

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**Sistema de control de Facturación
para empresa naviera Hapag-Lloyd Chile Spa**

Andrés Alejandro Kröll Guzmán

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

Diciembre 2018

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

Sistema de control de Facturación para empresa naviera Hapag-Lloyd Chile Spa

Andrés Alejandro Kröll Guzmán

Profesor Guía: Silvana Roncagliolo
Profesor Co-referente: Aldo Migliaro

Carrera: Ingeniería de Ejecución en Informática

Diciembre 2018

Resumen

En este informe se mostrarán las etapas que comprenden el desarrollo de un software que dará apoyo a la facturación automática de la empresa Hapag-Lloyd Chile Spa. El objetivo de este proyecto es desarrollar un programa que apoye el proceso de facturación automáticamente dentro de la plataforma FIS para dar apoyo y optimizar el tiempo de ejecución.

El paradigma utilizado para el desarrollo de este proyecto es el proceso evolutivo, que consta de realizar varias iteraciones en las cuales cada una desarrolla una parte del programa, se implementa, se entrega al cliente y luego se repite el proceso agregando un nuevo desarrollo del sistema, hasta llegar al producto final deseado por el cliente.

El producto final de este proyecto es un programa Excel que contiene una programación realizada en el entorno Visual Basic Application (propio de Excel), el cual realiza la tarea de procesar las facturas pagadas de distintas áreas de manera automática, además de enviar los resultados finales cuando termina su ejecución en un correo de Outlook.

Abstract

This report will show the stages that comprise the development of software that will support the automatic invoice of the company Hapag-Lloyd Chile Spa. The objective of this project is to develop a program that supports the invoice process automatically within the FIS platform to give support and optimize the execution time.

The paradigm used for the development of this project is the evolutionary process, which consists of performing several iterations in which each develops a part of the program, is implemented, delivered to the client and then the process is repeated adding a new development of the system, until reaching the final product desired by the client.

The final product of this project is an Excel program that contains a program made in the Visual Basic Application environment (Excel itself), which performs the task of automatically processing invoices paid from different areas, in addition to sending the final results when its execution ends in an Outlook mail.

Índice

Lista de Figuras	1
1 Introducción.....	2
2 Problemática y Solución.....	3
2.1 Descripción del problema.....	3
2.1 Solución propuesta	3
2.2 Objetivos.....	4
2.2.1 Objetivo General.....	4
2.2.2 Objetivos Específicos	4
2.3 Metodología a usar	4
3 Desarrollo	5
3.1 Modelo de Negocio	5
3.1.1 Reglas de Negocio.....	5
3.1.2 Modelo Actual de negocio.....	5
3.2 Iteración 1: Importación	8
3.2.1 Proceso	8
3.2.2 Implementación	16
3.3 Iteración 2: Exportación	17
3.3.1 Proceso	17
3.3.2 Implementación	22
3.4 Iteración 3: Bolivia	23
3.4.1 Proceso	23
3.4.2 Implementación	28
4 Conclusiones.....	29
5 Referencia Bibliográficas	30
Anexos	
A: Pseudocódigo Importación	
B: Pseudocódigo Exportación	
C: Pseudocódigo Bolivia	

Lista de Figuras

FIGURA 1 BPMN PROCESO DE NEGOCIO.....	5
FIGURA 2 BASE DE DATOS “GATE-IN”.....	7
FIGURA 3 DFD PROCESO DE IMPORTACIÓN.....	9
FIGURA 4 VENTANA “S7230”.....	10
FIGURA 5 VENTANA “S8100” CON ERROR.....	11
FIGURA 6 VENTANA “S8100”.....	12
FIGURA 7 VENTANA “F0030”.....	13
FIGURA 8 VENTANA “F0020” PESTAÑA “POSITIONS”.....	14
FIGURA 9 VENTANA “F0020” PESTAÑA “REVENUES”.....	15
FIGURA 10 PLANILLA DE IMPORTACIÓN.....	16
FIGURA 11 DFD PROCESO DE EXPORTACIÓN.....	18
FIGURA 12 VENTANA “S8100”.....	19
FIGURA 13 VENTANA “S8100” CON DATO.....	20
FIGURA 14 VENTANA “S8100” CON ERROR.....	20
FIGURA 15 PLANILLA “EXPO”.....	22
FIGURA 16 VENTANA “S7230”.....	24
FIGURA 17 VENTANA “S8100”.....	25
FIGURA 18 VENTANA “F0030”.....	26
FIGURA 19 VENTANA “F0020” PESTAÑA “CLAUSES”.....	27
FIGURA 20 PLANILLA “IMPO” CON DATOS DE BOLIVIA.....	28

1 Introducción

En la actualidad vivimos en un mundo globalizado en donde todas las partes están conectadas de una u otra manera, el ejemplo más claro es internet, medio por el cual se pueden conectar o enviar datos a través de todo el mundo. Pero para el envío de objetos físicos de un continente a otro, existen otros métodos, como las vías marítimas y aéreas.

Las empresas navieras se encargan de transportar los objetos de un continente a otro mediante la movilización de barcos a través del mar. Este método de transporte es el método más utilizado, debido a que los barcos tienen la capacidad de llevar grandes cantidades de contenedores apilados y distribuidos dentro del mismo, además de que los contenedores pueden tener una gran cantidad de objetos diferentes, desde ropa hasta alimentos refrigerados.

Hapag –Lloyd es una empresa dedicada al rubro de transporte marítimo desde 1850 e instalada en Chile hace más de 30. Mediante sus barcos y sus sedes repartidas por todo el mundo, tienen la capacidad de transportar grandes cantidades de contenedores a la mayoría de los puertos del mundo.

Con una empresa tan grande a nivel mundial y con un manejo de abundante información, se hace indispensable un sistema que apoye al proceso para cada país. El problema que abarca este proyecto corresponde a la sede de Chile de Hapag-Lloyd. La dificultad que tiene la sede es la de no contar con un sistema automatizado para procesar las facturas que ya han sido pagadas y reflejarlas dentro de su base de datos “FIS”. Además de recibir diariamente una cantidad grande de facturas y siendo el tiempo un factor tan importante para atraer y mantener a la clientela, se vuelve casi imprescindible contar con un sistema automatizado.

En el siguiente informe se abarcará sobre el desarrollo de un sistema de control de facturación que apoya los procesos de las áreas que reciben mayor cantidad de facturas, como son los casos de importación y exportación.

2 Problemática y Solución

2.1 Descripción del problema

Hapag-Lloyd Chile es una empresa naviera que se dedica al transporte de contenedores mediante sus barcos que llegan a los puertos del país desde todo el mundo, debido a que es una empresa con una gran cantidad de clientes e información de contenedores, esta recibe diariamente más de mil facturas sobre los pagos que se realizan.

El proceso por el cual debe pasar cada factura es realizado manualmente por los empleados de las áreas correspondientes a este trabajo, ellos realizan un proceso repetitivo para cada registro de factura pagada que reciben, siendo cada proceso especial para cada área.

Esto provoca un agotamiento en los recursos que dispone la empresa y a medida que crezca la misma, cada vez será mayor el tiempo requerido de procesamiento, siendo crucial los clientes para esta empresa, se vuelve un problema no contar con este apoyo al proceso.

Además, se tiene la necesidad de que cada área tenga la información de los registros procesado, para poder manejar los errores que ocurran además de tener un respaldo con la información que se completa.

2.1 Solución propuesta

Desarrollar un sistema de apoyo al área de facturación para los sectores que reciben una cantidad mayor de facturas diarias, este programa obtendrá los datos de una base de datos de Access ya establecida dentro de la red pública de la empresa Chilena y mediante una consulta SQL hará una comparación para obtener los datos que no han sido procesadas dentro de la base de datos "FIS" para su posterior ejecución.

También se contará con el envío de la planilla de Excel con los datos que se procesaron junto con los resultados finales para mantener un orden e informar a las áreas correspondiente sobre los resultados finales y puedan corregir los errores manualmente según sea requerido.

2.2 Objetivos

2.2.1 Objetivo General

Desarrollar un sistema que apoye a las áreas de importación, exportación y Bolivia en el proceso de facturación dentro de la base de datos “FIS” de los pagos realizados por los clientes.

2.2.2 Objetivos Específicos

Los objetivos específicos del proyecto son:

- Describir los procesos de las áreas de Facturación.
- Implementar cada módulo correspondiente a los diferentes procesos de las áreas.

2.3 Metodología a usar

En este proyecto se utilizará el modelo incremental, ya que al no saber cuántas áreas o procesos va a abarcar el programa que se está desarrollando, se separará en iteraciones, siendo la primera de importación y la segunda de exportación. Además, al final de cada iteración se implementará lo ya diseñado para su uso inmediato en la empresa. La ventaja que entrega esto es poder realizar una parte del software y poder realizar su implementación de manera inmediata, pudiendo validar que el programa cumple con los requerimientos del cliente.

3 Desarrollo

En la siguiente sección se describirá el análisis que se realizó a las reglas del negocio de la empresa, junto a las restricciones y pasos de cada proceso de facturación.

3.1 Modelo de Negocio

3.1.1 Reglas de Negocio

En el desarrollo de este sistema se respetarán las normas de programación impuestas desde la central de Hapag-Lloyd [1], ubicada en Hamburgo Alemania. Como restricción de programación se exige que se utilice Office 2013 y VBA (que viene incluido en Office) como entorno de programación. Esto es debido a que “Una macro de FIS es la posibilidad de controlar remotamente las aplicaciones de FIS desde un programa externo como Microsoft Excel. Esta posibilidad puede ser usada para mejorar el negocio con las soluciones de FIS”. Además, se exige que las pruebas de ejecución de la macro se realicen en “FIS TEST”, la cual es una plataforma de pruebas idéntica a la principal, la cual contiene información creada y tiene como objetivo hacer pruebas de ejecución de macros.

Al final de la programación se deberá enviar la macro al área de IT para que sea subida a la plataforma “FIS” principal y pueda ser utilizada en esta misma.

3.1.2 Modelo Actual de negocio

Como conocimiento previo a la programación del sistema, se va a dar información sobre el modelo de negocios por el cual se envía la información entre las entidades. La figura 1 muestra un BPMN con las tareas de las tres entidades (La nave, Hapag-Lloyd y Navesoft).

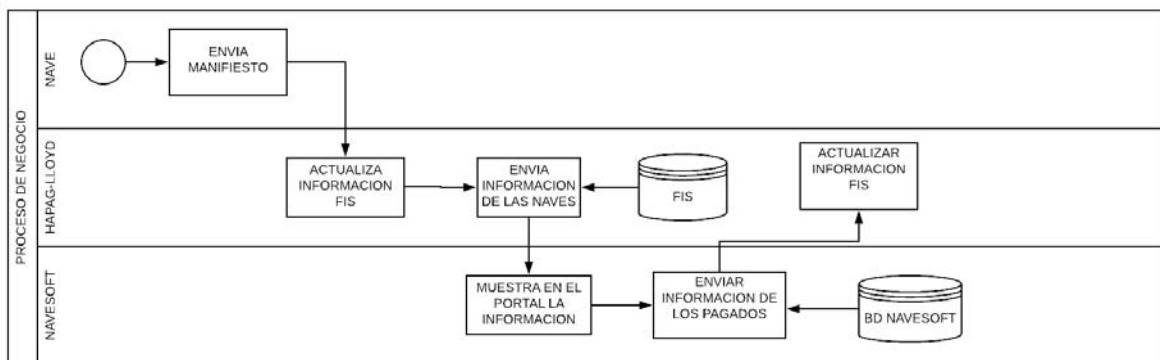


FIGURA 1 BPMN PROCESO DE NEGOCIO.

El primer paso corresponde al envío de los manifiestos de parte de los barcos. Cada manifiesto es un reporte en el cual declara que está transportando el barco, tiene información de cuantos contenedores trae y que objetos trae cada uno. Este reporte se envía a la sede de Hapag-Lloyd que corresponda dependiendo del país en el cual desembarcará próximamente. Este proyecto está enfocado en el proceso de la sede de Chile de Hapag-Lloyd.

A continuación, cuando la sede de Chile recibe el reporte(manifiesto), actualiza la información de “FIS” para que quede almacenado en la base de datos. Con esto queda disponible toda la información de lo que trae el barco.

Posteriormente, una macro programada para que funcione a las 12 del día automática y diariamente, se encarga de enviar los reportes de manifiesto de los barcos que están próximos a llegar al puerto de Chile a “Navesoft”. Navesoft es una empresa externa a Hapag-Lloyd que se encarga del portal de pago en el cual los clientes ingresan para pagar sus contenedores correspondientes. Navesoft diariamente envía un reporte en formato Excel a la sede de Chile de Hapag-Lloyd con la información de los contenedores que ya han sido pagados en el portal de pago, junto con datos de quien pago, fecha, etc.

Cuando la sede de Chile recibe el reporte de los pagos de Navesoft, estos son guardados en la carpeta pública además de ser reenviados a las áreas respectivas (importación, exportación), con esto los usuarios tienen la información para proceder a facturar dentro de la base de datos “FIS”. Cada área encargada de la facturación, con la información que reciben en el correo, proceden a facturar.

El área de TI está encargada de actualizar la base de datos “GATE IN”. Esta es una base de datos de Access que se encuentra en la red pública de la empresa. Esta base de datos tiene “Mapeados” (conectados archivos de Excel a la base de datos de Access) los archivos CSV guardados del correo que envía Navesoft con las reservas ya pagadas. Estos archivos se pueden ver dentro de la base de datos como tablas con toda la información listada de cada pago del reporte, pero no se pueden modificar. La base de datos también tiene tablas adicionales en donde se guardan los datos de los pagos ya procesados dentro de la base de datos de “FIS” que además fueron exitosos en su ejecución. La figura 2 muestra el archivo de Access “Gate-In”.

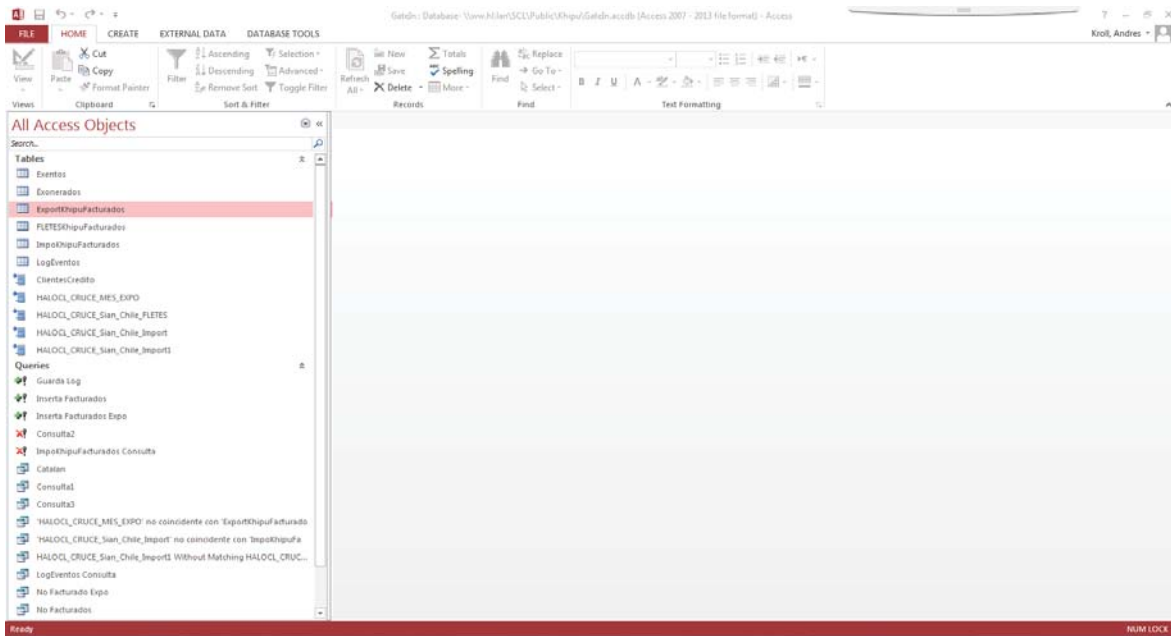


FIGURA 2 BASE DE DATOS “GATE-IN”.

En este proyecto se automatizará el proceso de facturar las reservas pagadas dentro de la base de datos de “FIS”. A continuación, en la siguiente sección se especificará sobre los pasos dentro del proceso de facturación para las áreas de Importación, Exportación y Bolivia.

3.2 Iteración 1: Importación

En la presente sección se mostrarán los pasos del proceso de Importación. Lo que hace diferente este proceso de los demás es que la información de las facturas viene separadas por contenedor, este proceso contempla un ordenamiento de los datos previo el cambio de información y facturación.

3.2.1 Proceso

En el siguiente texto se presenta el proceso de facturación que se realiza manualmente y los pasos que debe seguir el programa. La Figura 3 muestra el diagrama de flujo de datos para la mejor comprensión de los pasos a seguir dentro de la programación.

El proceso empieza ordenando los datos de la planilla Excel por el número de BL, esto se debe que la información de las facturas está separada por contenedor, por lo cual se deben tener todos los contenedores que corresponden a un BL agrupados previo a su paso por “FIS”.

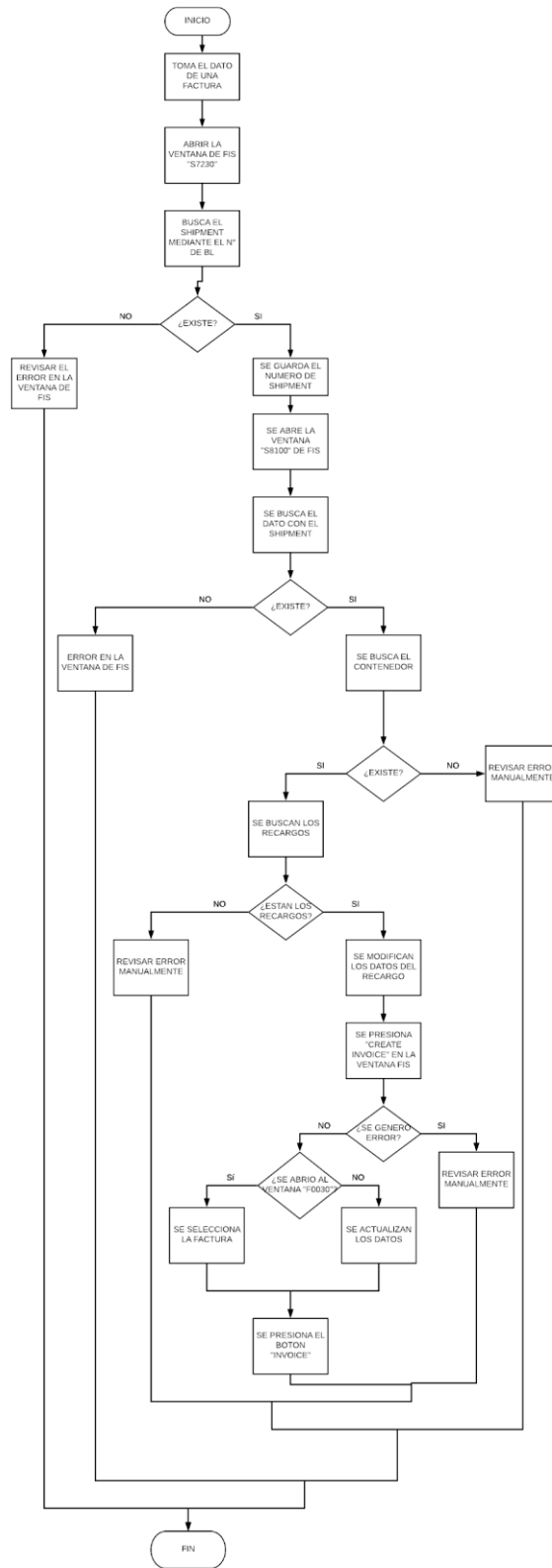


FIGURA 3 DFD PROCESO DE IMPORTACIÓN.

Luego se selecciona un grupo de datos que corresponda al mismo BL, y se procede al proceso dentro de la base de datos “FIS”. Para esto se debe abrir la ventana “S7230” y en la casilla “Document” escribir el número de BL (ver Figura 4) y buscar la información con el botón “Find”, con esto se mostrará la información del dato. Si el número de BL si existe dentro de la base de datos, este arrojará sus datos y el número de “Shipment” (número de reserva), este dato deberá guardarse para un próximo uso. En caso de que el dato no exista dentro de la base de datos, este error se verá en la ventana de “FIS” con una barra roja en la parte inferior, este dato deberá ser manejado manualmente por el departamento de importación, el tipo de error se puede ver mediante la misma ventana de “FIS”.

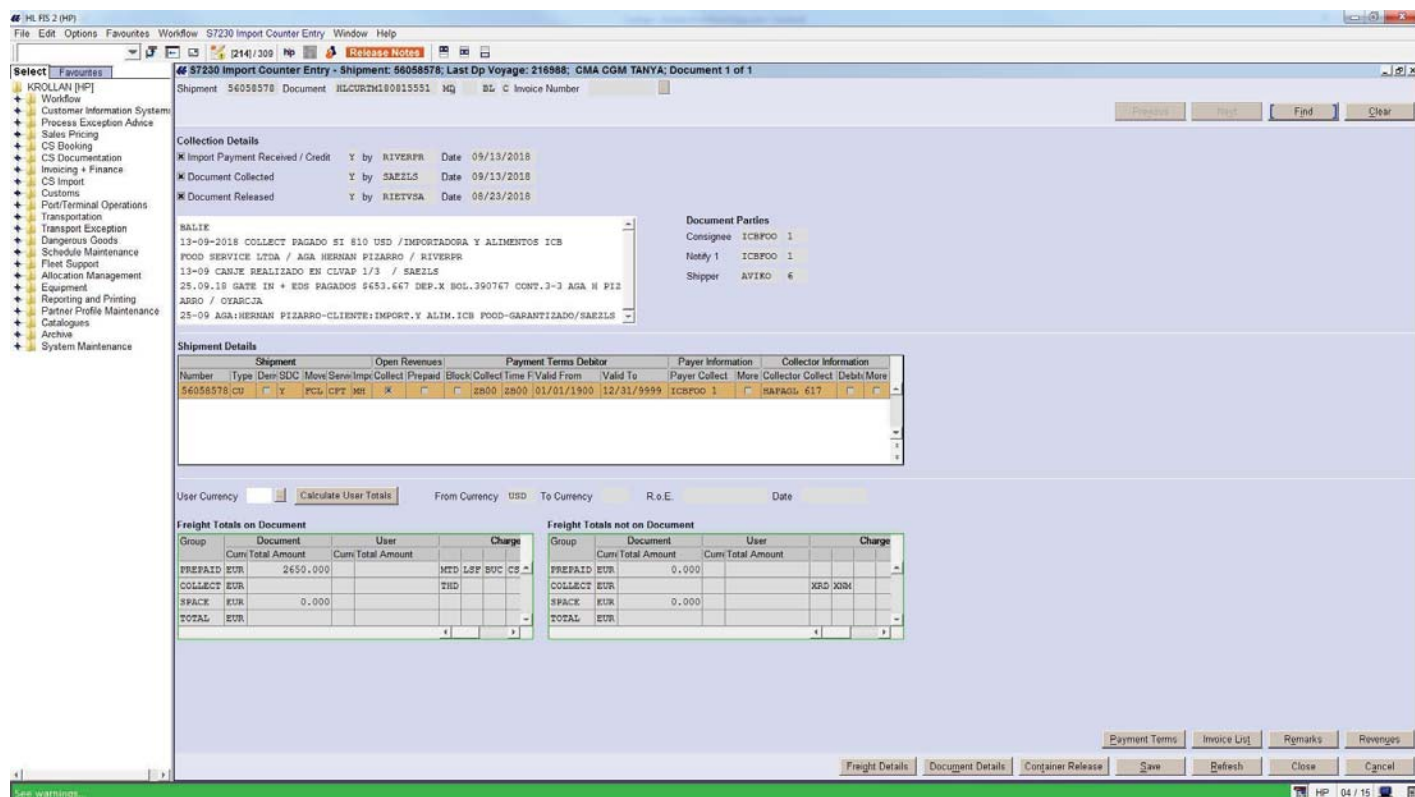


FIGURA 4 VENTANA “S7230”.

Luego cuando ya se tiene el dato del “Shipment”, se abre la ventana de “FIS” “S8100”, (ver Figura 5). La función principal de esta ventana es la de mostrar los recargos de los contenedores listados uno por uno y permitir la modificación de la información para su posterior factura. Se escribe el número de “Shipment” en el recuadro con el mismo nombre y se presiona el botón “Find” para que la base de datos busque su información. En caso de que no exista el dato, se mostrará un error y se solucionara manualmente por el departamento de importación (ver Figura 5).

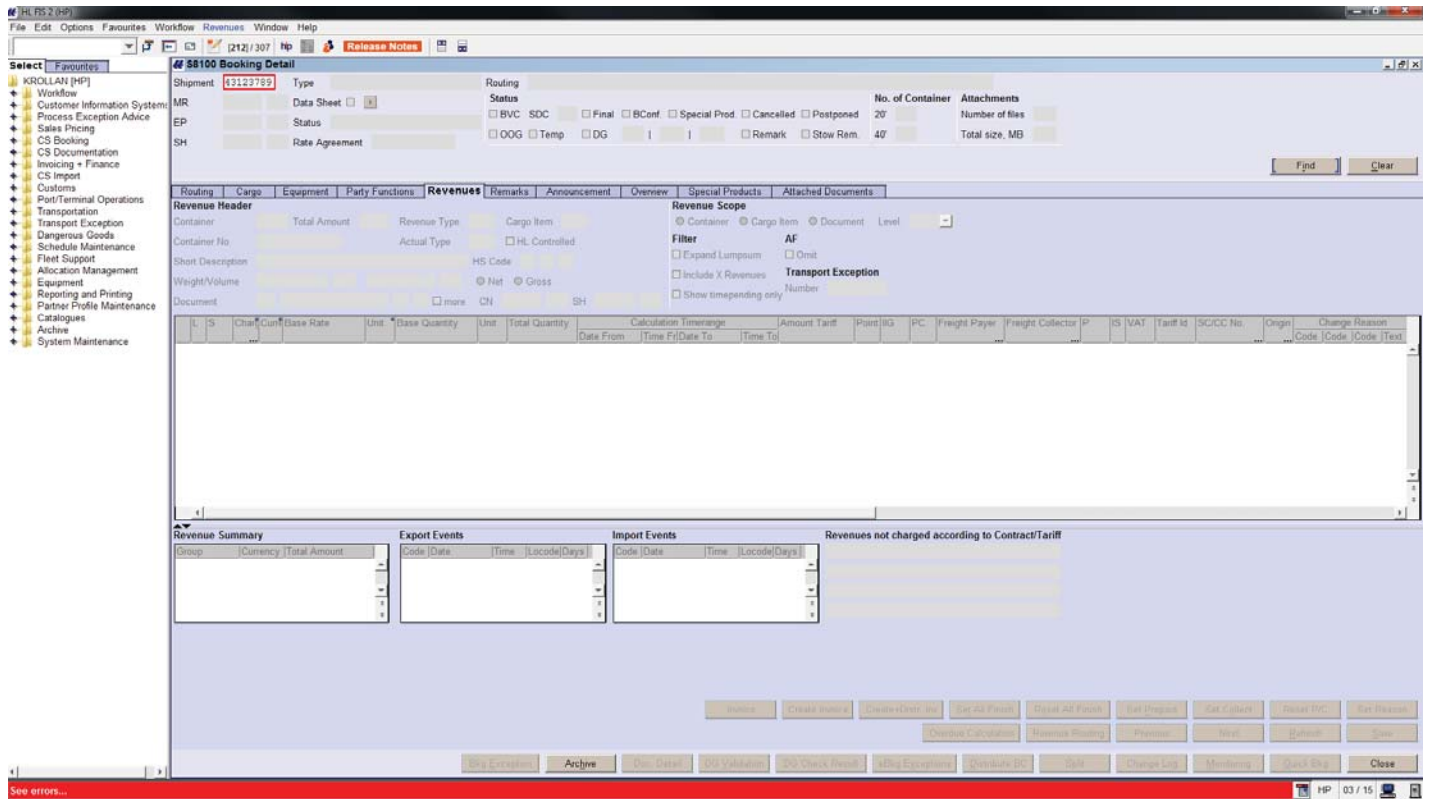


FIGURA 5 VENTANA “S8100” CON ERROR.

En caso de que, si exista el dato, la ventana mostrará la información del primer contenedor relacionado con el número de “Shipment” (ver Figura 6). Primero se deben identificar uno por uno los contenedores que se quieren facturar recorriendo la lista, cuando ya se ubica el contenedor mediante su número, se revisan el dato de los recargos que se quieren procesar, en caso de que el recargo ya este facturado, se mostrara en gris la fila y con el dato “I” (Invoiced) en la columna “S” (Status).

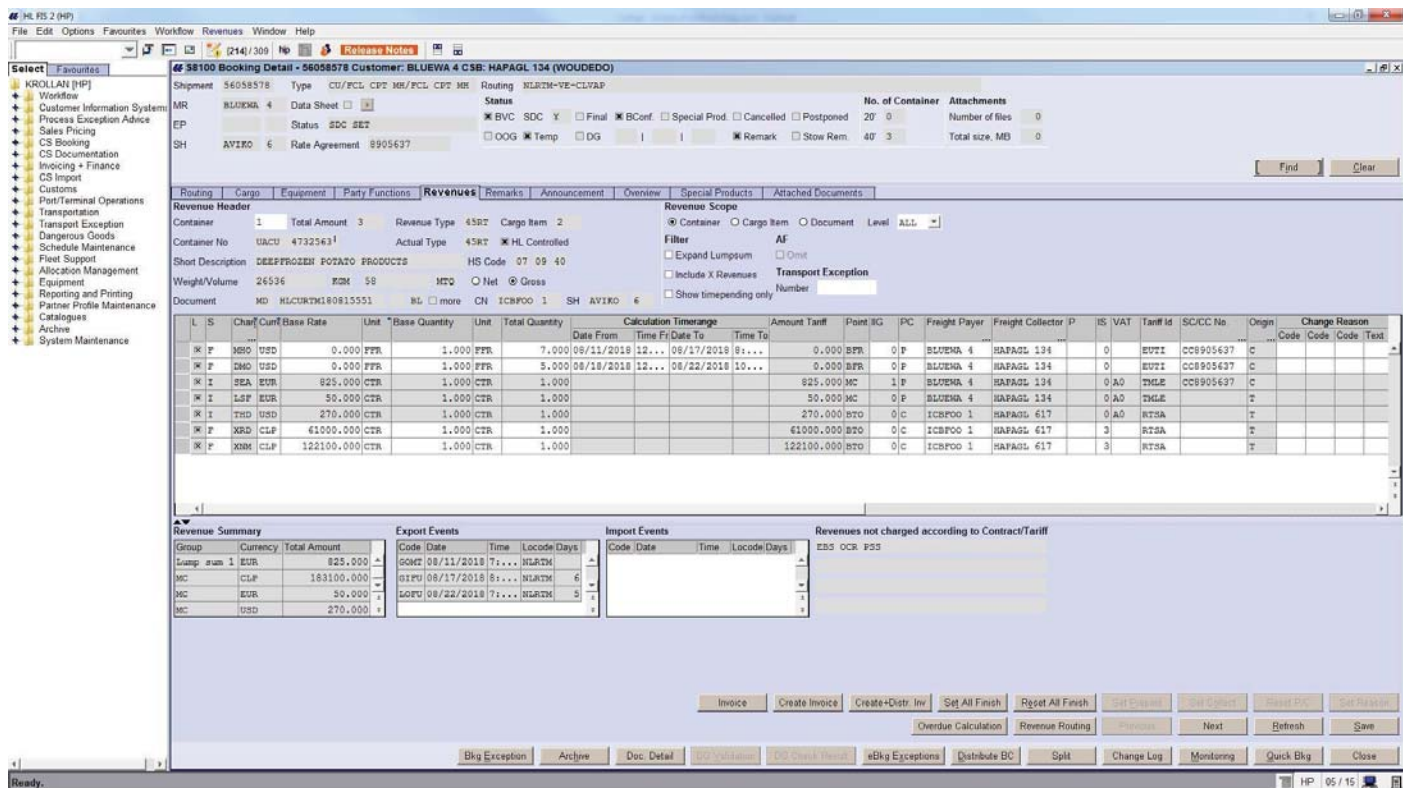


FIGURA 6 VENTANA “S8100”.

En caso de que aún no este facturado, se procede a actualizar los campos del recargo, cambiando el “Matchcode”, el “Invoice Separator” (dato para agrupar datos en una misma factura) y luego se guardan los cambios de ese contenedor, luego se repite el proceso por la cantidad que se estén pagando. Cuando se completan la actualización de datos de todos los datos, se procede a facturar presionando el botón “Create Invoice”.

En caso de que haya más de una factura lista pendiente para procesar, “FIS” mostrará la ventana “F0030” (ver Figura 7) con la lista de los datos pendientes que es posible proceder a facturar. Para seguir con la factura del dato sobre el cual se está trabajando, se debe buscar el dato que se quiere facturar, mediante la identificación del “Matchcode” y el “Invoice Separator” de los datos que trae la planilla, se selecciona la fila con la factura y se presiona el botón “Create Invoice”. En caso de que no exista algún otro dato pendiente, se omitirá este paso y emergerá la ventana “F0020” directamente.

F0030 Invoice Due List

Voyage
 DP Voyage
 Feeder MLV Both

Party
 Collector
 Payer

IRT
 All
 No IRT
 IRT Only

Port
 Locode
 Date Range
 Arrival Departure Both

Invoice Scope
 SH MD AN DO

Payability
 Prepaid Collect

Shipment 43123789 Document

[Find] [Clear]

Document		Shipment		Start Location	Port of Loading	Port of Discharge	End Location	Payer		Collector Matchcode	Payability	Invoice Currency	Invoice Group	Revenues Incomplete
Number	Suffix	Number	SDC					Matchcode	Country					
		43123789	<input type="checkbox"/>		NLRM	SGSIN	VNSGN	CARNIC 39	CL	HAPAGL 911	P	USD	7	<input checked="" type="checkbox"/>
		43123789	<input type="checkbox"/>		NLRM	SGSIN	VNSGN	QUIMIC 13	CL	HAPAGL 911	P	USD	7	<input checked="" type="checkbox"/>

[Booking Detail] [Create Invoice] [Create + Distribute Invoice]

[Refresh] [Cancel] [Close]

FIGURA 7 VENTANA “F0030”.

Luego, cuando emerge la ventana “F0020” (ver Figura 8) se actualizarán los datos de la ventana nuevamente. Primero se modifica la tasa de cambio a todos los contenedores de la lista, esto debido a que el cliente paga con CLP (Moneda Chilena). En primer lugar, se marca la casilla “Set All ROEs” (Rate of Exchange que significa Tasa de cambio en español) al primer dato de la lista y a continuación se vacían las dos casillas al lado derecho de este (“Rate of Exchange” y “Currency”), presionar el botón “Save” para guardar los cambios realizados y con esto se actualizarán las tasas de cambio de los contenedores.

F0020 Invoice Details - collect invoice on SH level, not booked (ID: 112658810)

Invoice Number: Invoice Preparation Number: 112658810 Invoice Scope: SH

Find Clear

Header **Positions** Clauses Free Invoice Positions

Single Invoice Based on
 Document Number:
 Shipment Number: 56058578

Collective Invoice Based on
 Voyage: Locode:
 Charges:
 From: To:

Rate of Exchange Rule
 1st Shipment: Tariff:
 Voyage: Source:
 Port: Date:
 Type: Buying Average Selling

Total Net Amount: 816.810 USD ROE Manually Changed

Invoice Positions

Pos	Position Name	Charge	Container			Revenue		Set all ROEs	First Conversion		Second Conversion		Invoice		Ship
			Seq	Type	Number	Amount	Cur		Rate of Exchange	Cur.	Rate of Exchange	Cur.	Amount	Cur	
1	GATE IN	XRD	1	45RT	UACU 4732563	61000.000	CLP	<input type="checkbox"/>	0.001487	USD			90.710	USD	560
2	DANOS CTR DEST-EDS	XNM	1	45RT	UACU 4732563	122100.000	CLP	<input type="checkbox"/>	0.001487	USD			181.560	USD	560
3	GATE IN	XRD	2	45RT	CPSU 5110934	61000.000	CLP	<input type="checkbox"/>	0.001487	USD			90.710	USD	560
4	DANOS CTR DEST-EDS	XNM	2	45RT	CPSU 5110934	122100.000	CLP	<input type="checkbox"/>	0.001487	USD			181.560	USD	560
5	GATE IN	XRD	3	45RT	UAEU 1241877	61000.000	CLP	<input type="checkbox"/>	0.001487	USD			90.710	USD	560
6	DANOS CTR DEST-EDS	XNM	3	45RT	UAEU 1241877	122100.000	CLP	<input type="checkbox"/>	0.001487	USD			181.560	USD	560

Set ROE Refresh Save

Copy Free Invoice Delete/Reverse Distribute Archive Cancel Close

FIGURA 8 VENTANA “F0020” PESTAÑA “POSITIONS”.

En el próximo paso se debe cambiar a la pestaña “Revenues” de la ventana “F0020” (ver Figura 9), se actualizan los datos de “Payment Base Date” a “I” y el dato “Currency” nuevamente a “CLP”, cuando se terminan de modificar estos campos se presiona el botón “Distribute” y con esto se terminará el proceso de facturación de un registro.

F0020 Invoice Details - collect invoice on SH level, not booked (ID: 112658810)

Invoice Number: [] Invoice Preparation Number: 112658810 Invoice Scope: SH

[Find] [Clear]

Header | Positions | Clauses | Free Invoice Positions

Basics

Invoice Number: 112658810 I Date: [] Payer: ICBFOO 1 1914786
 Referenced Invoice: [] Date: [] Mail to Party: [] [] []
 Total Amount: [] USD Collector: HAPAGL 617 483368
 Additional Number: [] [] [] Creator: HAPAGL 617 483368
 Accounting System: 0200 0078 R Post to AP
 Reason Codes: [] [] [] State: []
 Reason Text: [] Created by: KROLLAN 09/25/2018
 Changed by: KROLLAN 09/25/2018

Involved Organisations

Contact

Default Name: []
 Special Email Address: []
 None Telephone Number: []
 Manual Fax Number: []

ROE Adjustment

Invoice Paid
 Postponed
 Apply ROE Adjustment

Specials

Payment Base Date: V [] Payment Terms: ZB00 Payment Due Date: [] FAC Separate: Y
 Date of Activity: [] Currency: USD Collection Kind: S
 Date of ROE: E [] Language: ES Layout Type: L0
 Automatic Distribution

Withholding Tax

Type / Code: []
 Percentage: []
 Relevant Exempt

Print References

Payer PO Number
 Customer Shipment
 Shipper Container
 Consignee Cargo Item
 Forwarder Document
 Notify 1

General Print Options

Exchange Rates Freight Statement Vessel: F
 Container Grouped Shipper's Address Routing Type: R
 Local ROE Consignee's Address Payment Terms: N
 DG Details Invoice Positions Tariff Currency Summary
 Commodity Information Lines: []
 Marks and Numbers Lines: []

Decimal Separator: .
 Date Format: 1
 Currency Code: 1

[Refresh] [Save]

[Copy Free Invoice] [Delete/Reverse] [Distribute] [Archive] [Cancel] [Close]

FIGURA 9 VENTANA "F0020" PESTAÑA "REVENUES".

3.2.2 Implementación

El resultado de la programación resultó en un archivo Excel con programación hecha en Visual Basic Application, en el cual se tiene una pestaña dedicada para la escritura de los datos resultantes de la consulta SQL entre los datos del CSV enviado por “Navesoft” y la tabla de facturas ya procesadas por “FIS”. Al final de la ejecución de la facturación en “FIS”, la macro (Código de Excel) guardará los datos de las facturas que se hayan completado exitosamente dentro de la tabla de “Ya procesados” mediante una sentencia SQL para que en la siguiente ejecución la consulta no entregue el dato nuevamente.

Antes del final de la ejecución de la macro, esta enviara la planilla completa de los datos que fueron procesados al área de importación para que puedan tener una copia del resultado final del programa y poder manejar manualmente los datos que produjeron error.

En el Anexo A se muestra el pseudocódigo resultante de la programación del proceso basándose en las indicaciones de [2].

SHIPMENT	NRO. BOLETA	NRO. BI	CUPON	FECHA	CUPON	CONTENEDOR	NIT	FACTURAR	MATCHCODE	CONCEPTO	TOTAL CUPON	RESULTADO
390767	HLCURTM180815551	105018	13-sep-18	CPSU5110934	77965620-9	ICBFOO1	EDS	145299				
390767	HLCURTM180815551	105019	13-sep-18	UACU4732563	77965620-9	ICBFOO1	EDS	145299				
390767	HLCURTM180815551	105020	13-sep-18	UAEU1241877	77965620-9	ICBFOO1	EDS	145299				
390767	HLCURTM180815551	105021	13-sep-18	CPSU5110934	77965620-9	ICBFOO1	GATE IN	72590				
390767	HLCURTM180815551	105022	13-sep-18	UACU4732563	77965620-9	ICBFOO1	GATE IN	72590				
390767	HLCURTM180815551	105023	13-sep-18	UAEU1241877	77965620-9	ICBFOO1	GATE IN	72590				
398080	HLCURTM180815573	107116	21-sep-18	BMOU9216456	77965620-9	ICBFOO1	EDS	145299				
398080	HLCURTM180815573	107117	21-sep-18	CPSU5163127	77965620-9	ICBFOO1	EDS	145299				
398080	HLCURTM180815573	107118	21-sep-18	BMOU9216456	77965620-9	ICBFOO1	GATE IN	72590				
398080	HLCURTM180815573	107119	21-sep-18	CRLU1296599	77965620-9	ICBFOO1	EDS	145299				
398080	HLCURTM180815573	107120	21-sep-18	CPSU5163127	77965620-9	ICBFOO1	GATE IN	72590				
398080	HLCURTM180815573	107121	21-sep-18	CRLU1296599	77965620-9	ICBFOO1	GATE IN	72590				
398080	HLCURTM180815573	107122	21-sep-18	CRLU1824079	77965620-9	ICBFOO1	EDS	145299				
398080	HLCURTM180815573	107123	21-sep-18	TCLU1179096	77965620-9	ICBFOO1	EDS	145299				
398080	HLCURTM180815573	107124	21-sep-18	CRLU1824079	77965620-9	ICBFOO1	GATE IN	72590				
398080	HLCURTM180815573	107125	21-sep-18	TCLU1179096	77965620-9	ICBFOO1	GATE IN	72590				
398080	HLCURTM180815573	107126	21-sep-18	UACU4756550	77965620-9	ICBFOO1	EDS	145299				
398080	HLCURTM180815573	107127	21-sep-18	UACU4756550	77965620-9	ICBFOO1	GATE IN	72590				
398123	HLCULE118083298	107133	21-sep-18	HLEBU1933094	93435000-6	CITRO 9	EDS	74613				
398123	HLCULE118083298	107134	21-sep-18	HLEBU1933094	93435000-6	CITRO 9	GATE IN	72590				
398733	HLCUTLV180706743	107259	21-sep-18	TNIU5635240	96653350-1	AGROSY8	EDS	74613				
398733	HLCUTLV180706743	107260	21-sep-18	TNIU5635240	96653350-1	AGROSY8	GATE IN	72590				
398746	HLCUSZX1808ATXC3	107262	21-sep-18	AMFU8710825	76545533-2	URSUST3	EDS	74613				
398746	HLCUSZX1808ATXC3	107263	21-sep-18	AMFU8775657	76545533-2	URSUST3	EDS	74613				
398746	HLCUSZX1808ATXC3	107264	21-sep-18	BEAU4458431	76545533-2	URSUST3	EDS	74613				
398746	HLCUSZX1808ATXC3	107265	21-sep-18	AMFU8710825	76545533-2	URSUST3	GATE IN	72590				
398746	HLCUSZX1808ATXC3	107266	21-sep-18	AMFU8775657	76545533-2	URSUST3	GATE IN	72590				
398746	HLCUSZX1808ATXC3	107267	21-sep-18	BEAU4458431	76545533-2	URSUST3	GATE IN	72590				
398746	HLCUSZX1808ATXC3	107268	21-sep-18	BMOU4010436	76545533-2	URSUST3	EDS	74613				
398746	HLCUSZX1808ATXC3	107269	21-sep-18	BSIU9622240	76545533-2	URSUST3	EDS	74613				
398746	HLCUSZX1808ATXC3	107270	21-sep-18	BMOU4010436	76545533-2	URSUST3	GATE IN	72590				
398746	HLCUSZX1808ATXC3	107271	21-sep-18	BSIU9622240	76545533-2	URSUST3	GATE IN	72590				
398746	HLCUSZX1808ATXC3	107272	21-sep-18	CAIU4217254	76545533-2	URSUST3	EDS	74613				
398746	HLCUSZX1808ATXC3	107273	21-sep-18	CAIU4217254	76545533-2	URSUST3	GATE IN	72590				
398746	HLCUSZX1808ATXC3	107274	21-sep-18	DDV10000033	76545533-2	URSUST3	EDS	74613				

FIGURA 10 PLANILLA DE IMPORTACIÓN.

3.3 Iteración 2: Exportación

En la siguiente sección se mostrará el proceso de facturación de Exportación. En este proceso a diferencia del anterior (Importación), el registro no viene con los contenedores que se paga, al ser exportación, no se tiene contenedores asignados para la facturación, pero si una cantidad definida, además de ya venir con el número de “Shipment”, por lo cual se omite el paso de la ventana “S7230”.

3.3.1 Proceso

En la siguiente sección se presenta el proceso de facturación que se realiza manualmente para el área de exportación y que pasos deberá seguir el program. La Figura 11 muestra el diagrama de flujo de datos para la mejor comprensión de los pasos a seguir dentro de la programación.

El primer paso del proceso es el de realizar la consulta SQL a la base de datos “Gate-IN” (base de datos principal ubicada en la carpeta publica) para obtener los datos de las facturas no procesadas. La consulta SQL consta de un “LEFT JOIN” entre la tabla “HALOCRUCE_Sian_Chile_Export” (nombre como se tiene el archivo CSV recibido de parte de “Navesoft” dentro de la base de datos) con la tabla de “ExportKhipuFacturados” (tabla de Access que contiene los datos de las Facturas ya procesadas dentro de “FIS” y que fueron Exitosas), el resultado de este cruce es la de los datos que están contenidos en la primera tabla (reporte diario) pero que no están en la segunda (Tabla de datos ya procesados), obteniendo los pendientes que aun requieren pasar por la base de datos “FIS”.

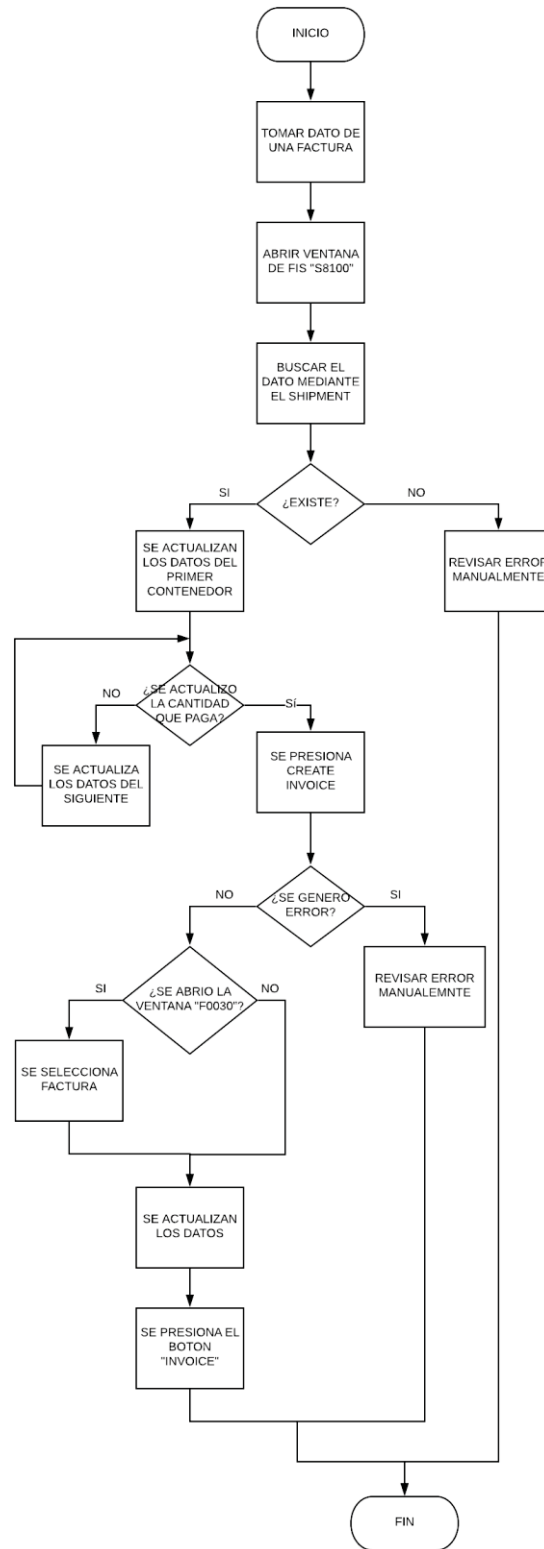


FIGURA 11 DFD PROCESO DE EXPORTACIÓN.

Luego se debe seleccionar las filas de datos de una factura (número de shipment, cantidad de contenedores a pagar, Matchcode), en exportación como se mencionó anteriormente los datos vienen en una fila con el número de “Shipment” como dato identificador y la cantidad de contenedores que se están pagando, lo cual es la diferencia con el proceso de importación.

A continuación, se debe abrir la ventana “S8100” (ver Figura 12), en la casilla “Shipment” se ingresa el número de “Shipment” y presionar el botón “Find”. En caso de que si exista el dato de reserva (“Shipment”) dentro de “FIS”, se mostrara la lista de contenedores uno por uno con sus respectivos recargos (ver Figura 13). En caso de que no exista el dato, se arrojará error de la misma manera que “Importación”, la ventana mostrara un error con una banda roja en la parte inferior (ver Figura 14).

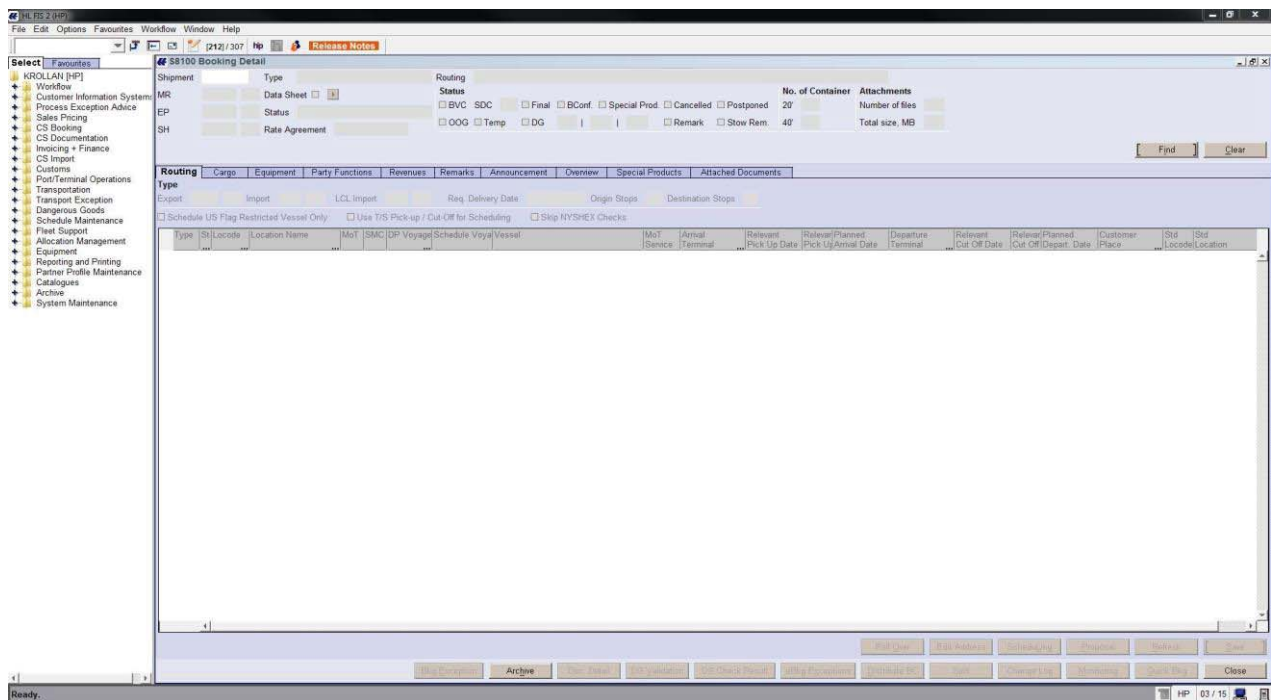


FIGURA 12 VENTANA “S8100”.

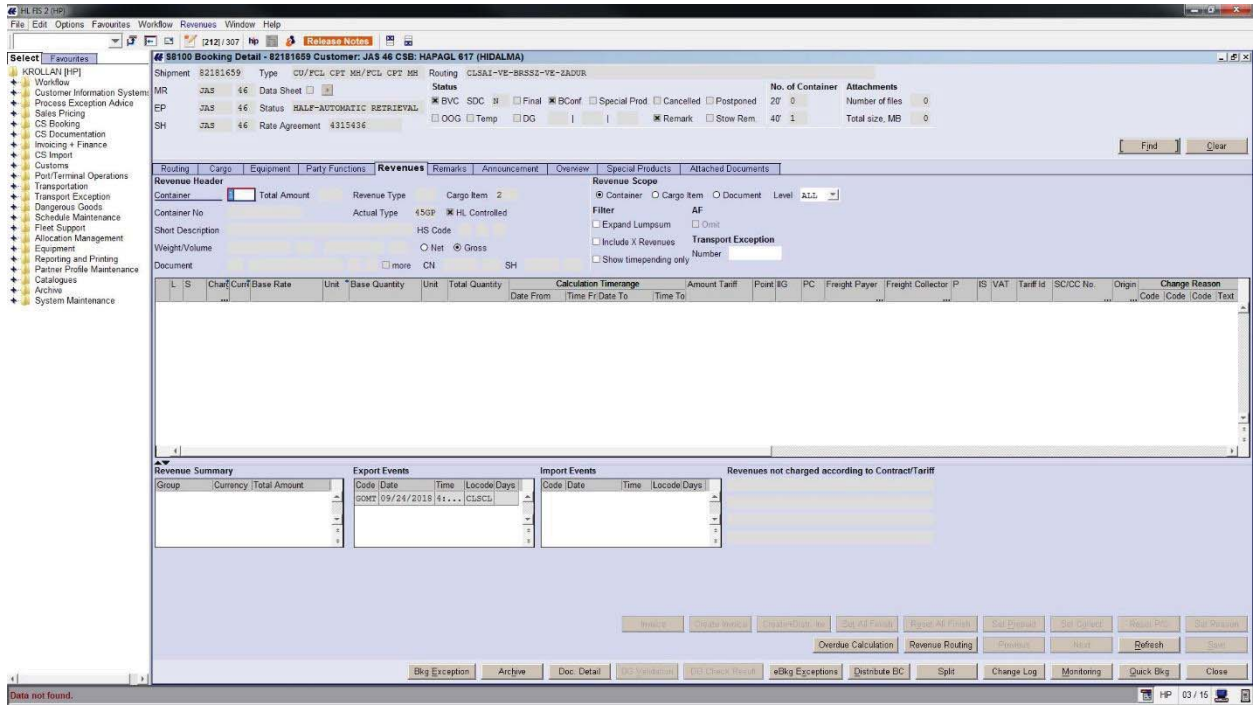


FIGURA 13 VENTANA “S8100” CON DATO.

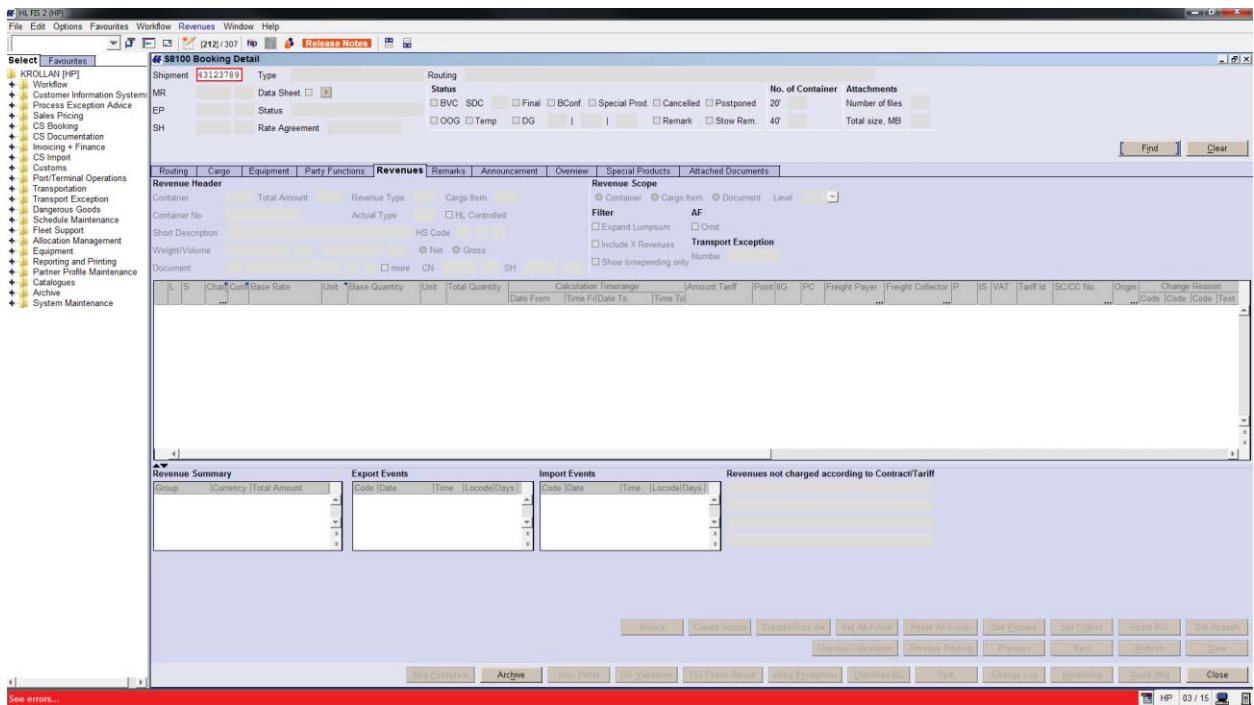


FIGURA 14 VENTANA “S8100” CON ERROR.

A continuación, debido a que en este proceso los datos de los contenedores que deben ser facturados no vienen identificados y solo vienen la cantidad que se están pagando, se procede a seleccionar los primeros contenedores que se encuentren sin facturar y se repite el

paso por la cantidad de contenedores. En caso de que no exista la cantidad suficiente de contenedores disponibles para facturar, se deberá marcar como error, situación que se solucionara manualmente por el departamento de importación.

Se repetirá el proceso similar a importación sobre el manejo de error en caso de que el dato del recargo ya este facturado, en caso de que ya este marcado (columna “S” con el dato “I”) se escribirá como completado en el resultado, agregando el detalle de que ya está facturado, para que con esto el dato no vuelva a ser retomado cuando se ejecute nuevamente la macro. En caso de que no esté facturado, se actualizan los datos del recargo (en este caso como ser exportación, siempre será el recargo que se actualiza “XRO”), se actualiza el “Matchcode”, el “Invoice Separator” (la función es identificar que contenedores se desea juntar al momento de realizar la facturación) y luego se guardan los cambios realizados al contenedor, este paso se repite por la cantidad de contenedores que se estén pagando. Cuando se completa la actualización de toda la factura, se procede al siguiente paso presionando el botón “Create Invoice”.

Luego en este paso se repite el proceso de la ventana “F0020” y “F0030” de la ejecución de importación. En caso de que haya más de una factura pendiente lista para procesar, “FIS” mostrara la ventana “F0030” con un listado los datos que están disponibles para continuar con el proceso. Para poder seguir con el dato con el cual se estaba trabajando, se debe buscar y seleccionar la fila que coincida con el “Matchcode” y el “Invoice Separator” actualizados previamente, y presionar el botón “Create Invoice” de la ventana de “F0030”. En caso de que no exista más de un dato pendiente para facturar, se omitirá este paso y abrirá directamente la ventana “F0020”.

Luego, cuando emerge la ventana “F0020” se actualizarán nuevamente los datos de la ventana vaciando los “ROE” de todos los contenedores. A continuación se cambia a la pestaña “Revenues” para cambiar los datos del “Payment Base Date” y el “Currency” para terminar con la facturación.

3.3.2 Implementación

El resultado de la programación resultó en un nuevo módulo dentro del archivo Excel, además de una pestaña nueva dedicada para el almacenamiento de los datos obtenidos de la consulta SQL y el guardado del resultado de la ejecución del programa. La consulta SQL cruza los datos que las facturas del archivo recibido de “Navesoft” con la tabla de las facturas ya procesados, obteniendo como resultado las facturas que aún no han sido procesadas.

Al final del proceso, los datos que hayan sido completados serán guardados en la tabla de los “Ya procesados” de la base de datos para que al momento de realizar nuevamente la consulta no se repitan los datos exitosos. Adicionalmente se generará un correo con la planilla completa de los datos que se procesaron, para que el área revise manualmente los datos procesados y analice los datos que generaron error.

En el anexo B se adjunta pseudolenguaje del resultado de la programación del proceso “Exportación”.

1	SHIPMENT	ORDEN SERVICIO	RUT	ORDEN COMPRA	MATCHCODE	PAGO	Tipo	CANTIDAD	CODIGO KHIPU	TOTAL FINAL	CREDITO	Resultado
2	50727974	421329	91449000-6		ND	PAGADO	DV 20	2	zndjwaptjpo	123760	31/AGO/2018	
3	51063811	422672	91449000-6		ND	PAGADO	DV 20	1	pfgedlupwmgd	61880	14-sep-18	
4	51394837	423156	76070534-9	204776	ND	PAGADO	HC 40	5	zbrfocvmaq3q	309400	24-sep-18	
5	51395657	421738	76041000-4		ARESTI2	PAGADO	DV 40	1	bgxyqg63xzcq	61880	20-sep-18	
6	51395657	421937	76041000-4		ARESTI2	PAGADO	DV 40	1	bgxyqg63xzcq	61880	20-sep-18	
7	51398621	423151	968990006	612-18 E	CTMCHI002	PAGADO	DV 20	1	dwtz1l0j62c	61880	24-sep-18	
8	52395405	421965	91449000-6		ND	PAGADO	DV 20	2	ptoodbth0nv	123760	07-sep-18	
9	52398915	423458	96866400-3		EXPORT235	PAGADO	DV 20	1	q1ducsczwal	61880	26-sep-18	
10	52728073	421947	91449000-6		ND	PAGADO	DV 20	6	3yjhudaxsd	371200	07-sep-18	
11	52731934	422997	93436000-1		DECAPA1	PAGADO	DV 20	1	zgh30nuwv4	61880	21-sep-18	
12	53064173	422901	99568800-7		VINA1N 2	PAGADO	HC 40	1	muechnloymic	61880	20-sep-18	
13	53065118	422950	99568800-7		VINA1N 2	PAGADO	HC 40	1	0yrs6ngxq3eo	61880	21-sep-18	
14	53396489	423145	7835548-4		ND	PAGADO	DV 40	1	gfgzmxkwl33g	61880	24-sep-18	
15	53730283	422724	93436000-1		DECAPA1	PAGADO	DV 20	1	ytpbwevtr97	61880	14-sep-18	
16	54397497	422862	94836000-4		LOESPE1	PAGADO	DV 20	1	wk8p9662h9w1	61880	21-sep-18	
17	56062569	422000	91449000-6		ND	PAGADO	DV 20	2	fxvoalgcqif	123760	07-sep-18	
18	56065606	423637	93436000-1		DECAPA1	PAGADO	DV 20	1	ib7vdxasit	61880	27-sep-18	
19	56397508	423065	99568800-7		VINA1N 2	PAGADO	HC 40	1	xqihap1ucdmc	61880	24-sep-18	
20	57397348	423647	77174670-5		AGEHCC392	PAGADO	HC 40	2	zplu0th0h24	123760	27-sep-18	
21	57398449	423058	99568800-7		VINA1N 2	PAGADO	HC 40	1	ntosadpm1ty	61880	24-sep-18	
22	57727082	421638	96907340-4		BODEGA364	PAGADO	DV 20	1	bgxyqg63xzcq	61880	20-sep-18	
23	58397113	422807	93436000-1		DECAPA1	PAGADO	DV 40	1	urwqf7ybxzd	61880	20-sep-18	
24	58728400	422918	76267890-K		VISTAM11	PAGADO	DV 20	1	5pboxpektvz	61880	26-sep-18	
25	59396092	421950	91449000-6		ND	PAGADO	DV 20	1	z0dzaehtne1	61880	07-sep-18	
26	69398452	422961	99568800-7		VINA1N 2	PAGADO	HC 40	1	yndfdajpc	61880	21-sep-18	
27	81879997	421131	76572558-5	EM497	ND	PAGADO	DV 20	20	mstkognuv66w	1237600	30/AGO/2018	
28	82007713	421129	94612000-6		DOLE 26	PAGADO	RE 40	3	axuqgd0nav	210630	30/AGO/2018	
29	82014944	421892	76228125-2		ND	PAGADO	DV 20	1	exjrgwv0hrd	61880	06-sep-18	
30	82030597	421783	76601808-4		ND	PAGADO	DV 20	1	bgxyqg63xzcq	61880	20-sep-18	
31	82039702	422281	8286926-K		ND	PAGADO	DV 20	1	mllmhm3klla	61880	11-sep-18	
32	82048459	421193	96659440-3		UNIPAC5	PAGADO	DV 20	1	meekdsccljze	61880	30/AGO/2018	
33	82048459	421193	96659440-3		UNIPAC5	PAGADO	HC 40	1	meekdsccljze	61880	30/AGO/2018	
34	82055114	421168	91449000-6		ND	PAGADO	HC 40	2	wtmk7nbyrwwz	123760	30/AGO/2018	
35	82055290	421652	91449000-6		ND	PAGADO	HC 40	1	y1fdqkqzls	61880	04-sep-18	

FIGURA 15 PLANILLA “EXPO”.

3.4 Iteración 3: Bolivia

En la siguiente sección, se mostrarán los pasos y la implementación de la tercera iteración de este proyecto, que corresponde a la facturación dentro de la base de datos de “FIS” que corresponde al proceso de Bolivia.

3.4.1 Proceso

En la presente sección se mostrarán los pasos que conlleva el proceso de facturación de Bolivia que son realizados por los encargados de esta. El proceso se realiza de manera similar de “Importación”, las diferencias son que los datos se cargan directamente de un CSV (Coma Separated Value) y que se añade una cláusula en la ventana “F0020”.

El primer paso es copiar los datos de las facturas desde el CSV. Notar la diferencia que en este caso, no hay una base de datos Access entremedio a la cual se le consulta los pagos. Este archivo contiene la información de los pagos correspondientes a Bolivia. La totalidad de los datos se copian a una pestaña que se crea automáticamente llamada “Paso”. A continuación, se procede a filtrar los datos para ser traspasados a la pestaña “Impo” y posteriormente a ser facturados. Los datos que se seleccionarán para facturarse son todos los que el pago se haya realizado hasta 3 días antes de la fecha actual. Como paso final del traspaso de información, se procede a eliminar la pestaña “Paso”.

El segundo paso consiste en tomar una línea con información de la pestaña “Impo” y proceder a actualizar la información. Se busca el número de “Shipment” dentro de la base de dato de “FIS”. Para esto se toma el dato “Numero de BL” del dato y se busca en la ventana “S7230”. El número de “Shipment” aparecerá al lado de la casilla que tiene como nombre “Shipment”(Ver Figura 16), el dato se guarda en la planilla en la misma fila para el siguiente paso. En caso de que no se encuentre el “Numero de BL”, “FIS” mostrará una barra roja abajo con la descripción del error.

47230 Import Counter Entry - Shipment: 52056603; Last Dp Voyage: 12191; SAN ANTONIO EXPRESS; Document 1 of 1

Shipment: 52056603 Document: HLCUBCL180728004 HQ BL C Invoice Number

Previous Next Find Clear

Collection Details

- * Import Payment Received / Credit Y by GUERESE Date 09/20/2018
- * Document Collected Y by ERASOAL Date 09/20/2018
- * Document Released Y by PALAINC Date 09/14/2018

18/09/18 CLIENTE CANCEO EN BOLIVIA CAC/GUEREME
 18/09/18 CLIENTE CANCEO EN BOLIVIA THD USD.-105/GUEREME
 LIBERACION 2436
 24/09/18 CLIENTE CANCELA EDS BTA 400237 / IGARRIDO
 09/10/18 CLIENTE CANCELA GATE IN BTA MANUAL 136144 / IGARRIDO

Document Parties

Consignee DEGENT 2
 Notify 1 DEGENT 2
 Shipper BSSELE 15

Shipment Details

Shipment Number	Type	Dem	SDC	Move	San	Imp	Collect	Prepaid	Block	Collect	Time	F	Valid	From	Valid	To	Payer	Information	Collector	Information
52056603	CU																DEGENT 2		ALMAGE 490	

User Currency: [] Calculate User Totals From Currency: USD To Currency: R.o.E. Date: []

Freight Totals on Document

Group	Document	User	Charge
PREFPAID	USD	2835.410	HTD CSF OCR TH
COLLECT	USD	105.000	THD
SPACE	USD	0.000	
TOTAL	USD	2940.410	

Freight Totals not on Document

Group	Document	User	Charge
PREFPAID	USD	0.000	
COLLECT	USD	1195.230	XRD XRD:MIID
SPACE	USD	0.000	
TOTAL	USD	1195.230	

Payment Terms Invoice List Remarks Revenues

Freight Details Document Details Container Release Save Refresh Close Cancel

FIGURA 16 VENTANA “S7230”.

Este proceso se repite por cada fila, hasta conseguir el número de “Shipment” de todos los datos que estén en la pestaña de “Impo”.

Luego, con el “Shipment” ya guardado de todos los datos, se procede a ir a la ventana “S8100” para la actualización de la información. Se abre la ventana “S8100” y luego en la casilla de Shipment se escribe el número obtenido del paso anterior(Ver Figura 17) . Se selecciona la pestaña “Revenues”, luego se selecciona el recargo(XNM o XRD dependiendo de la plantilla), y se modifica la información con los datos del pago. Con los datos ya actualizados, se procede a repetir el paso con la siguiente fila, hasta completar todas las filas.

S8100 Booking Detail - 52056603 Customer: JCV 2 CSB: HAPAGL 557 (MIQUEM)

Shipment 52056603 Type CU/FCL CPT HB/FCL CPT HB Routing ESRCH-YE-PECLL-YE-CLARI

MR JCV 2 Data Sheet Status BVC SDC Y Final BCont Special Prod Cancelled Postponed No. of Container 20 0 Attachments Number of files 0

EP Status SDC SKT DOG Temp DG I I Remark Stow Rem 40 1 Total size, MB 0

SH BSRELE 15 Rate Agreement

Find Clear

Routing Cargo Equipment Party Functions Revenues Remarks Announcement Overview Special Products Attached Documents

Revenue Header

Container 1 Total Amount 1 Revenue Type 450P Cargo Item 1

Container No USCF 5185347 Actual Type 450P HL Controlled

Short Description BOOSEBOLD AFFLIANCES HS Code 72 21 11

Weight/Volume 7938.378 EGM 57 MTO Net Gross

Document MD HLCUC1180728004 BL more CN DECENT 2 SH BSRELE 15

Revenue Scope

Container Cargo Item Document Level ALL

Filter AF

Expand Lumpsum Omit

Include X Revenues Transport Exception

Show timepending only

L	S	Char	Curr	Base Rate	Unit	Base Quantity	Unit	Total Quantity	Calculation	Time	Amount	Tariff	Point	IG	PC	Freight Payer	Freight Collector	P	IS	VAT	Tariff Id	SC/CC No.	Origin	Change Reason
K	I	WFO	EUR	43.000	CTR	1.000	CTR	1.000			43.000	BFR	0	P	JCV	2	HAPAGL	557	0	0	0		RTEU	T
K	F	DMO	EUR	0.000	FFR	1.000	FFR	3.000	08/08/2018	12...	0.000	BFR	0	P	JCV	2	HAPAGL	557	2	0	0		EUTI	T
K	I	SEA	EUR	2000.000	CTR	1.000	CTR	1.000			2000.000	MC	1	P	JCV	2	HAPAGL	557	0	0	0		TMLE	C
K	I	CSF	EUR	10.000	CTR	1.000	CTR	1.000			10.000	MC	0	P	JCV	2	HAPAGL	557	0	0	0		TMLE	T
K	I	OCR	EUR	94.000	CTR	1.000	CTR	1.000			94.000	MC	0	P	JCV	2	HAPAGL	557	0	0	0		TMLE	T
K	I	THD	USD	105.000	CTR	1.000	CTR	1.000			105.000	BTO	0	C	DECENT	2	ALMACE	690	0	0	0		RTSA	T
K	I	MID	USD	0.000	FFR	1.000	FFR	14.000	09/19/2018	12...	0.000	BTO	0	C	DECENT	2	HAPAGL	617	0	0	0		WEI1	C
K	I	MID	USD	145.000	DAY	1.000	DAY	7.000	10/03/2018	12...	1015.000	BTO	0	C	DECENT	2	HAPAGL	617	4	0	0		WEI1	T
K	F	XPD	CLP	61000.000	CTR	1.000	CTR	1.000			61000.000	BTO	0	C	DECENT	2	ALMACE	690	3	0	0		RTSA	T
K	F	XKM	CLP	62700.000	CTR	1.000	CTR	1.000			62700.000	BTO	0	C	DECENT	2	HAPAGL	617	3	0	0		RTSA	T

Revenue Summary

Group	Currency	Total Amount
Sum	EUR	2000.000
MC	CLP	123700.000
MC	EUR	367.000
MC	USD	1120.000

Export Events

Code	Date	Time	Locode	Days
DOMT	08/03/2018	8...	ESAZ	5
SIPO	08/08/2018	12...	ESRCH	5
LOPU	08/09/2018	10...	ESRCH	1

Import Events

Code	Date	Time	Locode	Days
DEPU	09/05/2018	2...	PECLL	3
LOPU	09/08/2018	8...	PECLL	3
DEPU	09/19/2018	5...	CLARI	11
GIMP	10/09/2018	4...	CLARI	20

Revenues not charged according to Contract/Tariff

SEC T30

Invoice Create Invoice Create+Distr. Inv Set All Finish Reset All Finish Set Prepaid Set Collect Reset PVC Set Reason

Override Calculation Revenue Rounding Invoice Split Defresh Save

Bkg Exception Archive Doc. Detail Get Validation DO Check Remark Bkg Exceptions Distribute BC Split Change Log Monitoring Quick Bkg Close

FIGURA 17 VENTANA "S8100".

Cuando se termina de actualizar toda la información de las facturas, se procede a facturar. Se agrupan los datos según "Shipment" para este paso. Se abre la ventana "S8100" y se busca por el mismo número(Ver Figura 17), luego se selecciona la opción "Create Invoice", esto abrirá la ventana "F0020". En caso de existir más de un dato a facturar, se abrirá la ventana "F0030", en la cual uno debe seleccionar el dato que quiere facturar y presionar "Create Invoice"(Ver Figura 18).

F0030 Invoice Due List

Voyage
 DP Voyage
 Feeder MLV Both

Party
 Collector
 Payer

IRT
 All
 No IRT
 IRT Only

Port
 Locode
 Date Range

Invoice Scope
 SH MD AN DO

Payability
 Prepaid Collect

Shipment 52056603 Document

Document		Shipment		Start Location	Port of Loading	Port of Discharge	End Location	Payer		Collector Matchcode	Payability	Invoice Currency	Invoice Group	Revenues Incomplete
Number	Suffix	Number	SDC					Matchcode	Country					
		52056603	<input checked="" type="checkbox"/>		ESBCN	PECLL	CLARI	DEGENT 2	BO	ALMACE 690	C	USD	3	<input checked="" type="checkbox"/>
		52056603	<input checked="" type="checkbox"/>		ESBCN	PECLL	CLARI	DEGENT 2	BO	HAPAGL 617	C	USD	3	<input checked="" type="checkbox"/>

FIGURA 18 VENTANA “F0030”.

Posteriormente, con la ventana “F0020” abierta, se procede a limpiar los campos de “ROE” de la lista que se despliega, de manera similar a las iteraciones anteriores. El paso que los diferencia es que posterior a limpiar los “ROE” de los recargos, se procede a la pestaña “Clauses” y se agrega una nueva fila, en la cual se escribirá “NO AFECTO A IVA SEGUN TRATADO DE PAZ CHILE/BOLIVIA AÑO 1904”, esto se hace para respetar el tratado y que la gente que lo vea en “FIS” sepa que es un contenedor de Bolivia.

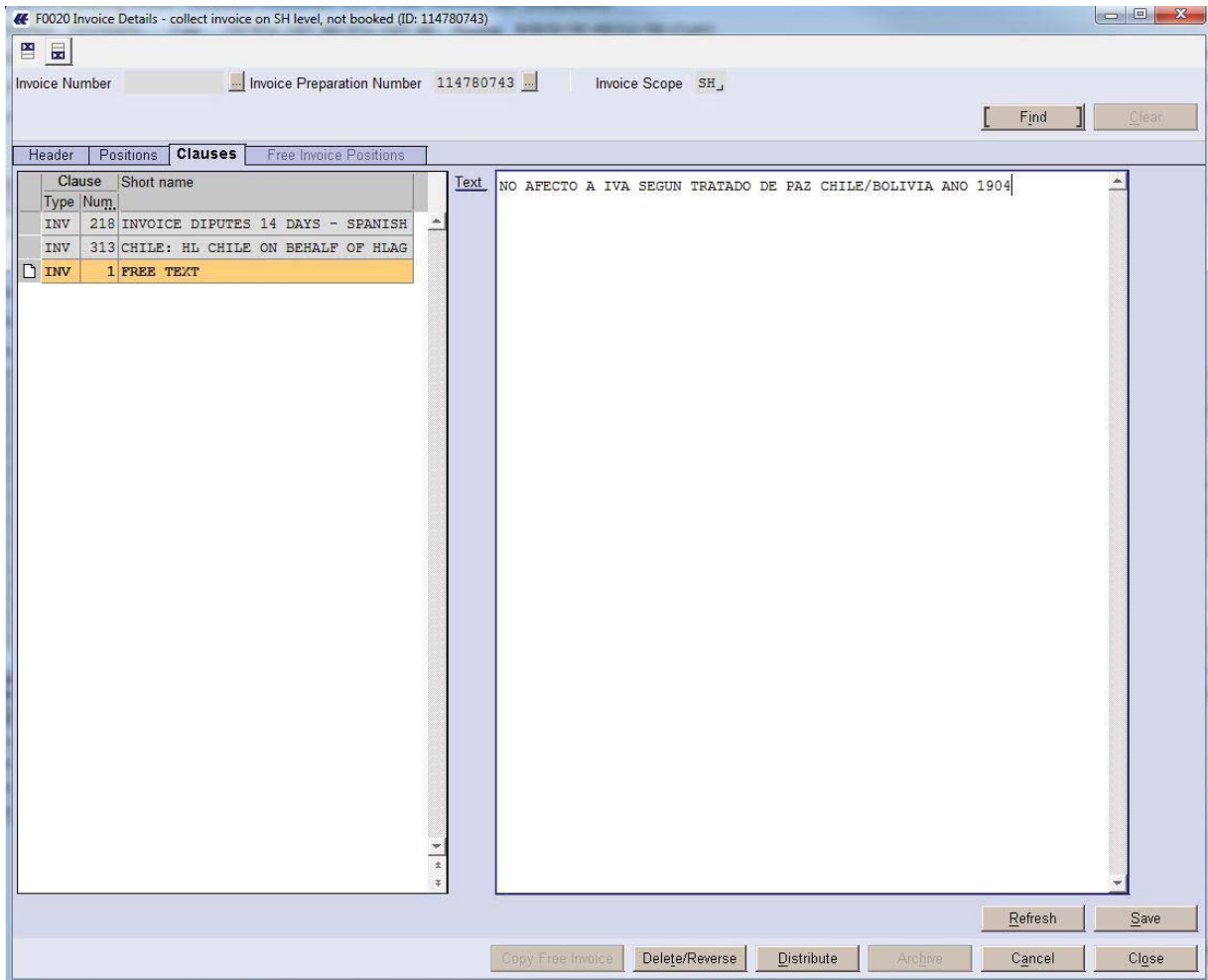


FIGURA 19 VENTANA “F0020” PESTAÑA “CLAUSES”.

Finalmente, se cambia a la pestaña “Header” para cambiar los datos finales (Payment Base Date y Currency), se presiona el botón “Distribuite” y se completa la facturación para el “Shipment” actual. Luego se repite el proceso por cada “Shipment” diferente que haya en la pestaña de “Impo”.

3.4.2 Implementación

El resultado de la programación resultó en la reutilización de la pestaña “Impo” debido a que los datos y el proceso es casi similar. Para copiar los archivos crea una nueva pestaña “Paso” para copiar el CSV completo y luego las filas que se puedan facturar se copian en la pestaña “Impo” para que puedan facturarse.

Este proceso, similar a las anteriores iteraciones, enviará la planilla con los resultados en un correo de Outlook a las personas encargadas del proceso de Bolivia para que tengan una constancia de los resultados y los errores generados durante la ejecución.

En el Anexo C se adjunta Pseudolenguaje de la programación resultante del proceso de Bolivia.

NRO_BOL	FECHA_B	CUPON	FECHA_C	NRO_BL	CONTENENT	FACTRAZON	S_NIT	PAGARAZON	S_CONCEPT	TIPO	CONESTADO	TOTAL	CI	FACTURA	NRO	REC	VALOR	B_MONEDA	FECHA_P	USUARIO	CARGA	RETRO	TASA	CAI	MATCHCODE			
100116	29/AGO/21	22821	29/AGO/21	HLCUSHA	TCKL25872	1609-7	GLTFER	J21609-7	GLTFER	JEDS	DV20	AUTORIZA	41500.00	8949	R1659617	83000.00	CLP	30/AGO/21	RTA	BOLIVIA	DIRECTO	0	ND					
100116	29/AGO/21	22822	29/AGO/21	HLCUSHA	UACLU332	21609-7	GLTFER	J21609-7	GLTFER	JEDS	DV20	AUTORIZA	41500.00	8950	R1659618	83000.00	CLP	30/AGO/21	RTA	BOLIVIA	DIRECTO	0	ND					
100118	29/AGO/21	22855	29/AGO/21	HLCUNQ2	HXLU85	14	2826-6	TRANSPO	2826-6	TRANSPO	EDES	HC40	AUTORIZA	62700.00	8957	R1659678	62700.00	CLP	30/AGO/21	RTA	BOLIVIA	DIRECTO	0	ND				
100122	29/AGO/21	22905	29/AGO/21	HLCUNQ1	GATU8644	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	92400.00	8971	R1659742	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22905	29/AGO/21	HLCUNQ1	GATU8644	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8971	R1659742	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22906	29/AGO/21	HLCUNQ1	GESU8661	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	DV20	AUTORIZA	41500.00	8972	R1659743	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22906	29/AGO/21	HLCUNQ1	GESU8661	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8972	R1659743	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22907	29/AGO/21	HLCUNQ1	HXLU8207	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8973	R1659744	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22908	29/AGO/21	HLCUNQ1	HXLU8791	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8976	R1659745	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22909	29/AGO/21	HLCUNQ1	HXLU8380	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8978	R1659746	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22910	29/AGO/21	HLCUNQ1	HXLU8245	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	DV20	AUTORIZA	41500.00	8974	R1659747	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22910	29/AGO/21	HLCUNQ1	HXLU8245	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8974	R1659747	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22911	29/AGO/21	HLCUNQ1	TCLU8070	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8977	R1659749	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100122	29/AGO/21	22912	29/AGO/21	HLCUNQ1	TCLU8114	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8975	R1659748	501600.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100123	29/AGO/21	22913	29/AGO/21	HLCUNQ1	AMFU8551	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8980	R1659751	250800.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100123	29/AGO/21	22914	29/AGO/21	HLCUNQ1	CHLU8770	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8979	R1659750	250800.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100123	29/AGO/21	22915	29/AGO/21	HLCUNQ1	TEMU7551	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8981	R1659752	250800.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100123	29/AGO/21	22916	29/AGO/21	HLCUNQ1	HXLU8090	13000-1	EMP	DE T	13000-1	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	8982	R1659753	250800.00	CLP	30/AGO/21	RTA	BOLIVIA	DESCONS	0	ND			
100124	30/AGO/21	23057	30/AGO/21	HLCUM4M	BMOU8669	12468-0	EMP	DE T	12468-0	EMP	DE T	EDES	HC40	AUTORIZA	62700.00	9003	R1660025	62700.00	CLP	31/AGO/21	RTA	BOLIVIA	DIRECTO	0	ND			
100125	31/AGO/21	23144	31/AGO/21	HLCUNQ1	CAIU9573	19626-6	TRANS	VII	19626-6	TRANS	VHEDS	HC40	AUTORIZA	62700.00	9081	R1660225	62700.00	CLP	01-sep-17	RTA	BOLIVIA	DIRECTO	0	ND				
100129	31/AGO/21	23221	31/AGO/21	HLCUNQ1	CLHU8770	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9075	R1660219	570.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100129	31/AGO/21	23223	31/AGO/21	HLCUNQ1	GESU8557	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9079	R1660223	570.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100129	31/AGO/21	23224	31/AGO/21	HLCUNQ1	HXLU8207	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9076	R1660220	570.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100129	31/AGO/21	23225	31/AGO/21	HLCUNQ1	HXLU8090	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9077	R1660222	570.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100129	31/AGO/21	23226	31/AGO/21	HLCUNQ1	TCLU8114	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9078	R1660221	570.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100129	31/AGO/21	23227	31/AGO/21	HLCUNQ1	TEMU7551	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9080	R1660224	570.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100131	31/AGO/21	23246	31/AGO/21	HLCUNQ1	HXLU8559	22469-K	TRANSPO	22469-K	TRANSPO	EDES	HC40	AUTORIZA	62700.00	9133	R1660438	62700.00	CLP	02-sep-17	RTA	BOLIVIA	DIRECTO	0	ND					
100137	31/AGO/21	23290	31/AGO/21	HLCUNQ1	AMFU8551	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9086	R1660248	475.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100137	31/AGO/21	23292	31/AGO/21	HLCUNQ1	HXLU8380	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9086	R1660246	475.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100137	31/AGO/21	23294	31/AGO/21	HLCUNQ1	GATU8644	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9088	R1660247	475.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND		
100137	31/AGO/21	23296	31/AGO/21	HLCUNQ1	CI INR1	TCLU8791	13000-1	EMP	DE T	13000-1	EMP	DE T	GATE	IN	HC40	AUTORIZA	95.00	9087	R1660248	475.00	CLP	01-sep-17	RTA	BOLIVIA	DESCONS	0	ND	

FIGURA 20 PLANILLA “IMPO” CON DATOS DE BOLIVIA.

4 Conclusiones

Como conclusión de este trabajo en primer lugar se puede decir que se aprendió sobre la empresa a la cual se le está desarrollando el programa, sobre las reglas de negocio que impone la sede central para el desarrollo de un nuevo Software. También se aprendió sobre el proceso que comprende el envío de la información de parte de los barcos para poder ponerla a disposición del cliente para que realice el pago.

La tarea más azarosa fue la de comprender los procesos que cumple cada área para llevar a cabo su trabajo de facturar en la base de datos "FIS". Esta tarea fue la que requirió más tiempo debido a que se debieron realizar muchas reuniones hasta comprender la totalidad de los pasos y posibles errores que se podían generar durante la ejecución.

En la etapa de desarrollo, se aprendió sobre el lenguaje de VBA, se investigó las posibles opciones que ofrece la programación de VBA, además de investigar las funciones que entregaba FIS para poder controlar sus componentes. También se aprendió cómo la programación de las macros de Excel se puede relacionar con otros programas de Microsoft Office, como Outlook o Access, esto permitiendo expandir los conocimientos de programas relacionados y pudiendo generar procesos automáticos sin la intervención de usuarios.

La implementación de los procesos que se llevaron a cabo fue exitosa, logrando un resultado satisfactorio para el cliente con lo cual se cumplen las expectativas que esperaba el cliente, realizando los procesos de facturación de manera correcta.

Como trabajo futuro se plantea seguir desarrollando iteraciones para incluir nuevos procesos de facturación a la macro, para poder ayudar más a las otras áreas encargadas y mejorar el tiempo en el cual se realiza la facturación en la base de datos "FIS".

5 Referencia Bibliográficas

- [1] [S/N] FIS 2 MACRO TRAINING, Hapag_Lloyd, Hamburgo, Alemania, 2015.
- [2] [J. M. Álvarez Palomo, D. Bueno Vallejo, J. Garralón Ruiz, J.A. Gómez Ruiz, P. López Olivas, L. Llopis Torres, F. Rus Mansilla, F.J. Veredas Navarro] *Un pseudolenguaje para la introducción a la programación*, E.T.S.I. Telecomunicación, Dpto. Lenguajes y Ciencias de la Computación, Universidad de Málaga, 2002

Anexos

A: Pseudocódigo Importación

NOMBRE: Impo

OBJETIVO: FACTURAS TODAS LAS FACTURAS YA PAGADAS QUE CORRESPONDAN A IMPORTACION

AREA: IMPORTACION

```
/'VARIABLES GLOBALES'/
```

```
Global hl
```

```
Global S8100_1
```

```
Global S8100A_2
```

```
Global S8100E_3
```

ALGORITMO Execute

```
Limpiar la planilla de excel
```

```
Limpiar colores de planilla de Excel
```

```
Sheet1.Cells.Range("A2:L999").Interior.ColorIndex = 0
```

```
LLAMAR LeeImpo
```

```
SI Sheet1.Cells(2, 2) <> "" And Sheet1.Cells(2, 3) <> "" ENTONCES
```

```
hl = CreateObject("HLBridge")
```

```
LLAMAR CheckMatchCode
```

```
LLAMAR GetRechargue
```

```
LLAMAR GetShipment
```

```
LLAMAR Check
```

```
LLAMAR InvoiceContainers
```

```
LLAMAR Mail.Send("Impo")
```

```
LLAMAR Database.SaveInvoiceImpo
```

```
LLAMAR hl.closesession
```

```
ASIGNAR hl = Nothing
```

```
FINSI
```

```
FIN
```

ALGORITMO Check

```
VAR
```

```
N row, separator /' N = Numero o Integer en VBA'/
```

```
C bl, matchcode /' C = Caracter o string en VBA'/
```

```
INICIO
```

```
Sheet1.Activate
```

```
Ordenar Hoja de excel (?)
```

```
CON Sheet1.Sort
```

```
.SortFields.Add Key:=Range("C:C"), SortOn:=xlSortOnValues,  
Order:=xlAscending, DataOption:=xlSortNormal
```

```
.SortFields.Add Key:=Range("H:H"), SortOn:=xlSortOnValues,  
Order:=xlAscending, DataOption:=xlSortNormal
```

```
.SetRange Range("A1:L999")
```

```
.Header = xlYes
```

```
.Apply
```

```
FIN CON
```

```

row = 2
separator = 7

MIENTRAS NO Sheet1.Cells(row, 2) = "" HACER

    bl = Sheet1.Cells(row, 3)
    matchcode = Sheet1.Cells(row, 8)

    Sheet1.Cells(row, 12) = separator
    SI InStr(Sheet1.Cells(row + 1, 3), bl) > 0 ENTONCES
        SI InStr(Sheet1.Cells(row + 1, 8), matchcode) = 0 ENTONCES
            separator = separator + 1
            SI separator = 10 ENTONCES
                separator = 7
            FINSI
        FINSI
    SINO
        separator = 7
    FINSI
    row = row + 1
REPETIR
FIN

```

ALGORITMO CheckMatchCode()

```

VAR
    N row
    C matchcode
    N row = 2
INICIO
    MIENTRAS NO Sheet1.Cells(row, 2) = "" HACER
        matchcode = Sheet1.Cells(row, 8)
        SI matchcode = "ND" ENTONCES
            LLAMAR WriteCell(row, "Match Code sin definir", 6)
        SINO
            SI matchcode = "" ENTONCES
                LLAMAR WriteCell(row, "Match Code vacio", 6)
            SINO
                MIENTRAS NO InStr(Sheet1.Cells(row, 8), " ") = 0 HACER
                    Sheet1.Cells(row, 8) = Replace(Sheet1.Cells(row, 8), " ", "")
                REPETIR
            FINSI
        FINSI
        row = row + 1
    REPETIR
FIN

```

ALGORITMO GetRecharge

```

/'Traducción de los recargos
'EDS = XNM
'GATE IN = XRD
'GATE OUT = XRO

```

```

\ /
VAR
  N row = 2
INICIO
  MIENTRAS NO Sheet1.Cells(row, 2) = "" HACER
    Sheet1.Cells(row, 9) = Trim(Sheet1.Cells(row, 9))
    SI Sheet1.Cells(row, 9) = "EDS" ENTONCES
      Sheet1.Cells(row, 9) = "XNM"
    SINO
      SI Sheet1.Cells(row, 9) = "GATE IN" ENTONCES
        Sheet1.Cells(row, 9) = "XRD"
      SINO
        SI Sheet1.Cells(row, 9) = "GATE OUT" ENTONCES
          Sheet1.Cells(row, 9) = "XRO"
          LLAMAR WriteCell(row, "Concepto no corresponde a IMPO", 6)
        FINSI
      FINSI
    FINSI
  row = row + 1
REPETIR
FIN

```

ALGORITMO GetShipment

```

VAR
  C S7230_1
  N row = 2
  C boleta, Contenedor, tipo, pago, Sql
INICIO
  /'PARA ABRIR LAS VENTANAS DE FIS SE USAN LOS SIGUIENTES COMANDOS'/
  /'EN CASO DE QUE YA ESTE ABIERTA LA VENTANA, ESTA FUNCIÓN ASIGNARA A LA
  VARIABLE EN NUMER DE VENTANA PARA MANEJAR LOS SIGUIENTES COMANDOS'/
  S7230_1 = hl.getApplicationId("S7230")
  /'EN CASO DE QUE LA APLICACION DEVUELVA VACIO, SIGNIFICA QUE NO ESTA ABIERTA Y
  SE DEBE ABRIR CON EL COMANDO OPENAPPLICATION'/
  SI S7230_1 = "" ENTONCES
    S7230_1 = hl.openApplication("S7230", "")
  FIN SI
  /'-----'/

  LLAMAR hl.setScreenLockingText("Facturando Impo")
  LLAMAR hl.proceedOnErrors(True)

  MIENTRAS NO Sheet1.Cells(row, 3) = "" And Sheet1.Cells(row, 2) = "" HACER
    bl = Sheet1.Cells(row, 3)

    LLAMAR hl.pressButton(S7230_1, "INTERNAL_CLEAR")

    /'Un error de que no deja hacer commit en la ventana genera el error'/
    SI hl.isDoYouWantToBoxOpened(S7230_1) ENTONCES
      LLAMAR hl.commitDoYouWantToBox(S7230_1, "NO")
    FIN SI

  LLAMAR hl.SetValue(S7230_1, "ESearchDocument_DocumentNo", bl)

```

```

LLAMAR hl.pressButton(S7230_1, "INTERNAL_FIND")

/'SI EL NÚMERO DE BL NO EXISTE, FIS ARROJARÁ UN ERROR Y SE CAPTURA DE LA
SIGUIENTE MANERA'/
SI hl.isErrorExists(S7230_1) ENTONCES
  LLAMAR WriteCell(row, "Error. Shipment no encontrado.", 3)
  /'NF SIGNIFICA NO ENCONTRADO'/
  Sheet1.Cells(row, 1) = "NF"
SINO
  /'GUARDA EL NUMERO DE SHIPMENT QUE OBTUVO MEDIANTE EN NUMERO DE BL'/
  Sheet1.Cells(row, 1) = hl.getFieldValue(S7230_1, "ESearchShipment_IdNumber")

  boleto = Sheet1.Cells(row, 2)
  Contenedor = Sheet1.Cells(row, 6)
  tipo = Sheet1.Cells(row, 9)
  pago = Sheet1.Cells(row, 10)
  Sql = ""

  /'OBTIENE LA INFORMACION QUE CONTIENE EL CUADRO DE TEXTO CON LOS
  COMENTARIOS'/
  aux = hl.getFieldValue(S7230_1, "EGeneralCounterText_Name")

  /'HLXU1234567-8_EDS_14/08/2018_CREDITO-Pagado_$85.000 FORMATO PARA LOS
  COMENTARIOS EN LA VENTANA S7230'/
  Sql = Contenedor & "_" & tipo

  /'EN CASO DE QUE LA CASILLA ESTE VACIA, ESCRIBIRA EL COMENTARIO'/
  SI Trim(aux) = "" ENTONCES
    SI pago = "0" Y tipo = "XNM" ENTONCES
      Sql = Sql & "_" & Date & "_CLIENTE EXENTO DE XNM_$" & pago & " -Nro
Boleta: " & Boleta
      SINO
        Sql = Sql & "_" & Date & "_PAGADO_$" & pago & " -Nro Boleta: " &
Boleta
      FIN SI
    LLAMAR hl.SetValue(S7230_1, "EGeneralCounterText_Name", Sql)
  SINO
    /'SI NO ESTA VACIA, LO AGREGARA AL FINAL DEL COMENTARIO ACTUAL'/
    SI InStr(aux, Sql) = 0 ENTONCE
      SI pago = "0" And tipo = "XNM" ENTONCES
        Sql = Sql & "_" & Date & "_CLIENTE EXENTO DE XNM_$" & pago & " -Nro
Boleta: " & Boleta
      SINO
        Sql = Sql & "_" & Date & "_PAGADO_$" & pago & " -Nro Boleta: " &
Boleta
      FIN SI

      Sql = aux & vbNewLine & Sql
      LLAMAR hl.SetValue(S7230_1, "EGeneralCounterText_Name", Sql)
    FINSI
  FINSI

LLAMAR hl.pressButton(S7230_1, "SaveButton")

```

```

    FINSI
    row = row + 1
REPETIR

    LLAMAR hl.pressButton(S7230_1, "INTERNAL_CLEAR")
    LLAMAR hl.pressButton(S7230_1, "CancelButton")
End Sub

ALGORITMO InvoiceContainers
VAR
    N row = 2
    C shipment, matchcode, container, recharge, separator, baserate
INICIO

    /'ABRIR VENTANA S8100 DE IGUAL MANERA QUE LA S7230'/
    S8100_1 = hl.getApplicationId("S8100")

    If S8100_1 = "" Then
        S8100_1 = hl.openApplication("S8100", "")
    End If
    /'-----'/
    LLAMAR hl.setScreenLockingText("Facturando Impo")
    LLAMAR hl.proceedOnErrors(True)

    MIENTRAS NO Sheet1.Cells(row, 1) = "" HACER
        /'ASIGNAR LAS VARIABLES INICIALES'/
        shipment = Sheet1.Cells(row, 1)
        matchcode = Sheet1.Cells(row, 8)
        container = Sheet1.Cells(row, 6)
        recharge = Sheet1.Cells(row, 9)
        baserate = Sheet1.Cells(row, 10)
        separator = Sheet1.Cells(row, 12)
        /'LA PRIMERA PARTE DE LA MACRO CAMBIA LA INFORMACION, LA SEGUNDA FACUTRA
        CADA SHIPMENT DISTINTO'/

        /'SI EL SHIPMENT NO EXISTE, FIS MOSTRARA UN ERROR Y ASI SE CAPTURA'/
        SI shipment = "NF" ENTONCES
            LLAMAR WriteCell(row, "Shipment no encontrado", 3)
        SINO
            SI matchcode = "" O matchcode = "ND" ENTONCES
                LLAMAR WriteCell(row, "Matchcode No Definido", 6)
            SINO
                LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")

            SI hl.isDoYouWantToBoxOpened(S8100_1) ENTONCES
                LLAMAR hl.commitDoYouWantToBox(S8100_1, "NO")
            SINO
                LLAMAR hl.SetValue(S8100_1, "EShipment_IdNumber", shipment)
                LLAMAR hl.pressButton(S8100_1, "INTERNAL_FIND")

            /'MANERA DE MANEJAR LOS ERRORES CUANDO EL SHIPMENT ESTA SPLIT'/
            s8000 = hl.getApplicationId("S8000")
            SI s8000 <> "" ENTONCES

```



```

        LLAMAR hl.closeApplication(s8000)
        LLAMAR WriteCell(row, "Shipment Splited", 3)
        S8100_1 = hl.openApplication("S8100", "")
        LLAMAR hl.setScreenLockingText("Facturando Impo")
    SINO
        /'
        SI hl.isErrorExists(S8100_1) ENTONCES
            LLAMAR WriteCell(row, hl.getFirstErrorMessage(S8100_1), 3)
        SINO
            LLAMAR SetContainer(matchcode, container, recharge, separator,
baseerate, row)
        FINSI
        FINSI
        FINSI
        FINSI
        FINSI
        row = row + 1
    REPETIR
'=====
row = 2

MIENTRAS NO Sheet1.Cells(row, 1) = "" HACER
    shipment = Sheet1.Cells(row, 1)
    matchcode = Sheet1.Cells(row, 8)
    result = Sheet1.Cells(row, 11)
    separator = Sheet1.Cells(row, 12)

    SI Sheet1.Cells(row, 11) = "Completed" ENTONCES
        LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")
        LLAMAR hl.SetValue(S8100_1, "EShipment_IdNumber", shipment)
        LLAMAR hl.pressButton(S8100_1, "INTERNAL_FIND")

        /'PRESIONAR DISTRIBUITE'/
        LLAMAR hl.pressButton(S8100_1, "BtnCreateInvoice")

    SI hl.isErrorExists(S8100_1) ENTONCES
        LLAMAR WriteCell(row, hl.getFirstErrorMessage(S8100_1), 3)
    SINO
        LLAMAR SelectInvoices(row, separator, matchcode)
    FIN SI
FINSI

MIENTRAS NO i = 2 HACER
    result = Sheet1.Cells(row, 11)
    /'ESCRIBE EL RESULTADO FINAL EN CADA SHIPMENT QUE SE HAYA COMPLETADO
    LA FACTURA'/
    SI InStr(Sheet1.Cells(row + 1, 1), shipment) > 0 Y
        InStr(Sheet1.Cells(row + 1, 8), matchcode) > 0 ENTONCES
        SI InStr(result, "Error") = 0 Then
            Sheet1.Cells(row + 1, 11) = result
        FINSI
        row = row + 1
    SINO

```

```

        SALIR MIENTRAS
    FINSI
    REPETIR

    row = row + 1
    REPETIR

    LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")
    LLAMAR hl.pressButton(S8100_1, "closeButton")
FIN

```

ALGORITMO SetContainer

ENTRADA(C matchcode, container, recharge, separator, baserate, N excel)

INICIO

```

    EN CASO DE ERROR IR A err_handler
    LLAMAR hl.selectTabForWindow(S8100_1, 4)
    S8100E_3 = hl.getTabApplicationId(S8100_1, 4)

    DESDE i = 1 HASTA hl.getFieldValue(S8100_1, "ERevContTotal_ShortInteger")
        //ESTA VALIDA SI CORRESPONDE AL CONTENEDOR \
        SI ValidateCtr(container) = True ENTONCES
            SI hl.getTableRowsNumber(S8100E_3, "EListRevenueComponents") = 0 ENTONCES
                LLAMAR WriteCell(excel, "Lista de Recargos Vacía", 6)
                SALIR
            SINO
                arr = hl.getTableColumnValuesByColumnName(S8100E_3,
                    "EListRevenueComponents", "ELineRevChargeType_ChargeCode")
                arr_status = hl.getTableColumnValuesByColumnName(S8100E_3,
                    "EListRevenueComponents", "ELineRevComponent_StateInvoicing")
                arr_freight = hl.getTableColumnValuesByColumnName(S8100E_3,
                    "EListRevenueComponents", "ELineFcOrg_MatchcodeName")
                arr_rate = hl.getTableColumnValuesByColumnName(S8100E_3,
                    "EListRevenueComponents", "ELineRevComponent_BaseRate")

                DESDE row = 0 HASTA UBound(arr)
                    'check if the row column have the same recharge
                    SI arr(row) = recharge ENTONCES
                        //REVISAR SI EL RECARGO YA ESTA FACTURADO//
                        SI arr_status(row) <> "I" ENTONCES
                            //CALCULA EL VALOR FINAL DEL PAGO SUMANDO EL IVA DEL 19%//
                            monto = Split(arr_rate(row), ".")(0) + (Split(arr_rate(row),
                                ".")(0) * 19) / 100
                            SI InStr(monto, baserate) > 0 ENTONCES
                                //ESTA FUNCION ES PARA MODIFICAR UN CONTENEDOR (EL ACTUAL)//
                                LLAMAR hl.SetValue(S8100E_3, "EListLevel", "THIS")

                                //ASIGNA LOS VALORES A EL RECARGO//
                                LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
                                    "EListRevenueComponents", row, "ELineFpOrg_MatchcodeName", matchcode)
                                LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
                                    "EListRevenueComponents", row, "ELineRevComponent_VatSign", "A4")

```

```

        LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineRevComponent_GroupForInvoice", separator)
        LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineRevComponent_StateInvoicing", "F")

        LLAMAR hl.pressButton(S8100E_3, "BtnSave")
        LLAMAR WriteCell(excel, "Completed", 4)
        IR A err_handler

    SINO
        LLAMAR WriteCell(excel, "Error. Diferencias de monto entre cupon
y FIS", 10)
        IR A err_handler
    FINSI
SINO
    LLAMAR WriteCell(excel, "Ya Facturado", 6)
    IR A err_handler
FINSI
SINO
    LLAMAR WriteCell(excel, "Rechargue No Encontrado", 6)
    IR A err_handler
FINSI
FINSI
SIGUIENTE row

    LLAMAR WriteCell(excel, "Rechargue No Encontrado", 6)
    IR A err_handler
FINSI
FINSI
/'EN CASO DE QUE NO SEA EL CONTENEDOR QUE SE BUSCA, SE PROCEDE A BUSCAR EL
SIGUIENTE CONTENEDOR DE LA LISTA'/
SI hl.isComponentEnabled(S8100_1, "BtnNext") ENTONCES
    LLAMAR hl.pressButton(S8100_1, "BtnNext")
FINSI
SIGUIENTE i

/'SI RECORRE TODOS LOS CONTAINERS, SIGNIFICA QUE NO LO ENCONTROO/
LLAMAR WriteCell(excel, "Container No Encontrado", 6)

err_handler:
    LLAMAR hl.proceedOnErrors(True)
    SI hl.isErrorExists(S8100E_3) ENTONCES
        LLAMAR WriteCell(excel, hl.getFirstErrorMessage(S8100E_3), 3)
    FINSI

    LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")

    SI hl.isDoYouWantToBoxOpened(S8100E_3) ENTONCES
        LLAMAR hl.commitDoYouWantToBox(S8100E_3, "NO")
    FINSI
FIN

```

ALGORITMO ValidateCtr
ENTRADA (C container)
SALIDA (B ValidateCtr)
VAR
C ctr

```

INICIO
  //OBTENER EL NUMERO DEL ACTUAL CONTENEDOR//
  ctr = hl.getFieldValue(S8100E_3, "ERevCont_ContainerNumber")

  //LIMPIAR DE ESPACIOS EL NOMBRE DEL CONTENEDOR DE FIS//
  MIENTRAS NO InStr(ctr, " ") = 0 HACER
    ctr = Replace(ctr, " ", "")
  REPETIR

  SI ctr = container ENTONCES
    ValidateCtr = True
  SINO
    ValidateCtr = False
  FINSI
FIN

ALGORITMO SelectInvoices
  ENTRADA (N excel, C separator, C matchcode)
VAR
  C F0030_5
INICIO
  F0030_5 = hl.getApplicationId("F0030")

  SI F0030_5 <> "" ENTONCES 'F0030 was opened
    DESDE i = 0 HASTA hl.getTableRowsNumber(F0030_5,
"EListDocsShipsToBeInvoiced") - 1
      SI hl.getTableCellValueByColumnName(F0030_5, "EListDocsShipsToBeInvoiced",
i, "ELineRevenue_GroupForInvoice") = separator ENTONCES

        arr = hl.getTableRowValues(F0030_5, "EListDocsShipsToBeInvoiced", i)

        DESDE j = 0 HASTA UBound(arr)
          aux = Replace(arr(j), " ", "")
          SI aux = matchcode ENTONCES
            LLAMAR hl.selectTableRow(F0030_5, "EListDocsShipsToBeInvoiced", i)
            LLAMAR hl.pressButton(F0030_5, "CreateInvoice")
            LLAMAR CreateInvoice(excel)
            LLAMAR hl.closeApplication(F0030_5)
            SALIR
          FINSI
        SIGUIENTE j
      FINSI
    SIGUIENTE i
    LLAMAR WriteCell(excel, "Matchcode no encontrado en ventana F0030", 6)
  SINO
    LLAMAR CreateInvoice(excel)
  FINSI
FIN

ALGORITMO CreateInvoice(excel As Integer)
  ENTRADA (N excel)
VAR
  C F0020_2

```

```

C F0020A_3
C F0020B_3
INICIO
F0020_2 = hl.getApplicationId("F0020")

SI F0020_2 = "" ENTONCES
  LLAMAR WriteCell(excel, "Error en la Ventana F0030", 6)
  SALIR
FINSI

F0020A_3 = hl.getTabApplicationId(F0020_2, 1)

LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineAllRoes_Flag", "Y")
LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineRevenueComponent_FirstConvRoeInv", "")
LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineRevenueComponent_FirstConvCurInv", "")

LLAMAR hl.pressButton(F0020A_3, "btnSave")

/'CAMBIAR A LA PESTAÑA HEADER'/
LLAMAR hl.selectTabForWindow(F0020_2, 0)
F0020B_3 = hl.getTabApplicationId(F0020_2, 0)

LLAMAR hl.SetValue(F0020B_3, "EInvoice_InvoiceCurrency", "CLP")

LLAMAR hl.pressButton(F0020B_3, "btnSave")

LLAMAR hl.pressButton(F0020_2, "btnDistribute")
LLAMAR WriteCell(excel, "Facturacion Completa", 4)
SALIR

err_handler:
  LLAMAR WriteCell(excel, "Error Facturando los Contenedores", 6)
FIN

```

ALGORITMO WriteCell

```

  ENTRADA ( N row, C data, N color)
INICIO
  Sheet1.Cells(row, 11) = data
  Sheet1.Cells(row, 11).Interior.ColorIndex = color
FIN

```

B: Pseudocódigo Exportación

NOMBRE: Expo

OBJETIVO: FACTURAS TODAS LAS FACTURAS YA PAGADAS QUE CORRESPONDAN A EXPORTACION

AREA: EXPORTACION

```
/'VARIABLES GLOBALES'/
```

```
Global hl
```

```
Global S8100_1
```

```
Global S8100A_2
```

```
Global S8100E_3
```

ALGORITMO Execute

INICIO

```
LIMPIAR CONTENIDOS DE LA PLANILLA
```

```
LIMPIAR COLORES DE LA PLANILLA
```

```
LLAMAR LeeExpo
```

```
LLAMAR InvoiceContainersExpo
```

```
LLAMAR Mail.Send("Expo")
```

```
LLAMAR Database.SaveInvoiceExpo
```

FIN

ALGORITMO InvoiceContainersExpo

VAR

```
C shipment, matchcode, oc, tipo, codigokhipu, total, fecha, S8000_1
```

```
N cant, row
```

INICIO

```
row = 2
```

```
Set hl = CreateObject("HLBridge")
```

```
'Test if FIS 2 is open
```

```
SI NO initBridge(True) ENTONCES
```

```
FIN ALGORITMO
```

```
FINSI
```

```
S8100_1 = getAppId("S8100")
```

```
LLAMAR hl.setScreenLockingText("Facturando Expo")
```

```
LLAMAR hl.proceedOnErrors(True)
```

```
MIENTRAS NO Sheet3.Cells(row, 1) = ""
```

```
shipment = Sheet3.Cells(row, 1)
```

```
oc = Sheet3.Cells(row, 4)
```

```
matchcode = Sheet3.Cells(row, 5)
```

```
tipo = Sheet3.Cells(row, 7)
```

```
cant = Sheet3.Cells(row, 8)
```

```
codigokhipu = Sheet3.Cells(row, 9)
```

```
total = Sheet3.Cells(row, 10)
```

```
fecha = Sheet3.Cells(row, 11)
```

```
SI matchcode = "VINACO6" O matchcode = "VINACO23" ENTONCES
```

```

LLAMAR WriteCell(row, "Completed. Matchcode for Concha y Toro", 4)
IR A Nxt
FINSI

SI matchcode <> "" Y matchcode <> "ND" ENTONCES

Call hl.pressButton(S8100_1, "INTERNAL_CLEAR")
Call hl.SetValue(S8100_1, "EShipment_IdNumber", shipment)
Call hl.pressButton(S8100_1, "INTERNAL_FIND")

/'ver lo que pasa cuando es un split
s8000 = hl.getApplicationId("S8000")
SI s8000 <> "" ENTONCES
    LLAMAR hl.closeApplication(s8000)
    LLAMAR WriteCell(row, "Shipment Splited", 3)
    S8100_1 = hl.openApplication("S8100", "")
    IR A nxt
FINSI

/'if the ship is not found, this will be caught it and write the error in
the excel sheet and not procees the data'/
SI hl.isErrorExists(S8100_1) ENTONCES
    LLAMAR WriteCell(row, hl.getFirstErrorMessage(S8100_1), 3)
/'if the ship exist and dont throw error, proceed to the rest of the
code'/
SINO
    SI hl.getCheckBoxState(S8100_1, "EShipment_CancelledFlag") = True
ENTONCES
    LLAMAR WriteCell(row, "Shipment Cancelled", 6)
SINO
    LLAMAR hl.selectTabForWindow(S8100_1, 4)
    S8100E_3 = hl.getTabApplicationId(S8100_1, 4)

    fiscontainer = hl.getFieldValue(S8100_1,
"ERevContTotal_ShortInteger")

    SI fiscontainer < cant ENTONCES
        LLAMAR WriteCell(row, "Number of containers to be invoiced is more
than the containers in FIS", 3)
    SINO
        SI fiscontainer = cant ENTONCES
            LLAMAR SetContainerAll(matchcode, "XRO", oc, "HAPAGL617", tipo,
cant, codigokhipu, total, fecha, row)
        SINO
            LLAMAR SetContainer(matchcode, "XRO", oc, "HAPAGL617", tipo,
cant, codigokhipu, total, fecha, row)
        FINSI
    FINSI
FINSI
FINSI
FINSI
LLAMAR WriteCell(row, "Matchcode not defined", 6)
FINSI
nxt:
    row = row + 1
REPETIR

```

```

LLAMAR hl.SetValue(S8100_1, "EShipment_IdNumber", "")
LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")

LLAMAR hl.pressButton(S8100_1, "closeButton")
LLAMAR hl.closesession
SET hl = Nothing
FIN

ALGORITMO SetContainer
ENTRADA( C matchcode, recharge , oc, freight, tipo, codigokhipu, monto, fecha,
N cant, excel)

VAR
  N i
  B flag

INICIO
  EN ERROR IR A err_handler

  i = 0

  HASTA QUE i = cant
    SI hl.getTableRowsNumber(S8100E_3, "EListRevenueComponents") <> 0 ENTONCES

      arr = hl.getTableColumnValuesByColumnName(S8100E_3,
"EListRevenueComponents", "ELineRevChargeType_ChargeCode")
      arr_status = hl.getTableColumnValuesByColumnName(S8100E_3,
"EListRevenueComponents", "ELineRevComponent_StateInvoicing")
      arr_freight = hl.getTableColumnValuesByColumnName(S8100E_3,
"EListRevenueComponents", "ELineFcOrg_MatchcodeName")

      DESDE row = 0 HASTA UBound(arr)
        /'CHEQUEA QUE SEA EL MISMO RECARGO'/
        SI arr(row) = recharge ENTONCES
          /'CHEQUEA SI EL RECARGO YA ESTA FACTURADO'/
          SI arr_status(row) <> "I" ENTONCES
            /'ESTO CAMBIA QUE LA MODIFICACION SOLO SEA PARA EL CONTENEDOR
ACTUAL'/
              LLAMAR hl.SetValue(S8100E_3, "EListLevel",
"THIS")

              /'ASIGNA LOS VALORES AL RECARGO'/
              LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineFpOrg_MatchcodeName", matchcode)
              LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineRevComponent_VatSign", "A4")
              LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineRevComponent_GroupForInvoice", separator)
              LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineRevComponent_StateInvoicing", "F")

              /'SELECCIONA LAS FACTURAS'/
              LLAMAR hl.pressButton(S8100E_3, "BtnSave")
              LLAMAR hl.selectTableRow(S8100E_3, "EListRevenueComponents", row)
              i = i + 1
              flag = False
            SINO

```



```

        flag = True
    FINSI
    IR A nxt
FINSI
SIGUIENTE row
LLAMAR WriteCell(excel, "Rechargue Not Found!", 6)
IR A err_handler
SINO
    LLAMAR WriteCell(excel, "No Rechargues", 6)
    IR A err_handler
FINSI
nxt:
/'CHEQUEA SI EL BOTON NEXT ESTA DISPONIBLE'/
SI hl.isComponentEnabled(S8100_1, "BtnNext") ENTONCES
    LLAMAR hl.pressButton(S8100_1, "BtnNext")
SINO
    SI i <> cant ENTONCES
        SI flag = True ENTONCES
            LLAMAR WriteCell(excel, "Already Invoiced", 3)
        SINO
            LLAMAR WriteCell(excel, "Not enough containers", 3)
        FINSI
    IR A err_handler
FINSI
FINSI
REPETIR

/'ASIGNA LA ORDEN DE COMPRA EN EL CAMPO 811'/
SI oc <> "X" Y oc <> "" ENTONCES
    SI Len(oc) < 18 ENTONCES
        LLAMAR SetQuickBooking(oc)
        LLAMAR hl.pressButton(S8100E_3, "BtnRefresh")
    SINO
        LLAMAR WriteCell(excel, "error in the OC", 3)
        IR A err_handler
    FINSI
FINSI

LLAMAR hl.pressButton(S8100_1, "BtnCreateInvoice")
SI hl.isErrorExists(S8100E_3) ENTONCES
    IR A err_handler
SINO
    LLAMARl SelectInvoices(excel)
FINSI

err_handler:
LLAMAR hl.proceedOnErrors(True)
SI hl.isErrorExists(S8100E_3) ENTONCES
    LLAMAR WriteCell(excel, hl.getFirstErrorMessage(S8100E_3), 3)
FINSI

/'agregar funcion para agregar el remarks con los datos'/
LLAMAR Remarks(rechargue, monto, codigokhipu, tipo, cant, fecha)
LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")

SI hl.isDoYouWantToBoxOpened(S8100E_3) ENTONCES
    LLAMAR hl.commitDoYouWantToBox(S8100E_3, "NO")

```

```
FINSI
End Sub
```

ALGORITMO SetContainerAll

ENTRADA(C matchcode, recharge , oc, freight, tipo, codigokhipu, monto, fecha, N cant, excel)

INICIO

```
EN ERROR IR A GoTo err_handler
```

```
SI hl.getTableRowsNumber(S8100E_3, "EListRevenueComponents") <> 0 ENTONCES
```

```
arr = hl.getTableColumnValuesByColumnName(S8100E_3,
```

```
"EListRevenueComponents", "ELineRevChargeType_ChargeCode")
```

```
arr_status = hl.getTableColumnValuesByColumnName(S8100E_3,
```

```
"EListRevenueComponents", "ELineRevComponent_StateInvoicing")
```

```
arr_freight = hl.getTableColumnValuesByColumnName(S8100E_3,
```

```
"EListRevenueComponents", "ELineFcOrg_MatchcodeName")
```

```
For row = 0 To UBound(arr)
```

```
/'CHEQUEA SI A COLUMNA TIENE EL MISMO RECARGO'/
```

```
SI arr(row) = recharge ENTONCES
```

```
/'CHEQUEA SI EL RECARGO YA ESTA FACTURADO'/
```

```
SI arr_status(row) <> "I" ENTONCES
```

```
/'ESTO ES PARA QUE EL CAMBIO SEA NIVEL CONTENEDOR ACTUAL'/1
```

```
LLAMAR hl.SetValue(S8100E_3, "EListLevel", "ALL")
```

```
/'ASIGNA LOS VALORES EN EL RECARGO'/
```

```
LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
```

```
"EListRevenueComponents", row, "ELineFpOrg_MatchcodeName", matchcode)
```

```
LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
```

```
"EListRevenueComponents", row, "ELineRevComponent_VatSign", "A4")
```

```
LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
```

```
"EListRevenueComponents", row, "ELineRevComponent_GroupForInvoice", separator)
```

```
LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
```

```
"EListRevenueComponents", row, "ELineRevComponent_StateInvoicing", "F")
```

```
LLAMAR hl.pressButton(S8100E_3, "BtnSave")
```

```
LLAMAR hl.selectTableRow(S8100E_3, "EListRevenueComponents", row)
```

```
IR A nxt
```

```
SINO
```

```
LLAMAR WriteCell(excel, "Already Invoiced", 6)
```

```
IR A err_handler
```

```
FINSI
```

```
FINSI
```

```
SIGUIENTE row
```

```
LLAMAR WriteCell(excel, "Recharge Not Found", 6)
```

```
IR A err_handler
```

```
SINO
```

```
LLAMAR WriteCell(excel, "No Recharges in the First", 6)
```

```
IR A err_handler
```

```
nxt:
```

```
/'ASIGNA LA OC EN EL CAMPO 811'/
```

```
SI oc <> "X" Y oc <> "" ENTONCES
```

```
SI Len(oc) < 18 ENTONCES
```

```
LLAMAR SetQuickBooking(oc)
```

```
LLAMAR hl.pressButton(S8100E_3, "BtnRefresh")
```

```

        SINO
            LLAMAR WriteCell(excel, "error in the OC", 3)
            IR A err_handler
        FINSI
    FINSI

    LLAMAR hl.pressButton(S8100_1, "BtnCreateInvoice")
    SI hl.isErrorExists(S8100E_3) ENTONCES
        IR A err_handler
    SINO
        LLAMAR SelectInvoices(excel)
    FINSI

err_handler:
    LLAMAR hl.proceedOnErrors(True)
    SI hl.isErrorExists(S8100E_3) ENTONCES
        LLAMAR WriteCell(excel, hl.getFirstErrorMessage(S8100E_3), 3)
    FINSI

    /'agregar funcion para agregar el remarks con los datos'/
    LLAMAR Remarks(rechargue, monto, codigokhipu, tipo, cant, fecha)

    LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")

    SI hl.isDoYouWantToBoxOpened(S8100E_3) ENTONCES
        LLAMAR hl.commitDoYouWantToBox(S8100E_3, "NO")
    FINSI
FIN

ALGORITMO Remarks
ENTRADA (C rechargue, monto, codigokhipu, tipo, fecha, N cant)

VAR
    C str, S8100K_2
INICIO
    str = "Fecha = " & fecha & " - Concepto = GATE OUT - " & tipo & " - PAGADO" &
vbNewLine
    str = str & "Monto = " & monto & vbNewLine
    str = str & "Codigo Khipu = " & codigokhipu & " - Cantidad Contenedores = " &
cant & " - Portal"

    LLAMAR hl.selectTabForWindow(S8100_1, 5)
    S8100K_2 = hl.getTabApplicationId(S8100_1, 5)

    HACER HASTA hl.isComponentEnabled(S8100K_2, "HLBtnNextExportRemark") = False

        aux = hl.getFieldValue(S8100K_2, "EGrpExportLines")

        /'EN CASO DE QUE SE REPITA LA LINEA CON EL COMENTARIO, SALDRA DE LA
FUNCION'/
        SI InStr(aux, UCase(codigokhipu)) > 0 Y InStr(aux, "- PORTAL") > 0 ENTONCES
            SI InStr(aux, tipo) > 0 ENTONCES
                SALIR
            SINO
                aux = hl.getFieldValue(S8100K_2, "EGrpExportLines") & vbNewLine & "(---
)" & vbNewLine & str

```

```

        LLAMAR hl.SetValue(S8100K_2, "EGrpExportLines", aux)
        LLAMAR hl.pressButton(S8100K_2, "HLBtnSaveRemarks")
        Exit Sub
    FINSI
FINSI

    LLAMAR hl.pressButton(S8100K_2, "HLBtnNextExportRemark")
REPETIR

SI Trim(hl.getFieldValue(S8100K_2, "EGrpExportLines")) = "" ENTONCES
    LLAMAR hl.SetValue(S8100K_2, "EGrpExportLines", str)
FINSI
LLAMAR hl.pressButton(S8100K_2, "HLBtnSaveRemarks")
LLAMAR hl.selectTabForWindow(S8100_1, 4)
FIN

```

ALGORITMO SetQuickBooking

ENTRADA (C HES)he quick booking windows and set the HES to external reference

```

VAR
    C S8020_2
INICIO
    LLAMAR hl.pressButton(S8100_1, "quickBookingButton")

    S8020_2 = hl.getApplicationId("S8020")

    LLAMAR hl.SetValue(S8020_2, "EReference_ReferenceString", "")
    LLAMAR hl.pressButton(S8020_2, "saveButton")

    LLAMAR hl.SetValue(S8020_2, "EReference_ReferenceQualifier", "PONO")
    LLAMAR hl.setRadioButtonGroupState(S8020_2, "EReferenceLevel_Integer", 1)
    LLAMAR hl.SetValue(S8020_2, "EReference_ReferenceString", HES)

    LLAMAR hl.pressButton(S8020_2, "saveButton")
    LLAMAR hl.pressButton(S8020_2, "closeButton")
FIN

```

ALGORITMO SelectInvoices

ENTRADA (N excel)

```

VAR
    C F0030_5
INICIO
    F0030_5 = hl.getApplicationId("F0030")

    SI F0030_5 <> "" ENTONCES /'LA VENTANA F0030 FUE ABIERTA'/
        DESDE i = 0 HASTA hl.getTableRowsNumber(F0030_5,
"EListDocsShipsToBeInvoiced") - 1
            SI hl.getTableCellValueByColumnName(F0030_5, "EListDocsShipsToBeInvoiced",
i, "ELineRevenue_GroupForInvoice") = separator ENTONCES
                LLAMAR hl.selectTableRow(F0030_5, "EListDocsShipsToBeInvoiced", i)
                LLAMAR hl.pressButton(F0030_5, "CreateInvoice")
                LLAMAR CreateInvoice(excel)
                LLAMAR hl.closeApplication(F0030_5)
                SALIR
            FINSI
        SIGUIENTE i

```

```
FINSI
LLAMAR CreateInvoice(excel)
FIN
```

```
ALGORITMO CreateInvoice
ENTRADA (N excel)
```

```
VAR
  C F0020_2, F0020A_3, F0020B_3, payment
```

```
INICIO
  EN ERROR IR A err_handler
```

```
F0020_2 = hl.getApplicationId("F0020")
```

```
SI F0020_2 = "" ENTONCES
  LLAMAR WriteCell(excel, "Error in Windows F0030", 6)
  SALIR
```

```
FINSI
```

```
F0020A_3 = hl.getTabApplicationId(F0020_2, 1)
```

```
/'ACTUALIZA LOS DATOS A TODOS LOS RECARGOS DE LA LISTA'/
LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineAllRoes_Flag", "Y")
LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineRevenueComponent_FirstConvRoeInv", "")
LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineRevenueComponent_FirstConvCurInv", "")
```

```
LLAMAR hl.pressButton(F0020A_3, "btnSave")
```

```
/'IR A LA PESTAÑA "header"'/
LLAMAR hl.selectTabForWindow(F0020_2, 0)
F0020B_3 = hl.getTabApplicationId(F0020_2, 0)
```

```
payment = hl.getFieldValue(F0020B_3, "EListPaymentTerms")
LLAMAR hl.SetValue(F0020B_3, "EInvoice_InvoiceCurrency", "CLP")
```

```
LLAMAR hl.pressButton(F0020B_3, "btnSave")
```

```
LLAMAR hl.pressButton(F0020_2, "btnDistribute")
LLAMAR WriteCell(excel, "Completed - Payment Terms: " & payment, 4)
SALIR
```

```
err_handler:
  LLAMAR WriteCell(excel, "Error Invoicing Container", 6)
  LLAMAR hl.closeApplication(F0020_2)
FIN
```

```
ALGORITMO WriteCell
ENTRADA (N row, color, C data)
```

```
INICIO
  Sheet3.Cells(row, 12) = data
  Sheet3.Cells(row, 12).Interior.ColorIndex = color
FIN
```

ALGORITMO LeeExpo

VAR

```
O cn
O rs
C strSql
C strConnection
N row
N column
```

INICIO

row = 2

```
cn = CreateObject("ADODB.Connection")
strConnection = "Provider=Microsoft.ACE.OLEDB.12.0;Mode=Read;" & _
  "Data Source=" & DatabaseLocation
strSql = "SELECT DISTINCT HALOCL_CRUCE_MES_EXPO.EMBARQUE,
HALOCL_CRUCE_MES_EXPO.ORDEN_SERVICIO, "
strSql = strSql & " HALOCL_CRUCE_MES_EXPO.RUT_A_FACTURAR,
HALOCL_CRUCE_MES_EXPO.ORDEN_COMPRA, HALOCL_CRUCE_MES_EXPO.MATCHCODE, "
strSql = strSql & " HALOCL_CRUCE_MES_EXPO.ESTADO, "
HALOCL_CRUCE_MES_EXPO.DETALLE_CUPON, HALOCL_CRUCE_MES_EXPO.CANTIDAD,
HALOCL_CRUCE_MES_EXPO.CODIGO_KHIPU, HALOCL_CRUCE_MES_EXPO.TOTAL_FINAL,
HALOCL_CRUCE_MES_EXPO.FECHA_PAGO "
strSql = strSql & " FROM HALOCL_CRUCE_MES_EXPO LEFT JOIN ExportKhipuFacturados
ON HALOCL_CRUCE_MES_EXPO.[ORDEN_SERVICIO] =
ExportKhipuFacturados.[ORDEN_SERVICIO] "
strSql = strSql & " WHERE (((HALOCL_CRUCE_MES_EXPO.ESTADO)='PAGADO') AND
((ExportKhipuFacturados.ORDEN_SERVICIO) Is Null) AND
((HALOCL_CRUCE_MES_EXPO.CODIGO_KHIPU) Is Not Null));"
cn.Open strConnection
rs = cn.Execute(strSql)
MIENTRAS NO rs.EOF
  column = 1
  DESDE i = 0 HASTA rs.Fields.Count - 1
    Sheet3.Cells(row, column) = rs.Fields(i).Value
    column = column + 1
  SIGUIENTE
  rs.MoveNext
  row = row + 1
VUELTA

On Error Resume Next
rs.Close
rs = Nothing
cn.Close
cn = Nothing
```

FIN

ALGORITMO SaveInvoiceExpo

VAR

```
L shipment //L ES EQUIVALENTE A LONG EN VBA//
L service
C rut
C oc
```

```

C matchcode
C stats
N row

INICIO
row = 2

HACER HASTA Sheet3.Cells(row, 1) = ""
shipment = Sheet3.Cells(row, 1)
service = val(Sheet3.Cells(row, 2))
rut = Sheet3.Cells(row, 3)
oc = Sheet3.Cells(row, 4)
matchcode = Sheet3.Cells(row, 5)
stats = Sheet3.Cells(row, 6)

SI InStr(Sheet3.Cells(row, 12), "Completed") > 0 O Sheet3.Cells(row, 12) =
"Not enough containers" Or Sheet3.Cells(row, 12) = "Already Invoiced" O
Sheet3.Cells(row, 12) = "Shipment Cancelled" ENTONCES
Call GrabaFacturadoExpo(shipment, service, rut, oc, matchcode, stats)
FINSI

row = row + 1
Loop
FIN

```

ALGORITMO GrabaFacturadoExpo

ENTRADA (L embarque, L servicio, C rut, C compra, C matchcode, C estado)

VAR

```

O cn
O rs
C strSql
C strConnection

```

ENTRADA

```

cn = CreateObject("ADODB.Connection")
strConnection = "Provider=Microsoft.ACE.OLEDB.12.0;" & _
"Data Source=" & DatabaseLocation
strSql = "INSERT INTO ExportKhipuFacturados"
strSql = strSql & " SELECT *"
strSql = strSql & " FROM HALOCL_CRUCE_MES_EXPO "
strSql = strSql & " WHERE HALOCL_CRUCE_MES_EXPO.EMBARQUE = " & embarque
strSql = strSql & " AND HALOCL_CRUCE_MES_EXPO.ORDEN_SERVICIO = " & servicio
strSql = strSql & " AND HALOCL_CRUCE_MES_EXPO.RUT_A_FACTURAR = '" & rut & "'"
strSql = strSql & " AND HALOCL_CRUCE_MES_EXPO.matchcode = '" & matchcode &
""
strSql = strSql & " AND HALOCL_CRUCE_MES_EXPO.estado = '" & estado & "'"

SI Trim(compra) = "" ENTONCES
strSql = strSql & " AND HALOCL_CRUCE_MES_EXPO.ORDEN_COMPRA IS NULL;"
SINO
strSql = strSql & " AND HALOCL_CRUCE_MES_EXPO.ORDEN_COMPRA = '" & compra &
"";
FINSI

```

```
EN ERROR IR A Resume Next
cn.Open strConnection
rs = cn.Execute(strSql)
rs = Nothing
cn.Close
cn = Nothing
FIN
```


C: Pseudocódigo Bolivia

NOMBRE: BOLIVIA

OBJETIVO: FACTURAS TODAS LAS FACTURAS YA PAGADAS QUE CORRESPONDAN A BOLIVIA

AREA: BOLIVIA

```
/'VARIABLES GLOBALES'/
```

```
CONST separator = 5
```

```
Global hl
```

```
Global S8100_1
```

```
Global S8100A_2
```

```
Global S8100E_3
```

ALGORITMO Complete

INICIO

```
LIMPIAR HOJA
```

```
AGREGAR HOJA PASO /'Sheets.Add.name = "Paso"'/
```

```
LLAMAR CSV
```

```
LLAMAR TextToColumn
```

```
LLAMAR SetData
```

```
ELIMINAR HOJA PASO /'Sheets("Paso").Delete'/'
```

```
LLAMAR GetShipment
```

```
LLAMAR InvoiceContainers
```

```
LLAMAR Send("Bolivia")
```

FIN

ALGORITMO GetShipment

VAR

```
C S7230_1
```

```
N row
```

INICIO

```
row = 2
```

```
hl = CreateObject("HLBridge")
```

```
/'PRUEBA SI FIS ESTA ABIERTO'/'
```

```
SI Not initBridge(True) ENTONCES
```

```
SALIR
```

```
FINSI
```

```
S7230_1 = hl.openApplication("S7230", "")
```

```
LLAMAR hl.setScreenLockingText("Facturando Bolivia")
```

```
LLAMAR hl.proceedOnErrors(True)
```

```
REPETIR HASTA Sheet1.Cells(row, 3) = ""
```

```
bl = Sheet1.Cells(row, 3)
```

```

LLAMAR hl.pressButton(S7230_1, "INTERNAL_CLEAR")

SI hl.isDoYouWantToBoxOpened(S7230_1) ENTONCES
    LLAMAR hl.commitDoYouWantToBox(S7230_1, "NO")
FINSI

LLAMAR hl.SetValue(S7230_1, "ESearchDocument_DocumentNo", bl)
LLAMAR hl.pressButton(S7230_1, "INTERNAL_FIND")

SI hl.isErrorExists(S7230_1) ENTONCES
    LLAMAR WriteCell(row, "Shipment not found.", 3)
    Sheet1.Cells(row, 1) = "NF"/'NF SIGNIFICA QUE NO LO ENCONTRO'/
SINO
    Sheet1.Cells(row, 1) = hl.getFieldValue(S7230_1,
"ESearchShipment_IdNumber")
    FINSI
    row = row + 1
Loop

LLAMAR hl.pressButton(S7230_1, "INTERNAL_CLEAR")
LLAMAR hl.pressButton(S7230_1, "CancelButton")
LLAMAR hl.closesession
hl = Nothing
FIN

ALGORITMO CSV
VAR
    C this
    WORKBOOK CSV /'WORKBOOK ES UNA VARIABLE DE EXCEL'/

INICIO
    Application.ScreenUpdating = False /'FUNCION PARA QUE NO ARROJE MENSAJES'/

    /'ASIGNO LA RUTA DE DONDE ESTA EL ARCHIVO QUE DEBO TOMAR'/
    rute = "\\ww.hl.lan\SCL\Public\Khipu\HALOCL_CRUCE_Sian_Bolivia.CSV"
    this = Application.ActiveWorkbook.name
    /'ABRO EL ARCHIVO'/
    CSV = Workbooks.Open(rute)

CON CSV
    /'SELECCIONO TODOS LOS DATOS'/
    ActiveSheet.Range("A1:A9999").Select
    Application.CutCopyMode = xlCopy
    Selection.Copy

    /'LOS PEGO EN LA PESTAÑA PASO'/
    Workbooks(this).Activate
    Sheets("paso").Select
    Sheets("paso").Range("A1:A9999").Select
    Sheets("paso").Paste

FIN CON

Application.DisplayAlerts = False
CSV.Close False

```

```
CSV = Nothing
FIN
```

ALGORITMO TextToColumnn

```
/'ALGORITMO PARA PASAR LA INFORMACION DE CSV A CASILLAS SEPARADAS DE EXCEL'/
INICIO
  Sheets("Paso").Activate
  Sheets("Paso").Range("A1:A9999").Select
  Selection.TextToColumns DataType:=xlDelimited, ConsecutiveDelimiter:=False,
other:=True, otherchar:=";"
FIN
```

ALGORITMO SetData

```
VAR
```

```
  D fecha
  N row
  N aux
```

```
INICIO
```

```
  dys = Sheet2.Cells(7, 7)
  row = 6
  aux = 2
```

```
DESDE i = 0 HASTA dys
```

```
  fecha = Date - i
  row = 6
```

```
  HACER HASTA Sheets("Paso").Cells(row, 1) = ""
```

```
    SI Sheets("Paso").Cells(row, 19) = fecha ENTONCES
```

```
      SI Sheets("Paso").Cells(row, 13) = "AUTORIZADO" ENTONCES
```

```
        SI Sheets("Paso").Cells(row, 11) <> "DEMORAS" ENTONCES
```

```
          SI Sheets("Paso").Cells(row, 24) <> "ND" ENTONCES
```

```
            'BL
```

```
            Sheet1.Cells(aux, 3) = Trim(Sheets("Paso").Cells(row, 5))
```

```
            'Chargue
```

```
            Sheet1.Cells(aux, 9) = Trim(FillDate(Sheets("Paso").Cells(row,
```

```
11)))
```

```
            'MC
```

```
            Sheet1.Cells(aux, 8) = Trim(Sheets("Paso").Cells(row, 24))
```

```
            'Container
```

```
            Sheet1.Cells(aux, 6) = Trim(Sheets("Paso").Cells(row, 6))
```

```
            aux = aux + 1
```

```
          End If
```

```
        End If
```

```
      End If
```

```
    End If
```

```
    row = row + 1
```

```
  Loop
```

```
Next i
```

```
FIN
```

ALGORITMO FillDate(chargue As String) As String

```
ENTRADA (C chargue)
```

```
SALIDA (C FillDate)
```

```
INICIO
```

```

/'EDS = XNM'/
SI cargue = "EDS" ENTONCES
  FillDate = "XNM"
FINSI

/'GATE IN = XRD'/
SI cargue = "GATE IN" ENTONCES
  FillDate = "XRD"
FINSI
FIN

```

ALGORITMO InvoiceContainers

```

VAR
  N row
  C matchcode
  C container
  C recharge
INICIO
  row = 2

  hl = CreateObject("HLBridge")

  SI Not initBridge(True) ENTONCES
    SALIR
  FINSI

  S8100_1 = getAppId("S8100")
  LLAMAR hl.setScreenLockingText("Facturando Bolivia")

  LLAMAR hl.proceedOnErrors(True)

  HACER HASTA Sheet1.Cells(row, 1) = ""
    shipment = Sheet1.Cells(row, 1)
    matchcode = Sheet1.Cells(row, 8)
    container = Sheet1.Cells(row, 6)
    recharge = Sheet1.Cells(row, 9)

    SI shipment <> "NF" ENTONCES
      SI matchcode <> "" Y matchcode <> "ND" ENTONCES
        LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")

        SI hl.isDoYouWantToBoxOpened(S8100_1) Then
          LLAMAR hl.commitDoYouWantToBox(S8100_1, "NO")
        FINSI

        LLAMAR hl.SetValue(S8100_1, "EShipment_IdNumber", shipment)
        LLAMAR hl.pressButton(S8100_1, "INTERNAL_FIND")

      SI hl.isErrorExists(S8100_1) ENTONCES
        LLAMAR WriteCell(row, hl.getFirstErrorMessage(S8100_1), 3)
      SINO
        LLAMAR SetContainer(matchcode, container, recharge, row)
      FINSI
    End If
  End If

```

```

    row = row + 1
REPETIR
/'====='/
row = 2

HACER HASTA Sheet1.Cells(row, 1) = ""
    shipment = Sheet1.Cells(row, 1)
    SI Sheet1.Cells(row, 11) = "Completed" ENTONCES
        LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")
        LLAMAR hl.SetValue(S8100_1, "EShipment_IdNumber", shipment)
        LLAMAR hl.pressButton(S8100_1, "INTERNAL_FIND")

/'distribuite'/
    LLAMAR hl.pressButton(S8100_1, "BtnCreateInvoice")

    SI hl.isErrorExists(S8100_1) ENTONCES
        LLAMAR WriteCell(row, hl.getFirstErrorMessage(S8100_1), 3)
    SINO
        LLAMAR SelectInvoices(row)
    FINSI
FINSI
HACER HASTA i = 2
    SI shipment = Sheet1.Cells(row + 1, 1) ENTONCES
        row = row + 1
    SINO
        SALIR HACER HASTA
    FINSI
REPETIR

    row = row + 1
REPETIR
/'====='/

    LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")

    LLAMAR hl.pressButton(S8100_1, "closeButton")
    LLAMAR hl.closesession
    hl = Nothing
FIN

```

ALGORITMO SetContainer

ENTRADA (C matchcode, C container, C recharge, N excel)

INICIO

```

EN ERROR IR A err_handler
LLAMAR hl.selectTabForWindow(S8100_1, 4)
S8100E_3 = hl.getTabApplicationId(S8100_1, 4)

DESDE i = 1 HASTA hl.getFieldValue(S8100_1, "ERevContTotal_ShortInteger")
/'FUNCION PARA VALIDAR QUE SEA EL CONTENEDOR CORRECTO'/
    SI ValidateCtr(container) = True ENTONCES

    SI hl.getTableRowsNumber(S8100E_3, "EListRevenueComponents") = 0 ENTONCES
        LLAMAR WriteCell(excel, "No Recharges", 6)
    SALIR
    SINO

```

```

arr = hl.getTableColumnValuesByColumnName(S8100E_3,
"EListRevenueComponents", "ELineRevChargeType_ChargeCode")
arr_status = hl.getTableColumnValuesByColumnName(S8100E_3,
"EListRevenueComponents", "ELineRevComponent_StateInvoicing")

DESDE row = 0 HASTA UBound(arr)
/'CHEQUEA QUE LA COLUMNA TENGA EL MISMO RECARGO QUE SE BUSCA'/
SI arr(row) = recharge ENTONCES
/'CHEQUEA SI YA ESTA FACTURADO'/
SI arr_status(row) <> "I" ENTONCES

/'ESTO ES PARA QUE SOLO HAGA LOS CAMBIO EN EL ACTUAL
CONTENEDOR'/
LLAMAR hl.SetValue(S8100E_3,
"EListLevel", "THIS")

LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineFpOrg_MatchcodeName", matchcode)
LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineRevComponent_VatSign", "A0")
LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineRevComponent_GroupForInvoice", separator)
LLAMAR hl.setTableCellValueByColumnName(S8100E_3,
"EListRevenueComponents", row, "ELineRevComponent_StateInvoicing", "F")

LLAMAR hl.pressButton(S8100E_3, "BtnSave")
LLAMAR WriteCell(excel, "Completed", 4)
IR A err_handler
SINO
LLAMAR WriteCell(excel, "Already Invoiced", 6)
IR A err_handler
FINSI
FINSI
SIGUIENTE row

LLAMAR WriteCell(excel, "Recharge Not Found", 6)
IR A err_handler
FINSI
FINSI
/'VERIFICA SI EL BOTON NEXT ESTA HABILITADO'/
SI hl.isComponentEnabled(S8100_1, "BtnNext") ENTONCES
LLAMAR hl.pressButton(S8100_1, "BtnNext")
FINSI
SIGUIENTE i

LLAMAR WriteCell(excel, "Container Not Found", 6)

err_handler:
LLAMAR hl.proceedOnErrors(True)
SI hl.isErrorExists(S8100E_3) ENTONCES
LLAMAR WriteCell(excel, hl.getFirstErrorMessage(S8100E_3), 3)
FINSI

LLAMAR hl.pressButton(S8100_1, "INTERNAL_CLEAR")

SI hl.isDoYouWantToBoxOpened(S8100E_3) Then
LLAMAR hl.commitDoYouWantToBox(S8100E_3, "NO")

```

```
FINSI
FIN
```

```
ALGORITMO ValidateCtr
ENTRADA (C container)
SALIDA (B ValidateCtr) //B IS FOR BOOLEAN//
```

```
VAR
  C Ctr
```

```
INICIO
  ctr = hl.getFieldValue(S8100E_3, "ERevCont_ContainerNumber")
```

```
  HACER HASTA InStr(ctr, " ") = 0
    ctr = Replace(ctr, " ", "")
```

```
  REPETIR
```

```
  SI ctr = container ENTONCES
```

```
    ValidateCtr = True
```

```
  SINO
```

```
    ValidateCtr = False
```

```
  FINSI
```

```
FIN
```

```
ALGORITMO SelectInvoices
```

```
ENTRADA (N excel)
```

```
VAR
```

```
  C F0030_5
```

```
  N CANT
```

```
INICIO
```

```
  EN ERROR IR A err_handler
```

```
  F0030_5 = hl.getApplicationId("F0030")
```

```
  SI F0030_5 <> "" ENTONCES //SE ABRIO LA VENTANA F0030//
```

```
    cant = hl.getTableRowsNumber(F0030_5, "EListDocsShipsToBeInvoiced") - 1
```

```
    DESDE i = 0 HASTA cant
```

```
      SI hl.getTableCellValueByColumnName(F0030_5, "EListDocsShipsToBeInvoiced",  
i, "ELineRevenue_GroupForInvoice") = separator ENTONCES
```

```
        LLAMAR hl.selectTableRow(F0030_5, "EListDocsShipsToBeInvoiced", i)
```

```
        LLAMAR hl.pressButton(F0030_5, "CreateInvoice")
```

```
        LLAMAR CreateInvoice(excel)
```

```
        IR A err_handler
```

```
      FINSI
```

```
      SIGUIENTE i
```

```
    FINSI
```

```
    LLAMAR CreateInvoice(excel)
```

```
err_handler:
```

```
  F0030_5 = hl.getApplicationId("F0030")
```

```
  SI F0030_5 <> "" ENTONCES //CERRARLA EN CASO DE QUE SIGA ABIERTA//
```

```
    LLAMAR hl.pressButton(F0030_5, "Close")
```

```
  FINSI
```

```
FIN
```

```
ALGORITMO CreateInvoice
```

ENTRADA (N excel)

VAR

```
C F0020_2
C F0020A_3
C F0020B_3
C F0020C_2
```

INICIO

```
EN ERROR IR A err_handler
```

```
F0020_2 = hl.getApplicationId("F0020")
```

```
SI F0020_2 = "" ENTONCES
```

```
LLAMAR WriteCell(excel, "Error in Windows F0030", 6)
```

```
SALIR
```

```
FINSI
```

```
F0020A_3 = hl.getTabApplicationId(F0020_2, 1)
```

```
'set for all the rows of containers
```

```
LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineAllRoes_Flag", "Y")
```

```
LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineRevenueComponent_FirstConvRoeInv", "")
```

```
LLAMAR hl.setTableCellValueByColumnName(F0020A_3, "EListPositions", 0,
"ELineRevenueComponent_FirstConvCurInv", "")
```

```
LLAMAR hl.pressButton(F0020A_3, "btnSave")
```

```
/'IR A LA PESTAÑA HEADER "header"'/
```

```
LLAMAR hl.selectTabForWindow(F0020_2, 0)
```

```
F0020B_3 = hl.getTabApplicationId(F0020_2, 0)
```

```
LLAMAR hl.SetValue(F0020B_3, "EInvoice_InvoiceCurrency", "CLP")
```

```
LLAMAR hl.pressButton(F0020B_3, "btnSave")
```

```
/'AGREGAR LA CLAUSULA DE BOLIVIA'/'
```

```
LLAMAR hl.selectTabForWindow(F0020_2, 2)
```

```
F0020C_2 = hl.getTabApplicationId(F0020_2, 2)
```

```
LLAMAR hl.tableAppendRow(F0020C_2, "EListClauses")
```

```
LLAMAR hl.SetValue(F0020C_2, "EListClauseLines", "NO AFECTO A IVA SEGUN
TRATADO DE PAZ CHILE/BOLIVIA ANO 1904")
```

```
LLAMAR hl.pressButton(F0020_2, "btnSave")
```

```
LLAMAR hl.pressButton(F0020_2, "btnDistribute")
```

```
LLAMAR WriteCell(excel, "Completed Invoice", 4)
```

```
SALIR
```

err_handler:

```
LLAMAR WriteCell(excel, "Error Invoicing Containers", 6)
```

FIN


```
ALGORITMO WriteCell  
ENTRADA (N row, C data, N color)  
INICIO  
    Sheet1.Cells(row, 11) = data  
    Sheet1.Cells(row, 11).Interior.ColorIndex = color  
FIN
```