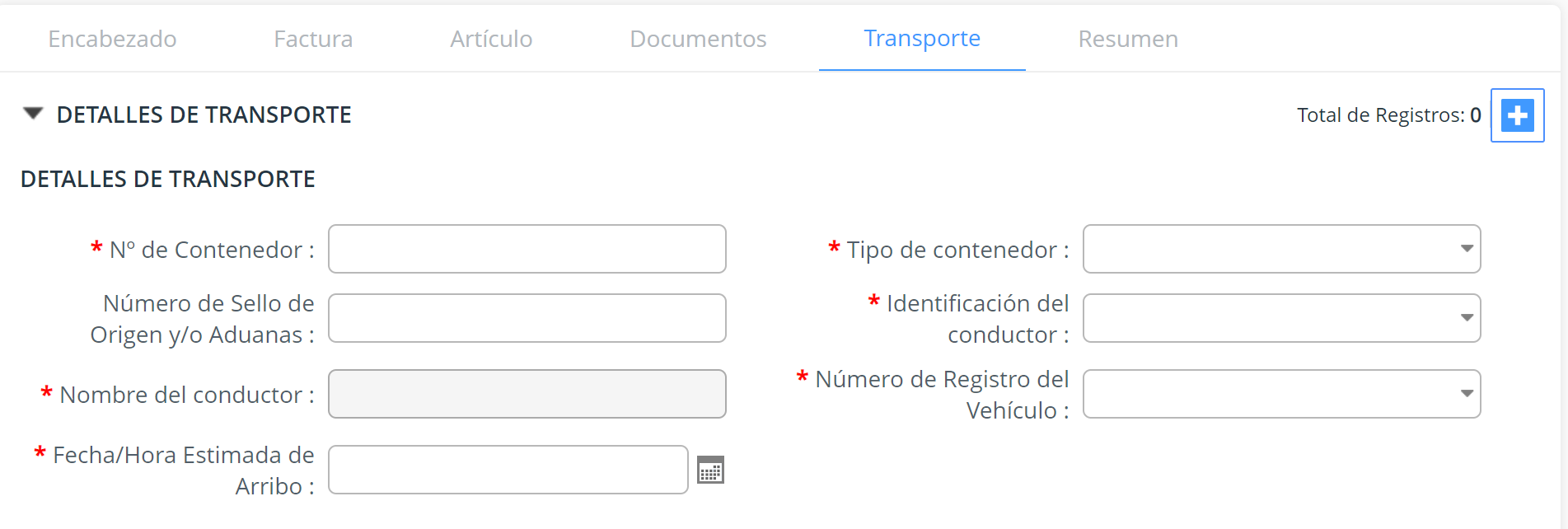
**Mantenga el elemento en blanco abriendo y cerrando la etiqueta listadevagon sin colocar contenido ejemplo, ejemplo:**

****

Ejemplo con los datos de transporte:  
  
Ejemplo sin los datos de transporte:

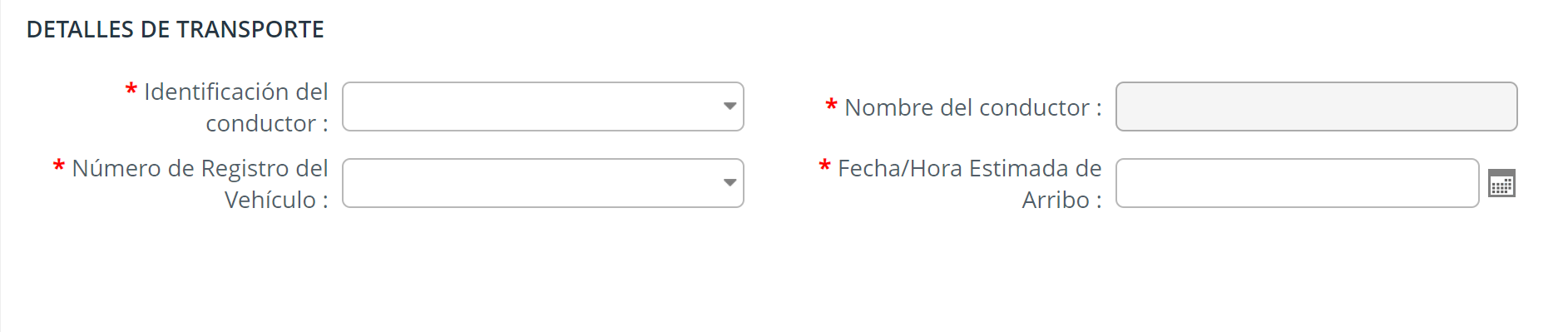
****

1. **DEFINICION DE CAMPOS PARA TIPO DE CARGA CONTEDORES**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Campo | Tipo de  Data | Maximo de Dato | Valor de Dato | Obligatorio | Valor en XML | Descripción |
|  | **Identificador de Formulario** | integer | 13 |  | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | Para el formato texto es un identificador de a que formulario pertenece esa línea.  Puede haber más de 1 formulario en 1 archivo de carga de interface. |
|  | **Record Type** | string | - | CONTAINER | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | **RESPETAR CONSTANTE** |
|  | **Numero de Contenedor** | string | 20 |  | Sí | nodecontenedor | Dígitos de código arancelario sin puntos.  Estos códigos pueden ser consultados en  “código de descripción” o libro de aranceles de la autoridad de aduanas actualizado. |
|  | **Tipo de Contenedor** | string | - | - 20'  - 40'  - 45’  - 48’  - 53’  - 10’  - HQ 45'  - HQ 40'  - Frozen 20'  - Frozen 40' | Sí | tipodecontenedor | El usuario registra de número de contenedor que se va utilizar en la declaración  **Este campo es válido para el tipo de carga:**  **Contenedor**  Las constantes son: 45’ = 45 Footer Container, 48’ = 45 Footer Container, 53’ = 53 Footer Container, 10’ = 10 Footer Container, 20’ = 20 Footer Container, 40’ = 40 Footer Container, HQ 45’ = 45 Footer Container, Frozen 20’ = 20 Footer Container, Frozen 40’ = 40 Footer Container, HQ 40’ = 40 Footer Container  RESPETAR CONSTANTES |
|  | **Identificación del conductor** | string | 20 |  | Sí | iddelconductor | Poner el número de cédula del Conductor  Este campo es válido para el tipo de carga:  **Contenedor, Bulto**  El campo Nombre de conductor se llena automáticamente basándose de la Identificación del conductor. |
|  | **Número de Registro del vehículo** | string | 20 |  | Sí | noderegistrovehicular | Poner el número de la identificación del vehículo que describe para Panamá como Placa del Vehículo. |
|  | **Sello de Aduanas** | string | 20 |  | No | selloaduanal | Se registra el número de sello de origen que es colocado por la empresa embarcadora y/o el número sello de Aduanas que corresponde a el Tránsito Interno o Marchamo.  Este campo es opcional en este caso se debe cumplir con las siguientes reglas: **Archivo de texto (\* .txt):** Mantenga el pipe “|” con valor blanco  **Archivo de xml (\* .xml):** Mantenga el elemento en blanco abriendo y cerrando la etiqueta sin colocar contenido, ejemplo: **<ejemplo></ejemplo>** |
|  | **Fecha y Hora de Entrada** | date time |  | yyyy-MM-ddTHH:mm:ss  e.g. 2008-08-17T09:00:00 | Sí | fechahoraestimada | ES NECESARIO RESPETAR EL FORMATO:  yyyy-MM-ddTHH:mm:ss  ejemplo : 2008-08-17T09:00:00 |

1. **DEFINICION DE CAMPOS PARA TIPO DE CARGA EN BULTO**



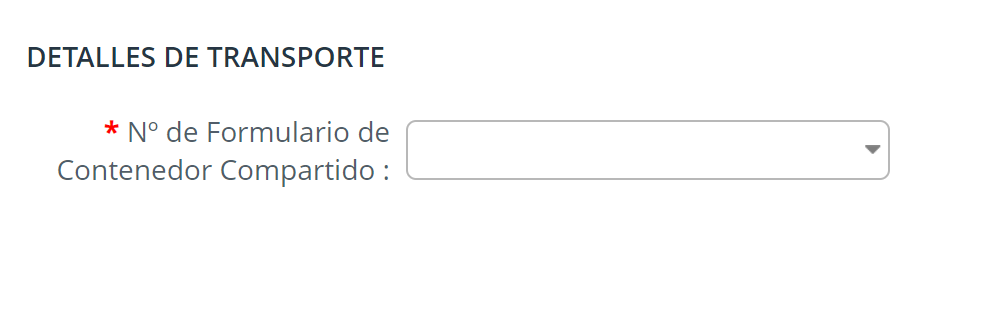
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Campo | Tipo de  Data | Maximo de Dato | Valor de Dato | Obligatorio | Valor en XML | | Descripción |
|  | **Identificador de Formulario** | integer | 13 |  | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | Para el formato texto es un identificador de a que formulario pertenece esa línea.  Puede haber más de 1 formulario en 1 archivo de carga de interface. | |
|  | **Record Type** | string | - | BULKCARGO | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | RESPETAR CONSTANTE | |
|  | **Identificación del conductor** | string | 20 |  | Sí | iddelconductor | Poner el número de cédula del Conductor  Este campo es válido para el tipo de carga:  **Contenedor, Bulto**  El campo Nombre de conductor se llena automáticamente basándose de la Identificación del conductor. | |
|  | **Número de Registro del vehículo** | string | 20 |  | Sí | noderegistrovehicular | Poner el número de la identificación del vehículo que describe para Panamá como Placa del Vehículo. | |
|  | **Fecha y Hora de Entrada** | date time |  | yyyy-MM-ddTHH:mm:ss  e.g. 2008-08-17T09:00:00 | Sí | fechahoraestimada | Se registra fecha y hora de entrada/salida estimada por la empresa ES NECESARIO RESPETAR EL FORMATO:  yyyy-MM-ddTHH:mm:ss  ejemplo : 2008-08-17T09:00:00 | |

1. **DEFINICION DE CAMPOS PARA TIPO DE CARGA EQUIPAJE ACOMPAÑADO**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Campo | Tipo de  Data | Maximo de Dato | Valor de Dato | Obligatorio | Valor en XML | Descripción |
|  | **Identificador de Formulario** | integer | 13 |  | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | Para el formato texto es un identificador de a que formulario pertenece esa línea.  Puede haber más de 1 formulario en 1 archivo de carga de interface. |
|  | **Record Type** | string | - | HANDLUGGAGE | Sí |  | RESPETAR CONSTANTE |
|  | **Fecha y Hora de Entrada** | date time | yyyy-MM-ddTHH:mm:ss  e.g. 2008-08-17T09:00:00 |  | Sí | fechahoraestimada | ES NECESARIO RESPETAR EL FORMATO:  yyyy-MM-ddTHH:mm:ss  ejemplo : 2008-08-17T09:00:00 |

1. **DEFINICION DE CAMPOS PARA TIPO DE CARGA CONTENEDOR COMPARTIDO (ENTRADA, SALIDAS Y MERMAS)**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Campo | Tipo de  Data | Maximo de Dato | Valor de Dato | Obligatorio | Valor en XML | Descripción |
|  | **Identificador de Formulario** | integer | 13 |  | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | Para el formato texto es un identificador de a que formulario pertenece esa línea.  Puede haber más de 1 formulario en 1 archivo de carga de interface. |
|  | **Record Type** | string | - | SHARED | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | RESPETAR CONSTANTE |
|  | **Numero de Formulario** | string | 20 |  | Sí | noformulariodecontenedor | EL FORMULARIO DEBE EXISTIR COMO  ENTRADA DE CONTENDOR COMPARTIDO Y  DEBE HABER SIDO SELECCIONADO DE ESTA  MANERA (VER INSTRUCCIONES INICIALES  DE CONTENEDOR COMPARTIDO) |

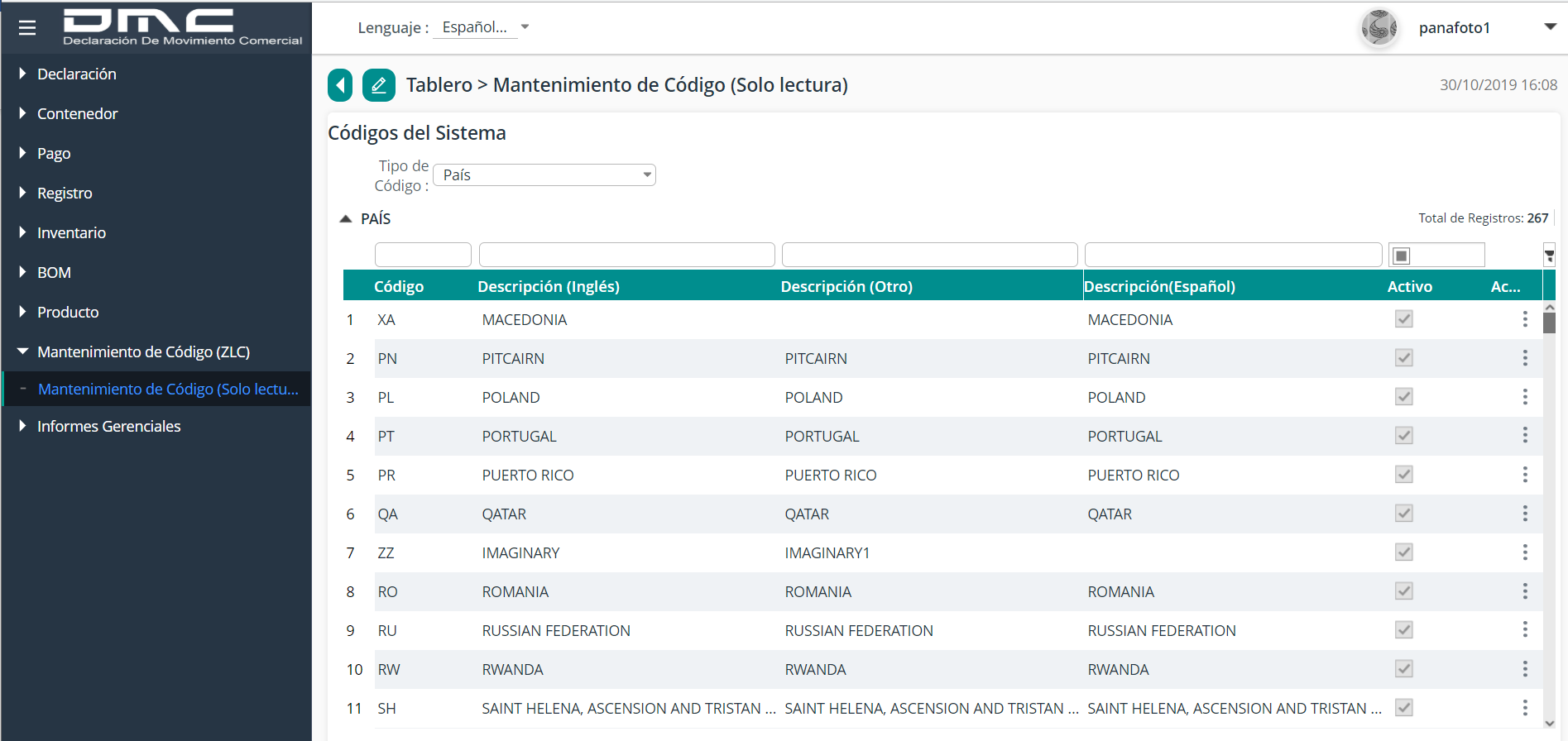
1. **DEFINICION DE CAMPOS PARA TIPO DE CARGA VENTAS FUERA DEL PAIS**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Campo | Tipo de  Data | Maximo de Dato | Valor de Dato | Obligatorio | Valor en XML | Descripción |
|  | **Identificador de Formulario** | integer | 13 |  | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | Para el formato texto es un identificador de a que formulario pertenece esa línea.  Puede haber más de 1 formulario en 1 archivo de carga de interface. |
|  | **Record Type** | string | - | OFFSHORE | Sí | Este campo solo es valido para los archivos enviados en formato **TXT** | RESPETAR CONSTANTE |
|  | **Fecha y Hora de Entrada** | date time | yyyy-MM-ddTHH:mm:ss  e.g. 2008-08-17T09:00:00 |  | Sí | fechahoraestimada | ES NECESARIO RESPETAR EL FORMATO:  yyyy-MM-ddTHH:mm:ss  ejemplo : 2008-08-17T09:00:00 |

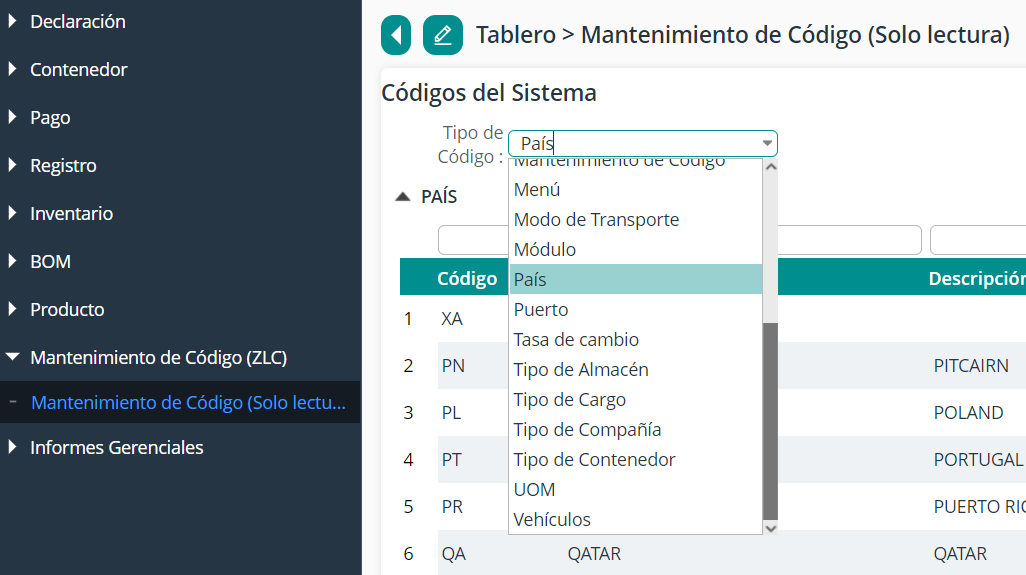
## Ver valores de codigos dentro del sistema

A continuación, vamos a mostrar cómo podemos consultar la información que vayamos a utilizar dentro del módulo de declaración. Esta información puede ser accedida desde la página haciendo clic en el modulo **Mantenimiento de Código (ZLC) => Mantenimiento de Código (Solo lectura)**.



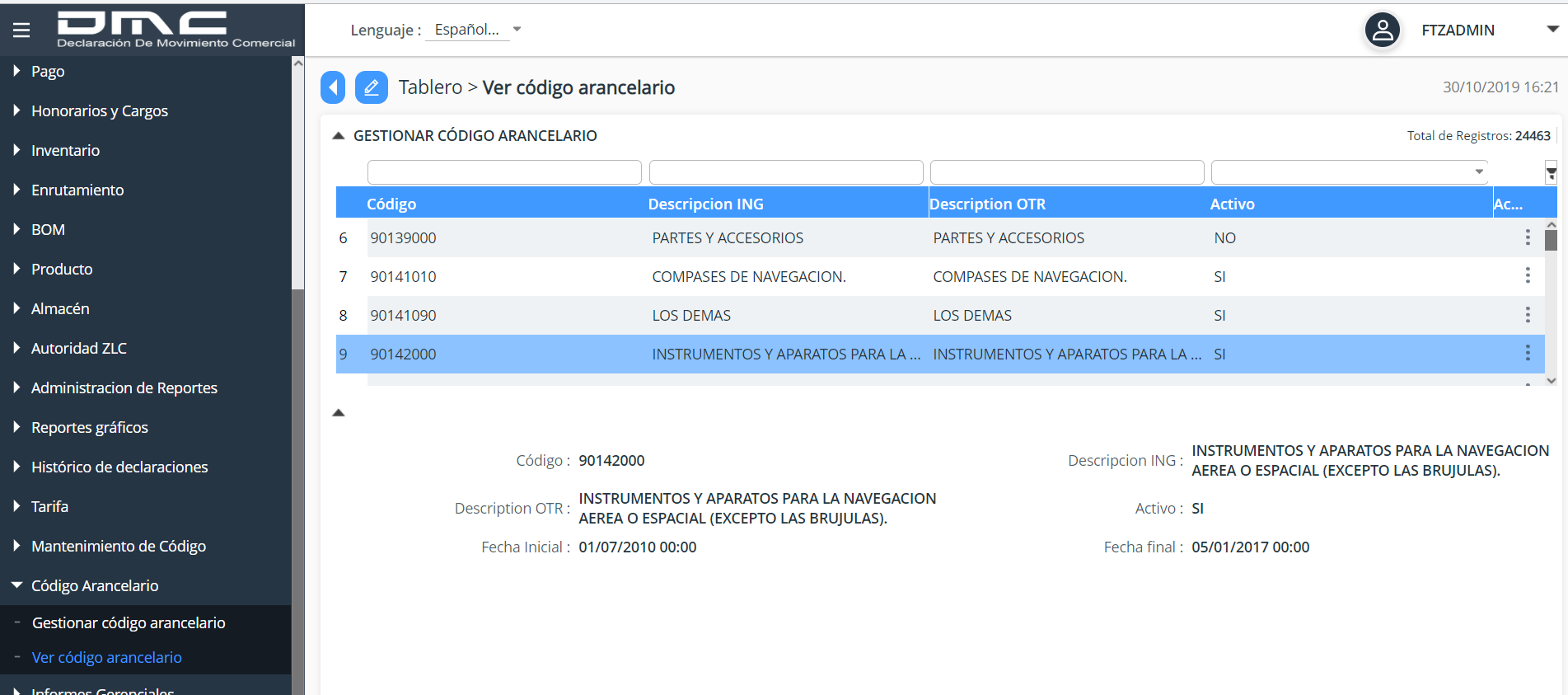
***FIGURA 22: Mantenimiento de Código***

La pagina presentaran datos iniciales que pueden ser cambiados mas adelante. Usted para seleccionar el tipo de codigo que desea buscar, debera abrir la lista y seleccionar el tipo de codigo deseado:



***FIGURA 23: Dropdown list para seleccionar el tipo de código a consultar***

El único tipo de código que no aparece en la lista es el código arancelario. Este debe accederse desde otro módulo de la página. Se debe seleccionar el modulo **Código Arancelario => Ver código arancelario**:



***FIGURA 24: Código Arancelario***

## Ver estado de procesamiento de los archivos subidos

Para poder ver el estado de procesamiento de los archivos subidos, usted deberá iniciar sesión con un usuario administrador de la empresa, o bien, un usuario con el rol **FUSERV** asignado. Para abrir esta tabla, diríjase al modulo de **Informes Gerenciales** -> **Procesamiento de Archivos**:



***FIGURA 25: Ver estados de procesamiento de los archivos***

Dentro de los registros, si fuese el caso de que algún archivo no se pudo procesar correctamente, usted deberá hacer doble clic sobre el registro para poder visualizar los errores que se produjeron durante el procesamiento.

* **Deberá solicitar la cuenta de usuario de administración de empresa a la administración de la Zona Libre de Colón para poder visualizar esta pantalla.**
* **El administrador de la empresa deberá crear el usuario con el rol FUSERV para poder consumir el servicio web y poder visualizar esta pantalla.**