

1. OBJETIVO

Documentar las actividades para la elaboración del avance de comercio exterior, en concordancia con los lineamientos de la política de gestión de la información estadística y la normatividad aduanera vigente, con el propósito de garantizar su calidad y estandarización.

2. DOCUMENTOS RELACIONADOS

Tipo de documento	Código	Título	Modo de uso	Clasificación documento
Procedimiento	PR-IIT-0244	Producción y análisis de estadísticas	Digital	Interno
Formato	FT-PEC-1463	Solicitud de Publicación, Actualización o Eliminación en el Portal WEB de la Entidad		
Formulario	500	Declaración de Importación	Digital	Interno
Formulario	600	Declaración de Exportación	Digital	Interno
Plantilla		Formato de publicación Avance de Comercio Exterior	Digital	Interno
Documento externo		Total de partidas arancelarias por series históricas 1993 – 2023 TOTPART Versión 75.	Digital	Externo

3. DEFINICIONES Y SIGLAS

- CETCE:** Coordinación de Estadística Tributaria y de Comercio Exterior.
- Dirección seccional:** estructura de la DIAN para el cumplimiento de sus funciones en el territorio nacional, en los niveles local, delegado y de grandes contribuyentes, correspondiente a las Direcciones Seccionales, a las Direcciones Delegadas de Impuestos, Aduanas e Impuestos y Aduanas, y a la Dirección Operativa de Grandes Contribuyentes respectivamente. Los declarantes y sus respectivas declaraciones estarán asociadas a la Dirección Seccional o Delegada que corresponda al domicilio principal de su actividad o negocio, excepto, quienes sean calificados como grandes contribuyentes. Fuente. U.A.E Dirección Impuestos y Aduanas Nacionales- DIAN – Dirección de Gestión Estratégica y de Analítica - Subdirección de Estudios Económicos – Coordinación de Estadística Tributaria y de Comercio Exterior.
- DGEA:** Dirección de Gestión Estratégica y de Analítica.
- DataFrame:** estructura de datos bidimensionales, de tamaño variable y potencialmente heterogéneo, compuesto por ejes etiquetados (filas y columnas). Fuente: pandas.DataFrame (2024). <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>
- Exportación:** es la salida de mercancías del territorio aduanero nacional con destino a otro país. Fuente: Presidencia de la República de Colombia (2019). Decreto 1165.
- Importación:** es la introducción de mercancías de procedencia extranjera al territorio aduanero nacional cumpliendo con los términos y condiciones previstos en el presente decreto. Fuente: Presidencia de la República de Colombia (2019). Decreto 1165.

- **Merge:** operación en Python que combina dos DataFrame a partir de una o más variables en común. U.A.E Dirección Impuestos y Aduanas Nacionales- DIAN – Dirección de Gestión Estratégica y de Analítica - Subdirección de Estudios Económicos – Coordinación de Estadística Tributaria y de Comercio Exterior.
- **MUISCA:** Modelo Único de Ingresos, Servicio y Control Autorizado.
- **Precio implícito:** valor estimado de una mercancía por kilogramo. Resulta de la división del valor FOB entre el peso neto. Fuente: U.A.E Dirección Impuestos y Aduanas Nacionales- DIAN – Dirección de Gestión Estratégica y de Analítica - Subdirección de Estudios Económicos – Coordinación de Estadística Tributaria y de Comercio Exterior.
- **PST:** Punto único de Soluciones Tecnológicas
- **Script:** conjunto de instrucciones de programación que se ejecutan dentro de un entorno específico. A diferencia de un programa tradicional, que es independiente y más complejo, un script suele ser más sencillo y depende de otro programa (como un navegador web) para su ejecución. Fuente: InboundCycle (2024). <https://www.inboundcycle.com/diccionario-marketing-online/script>.
- **SEE:** Subdirección de Estudios Económicos.
- **Spyder:** entorno científico escrito en Python, para Python, y diseñado por y para científicos, ingenieros y analistas de datos. Ofrece una combinación de la funcionalidad avanzada de edición, análisis, depuración y creación de perfiles de una herramienta de desarrollo integral con la exploración de datos, la ejecución interactiva, la inspección profunda y las capacidades de visualización de un paquete científico. Fuente: PyPi.org (2024). <https://pypi.org/project/spyder> .
- **SYGA:** Sistema de información y Gestión Aduanera
- **U.A.E DIAN:** Unidad Administrativa Especial Dirección de Impuestos y Aduanas Nacionales.
- **TOTPART:** Total de Partidas Arancelarias.
- **Valor CIF (Cost, Insurance, Freight, es decir, costo, seguro y flete):** costo que incluye el valor de la transacción de los bienes, el de los servicios suministrados para entregarlos en la frontera del país exportador y el de los servicios prestados para la entrega de los bienes desde la frontera del país exportador hasta la frontera del país importador. Fuente: Organización de las Naciones Unidas (ONU) (2010). Departamento de Asuntos Económicos y Sociales, División de Estadística. Estadísticas del comercio internacional de mercancías. Conceptos y Definiciones.
- **Valor FOB (Free On Board, es decir, Franco a Bordo):** corresponde al precio de venta de los bienes embarcados a otros países, puestos en el medio de transporte, sin incluir valor de seguro y fletes. Las valoraciones FOB incluyen el valor de transacción de los bienes y el valor de los servicios suministrados para entregar las mercancías en la frontera del país exportador. Fuente: Organización de las Naciones Unidas (ONU) (2010). Departamento de Asuntos Económicos y Sociales, División de Estadística. Estadísticas del comercio internacional de mercancías. Conceptos y Definiciones.

4. DESARROLLO DEL TEMA

El Avance de Comercio Exterior es un reporte, de publicación semanal, que se realiza a partir de las bases de las declaraciones de importación (Formulario 500) y de exportación (Formulario 600). El periodo de análisis corresponde al rango entre el primer día del mes al sábado inmediatamente anterior a la semana de elaboración (por ejemplo, del 1 de octubre al 21 de octubre de 2023, o 1 de octubre a 28 de octubre de 2023).

Los Avances de Comercio Exterior se elaboran con el propósito de brindar información oportuna para el monitoreo semanal del comercio internacional colombiano. No obstante, en consideración con la metodología que se aplica para la elaboración del producto, los Avances de Comercio Exterior no pueden ser interpretados como un valor oficial de las cifras de comercio exterior de Colombia, toda vez que, a este nivel, no se han aplicado todas las validaciones y consolidaciones necesarias sobre el total de variables que se presentan en las bases oficiales de comercio exterior, por lo cual, las bases usadas para la construcción de los avances se clasifican en estado preliminar.

El proceso para la elaboración del avance inicia con la solicitud de cargue de bases de datos semanales en el repositorio administrador por la Coordinación de Estadística Tributaria y de Comercio Exterior, con el apoyo de la Subdirección de Procesamientos de Datos. Posteriormente se aplican las validaciones y cálculos que se describen en el presente Instructivo y finalmente, se publica un reporte en formato PDF, disponible en la sección de Cifras >>Estadísticas>>Estadísticas de Comercio Exterior >>Avance de Comercio Exterior: <https://www.dian.gov.co/dian/cifras/Paginas/EstadisticasComEx.aspx>

Este instructivo expone detalladamente las actividades para la obtención de la información, su procesamiento y el cálculo de los resultados a publicar en el Avance de Comercio Exterior, utilizando el lenguaje de programación Python a través de la herramienta Spyder.

4.1 Integración de datos

Para la elaboración del Avance de Comercio Exterior el primer paso a seguir es acopiar los datos correspondientes al periodo de referencia, estos provienen de las bases de datos que la entidad recopila a partir de las declaraciones de importaciones y exportaciones. A continuación, se describe el proceso que se sigue para reunir estos datos.

4.1.1 Alistamiento de bases de datos

Como se indicó previamente, es necesario realizar el acopio de los datos que se van a emplear en la elaboración del avance, esto comprende el cargue que se debe realizar cada semana y la revisiones y validaciones de datos para contar con datos depurados.

4.1.1.1 Cargue semanal

Como se indicó en el apartado anterior, la elaboración del Avance de Comercio Exterior inicia con la solicitud que hace la Coordinación de Estadísticas Tributarias y de Comercio Exterior a la Subdirección de Procesamiento de Datos del cargue de la información en bodega de datos. Para tal fin, el primer día hábil de la semana, el servidor público encargado envía un requerimiento a través del Punto único de Soluciones Tecnológicas (PST) de la entidad, en el cual solicita la carga de la información de las declaraciones de importaciones y exportaciones presentadas electrónicamente de la semana anterior (de domingo a sábado) (ver figura 1).

Figura 1 Ejemplo de solicitud de cargue semanal

Asunto:	
Información semanal de importaciones y exportaciones del periodo 30 de abril al 06 de mayo de 2023	
Descripción	Solución
<p>Por instrucciones del jefe de la Subdirección de Estudios Económicos, atentamente solicito nos colaboren por favor con el cargue de la información semanal de importaciones y exportaciones del periodo 30 de abril al 06 de mayo de 2023, se realiza por análisis de operación.</p> <p>Adicionalmente solicito el envío del archivo con los totales de control del referido período a lpicos@dian.gov.co</p>	

Fuente: U.A.E. DIAN. CETCE.

4.1.1.1.1 Revisión y corrección de errores en cargue semanal

En el evento que el funcionario encargado de atender el requerimiento desde la Subdirección de Procesamientos de Datos, encuentre algún error en el proceso de cargue, enviará dos archivos al funcionario encargado en la Coordinación de Estadística Tributaria y de Comercio Exterior, un archivo denominado *cargue.log* y otro *cargue.bad*. En el archivo *.log* se podrán identificar las variables del archivo y el listado de registros que presentaron error con su detalle (ver Figura 2).

Figura 2 Ejemplo archivo cargue.log

IVA_LIQUIDADDO_DOLARES	NEXT	*	CHARACTER
PORCENTAJE_SALVAGUARDIA	NEXT	*	CHARACTER
BASE_SALVAGUARDIA	NEXT	*	CHARACTER
VALOR_TOTAL_SALVAGUARDIA	NEXT	*	CHARACTER
SALVAGUARDIA_PAGADA	NEXT	*	CHARACTER
SALVAGUARDIA_LIQUIDADA_DOLARES	NEXT	*	CHARACTER
PORCENTAJE_DERECHOS_COMPENSATO	NEXT	*	CHARACTER
BASE_DERECHOS_COMPENSATORIOS	NEXT	*	CHARACTER
VALOR_TOTAL_DERECHOS_COMPENSAT	NEXT	*	CHARACTER
DERECHOS_COMPENSATORIOS_PAGADOS	NEXT	*	CHARACTER
DERECHO_COM_LIQUIDADOS_DOLARES	NEXT	*	CHARACTER
VALOR_TOTAL_ANTIDUMPING	NEXT	*	CHARACTER
ANTIDUMPING_PAGADO	NEXT	*	CHARACTER
ANTIDUMPING_LIQUIDADO_DOLARES	NEXT	*	CHARACTER
VALOR_TOTAL_RESCATE	NEXT	*	CHARACTER
RESCATE_PAGADO	NEXT	*	CHARACTER
TOTAL_LIQUIDADO_DOLARES	NEXT	*	CHARACTER

Record 3: Rejected - Error on table "ROOT"."COMEX_SEMANA".
ORA-01401: inserted value too large for column

Record 4: Rejected - Error on table "ROOT"."COMEX_SEMANA", column PORCENTAJE_DERECHOS_COMPENSATO.
ORA-01438: value larger than specified precision allows for this column

Record 6: Rejected - Error on table "ROOT"."COMEX_SEMANA", column FAX_DECLARANTE.
ORA-01722: invalid number

Record 7: Rejected - Error on table "ROOT"."COMEX_SEMANA", column FAX_DECLARANTE.
ORA-01722: invalid number

Record 8: Rejected - Error on table "ROOT"."COMEX_SEMANA", column FAX_DECLARANTE.
ORA-01722: invalid number

Record 9: Rejected - Error on table "ROOT"."COMEX_SEMANA", column FAX_DECLARANTE.
ORA-01722: invalid number

Fuente: U.A.E. DIAN. CETCE.

Cabe resaltar que es necesario que en cada uno de los registros aparezca el detalle de la variable que presenta error, de lo contrario (como se evidencia en el recuadro azul de la Figura 2), se dificulta la identificación del campo que debe arreglarse y es necesario revisar la línea completa para su mejora, como la identificación y reemplazo de los caracteres especiales.

En el mismo archivo, y al final, aparece el total de registros que presentaron error y deben ajustarse (ver Figura 3).

Figura 3 Ejemplo de archivo cargue.log

```
Record 68: Rejected - Error on table "ROOT"."COMEX_SEMANA", column FAX_DECLARANTE.  
ORA-01722: invalid number  
  
Record 69: Rejected - Error on table "ROOT"."COMEX_SEMANA", column FAX_DECLARANTE.  
ORA-01722: invalid number  
  
Table "ROOT"."COMEX_SEMANA":  
    7 Rows successfully loaded.  
    62 Rows not loaded due to data errors.  
    0 Rows not loaded because all WHEN clauses were failed.  
    0 Rows not loaded because all fields were null.  
  
Space allocated for bind array:           61662 bytes(1 rows)  
Space allocated for memory besides bind array:      0 bytes  
  
Total logical records skipped:          0  
Total logical records read:            69  
Total logical records rejected:        62  
Total logical records discarded:       0  
  
Run began on Mon Jan 22 09:54:56 2024  
Run ended on Mon Jan 22 09:54:56 2024  
  
Elapsed time was:   00:00:00.13  
CPU time was:      00:00:00.02
```

Fuente: U.A.E. DIAN. CETCE.

Por su parte, el archivo *.bad* contiene los registros que presentaron error, este es el archivo que debe ser ajustado y enviado en el mismo formato (*.bad*) al encargado de atender el caso. Se sugiere que el archivo *.bad* sea abierto en el software WordPad dado que este permite visualizar todos los errores, a diferencia de Bloc de Notas en donde algunos detalles permanecen ocultos.

El funcionario de la CETCE deberá contrastar los datos, de preferencia contra la copia de la declaración electrónica y proceder a realizar los ajustes a los errores recibidos. Es de tener en cuenta que estos errores pueden estar asociados a formato (por ejemplo, espacios adicionales o caracteres especiales, ver resaltado en Figura 4), o pueden corresponder a datos inconsistentes o fuera de su casilla, datos faltantes, entre otros. En el siguiente Figura pueden observarse algunos de estos:

Figura 4 Ejemplo de archivo carque.bad

Fuente: U.A.E., DIAN, CETCE.

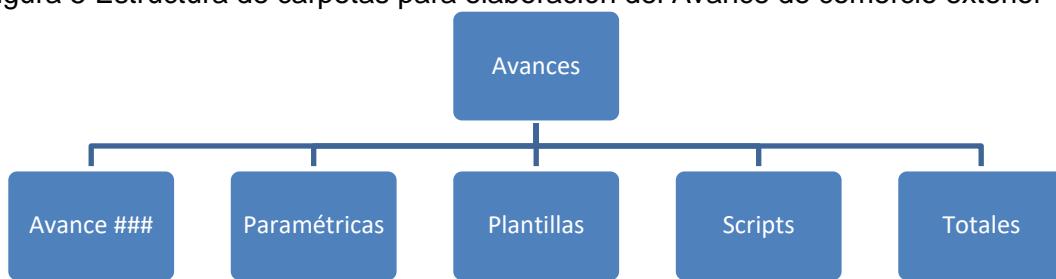
Luego de realizar los ajustes, se debe responder el correo al funcionario de la Subdirección de Procesamiento de datos, adjuntando el archivo **.bad** corregido.

Si los errores persisten, se volverá a recibir los dos archivos (.log y .bad) desde la Subdirección de procesamiento de Datos y se deberá realizar y enviar de nuevo los ajustes correspondientes hasta recibir el visto bueno por parte del funcionario de la Subdirección de Procesamiento de Datos. Cuando ya no se cuente con errores en el cargue, deberá continuar con la siguiente sección.

4.1.1.2 Confirmación de los totales del carque semanal

Para la organización de la información relacionada con la confirmación de totales y la elaboración del avance de comercio exterior, se deberá considerar la siguiente estructura de carpetas (ver figura 5), en el espacio de OneDrive destinado por la Coordinación:

Figura 5 Estructura de carpetas para elaboración del Avance de comercio exterior



Fuente: U.A.E, DIAN, CETCE.

Avances: corresponde a la carpeta raíz donde se almacena toda la información de esta operación.

Avance ###: corresponde a la carpeta específica para el número de avance a desarrollar, por ejemplo, si el avance es el 975, se reemplaza los símbolos de numeral por el número correspondiente.

Paramétricas: carpeta en la cual se almacenan las tablas paramétricas necesarias para la ejecución de los scripts, en específico, la tabla de Aduanas, Cuode, Departamentos, Países, Totpart y Usuario calificados de zonas francas. Estas tablas únicamente deben ser actualizadas por el funcionario de la Coordinación que sea delegado, pues los cambios en las paramétricas podrían generar error en la ejecución de los scripts.

Plantillas: espacio en el cual se encuentran las plantillas para los cuadros de salida y el documento a publicar.

Scripts: almacena la última versión aprobada de los scripts necesarios para la elaboración del avance, los cuales están diseñados para ejecución en Spyder.

Totales: contiene los archivos con los totales, mes a mes, de las exportaciones e importaciones, en valores FOB, CIF, Peso neto y número de registros. Deben ser actualizados previo a la generación del avance y según se hayan o no publicado las cifras oficiales.

Para los archivos que se generan en el proceso, es importante respetar los nombres y las extensiones, las cuales se aclaran en cada paso de este instructivo. Cualquier cambio puede originar errores en la ejecución de los scripts.

Una vez se haya realizado el cargue de la base en el repositorio Cognos 5.0 Impromptu por parte de la Subdirección de Procesamiento de Datos, el funcionario de la CETCE que radique la solicitud por PST recibirá un archivo Excel con los totales, por dirección seccional, del valor FOB y peso neto, tanto para importaciones como exportaciones (ver Figura 6).

Figura 6 Ejemplo cuadro con totales cargue semanal

A	B	C	D	E	F
EXPORTACIONES					
COD_ADMINISTRACION_PR NOMBRE_ADMINISTRA CASOS VAL_TOTAL_FOB_US					
3	ESPECIAL DE ADUANA:	7.856	240.682.142,83		
4	BUCARAMANGA	6	2.917.297,03		
16	PEREIRA	39	2.457.010,10		
19	SANTA MARTA	435	68.390.557,12		
25	RIOHACHA	3	22.359.550,00		
27	SAN ANDRES	1	51.152,27		
35	BUENAVENTURA	2.840	73.340.255,65		
37	IPIALES	1.989	20.106.331,86		
39	MAICAO	498	8.451.612,61		
41	TURBO	180	10.732.347,17		
46	PUERTO ASIS	19	940.982,39		
48	CARTAGENA ADU	7.110	473.784.733,54		
87	ADUANAS BARRANQUI	802	33.231.096,86		
88	ADUANAS CALI	655	15.962.793,64		
89	ADUANAS CUCUTA	321	2.328.026,13		
90	MEDELLIN	3.023	55.025.747,34		
IMPORTACIONES					
ADM_CODIGO_ADMINISTRATIVO	NOMBRE_ADMINISTRA	CASOS	VAL_TOTAL_FOB_US		
1	ARMENIA	939	2.506.526,97		
3	ESPECIAL DE ADUANA:	30.304	404.763.335,12		
4	BUCARAMANGA	63	214.098,79		
10	MANIZALES	7	402.241,67		
16	PEREIRA	451	5.558.465,28		
19	SANTA MARTA	1.777	100.502.390,50		
25	RIOHACHA	280	1.439.874,17		
35	BUENAVENTURA	11.730	277.107.948,53		
37	IPIALES	404	9.926.991,66		
39	MAICAO	75	612.987,38		
41	TURBO	117	2.167.008,06		
46	PUERTO ASIS	15	570.833,02		
48	CARTAGENA ADU	11.292	323.758.208,84		
87	ADUANAS BARRANQUI	2.717	91.912.831,59		
88	ADUANAS CALI	1.714	22.278.093,81		
89	ADUANAS CUCUTA	88	1.478.845,78		
90	MEDELLIN	4.190	100.801.531,13		

Fuente: U.A.E. DIAN. CETCE.

El funcionario de la CETCE que realiza el cargue semanal, deberá enviar el cuadro de totales al funcionario de la Coordinación que tenga a cargo la elaboración del Avance de Comercio de la semana para que realice la validación de los totales.

Para comparar los totales de la semana, cargados previamente, se deberá realizar en Cognos 5 Impromptu, las siguientes consultas:

- Exportaciones:** ejecutar una consulta sobre la carpeta **Exportaciones Semana**, con las variables *Bdtr Identificador Documento*, *Val Total FOB Us*, *Cod Administracion Pres*, *Fec Declaración Exportacion* (ver figura 7), asegurándose de ajustar el filtro para tomar únicamente la semana que corresponda al periodo de referencia (de domingo a sábado) de la semana inmediatamente anterior. El resultado debe ser exportado a la carpeta del avance en curso (ver Figura 5) en un archivo .cvs (con separador | pipe) y respetando la siguiente estructura del nombre: prueba_expo_A###.csv (### corresponde al número del Avance a elaborar).

Figura 7 Ejemplo consulta Exportaciones Semana

The screenshot shows the Impromptu software interface. On the left is a results grid with the following data:

Bdtr Identificador Documento	Val Total Fob Us	Cod Administracion Pres	Fec Declaracion Exportacion
6007721829077	28060.05	35	20231015
6007721829077	2178.88	35	20231015
6007721829077	21217.44	35	20231015
6007721829077	12470.33	35	20231015
6007722275933	97965.35	35	20231015
6007722275933	9590.20	35	20231015
6007722275958	3950.32	35	20231015
6007722896872	19440.00	3	20231015
6007722962965	39560.00	48	20231015
6007722962972	6337.00	48	20231015
6007722990837	147637.50	37	20231015
6007722990890	147637.50	37	20231015
6007722970266	155488.22	90	20231015

Fuente: U.A.E. DIAN. CETCE.

- Importaciones:** ejecutar una consulta sobre la carpeta **Comex Semana**, con las variables *Identificador Documento*, *Admin Código Administracion*, *Valor FOB1*, *Fecha Presentacion* (ver figura 8), asegurándose de ajustar el filtro para tomar únicamente la semana que corresponda al periodo de referencia (de domingo a sábado) de la semana inmediatamente anterior. El resultado debe ser exportado a la carpeta del avance en curso (ver Figura 5) en un archivo .cvs (con separador | pipe) y respetando la siguiente estructura del nombre: prueba_impo_A###.csv (### corresponde al número del Avance a elaborar).

Figura 8 Ejemplo consulta Comex Semana

The screenshot shows a software window titled "Impromptu - [Prueba impo_2023]". The main area displays a table with columns: Identificador Documento, Adm1 Código Administración, Valor Fob1, and Fecha Presentación. The data includes various entries such as 37920346, 48, 4973.93, 20231015. To the right of the table is a "Query" panel with a "Filter definition" section containing the expression "Fecha Presentacion between 20231015 and 20231021". Below this are sections for "Available components", "Filter options", and "Tips". Buttons for "OK" and "Cancel" are at the bottom right.

Fuente: U.A.E. DIAN. CETCE.

Con los anteriores archivos .csv resultantes de las dos consultas desde Cognos 5 Impromptu, se procede a realizar la importación de estos en Python, para generar los totales del valor FOB y peso neto, por dirección seccional, tanto para exportaciones como importaciones, por separado. La siguiente imagen muestra una captura de pantalla del proceso realizado en Python (ver figura 9):

Figura 9 Confirmar totales cargue semana

```

1  import pandas as pd
2  import math
3
4  ##### Parámetros
5  avance="981"
6  fecha="Octubre 1 al 21 de 2022 y 2023"
7
8  ##### Conectar bases #####
9  path="D:/Alejandro Pico\OneDrive - Direccion de Impuestos y Aduanas Nacionales de Colombia/5. Avances/Avance "+avance
10 totales_cargue= pd.read_excel(path+"IMPO_EXP0_SEMANA_A"+avance+".xlsx",usecols=[0,1,2,3,4],skiprows=1,decimal=',')
11 totales_cognos_expo= pd.read_csv(path+"/prueba_expo_A"+avance+".csv",sep=';',decimal=',')
12 totales_cognos_impo= pd.read_csv(path+"/prueba_impo_A"+avance+".csv",sep=';',decimal=',')
13 totales_cognos_impo = totales_cognos_impo.rename(columns={'Adm1 Código Administración': 'Cod Administracion Pres'})
14
15 #####Confirmar totales semana #####
16 totales_expo_cargue=[]
17 totales_impo_cargue=[]
18 for i in range(len(totales_cargue['CASOS'])):
19     if math.isnan(totales_cargue['CASOS'][i]):
20         totales_expo_cargue=totales_cargue[0:i]
21         totales_impo_cargue=totales_cargue[i+3:]
22         break
23
24 totales_expo_cargue = totales_expo_cargue.rename(columns={'COD_ADMINISTRACION_PRES': 'Cod Administracion Pres', 'CASOS': 'C'})
25 totales_impo_cargue = totales_impo_cargue.rename(columns={'COD_ADMINISTRACION_PRES': 'Cod Administracion Pres', 'CASOS': 'C'})
26
27 resumen_totales_cognos_expo = totales_cognos_expo.groupby('Cod Administracion Pres')[ 'Val Total Fob Us'].agg(['count', 'sum'])
28 resumen_totales_cognos_impo = totales_cognos_impo.groupby('Cod Administracion Pres')[ 'Valor Fob1'].agg(['count', 'sum']).re
29
30 seccionales_impo=resumen_totales_cognos_impo['Cod Administracion Pres'].unique()
31 seccionales_expo=resumen_totales_cognos_expo['Cod Administracion Pres'].unique()
32 seccionales_cuadro = list(set(seccionales_impo) | set(seccionales_expo))

```

Fuente: U.A.E. DIAN. CETCE.

Una vez procesada la información, se debe realizar el contraste con los totales enviados por la Subdirección de Procesamiento de Datos. Para este punto, además de validar que los totales coincidan, también se debe verificar que se encuentren las direcciones seccionales completas, por ejemplo, un error frecuente es que el cargue se realiza para un número de seccionales, dejando por fuera la

información de otras seccionales que, si bien en registros es pequeño, es necesario que se cuente con toda la información disponible (ver figura 10). En caso de identificarse algún error o novedad en los totales o cantidad de direcciones seccionales, se debe reportar el hecho al funcionario de la CETCE encargado del cargue semanal para que revise con la Subdirección de Procesamiento de Datos. En el evento que los totales coincidan y no se identifique novedad alguna, se debe enviar correo al funcionario de la CETCE encargado del cargue semanal, confirmando la coincidencia de la información reportada por la SPD y la dispuesta en bodega de datos.

Figura 10 Ejemplo cruce totales semana

B	C	D	E	F	G	H	I	J	K	L	M	N
Administracion	Conteo Expo cargue	Conteo Expo	FOB Expo cargue	FOB Expo	Conteo Impo cargue	Conteo Impo	FOB Impo cargue	FOB Impo	V_conteo_Expo	V_conteo_Impo	V_FOB_Expo	V_FOB_Impo
1												
2	ESPECIAL DE ADUANAS	7.856,00	7.856,00	240.682.142,63	240.682.142,63	30.304,00	30.304,00	404.763.335,12	404.763.335,12	VERDADERO	VERDADERO	VERDADERO
3	BUCARAMANGA	6,00	6,00	2.917.297,03	2.917.297,03	63,00	63,00	214.098,79	214.098,79	VERDADERO	VERDADERO	VERDADERO
4	PEREIRA	39,00	39,00	2.457.010,10	2.457.010,10	451,00	451,00	5.558.465,28	5.558.465,28	VERDADERO	VERDADERO	VERDADERO
5	SANTA MARTA	435,00	435,00	68.390.557,12	68.390.557,12	1.777,00	1.777,00	100.502.390,50	100.502.390,50	VERDADERO	VERDADERO	VERDADERO
6	RIOHACHA	3,00	3,00	22.359.550,00	22.359.550,00	280,00	280,00	1.439.874,17	1.439.874,17	VERDADERO	VERDADERO	VERDADERO
7	BUENAVENTURA	2.840,00	2.840,00	73.340.255,65	73.340.255,65	11.730,00	11.730,00	277.107.948,53	277.107.948,53	VERDADERO	VERDADERO	VERDADERO
8	IPIALES	1.989,00	1.989,00	20.106.331,86	20.106.331,86	404,00	404,00	9.926.991,66	9.926.991,66	VERDADERO	VERDADERO	VERDADERO
9	MAICAO	498,00	498,00	8.451.612,61	8.451.612,61	75,00	75,00	612.987,38	612.987,38	VERDADERO	VERDADERO	VERDADERO
10	TURBO	180,00	180,00	10.732.347,17	10.732.347,17	117,00	117,00	2.167.008,06	2.167.008,06	VERDADERO	VERDADERO	VERDADERO
11	PUERTO ASIS	19,00	19,00	940.982,39	940.982,39	15,00	15,00	570.833,02	570.833,02	VERDADERO	VERDADERO	VERDADERO
12	CARTAGENA ADU	7.110,00	7.110,00	473.784.733,54	473.784.733,54	11.292,00	11.292,00	323.758.208,84	323.758.208,84	VERDADERO	VERDADERO	VERDADERO
13	ADUANAS BARRANQUILLA	802,00	802,00	33.231.096,86	33.231.096,86	2.717,00	2.717,00	91.912.831,59	91.912.831,59	VERDADERO	VERDADERO	VERDADERO
14	ADUANAS CALI	655,00	655,00	15.962.793,64	15.962.793,64	1.714,00	1.714,00	22.278.093,81	22.278.093,81	VERDADERO	VERDADERO	VERDADERO
15	ADUANAS CUCUTA	321,00	321,00	2.328.026,13	2.328.026,13	88,00	88,00	1.478.845,78	1.478.845,78	VERDADERO	VERDADERO	VERDADERO
16	MEDELLIN	3.023,00	3.023,00	55.025.747,94	55.025.747,94	4.190,00	4.190,00	100.801.531,13	100.801.531,13	VERDADERO	VERDADERO	VERDADERO
17		25.776,00	25.776,00	1.030.710.484,67	1.030.710.484,67	65.217,00	65.217,00	1.343.093.443,66	1.343.093.443,66	VERDADERO	VERDADERO	VERDADERO

Fuente: U.A.E. DIAN. CETCE.

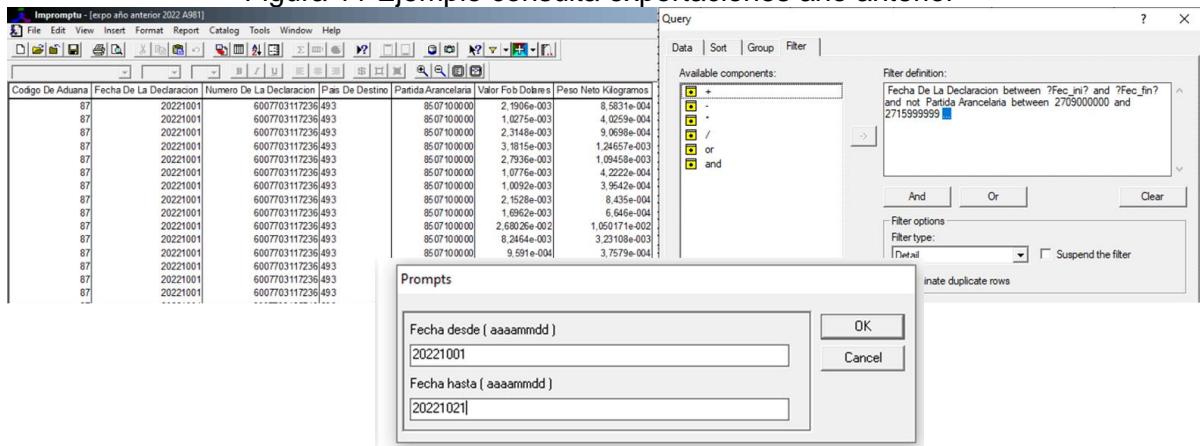
Una vez recibido el correo de confirmación indicado anteriormente, el funcionario de la CETCE encargado del cargue semanal informa el cargue exitoso a la Subdirección de Procesamiento de Datos. Posteriormente, se procede al diligenciamiento de la encuesta del caso en PST correspondiente.

4.1.1.2 Descarga de bases de datos para elaboración del Avance

Una vez se haya realizado el proceso de cargue semanal, se continúa con la descarga de las bases de datos que son utilizadas en el proceso de elaboración del avance, de la siguiente manera:

- i. **Exportaciones año anterior:** ejecutar una consulta sobre la carpeta **Exportaciones**, con las variables *Codigo De Aduana, Fecha De La Declaracion, Numero De La Declaracion, País De Destino, Partida Arancelaria, Valor FOB Dolares y Peso Neto Kilogramos* (ver figura 11), asegurándose de ajustar el filtro para tomar únicamente el mes de referencia (01 del mes a la fecha calendario que corresponda al sábado de la semana precedente) del año anterior. El resultado debe ser exportado a la carpeta del avance en curso (ver Figura 5) en un archivo .cvs (con separador | pipe) y respetando la siguiente estructura del nombre: *expo_anterior_A###* (### corresponde al número del Avance a elaborar).

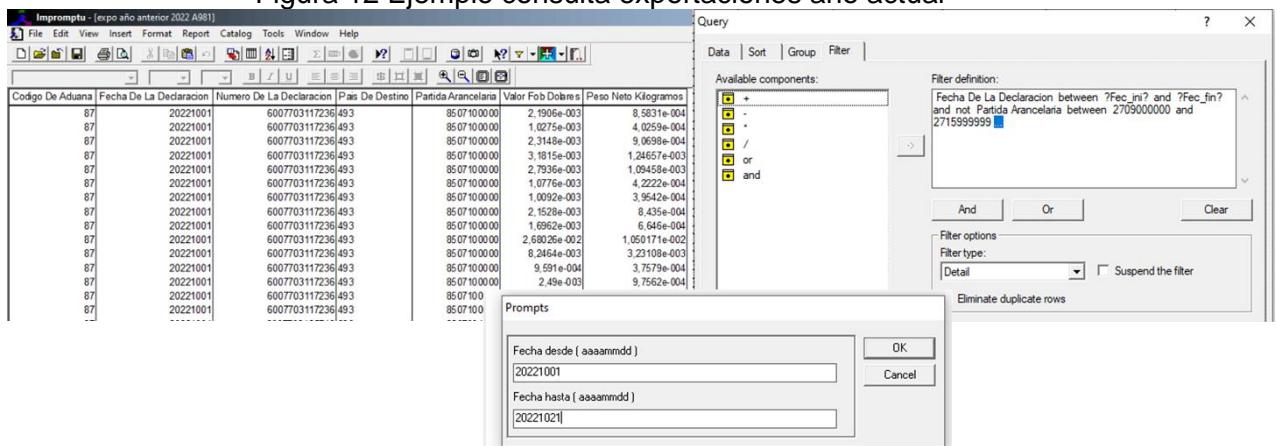
Figura 11 Ejemplo consulta exportaciones año anterior



Fuente: U.A.E. DIAN. CETCE.

- ii. **Exportaciones año actual:** ejecutar una consulta sobre la carpeta **Exportaciones**, con las variables *Código De Aduana*, *Fecha De La Declaracion*, *Número De La Declaracion*, *País De Destino*, *Partida Arancelaria*, *Valor FOB Dolares* y *Peso Neto Kilogramos* (ver figura 12), asegurándose de ajustar el filtro para tomar únicamente el mes de referencia (01 del mes a la fecha calendario que corresponda al sábado de la semana precedente) del año actual. El resultado debe ser exportado a la carpeta del avance en curso (ver Figura 5) en un archivo .cvs (con separador | pipe) y respetando la siguiente estructura del nombre: *expo_anterior_A####* (#### corresponde al número del Avance a elaborar).

Figura 12 Ejemplo consulta exportaciones año actual

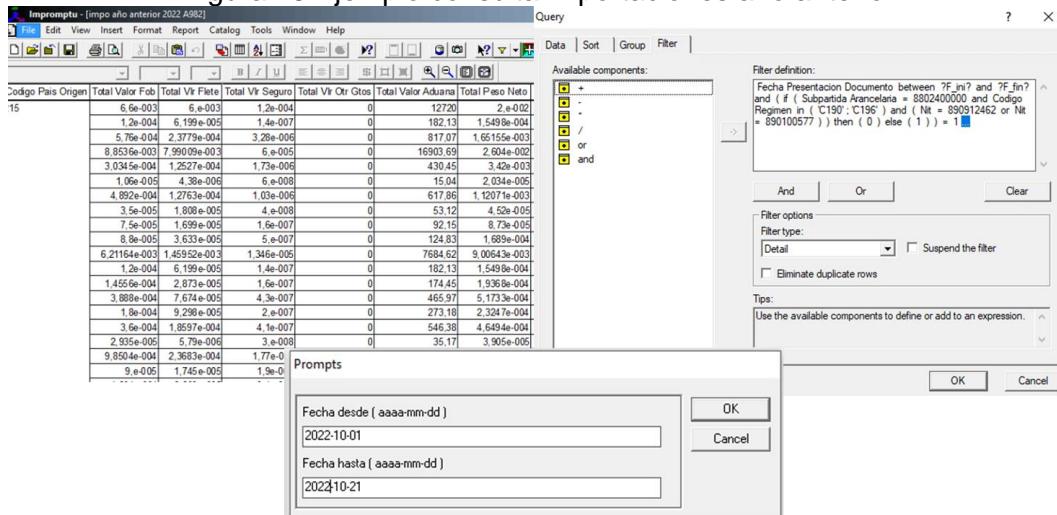


Fuente: U.A.E. DIAN. CETCE.

- iii. **Importaciones año anterior:** ejecutar una consulta sobre la carpeta **V Declaraciones Aduanas**, con las variables *Identificador Documento*, *Admin Código Administracion*, *Valor FOB1*, *Fecha Presentacion* (ver figura 13), asegurándose de ajustar el filtro para tomar únicamente la semana que corresponda al periodo de referencia (de domingo a sábado) de la semana inmediatamente anterior y para el año anterior. El resultado debe ser exportado a la carpeta del avance en curso (ver Figura 5) en un archivo .cvs (con separador | pipe) y

respetando la siguiente estructura del nombre: prueba_impo_A### (### corresponde al número del Avance a elaborar).

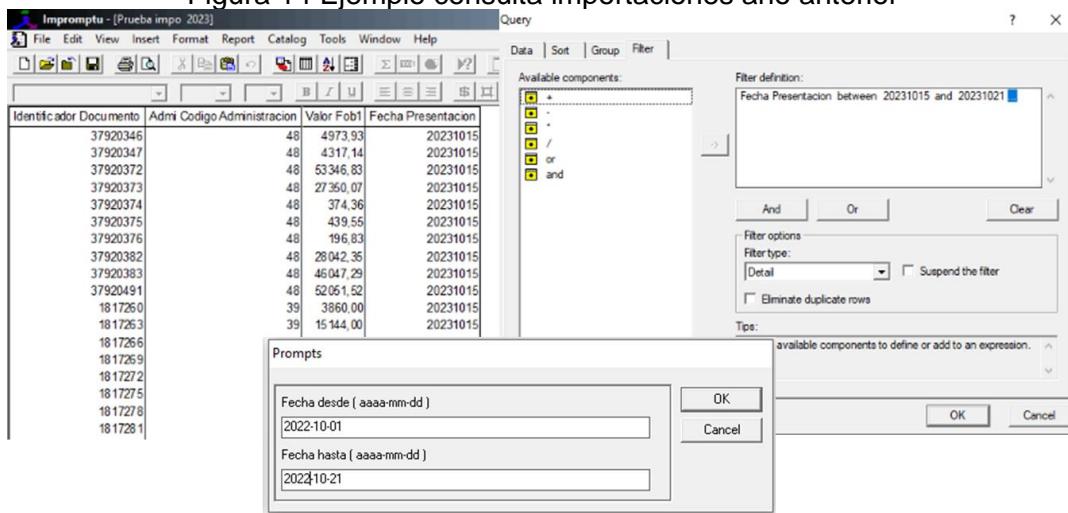
Figura 13 Ejemplo consulta importaciones año anterior



Fuente: U.A.E. DIAN. CETCE.

- iv. **Importaciones año actual:** ejecutar una consulta sobre la carpeta **V Declaraciones Aduanas**, con las variables *Fecha Presentacion Documento*, *Cod Administracion Presentada*, *Codigo Pais Origen*, *Subpartida Arancelaria*, *Total Valor FOB*, *Total Peso Neto*, *Total Vr CIF* (ver figura 14), asegurándose de ajustar el filtro para tomar únicamente la semana que corresponda al periodo de referencia (de domingo a sábado) de la semana inmediatamente anterior y del año actual. El resultado debe ser exportado a la carpeta del avance en curso (ver Figura 5) en un archivo .cvs (con separador | pipe) y respetando la siguiente estructura del nombre: prueba_impo_A### (### corresponde al número del Avance a elaborar).

Figura 14 Ejemplo consulta importaciones año anterior

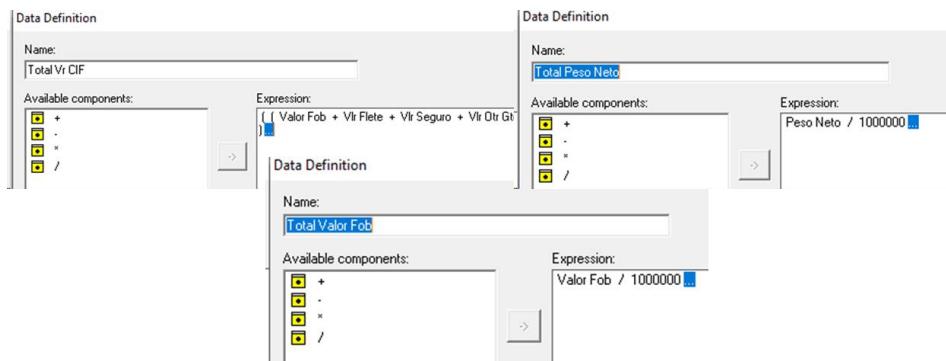


Fuente: U.A.E. DIAN. CETCE.

Nota: en la consulta de Cognos, se debe realizar las siguientes parametrizaciones, antes de generar el reporte:

- **Total Vr CIF** debe ser construida con la siguiente expresión: $(Total \ Valor \ FOB + Total \ Vlr \ Flete + Vlr \ Seguro) / 1000000$.
- **Total Valor FOB y Total Peso Neto** deben ser construidas con la siguiente expresión: $Total \ Valor \ FOB / 1000000$ y $Total \ Peso \ Neto / 1000000$.

Figura 15 Ejemplo expresiones para variables importación año actual



Fuente: U.A.E. DIAN. CETCE.

4.1.2 Revisión de la estructura de la información

Para asegurar que la información a utilizar en la elaboración del avance sea consistente, coherente y completa, se llevan a cabo revisiones tanto horizontal como vertical y se depuran aquellos datos que no harán parte del avance.

4.1.2.1 Revisión horizontal

Con el propósito de verificar que las bases se encuentren completas y bajo las características necesarias para el procesamiento, se deberá verificar que las bases contengan todas las columnas (variables necesarias).

- Para la base Importaciones año actual (ver figura 16):

Figura 16 Revisión horizontal para importaciones año actual

```
### Revisión horizontal
variables=['Fecha Presentacion', 'Codigo Admon Aduana', 'Codigo Pais Origen', 'Total Valor Fobl', 'Total Valor cif', 'Total Peso Netol',
          'Total Valor Cif Pesos', 'Total Subtotal Aranceli', 'Total Subtotal Ival', 'Numero Aceptacion Declaracion', 'Codigo Regimenl',
          'Valor Ajustel', 'Nume Declara Exportacion', 'Número Factura', 'Número Declaracion Anteri', 'Fecha Declaracion Anterio',
          'Subpartida Arancelaria', 'Número Levante', 'Fecha Levante']
var_df=list(impo_actual.columns)

variables_faltantes = []
for variable in variables:
    if variable not in var_df:
        variables_faltantes.append(variable)

if len(variables_faltantes)>0:
    print("Por favor revise la estructura de las variables de la base de datos")
    for variable in variables_faltantes:
        archivo_log.write("Variable faltante en Impo actual " + variable+"\n")
    sys.exit()
else:
    print("Revisión horizontal realizada correctamente")
    archivo_log.write("Revisión horizontal realizada correctamente " + variable+"\n")

### Revisión vertical
print("Valor FOB inicial: " + str(impo_actual['Total Valor Fobl'].sum()))
print("Peso neto inicial: " + str(impo_actual['Total Peso Netol'].sum()))
print("Registros iniciales: " + str(impo_actual['Codigo Admon Aduana'].count()))
archivo_log.write("Valor FOB inicial: " + str(impo_actual['Total Valor Fobl'].sum())+"\n")
archivo_log.write("Peso neto inicial: " + str(impo_actual['Total Peso Netol'].sum())+"\n")
archivo_log.write("Registros iniciales: " + str(impo_actual['Codigo Admon Aduana'].count())+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

- Para la base Exportaciones año actual (ver figura 17):

Figura 17 Revisión horizontal para exportaciones año actual

```
### Revisión horizontal
variables=[ 'Cod Administracion Pres', 'Ide Declaracion Exportacion', 'Num Item', 'Fec Declaracion Exportacion', 'Cod Colombia',
           'Val Total Fob Us', 'Num Peso Neto', 'Cod País Destino', 'Nit Exportador']

var_df=list(expo_actual.columns)

variables_faltantes = []
for variable in variables:
    if variable not in var_df:
        variables_faltantes.append(variable)

if len(variables_faltantes)>0:
    print("Por favor revise la estructura de las variables de la base de datos")
    for variable in variables_faltantes:
        archivo_log.write("Variable faltante en Expo actual " + variable+"\n")
    sys.exit()
else:
    print("Revisión horizontal realizada correctamente")
    archivo_log.write("Revisión horizontal realizada correctamente " + variable+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

4.1.2.2 Revisión vertical

Con el propósito de verificar que los registros correspondan a la base de datos cargada por parte de la Subdirección de Procesamiento de Datos, se realiza la suma de las variables Valor FOB y Peso Neto, además, el conteo del total de registros.

- Para la base Importaciones año actual (ver figura 18):

Figura 18 Revisión vertical para importaciones año actual

```
### Revisión vertical
print("Valor FOB inicial: " + str(impo_actual['Total Valor Fob1'].sum()))
print("Peso neto inicial: " + str(impo_actual['Total Peso Netol'].sum()))
print("Registros iniciales: " + str(impo_actual['Codigo Admon Aduana'].count()))
archivo_log.write("Valor FOB inicial: " + str(impo_actual['Total Valor Fob1'].sum())+"\n")
archivo_log.write("Peso neto inicial: " + str(impo_actual['Total Peso Netol'].sum())+"\n")
archivo_log.write("Registros iniciales: " + str(impo_actual['Codigo Admon Aduana'].count())+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

- Para la base Exportaciones año actúa (ver figura 19):

Figura 19 Revisión vertical para exportaciones año actual

```
### Revisión vertical
print("Valor FOB inicial: " + str(expo_actual['Val Total Fob Us'].sum()))
print("Peso neto inicial: " + str(expo_actual['Num Peso Neto'].sum()))
print("Registros iniciales: " + str(expo_actual['Ide Declaracion Exportacion'].count()))
archivo_log.write("Valor FOB inicial: " + str(expo_actual['Val Total Fob Us'].sum())+"\n")
archivo_log.write("Peso neto inicial: " + str(expo_actual['Num Peso Neto'].sum())+"\n")
archivo_log.write("Registros iniciales: " + str(expo_actual['Ide Declaracion Exportacion'].count())+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

4.2 Clasificación, codificación, revisión y validación

Una vez se cuente con las bases de datos que se indican en el punto anterior, se continúa con el proceso de validación para las bases correspondientes al año actual (las bases relacionadas con el año anterior no tendrán validación en esta operación, dado que ya se encuentran en la condición de certificadas), de acuerdo con los puntos que se describen a continuación:

4.2.1 Validación importaciones año actual

La información correspondiente a las importaciones del año actual, tal como se parametrizó anteriormente, desde el día 01 del mes hasta el sábado de la semana precedente a la fecha de elaboración del avance, contiene los registros de las declaraciones presentadas en el periodo y que no han surtido un proceso de validación estadística. Por consiguiente, se deben desarrollar las siguientes acciones sobre la base (**nota: este instructivo describe la lógica que se debe aplicar en las validaciones, no obstante, los scripts harán la mayor parte del proceso automáticamente. Los pasos en cursiva y subrayados indicarán aquellas acciones que debe ejecutar el funcionario a cargo:**).

4.2.1.1 Eliminar registros en blanco

El script de consulta en Cognos para la generación de la base de datos de importaciones del año actual excluye declaraciones que cuenten con correcciones en el mismo periodo. En consecuencia, en la base exportada, los registros anteriores a las correcciones son modificados y reportados con valores en cero (denominados para este ejercicio como registros en blanco). Por consiguiente, en la elaboración del Avance, se procede a la eliminación de estos, identificando para tal fin, todos los registros con valores en cero (0) para las variables *Total Valor FOB1*, *Total Valor cif* y *Total Peso Neto1*, tal como se observa a continuación en la figura 20:

Figura 20 Eliminar registros en blanco

```
#### Eliminar registros en blanco
impo_actual = impo_actual.sort_values(by=['Total Valor Fobl', 'Total Valor cif', 'Total Peso Netol'], ascending=[True, True, True])
reg_impo_anter = impo_actual.shape[0]
impo_actual = impo_actual.loc[~(impo_actual['Total Valor Fobl'] == 0) | ~(impo_actual['Total Valor cif'] == 0) |
                           ~(impo_actual['Total Peso Netol'] == 0)]
reg_impo_despues = impo_actual.shape[0]
archivo_log.write("Registros en blanco eliminados: " + str(reg_impo_anter - reg_impo_despues)+"\n")
print("Registros en blanco eliminados: " + str(reg_impo_anter - reg_impo_despues))
print("Registros en base: " + str(len(impo_actual))+"\n")
archivo_log.write("Registros luego de eliminar registros en blanco eliminados: " + str("Registros en base: " + str(len(impo_actual)))+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

El script almacenará en el archivo log, un conteo del total de registros que son eliminados en este paso.

4.2.1.2 Aplicar ajustes anteriores

Antes de continuar con los siguientes pasos de validación, se deben aplicar los ajustes que fueron identificados en las validaciones de anteriores avances pero que correspondan a la vigencia del mismo mes (es decir, del día 01 al día correspondiente al último sábado a la semana anterior del mismo mes). Si en el periodo de análisis se cierra el mes, el proceso se realiza en dos partes: la primera parte entre el 01 al 28,29,30,31 según corresponda y la segunda, para el día 01 al día que corresponda al sábado de la semana inmediatamente anterior. Por ejemplo, para octubre de 2023, la semana anterior comprendió entre el domingo 29 de octubre y el sábado 04 de noviembre; por consiguiente, la validación se realizó, primero para una base del 01 al 31 de octubre y luego para una base del 01 al 04 de noviembre (esto solo aplica cuando la semana anterior incluya cuatro o más días del siguiente mes, en caso contrario, la validación solo se ejecuta hasta el cierre del mes).

Para identificar los registros y realizar los ajustes que correspondan, se debe:

1. Copie el archivo reporte ajustes A### del Avance de la semana inmediatamente anterior. Pégue el archivo en la carpeta del Avance que está elaborando (ver Figura 5) y ajuste el número para que corresponda con la edición que está elaborando.
2. Abra el archivo de ajustes del último avance. Este archivo corresponde a un formato Excel en que encontrará dos hojas. Ubíquese en la hoja “Importaciones”, la cual debe ser similar a la que se muestra en la siguiente captura de pantalla, figura 21:

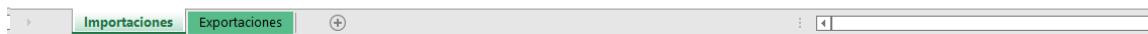
Figura 21 Ejemplo del reporte de ajustes estadísticos - Importaciones



AVANCE DE COMERCIO EXTERIOR # 984
REPORTE DE AJUSTES ESTADÍSTICOS
Periodo: 01 al 11 de noviembre de 2023

Importaciones				
Declaración	Inconsistencia identificada	Ajuste realizado	Profesional que ajusta	Observación
352023000546570	Duplicado de 352023000541242	Eliminar registro	Luis Alejandro Pico	
352023000549900	Possible duplicado método 2	Registro validado	Luis Alejandro Pico	
352023000549899	Possible duplicado método 2	Registro validado	Luis Alejandro Pico	
352023000562079	Possible duplicado método 2	Registro validado	Luis Alejandro Pico	
352023000562078	Possible duplicado método 2	Registro validado	Luis Alejandro Pico	
352023000562080	Possible duplicado método 2	Registro validado	Luis Alejandro Pico	
352023000562083	Possible duplicado método 2	Registro validado	Luis Alejandro Pico	
352023000562082	Possible duplicado método 2	Registro validado	Luis Alejandro Pico	
352023000562081	Possible duplicado método 2	Registro validado	Luis Alejandro Pico	

Elaboró: Coordinación de Estadística Tributaria y de Comercio Exterior - Subdirección de Estudios Económicos. DGEA. DIAN.



Fuente: U.A.E. DIAN. CETCE.



3. En la base, aplique los cambios que se indica en la columna “Ajuste realizado”. Para ello, busque la declaración de importación en la base. Busque el número que encuentra en la columna “Declaración” del archivo de ajustes en la columna “Numero Aceptacion Declaración” en la base de importaciones año actual. Si en el ajuste realizado se encuentra el texto “Registro validado”, significa que esa declaración ya fue revisada y no debe realizarse ningún ajuste (ver figura 22).

Figura 22 Aplicar ajustes identificados en avances anteriores

```
#### Eliminar registros en blanco
impo_actual = impo_actual.sort_values(by=['Total Valor Fob1', 'Total Valor cif', 'Total Peso Netol'], ascending=[True, True, True])
reg_impo_anter = impo_actual.shape[0]
impo_actual = impo_actual.loc[(impo_actual['Total Valor Fob1'] == 0) | ~(impo_actual['Total Valor cif'] == 0) |
~(impo_actual['Total Peso Netol'] == 0)]
reg_impo_despues = impo_actual.shape[0]
archivo_log.write("Registros en blanco eliminados: "+ str(reg_impo_anter - reg_impo_despues)+"\n")
print("Registros en blanco eliminados: "+ str(reg_impo_anter - reg_impo_despues))
print("Registros en base: "+ str(len(impo_actual))+"\n")
archivo_log.write("Registros luego de eliminar registros en blanco eliminados: "+ str("Registros en base: "+ str(len(impo_actual)))+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

4.2.1.3 Identificación y eliminación de registros repetidos

Para identificar y eliminar los registros duplicados en la base de datos, se desarrollan los siguientes dos procesos:

4.2.1.3.1 Duplicados método 1

En este paso se busca identificar las declaraciones que presentan declaración de corrección en el mismo mes de análisis, en tal sentido, se debe eliminar la primera declaración y dejar la última, es decir, reemplazar la inicial por la corrección.

Para identificar los registros y eliminarlos, se debe:

1. Filtre los registros en la base, que en la columna “Numero Declaracion Anteri” tenga valores diferentes a nulos o vacíos. En este caso, indica que existen dos declaraciones, una inicial y una anterior.
2. Valide que la columna “Fecha Declaracion Anterior” solo tenga fechas para el mes de análisis. Si la base tiene registros fuera de estas fechas, es posible que se haya presentado algún error en la parametrización de la consulta para descargar de la base. En caso de error, revise la consulta realizada en Cognos, corrija lo que corresponda y vuelva a iniciar el proceso de validación.
3. Guarde, en una nueva hoja o en un nuevo DataFrame los números de las declaraciones que se identificaron en el primer paso de este numeral, siempre y cuando cumplan lo indicado en el punto 2.
4. Realice un cruce (en Python, merge) en la que se busque los “Numero Aceptacion Declaracion” de la base de importaciones año actual en la nueva hoja o DataFrame del punto 3.
5. Guarde un log con los registros que el “Numero Aceptacion Declaracion” fue encontrara en la nueva hoja o DataFrame.
6. Borre de la base de importaciones del año actual, los registros del punto 5.

Figura 23 Depuración duplicados método 1

```
#### Duplicados método 1
m1 = impo_actual.loc[pd.notna(impo_actual['Número Declaracion Anterior'])]
fecha_m1 = m1.loc[(m1['Fecha Declaracion Anterior'] < fecha_inicio) | (m1['Fecha Declaracion Anterior'] > fecha_fin)]
archivo_log.write("Registros en método 1 que están fuera del rango de fechas: " + str(len(fecha_m1))+"\n")
if len(fecha_m1)>0:
    print("Error en validación del método 1, revise el log y la base ingresada")
    archivo_log.close()
    sys.exit()

metodo1 = impo_actual.merge(m1[['Número Declaracion Anterior']], left_on='Número Aceptacion Declaracion',
                             right_on='Número Declaracion Anterior', how='inner')
metodo1 = metodo1 .drop_duplicates()

impo_actual=impo_actual.merge(metodo1[['Número Aceptacion Declaracion']], left_on='Número Aceptacion Declaracion',
                               right_on='Número Aceptacion Declaracion', indicator=True, how='left')
reg_impo_anteriores=impo_actual.shape[0]
impo_actual = impo_actual.loc[impo_actual['_merge'] != 'both']
reg_impo_despues = impo_actual.shape[0]
metodo1['Número Aceptacion Declaracion'].to_excel(path+"/Metodo1.xlsx", index=False)
archivo_log.write("Registros eliminados por método 1: " + str(reg_impo_anteriores - reg_impo_despues)+"\n")
print("Registros eliminados por método 1: " + str(reg_impo_anteriores - reg_impo_despues))
print("Registros en base: " + str(len(impo_actual))+"\n")
archivo_log.write("Registros luego de aplicar método 1: " + str("Registros en base: " + str(len(impo_actual)))+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

Nota: los registros que se eliminan en este paso son almacenados automáticamente en la carpeta del Avance que se está desarrollando, con el nombre “Método1.xlsx”, tanto para trazabilidad como posibles validaciones posteriores.

4.2.1.3.2 Duplicados método 2

En este paso se busca identificar las declaraciones duplicadas a partir de la información de la factura, país de origen y subpartida arancelaria. En este paso se evidencian registros con diferente número de aceptación, pero misma información en la descripción y valores para las variables antes señaladas; situación que se debe al registro en momentos del tiempo diferente, uno de ellos finalizado (con levante) y los demás hasta el estado presentada en bancos.

Para identificar los registros y eliminarlos, se debe:

1. Organice la base de manera ascendente por la variable “Número Factura”.
2. Seleccione los registros cuyo valor FOB sea mayor a 0.5 millones de dólares.
3. Con los registros resultantes del paso anterior, identifique cada registro que, con respecto a la fila anterior, tengan los mismos valores en las columnas “Código País Origen1”, “Subpartida Arancelaria1” y “Número Factura”.
4. De los registros que se generan en el paso anterior, se excluyen automáticamente aquellos candidatos que cuenten con número de levante. Es decir, si hay dos posibles declaraciones duplicadas y ambas tienen levante, el script las omitirá; este paso solo dejará las parejas o grupos en los cuales existan candidatos a duplicados y al menos uno de ellos no tenga levante.
5. Guarde, en un archivo de Excel o en un nuevo DataFrame, los registros candidatos a ser duplicados del método 2.
6. Consulte en SYGA cada una de las declaraciones que son consideradas como posibles duplicados y en función de la descripción y demás valores allí presentes, determine si se trata o no de un duplicado.
7. En caso de identificarse como duplicado, regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Eliminar registro”.

		Importaciones	
Declaración	Inconsistencia identificada	Ajuste realizado	Profesional que ajusta
352023000546570	Duplicado de 352023000541242	Eliminar registro	Luis Alejandro Pico

8. *Si la validación da como resultado que el registro NO es un duplicado, regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Registro validado”.*

		Importaciones	
Declaración	Inconsistencia identificada	Ajuste realizado	Profesional que ajusta
352023000549899	Possible duplicado método 2	Registro validado	Luis Alejandro Pico

9. *Una vez revisados todos los candidatos a duplicados por método 2, guarde los cambios en el archivo de Reporte de ajustes de avance y ciérrelo.*
10. *Vuelva a correr el script para el procesamiento de importaciones.*
11. *Si la validación genera de nuevo candidatos a duplicados por el método 2, vuelva a ejecutar los pasos del 1 al 11. En caso contrario, puede continuar con el siguiente paso de validación de la base.*

El siguiente, figura 24, corresponde al código de implementación de la validación del método 2:

Figura 24 Depuración duplicados método 2

```
#### Duplicados método 2
m2 = impo_actual.sort_values(by=['Número Factura'], ascending=[True])
m2 = m2.drop(columns=[`merge`, 'Declaracion', 'Tipo Declaracion', 'Fecha Declaracion Anterior', 'Número Declaracion Anterior', 'Número Declara Exportacion', 'Valor Ajustel', 'Codigo Regimen'])
m2 = m2.drop(columns=[`Declaración`, 'Tipo Declaracion', 'Fecha Declaracion Anterior', 'Número Declaracion Anterior', 'Número Declara Exportacion', 'Valor Ajustel', 'Codigo Regimen'])
m2 = m2.loc[(m2['Total Valor Fob'] >= 0.5) & (m2['Ajuste realizado'] != 'Registro validado')]

m2_iguales = [False] * len(m2)
m2[`Iguales`] = False
m2[`verificado`] = 'No'
for i in range(len(m2)-1):
    fila_actual = m2.iloc[i]
    fila_siguiente = m2.iloc[i + 1]

    if (fila_actual['Codigo País Origen'] == fila_siguiente['Codigo País Origen']) and
       fila_actual['Subpartida Arancelaria'] == fila_siguiente['Subpartida Arancelaria'] and
       fila_actual['Número Factura'] == fila_siguiente['Número Factura']):
        m2_iguales[i] = math.isnan(fila_actual['Número Levante'])
        m2_iguales[i + 1] = math.isnan(fila_siguiente['Número Levante'])

m2[`Iguales`] = m2_iguales

posibles_m2 = pd.DataFrame()
posibles_m2[`Llave`] = m2[m2[`Iguales`] == True][`Número Factura`].unique()
validar_m2=m2.merge(posibles_m2[[`Llave`]], left_on='Número Factura', right_on='Llave', how='right')
validar_m2.to_excel(path+`validar_método2.xlsx`, index=False)
if len(validar_m2)>0:
    print("Existen registro del método 2 sin validar. Revise, ajuste y vuelva a ejecutar")
    sys.exit()
else:
    print("Validación método 2 ejecutada correctamente")
```

Fuente: U.A.E. DIAN. CETCE.

4.2.1.4 Reparaciones - Régimen C605, C606 y C611

En este paso se busca identificar y eliminar las declaraciones correspondientes a registros con régimen C605 (Reimportación de mercancías reparadas o reemplazadas, sin garantía vigente del fabricante o proveedor), C606 (Reimportación de mercancías reparadas o reemplazadas, sin garantía vigente del fabricante o proveedor, excluidos del impuesto sobre las ventas, clasificables en una subpartida

arancelaria parcialmente sujeta a esta.) o C611(Reimportación de mercancías exportadas temporalmente para elaboración, reparación o transformación sujetas a un diferimiento arancelario del 0%, exentas o excluidas de arancel), con ajuste negativo y declaración de exportación. En los registros que cumplen estas tres condiciones, se realiza la trazabilidad y análisis para identificar si corresponden a importaciones de bienes reparados. Se analizan los registros con valor FOB mayores a 0,1 millones de dólares.

Para identificar los registros y eliminarlos, se debe:

1. Filtre las declaraciones que en la variable “Código Regimen1” correspondan a C605, C606 o C611.
2. Seleccione los registros cuyo valor FOB sea mayor a 0.1 millones de dólares.
3. Guarde, en un archivo de Excel o en un nuevo DataFrame, los registros candidatos a ser eliminados en este paso.
4. Consulte en SYGA cada una de las declaraciones que son candidatas para ser eliminadas en este paso y en función de la descripción, determine si corresponde a bienes que fueron reparados.
5. Para los registros que se confirme en la descripción que fueron reparados, realice una consulta en la carpeta de Exportaciones, tomando el número de la declaración de exportación.
6. Si el registro cumple el paso 4 y no se encuentra información al realizar la consulta del paso 5, regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Eliminar registro”.

Importaciones			
Declaración	Inconsistencia identificada	Ajuste realizado	Profesional que ajusta
192023000073227		Eliminar registro	Luis Alejandro Pico Silva
482023000623926		Eliminar registro	Luis Alejandro Pico Silva
192023000073663		Eliminar registro	Luis Alejandro Pico Silva
192023000073665		Eliminar registro	Luis Alejandro Pico Silva
192023000073926		Eliminar registro	Luis Alejandro Pico Silva
32023001522138		Eliminar registro	Luis Alejandro Pico Silva
192023000073662		Eliminar registro	Luis Alejandro Pico Silva
32023001545610		Eliminar registro	Luis Alejandro Pico Silva
32023001536705		Eliminar registro	Luis Alejandro Pico Silva

7. Si el registro no cumple el paso 4 (es decir, no corresponde a una reparación) o si cumple el paso 4 pero se encuentra traza de información al realizar la consulta del paso 5, regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Registro validado”.

Importaciones				
Declaración	Inconsistencia identificada	Ajuste realizado	Profesional que ajusta	Observación
32023001494915	Posible reparación	Registro validado	Luis Alejandro Pico Silva	No identifica si se trata de reparación
482023000682245	Posible reparación	Registro validado	Luis Alejandro Pico Silva	No identifica si se trata de reparación

8. Una vez revisados todos los registros candidatos a ser eliminados, guarde los cambios en el archivo de Reporte de ajustes de avance y ciérrelo.
9. Vuelva a correr el script de procesamiento de importaciones.
10. Si la validación genera nuevos candidatos a ser eliminados, vuelva a ejecutar los pasos del 1 al 9. En caso contrario, puede continuar con el siguiente paso de validación de la base.

El siguiente, figura 25, corresponde al código de implementación de la validación de Reparaciones - Régimen C605, C606 y C611:

Figura 25 Depuración modalidad C605, C606 y C611

```
#### Reparaciones - Régimen C605, C606 y C611
reparacion=impo_actual.loc[((impo_actual['Codigo Regimen1'] == 'C605') | (impo_actual['Codigo Regimen1'] == 'C606') |
                           (impo_actual['Codigo Regimen1'] == 'C611')) & (impo_actual['Valor Ajustel'] != 0) &
                           (impo_actual['Total Valor Fob1'] >= 0.1) & (impo_actual['Ajuste realizado']=="")]
reparacion.to_excel(path+"/validar_reparaciones.xlsx", index=False)
if len(reparacion)>0:
    print("Existen registros del Régimen C605, C606 y C611 sin validar. Revise, ajuste y vuelva a ejecutar")
    sys.exit()
else:
    print("Validación del régimen C605, C606 y C611 ejecutada correctamente")
```

Fuente: U.A.E. DIAN. CETCE.

4.2.2 Validación exportaciones año actual

La información correspondiente a las exportaciones del año actual, tal como se parametrizó, desde el día 01 del mes hasta el sábado de la semana precedente a la fecha de elaboración del avance, contiene los registros de las declaraciones presentadas en el periodo y que no han surtido un proceso de validación estadística. Por consiguiente, se deben desarrollar las siguientes acciones sobre la base (**nota: este instructivo describe la lógica que se debe aplicar en las validaciones, no obstante, los scripts harán la mayor parte del proceso automáticamente. Los pasos en cursiva y subrayados indicarán aquellas acciones que debe ejecutar el funcionario a cargo**):

4.2.2.1 Ajustes en peso neto

Los estándares para el procesamiento de la información estadística de comercio exterior establecen que los registros cuya unidad comercial corresponda a kilogramos, el peso neto debe ser igual a este (en millones). Por consiguiente, se deben realizar los siguientes pasos:

1. Identifique los registros cuya unidad comercial corresponde a kilogramos.
2. De los registros anteriores, identifique aquellos que estén en blanco (vacíos) o su valor absoluto corresponda a cero (0) – sin aproximación.
3. Para los registros que coincidan con los anteriores criterios, se debe reemplazar el valor neto por el valor que se registre en la cantidad, dividido en 1000000.

Figura 26 Ajuste peso neto

```
##Ajuste peso neto##
ajuste_peso_expo = expo_actual.loc[(expo_actual['Cod Unidad Comercial'] == 'kg') & (expo_actual['Num Peso Neto'] == 0) |
                                      & (expo_actual['Num Cantidad Productos'] > 0)]
ajuste_peso_expo.to_excel(path+"/ajustes_peso.xlsx", index=False)
ajustes_realizar=len(ajuste_peso_expo)
print("Número de registros ajustados por peso neto: "+str(ajustes_realizar))
archivo_log.write("Número de registros a ajustar por peso neto: "+str(ajustes_realizar)+"\n")
expo_actual.loc[(expo_actual['Cod Unidad Comercial'] == 'kg') & (expo_actual['Num Peso Neto'] == 0) |
                 & (expo_actual['Num Cantidad Productos'] > 0),
                 'Num Peso Neto'] = expo_actual['Num Cantidad Productos'] / 1000000
```

Fuente: U.A.E. DIAN. CETCE.

Nota: los registros que se eliminan en este paso son almacenados automáticamente en la carpeta del Avance que se está desarrollando, con el nombre “ajustes_peso.xlsx”, tanto para trazabilidad como posibles validaciones posteriores.

4.2.2.2 Aplicar ajustes anteriores

Antes de continuar con los siguientes pasos de validación, se deben aplicar los ajustes que fueron identificados en las validaciones de anteriores avances. Tenga presente que únicamente se aplican ajustes para el mes de referencia, es decir, del día 01 al día correspondiente al último sábado a la semana anterior del mismo mes. Si en el periodo de análisis se cierre el mes, el proceso se realiza en dos partes: la primera parte entre el 01 al 28,29,30,31 según corresponda y la segunda, para el día 01 al día que corresponda al sábado de la semana inmediatamente anterior. Por ejemplo, para octubre de 2023, el avance comprendió entre el domingo 29 de octubre y el sábado 04 de noviembre; por consiguiente, la validación se realizó, primero para una base del 01 al 31 de octubre y luego para una base del 01 al 04 de noviembre.

Para identificar los registros y realizar los ajustes que corresponda, se debe:

1. Abrir el archivo de ajustes del último avance. Este archivo corresponde a un formato Excel en que encontrará dos hojas. Ubíquese en la hoja “Exportaciones”, la cual debe ser similar a la que a siguiente captura de pantalla:

AVANCE DE COMERCIO EXTERIOR # 986 REPORTE DE AJUSTES ESTADÍSTICOS Periodo: 01 al 25 de noviembre de 2023						
Exportaciones						
Llave	Declaración	Item	Inconsistencia identificada	Ajuste realizado	Nuevo valor	Profesional que ajusta
6007724603389-1	6007724603389	1	Software	Eliminar registro		Luis Alejandro Pico Silva
6007724825824-1	6007724825824	1	Aviones	Eliminar registro		Luis Alejandro Pico Silva
6007723543719-1	6007723543719	1	Valor FOB en pesos	Se ajusta el Valor FOB	0.00436163408797305	Luis Alejandro Pico Silva
6007724653341-1	6007724653341	1	Valor FOB en pesos	Se ajusta el Valor FOB	0.00135845368084572	Luis Alejandro Pico Silva

Fuente: U.A.E. DIAN. CETCE.

2. En la base, aplique los cambios que se indica en la columna “Ajuste realizado”. Para ello, debe crear en la base una columna “Llave” conformada por la estructura “número de declaración – ítem”. En la base, busque la llave que se indica en el reporte de ajustes y luego aplique los cambios que allí se indiquen. Si en el ajuste realizado se encuentra el texto “Registro validado”, significa que esa declaración ya fue revisada y no debe realizarse cambio alguno, (ver figura 27).

Figura 27 Aplicar ajustes identificados en avances anteriores- Exportaciones

```
###Aplicar ajustes identificados en avances anteriores###
expo_actual['Llave'] = expo_actual['Id Declaracion Exportacion'].astype(str) + "-" + expo_actual['Num Item'].astype(str)
expo_actual = expo_actual.merge(ajustes_expo[['Llave', 'Ajuste realizado', 'Nuevo valor']], left_on='Llave', right_on='Llave', how='left')
req_expo_anteriores = expo_actual.shape[0]
expo_actual = expo_actual.loc[expo_actual['Ajuste realizado'] != 'Eliminar registro']
req_expo_despues = expo_actual.shape[0]
print("Número de registros eliminados por ajustes anteriores: "+str(req_expo_anteriores-req_expo_despues))
archivo_log.write("Número de registros eliminados por ajustes anteriores: "+str(req_expo_anteriores-req_expo_despues)+"\n")
ajustes_realizar=(expo_actual['Ajuste realizado']=='Se ajusta el Valor FOB').sum()
expo_actual.loc[expo_actual['Ajuste realizado']=='Se ajusta el Valor FOB', ['Val Total Fob Us']] = expo_actual['Nuevo valor']
print("Número de registros ajustados por valor FOB: "+str(ajustes_realizar))
archivo_log.write("Número de registros ajustados por valor FOB: "+str(ajustes_realizar)+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

4.2.2.3 Registros correspondientes a software

Las bases elaboradas con este instructivo corresponden a registros de mercancías y no de servicios. Por consiguiente, se deben verificar que en las subpartidas arancelarias 8523.49.90.00, 8523.51.00.00 y 8523.41.00.00 no se encuentren registros relacionados con software.

Para identificar los registros y realizar los ajustes que correspondan, se debe:

1. Filtrar la base por las subpartidas arancelarias 8523.49.90.00, 8523.51.00.00 y 8523.41.00.00.
2. Seleccione los registros cuyo valor FOB sea mayor a 0.1 millones de dólares.
3. Guarde, en un archivo de Excel o en un nuevo DataFrame, los registros candidatos a ser eliminados en este paso.
4. Consulte en MUISCA cada una de las declaraciones que son candidatas para ser eliminadas en este paso y en función de la descripción, determine si corresponde o no a un software.
5. Si la declaración corresponde a una exportación de software, regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Eliminar registro”.

Exportaciones						
Llave	Declaración	Item	Inconsistencia identificada	Ajuste realizado	Nuevo valor	Profesional que ajusta
6007724603389-1	6007724603389	1	Software	Eliminar registro		Luis Alejandro Pico Silva
6007724825824-1	6007724825824	1	Aviones	Eliminar registro		Luis Alejandro Pico Silva

6. Si en la declaración se identifica que realmente corresponde a una mercancía (bien material), regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Registro validado”.

Exportaciones						
Llave	Declaración	Item	Inconsistencia identificada	Ajuste realizado	Nuevo valor	Profesional que ajusta
6007724603388-1	6007724603388	1	Possible software	Registro validado		Luis Alejandro Pico Silva

7. Una vez revisados todos los registros candidatos a ser eliminados, guarde los cambios en el archivo de Reporte de ajustes de avance y ciérrelo.
8. Vuelva a correr el script de validación de exportaciones.
9. Si la validación genera de nuevo candidatos a ser eliminados, vuelva a ejecutar los pasos del 1 al 7. En caso contrario, puede continuar con el siguiente paso de validación de la base.

Figura 28 Ajustes para validación de software

```
###Ajuste - software###
subpartidas_software = [8523499000, 8523510000, 8525602000, 8523410000]
ajuste_software = expo_actual.loc[expo_actual['Cod Colombia'].isin(subpartidas_software)& (expo_actual['Val Total Fob Us'] >= 0.1)]
ajuste_software.to_excel(path+"/validar_software.xlsx", index=False)
if len(ajuste_software)>0:
    print("Hay declaraciones sin validación (software). Revise, ajuste y vuelva a correr")
    sys.exit()
else:
    archivo.log.write("Validación 'software' realizada\n")
    print("Validación 'software' realizada")

###Ajuste - Aviones###
ajuste_aviones = expo_actual.loc[expo_actual['Cod Colombia']==8802400000]
ajuste_aviones.to_excel(path+"/validar_aviones.xlsx", index=False)
if len(ajuste_aviones)>0:
    print("Hay declaraciones sin validación (aviones). Revise, ajuste y vuelva a correr")
    sys.exit()
else:
    archivo.log.write("Validación 'aviones' realizada\n")
    print("Validación 'aviones' realizada")
```

Fuente: U.A.E. DIAN. CETCE.

4.2.2.4 Registros correspondientes a aviones

En la base deben ser eliminados los registros que contengan la subpartida arancelaria 8802.40.00.00 (Aviones y demás aeronaves, de peso en vacío, superior a 15000 kilogramos), que correspondan a las empresas Avianca o Tampa; teniendo en cuenta que en la base de datos el código de modalidad de la importación se encuentra con C100 (Importación Ordinaria) y en la declaración de exportación con C190 (Importación temporal para perfeccionamiento activo de materias primas e insumos) o C196 (Importación temporal de bienes de capital sin pago de tributos aduaneros destinados a ser reexportados de conformidad con lo establecido en el artículo 163 del decreto 2685 de 1999).

Para identificar los registros y realizar los ajustes que correspondan, se debe:

1. Filtrar la base por la subpartida arancelaria 8802.40.00.00
2. Guarde, en un archivo de Excel o en un nuevo DataFrame, los registros candidatos a ser eliminados en este paso.
3. Consulte en MUISCA cada una de las declaraciones que son candidatas para ser eliminadas en este paso y en función de la descripción, determine si corresponde o no a las empresas Avianca o Tampa. Verifique que el régimen precedente corresponda a la modalidad C190 (Importación temporal para perfeccionamiento activo de materias primas e insumos) o C196 (Importación temporal de bienes de capital sin pago de tributos aduaneros destinados a ser reexportados de conformidad con lo establecido en el artículo 163 del decreto 2685 de 1999)

113. Unidad de medida plazo	Cód.	114. Plazo	115. País de origen	Cód.	116. Región de origen	Cód.
ALEMANIA	DE					
Régimen precedente						
118. Aduana precedente	Cód.	119. No. Declaración precedente	120. Año aceptación	121. Régimen precedente	Cód.	122. Cód. Modalidad precedente
Aduana Medellín	9 0	9020190001851103	2019	Importacion temporal para perf	C	C 1 9 0
						1

4. Si la declaración cumple con los criterios antes señalados, regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Eliminar registro”.

Llave	Declaración	Item	Inconsistencia identificada	Exportaciones	Nuevo valor	Profesional que ajusta
6007722786280-1	6007722786280	1	Aeronave Avianca con C190	Eliminar registro		Luis Alejandro Pico Silva

5. Si la declaración no cumple con los criterios señalados anteriormente, regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Registro validado”.

Llave	Declaración	Item	Inconsistencia identificada	Exportaciones	Nuevo valor	Profesional que ajusta
6007723279328-1	6007723279328	1	Possible Aeronave Avianca con C190	Registro validado		Luis Alejandro Pico Silva

6. Una vez revisados todos los registros candidatos a ser eliminados, guarde los cambios en el archivo de Reporte de ajustes de avance y ciérrelo.
7. Vuelva a correr el script de validación de exportaciones.
8. Si la validación genera de nuevo candidatos a ser eliminados, vuelva a ejecutar los pasos del 1 al 7. En caso contrario, puede continuar con el siguiente paso de validación de la base.

Figura 29 Ajustes para validación de aviones

```
###Ajuste - Aviones###
ajuste_aviones = expo_actual.loc[expo_actual['Cod Colombia']==8802400000]
ajuste_aviones.to_excel(path+"/validar_aviones.xlsx", index=False)
if len(ajuste_aviones)>0:
    print("Hay declaraciones sin validación (aviones). Revise, ajuste y vuelva a correr")
    sys.exit()
else:
    archivo_log.write("Validación 'aviones' realizada\n")
    print("Validación 'aviones' realizada")
```

Fuente: U.A.E. DIAN. CETCE.

4.2.2.5 Registros zonas francas

Con ocasión de la entrada en vigor de la Resolución 0000046 del 26 de julio de 2019, sobre los nuevos lineamientos para la “exportación de joyas, oro, esmeraldas y demás piedras preciosas, así como sus derivados”, capítulo 71 del Arancel de Aduanas, y con el fin de mantener las estadísticas conforme a la metodología de la ONU, se deben eliminar de la base “Exportaciones no certificadas año actual” todas las exportaciones del capítulo 71 que se hayan efectuado desde la zona franca. Para esto, es necesario organizar la base por subpartida arancelaria y luego identificar los NIT de las empresas exportadoras clasificadas como usuarios de la zona franca y eliminar estos registros.

Para identificar los registros y realizar los ajustes que corresponda, se debe:

1. Derivar una nueva variable “Capítulo”, a partir del código de la subpartida arancelaria.
2. Realizar un *merge* entre la base y el directorio de usuarios calificados de zonas francas (ubicado xxx), tomando como llave el número de NIT.
3. Filtrar la base por el capítulo 71.
4. Identificar los registros que correspondan al capítulo 71 y que a su vez el exportador corresponda a un usuario calificado de zona franca.
5. Guarde, en un archivo de Excel o en un nuevo DataFrame, los registros que serán eliminados en este paso.

Nota: cuando se esté elaborando el avance correspondiente al cierre del mes, el listado de las declaraciones eliminadas en este paso debe ser remitidas a la persona encargada del procesamiento de las bases certificadas.

Figura 30 Ajustes para zonas francas

```
###Ajuste - Zonas Francas###
expo_actual['Capítulo']=expo_actual['Cod Colombia'].apply(lambda x: int(str(x)[:2]) if len(str(x)) == 10 else int(str(x)[:1]))
expo_actual['Nit_A'] = pd.to_numeric(expo_actual['Nit Exportador'], errors='coerce').fillna(0).astype(int)
expo_actual = expo_actual.merge(usuario_zona_francia[['NIT']], left_on='Nit_A', right_on='NIT', how='left')
ajuste_zonas_francas = expo_actual.loc[(expo_actual['Capítulo'] == 71) & (pd.notna(expo_actual['NIT']))]
ajuste_zonas_francas.to_excel(path+"/ajustes_francas.xlsx", index=False)
print("Ajustes de zonas francas realizado: "+str(len(ajuste_zonas_francas)))
expo_actual = expo_actual.loc[(expo_actual['Capítulo'] != 71) & expo_actual['NIT'].isna()]
```

Fuente: U.A.E. DIAN. CETCE.

Nota: los registros que se eliminan en este paso son almacenados automáticamente en la carpeta del Avance que se está desarrollando, con el nombre “ajustes_zonas_francas.xlsx”, tanto para trazabilidad como posibles validaciones posteriores.

Identificación y ajustes de valores atípicos

En la base de exportaciones, es probable que se encuentren errores en los valores FOB o en el peso neto. Por consiguiente, se procede a generar el precio implícito de cada declaración, con el fin de identificar las posibles inconsistencias en la información.

Para identificar los registros y realizar los ajustes que corresponda, se debe:

1. Calcule una nueva variable “Implícito” correspondiente a la relación entre el Total del valor FOB y el peso neto.
2. Abra el DataFrame e identifique aquellos registros con los mayores o menores valores.
3. Consulte en MUISCA cada una de las declaraciones que usted considere presentan inconsistencias. Importante, detalle todas las características posibles.
4. Si la declaración presenta alguna novedad (normalmente en el valor FOB), regístrelo en el archivo de Reporte de ajustes de avance correspondiente, tal como se muestra a continuación y con el ajuste “Se ajusta el Valor FOB”. Importante: en la columna “Nuevo valor”, registro el valor que debe quedar en la base, ya convertido en unidad de millones.

Exportaciones						
Llave	Declaración	Item	Inconsistencia identificada	Ajuste realizado	Nuevo valor	Profesional que ajusta
6007723543719-1	6007723543719	1	Valor FOB en pesos	Se ajusta el Valor FOB	0.00436163408797306	Luis Alejandro Pico Silva
6007724659341-1	6007724659341	1	Valor FOB en pesos	Se ajusta el Valor FOB	0.0013584598084572	Luis Alejandro Pico Silva

5. Una vez revisados todos los registros de los cuales usted consideraba podrían tener inconsistencias, guarde los cambios en el archivo de Reporte de ajustes de avance y ciérrelo.
6. Vuelva a correr el script de validación de exportaciones.
7. Vuelva y revise el DataFrame para garantizar que los cambios se hayan ejecutado. Si la validación genera de nuevo candidatos a ser ajustados, vuelva a ejecutar los pasos del 2 al 6. En caso contrario, puede continuar con el siguiente paso de validación de la base.

Figura 31 Cálculo de valores implícitos

```
###Ajuste - Implícitos###
expo_actual['Implicito']=expo_actual['Val Total Fob Us']/expo_actual['Num Peso Neto']
expo_actual.sort_values(by=['Implicito'], ascending=[False])
```

Fuente: U.A.E. DIAN. CETCE.

4.2.3 Verificación de consistencia de la información

Una vez ejecutadas las validaciones y ajustes detallados en los numerales 4.2.1 y 4.2.2, se realizan las verificaciones de los totales resultantes, con el fin de garantizar su consistencia:

- Para importaciones año actual:

Figura 32 Revisión de consistencia de la información - Importaciones

```
### Revisión consistencia de la información
print("Valor FOB validado: " + str(impo_actual['Total Valor Fob1'].sum()))
print("Peso neto validado: " + str(impo_actual['Total Peso Netol'].sum()))
print("Registros validados: " + str(impo_actual['Codigo Admon Aduana'].count()))
archivo_log.write("Valor FOB validado: " + str(impo_actual['Total Valor Fob1'].sum())+"\n")
archivo_log.write("Peso neto validado: " + str(impo_actual['Total Peso Netol'].sum())+"\n")
archivo_log.write("Registros validados: " + str(impo_actual['Codigo Admon Aduana'].count())+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

- Para exportaciones año actual:

Figura 33 Revisión de consistencia de la información - Exportaciones

```
### Revisión consistencia de la información
print("Valor FOB validado: " + str(expo_actual['Val Total Fob Us'].sum()))
print("Peso neto validado: " + str(expo_actual['Num Peso Neto'].sum()))
print("Registros validados: " + str(expo_actual['Ide Declaracion Exportacion'].count()))
archivo_log.write("Valor FOB validado: " + str(expo_actual['Val Total Fob Us'].sum())+"\n")
archivo_log.write("Peso neto validado: " + str(expo_actual['Num Peso Neto'].sum())+"\n")
archivo_log.write("Registros validados: " + str(expo_actual['Ide Declaracion Exportacion'].count())+"\n")
```

Fuente: U.A.E. DIAN. CETCE.

4.3 Elaboración del producto para publicación

Con las bases de datos validadas y las bases complementarias generadas en el numeral 4.1, se procede a elaborar los cuadros de salida, de acuerdo con los pasos que se describen a continuación:

4.3.1 Derivación de nuevas variables y unidades

Con el propósito de contar con la información de las tablas paramétricas que complementan la información para la elaboración del avance, se realiza un merge con las paramétricas TotPart (última versión vigente), países, aduanas y clasificación Cuode, que se ubican en la carpeta *Paramétricas* del directorio Avances del OneDrive de la CETCE. Este proceso agrega las descripciones de cada una las tablas a las bases procesadas, para ello, importe las tablas en su última versión y en el paso que corresponda, realice la operación de merge:

Figura 34 Importación de tablas paramétricas

```
paises_BI = pd.read_excel(path+"Parametricas/paises_BI.xlsx",sheet_name="Hojal", header=0, usecols=[0,2])
paises = pd.read_excel(path+"Parametricas/Paises.xlsx",sheet_name="Paises", header=0, usecols=[0,1])
aduanas = pd.read_excel(path+"Parametricas/aduanas.xlsx",sheet_name="Hojal", header=0, usecols=[0,2])
aduanas.rename(columns={'COD ADMINISTRACION_PRESENTADA': 'Cod Administracion Presentada'}, inplace=True)
totpar = pd.read_excel(path+"Parametricas/Totparv75.xlsx",sheet_name="Sin espacios", header=0, usecols=[0,2,5])
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2 Cálculo de agregaciones - Cuadros de salida

A continuación, se describe el proceso para la generación de los cuadros de salida, previos a la construcción del documento. Estos procesamientos se realizan con el apoyo de Python y los resultados son almacenados en un archivo de Excel con el nombre estándar ANEXOS_AXXXX (XXXX corresponde al número del avance a elaborar).

Nota: este instructivo describe la lógica que se debe aplicar en la elaboración de los cuadros, no obstante, los scripts harán la mayor parte del proceso automáticamente. Los pasos en cursiva y subrayados indicarán aquellas acciones que debe ejecutar el funcionario a cargo.

4.3.2.1 Cuadro 1 - Comportamiento del comercio exterior de Colombia (Valor)

Contiene información histórica, acumulada a nivel mensual y agregado al periodo actual, del valor CIF y FOB de las importaciones y del valor FOB de las exportaciones (para las cifras del periodo sin

certificar, corresponde a los totales sin incluir petróleo). Para su construcción debe realizarse los siguientes pasos:

- Actualice los cuadros con los totales de las importaciones y exportaciones, de acuerdo con los últimos valores publicados y/o actualizados. En importaciones, si la cifra ya se encuentra como certificada, se debe poner "c" en la columna correspondiente, en caso contrario, se debe poner la letra "o", de observadas.

Figura 35 Ejemplo actualización totales - Importación

A	B	C	D	E	F
MES	CIF US\$	FOB US\$	PESO NETO	REGISTROS	certificado
1	5.529.665.230,76	5.174.272.035,65	3.384.293.231,71	273.092	c
2	5.058.145.403,66	4.757.580.543,03	2.946.086.299,25	266.081	c
3	5.856.797.852,20	5.547.248.236,46	3.264.239.931,42	294.790	c
4	4.942.584.913,10	4.661.857.293,99	3.322.148.159,16	261.128	c
5	5.418.513.280,35	5.130.399.359,97	3.266.003.929,03	290.436	c
6	5.015.211.672,38	4.752.059.285,70	2.824.169.858,23	284.161	c
7	4.945.200.918,60	4.694.962.525,50	2.765.710.704,20	278.629	c
8	5.288.179.351,77	5.008.159.977,77	3.558.686.579,97	310.715	c
9	4.944.207.348,91	4.702.379.941,64	2.870.150.074,70	296.972	c
10	5.343.857.091,78	5.083.183.782,85	3.235.365.102,37	0	

Fuente: U.A.E. DIAN. CETCE.

Para el caso de exportaciones, si la cifra aún no se ha certificado, se debe poner los totales (sin incluir petróleo) en las columnas F y G, según corresponda.

Figura 36 Ejemplo actualización totales - Exportación

A	B	C	D	E	F	G
FECHA	FOB US\$	PESO NETO KGM	REGISTROS	certificado	fob_sin_petroleo	pnk_sin_petroleo
202301	3.696.188.002,73	7.287.047.052,30	83.942	d		
202302	4.202.233.977,09	8.208.930.235,36	96.263	d		
202303	4.431.911.358,34	8.565.127.996,82	109.566	d		
202304	3.739.213.832,49	6.753.843.218,29	95.467	d		
202305	4.497.862.414,31	9.359.080.783,28	110.405	d		
202306	4.020.915.678,23	8.773.184.063,16	102.445	d		
202307	4.131.026.665,06	8.681.195.074,29	102.233	d		
202308	3.944.147.222,46	7.170.557.853,07	105.597	d		
202309	4.127.668.087,77	8.393.436.074,97	103.377	d		
202310	2.586.761.950,10	5.713.734.808,66	105434	o	3.012.222.747,40	5.173.954.093,07

Fuente: U.A.E. DIAN. CETCE.

- Ejecute el script para rellenado del cuadro 1, en el que se extraen y organizan los totales de los cuadros relacionados en el punto anterior; mientras que el acumulado al periodo en curso se toma de las bases procesadas en la sección 4.2. El código que ejecuta este paso en Python es el siguiente:

Figura 37 Script para llenado el cuadro 1

```
###Llenado de Cuadro 1#####
for indice, valor in enumerate(totales_impo_actual['CIF US$']):
    valor_anterior = totales_impo_anterior['CIF US$'][indice]
    archivo_anexo['Cuadro 1']['D' + str(indice + 9)].value = valor_anterior / 1000000
    archivo_anexo['Cuadro 1']['D' + str(indice + 11)].value = impo_anterior['Total Vr CIF'].sum()

for indice, valor in enumerate(totales_impo_actual['CIF US$']):
    archivo_anexo['Cuadro 1']['E'+str(indice+9)].value=valor/1000000
    certificado = totales_impo_actual['certificado'][indice]
    archivo_anexo['Cuadro 1']['J'+str(indice+9)].value=certificado

archivo_anexo['Cuadro 1']['E' + str(indice + 11)].value = impo_actual['Total Valor cif'].sum()
archivo_anexo['Cuadro 1']['J'+str(indice+11)].value='o'

for indice, valor in enumerate(totales_impo_actual['FOB US$']):
    valor_anterior = totales_impo_anterior['FOB US$'][indice]
    archivo_anexo['Cuadro 1']['G' + str(indice + 9)].value = valor_anterior / 1000000
    archivo_anexo['Cuadro 1']['G' + str(indice + 11)].value = impo_anterior['Total Valor Fob'].sum()

for indice, valor in enumerate(totales_impo_actual['FOB US$']):
    archivo_anexo['Cuadro 1']['H'+str(indice+9)].value=valor/1000000
    archivo_anexo['Cuadro 1']['H' + str(indice + 11)].value = impo_actual['Total Valor Fobi'].sum()

for indice, valor in enumerate(totales_expo_actual['FOB US$']):
    valor_anterior = totales_expo_anterior['FOB US$'][indice]
    archivo_anexo['Cuadro 1']['K' + str(indice + 9)].value = valor_anterior / 1000000
    archivo_anexo['Cuadro 1']['K' + str(indice + 11)].value = expo_anterior['Valor Fob Dolares'].sum()

for indice, valor in enumerate(totales_expo_actual['FOB US$']):
    archivo_anexo['Cuadro 1']['L'+str(indice+9)].value=valor/1000000
    certificado = totales_expo_actual['certificado'][indice]
    archivo.anexo['Cuadro 1']['N'+str(indice+9)].value=certificado
    valor_anterior = totales_expo_actual['fob_sin_petroleo'][indice]
    if certificado=='o':
        archivo_anexo['Cuadro 1']['K' + str(indice + 9)].value = valor_anterior / 1000000
    archivo_anexo['Cuadro 1']['L' + str(indice + 11)].value = expo_actual['Val Total Fob Us'].sum()
    archivo_anexo['Cuadro 1']['N'+str(indice+11)].value='o'
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.2 Cuadro 2 - Comportamiento del comercio exterior de Colombia (Peso neto)

Contiene información histórica, acumulada a nivel mensual y agregado al periodo actual, para el peso neto (en millones) de las importaciones y exportaciones (para las cifras del periodo sin certificar, corresponde a los totales sin incluir petróleo). Para su construcción debe realizarse los siguientes pasos:

1. Garantice que se haya realizado la actualización de los cuadros de totales, realizando el mismo paso 1 del numeral 4.3.2.1
2. Ejecute el script para relleno del cuadro 2, en el que se extraen y organizan los totales de los cuadros relacionados en el punto anterior; mientras que el acumulado del periodo en curso, se toma de las bases procesadas en la sección 4.2. El código que ejecuta este paso en Python es el siguiente:

Figura 38 Script para llenado el cuadro 2

```
###Llenado de Cuadro 2###
for indice, valor in enumerate(totales_impo_actual['PESO NETO']):
    valor_anterior = totales_impo_anterior['PESO NETO'][indice]
    archivo_anexo['Cuadro 2']['D' + str(indice + 10)].value = valor_anterior / 1000000
    archivo_anexo['Cuadro 2']['D' + str(indice + 12)].value = impo_anterior['Total Peso Neto'].sum()

for indice, valor in enumerate(totales_impo_actual['PESO NETO']):
    archivo_anexo['Cuadro 2']['E'+str(indice + 10)].value=valor/1000000
    archivo_anexo['Cuadro 2']['E' + str(indice + 12)].value = impo_actual['Total Peso Neto'].sum()

for indice, valor in enumerate(totales_expo_actual['PESO NETO KGM']):
    valor_anterior = totales_expo_anterior['PESO NETO KGM'][indice]
    archivo_anexo['Cuadro 2']['H' + str(indice + 10)].value = valor_anterior / 1000000
    archivo_anexo['Cuadro 2']['H' + str(indice + 12)].value = expo_anterior['Peso Neto Kilogramos'].sum()

for indice, valor in enumerate(totales_expo_actual['PESO NETO KGM']):
    archivo_anexo['Cuadro 2']['I'+str(indice+10)].value=valor/1000000
    certificado = totales_expo_actual['certificado'][indice]
    valor_anterior = totales_expo_actual['pnk_sin_petroleo'][indice]
    if certificado=='o':
        archivo_anexo['Cuadro 2']['H' + str(indice + 10)].value = valor_anterior / 1000000
    archivo_anexo['Cuadro 2']['I' + str(indice + 12)].value = expo_actual['Num Peso Neto'].sum()
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.3 Cuadro 3 - Precios implícitos del comercio exterior de Colombia

Contiene información histórica, acumulada a nivel mensual y agregado al periodo actual, para el precio implícito (Valor CIF y FOB con respecto al peso neto, para importaciones, y el valor FOB con respecto al peso neto para exportaciones. El cálculo se realiza de manera automática por la plantilla de Excel (Archivo ANEXOS), por tanto, no debe ejecutarse script alguno.

4.3.2.4 Cuadro 4 - Declaraciones de importación por Dirección Seccional

Contiene, tanto para el periodo de análisis como para el mismo periodo en el año anterior, los valores FOB, CIF, peso neto y número de declaraciones por Dirección Seccional. Para el periodo actual, se agrega a variable CIF en pesos. De igual manera, se calculan las variaciones entre el periodo actual y el mismo periodo en el año anterior.

Para su construcción debe realizarse los siguientes pasos:

1. Realice el *merge* de las bases con la tabla paramétrica de Aduanas.
2. Cuando el número de adunas para el periodo sea mayor al número de aduanas que tiene la plantilla ANEXOS A###, deberá ajustar el número de filas del cuadro, según lo indique la ventana de resultados en la ejecución del script en Spyder.
3. Genere los valores agregados, por dirección seccional, de valor FOB, CIF, peso neto y conteo en la cantidad de registros. Lo anterior para el periodo actual y el mismo periodo del año anterior. En caso del periodo de análisis, incluya la sumatoria de la variable CIF pesos.
4. Organice la tabla de manera descendente por el valor CIF para el año actual.

5. Ejecute el script para llenado del cuadro 4, en el que se calculan los totales y se organizan en el formato definido. El código que ejecuta este paso en Python es el siguiente:

Figura 39 Script para llenado el cuadro 4

```
###Llenado de Cuadro 4###
seccionales_impo_actual=impo_actual['Codigo Admón Aduana'].unique()
seccionales_impo_anterior=impo_anterior['Cod Administración Presentada'].unique()
seccionales_cuadros_impo = list(set(seccionales_impo_actual) | set(seccionales_impo_anterior))

if archivo_anexo['Cuadro 4'][‘A9’].value<len(seccionales_cuadros_impo):
    print("Aduanas impo actual: " + str(impo_actual['Codigo Admón Aduana'].nunique()))
    print("Aduanas impo anterior: " + str(impo_anterior['Cod Administración Presentada'].nunique()))
    print("Ajuste el Cuadro 4 en el archivo Anexos antes de continuar")
    sys.exit()

cuadro4 = pd.DataFrame({'Seccional': seccionales_cuadros_impo}, columns=['Seccional'])

c4_1 = impo_anterior.groupby('Cod Administración Presentada')[['Total Valor Fob', 'Total Vr CIF', 'Total Peso Neto']].sum().reset_index()
c4_4 = impo_anterior.groupby('Cod Administración Presentada')[['Fecha Presentacion Documento']].count().reset_index()
c4_5 = impo_anterior.groupby('Codigo Admón Aduana')[‘Total Valor Fob’].sum().reset_index()
c4_5.rename(columns={‘Codigo Admón Aduana’: ‘Cod Administración Presentada’}, inplace=True)
c4_6 = impo_anterior.groupby('Codigo Admón Aduana')[‘Total Valor cif’].sum().reset_index()
c4_6.rename(columns={‘Codigo Admón Aduana’: ‘Cod Administración Presentada’}, inplace=True)
c4_7 = impo_anterior.groupby('Codigo Admón Aduana')[‘Total Peso Neto’].sum().reset_index()
c4_7.rename(columns={‘Codigo Admón Aduana’: ‘Cod Administración Presentada’}, inplace=True)
c4_8 = impo_anterior.groupby('Codigo Admón Aduana')[‘Fecha Presentacion’].count().reset_index()
c4_8.rename(columns={‘Codigo Admón Aduana’: ‘Cod Administración Presentada’}, inplace=True)
c4_9 = impo_anterior.groupby('Codigo Admón Aduana')[‘Total Valor Cif Pesos’].sum().reset_index()
c4_9.rename(columns={‘Codigo Admón Aduana’: ‘Cod Administración Presentada’}, inplace=True)

cuadro4 = cuadro4.merge(c4_1, left_on='Seccional', right_on='Cod Administración Presentada', how='left')
cuadro4 .drop('Cod Administración Presentada', axis=1, inplace=True)
cuadro4 = cuadro4.merge(c4_4, left_on='Seccional', right_on='Cod Administración Presentada', how='left')
cuadro4 .drop('Cod Administración Presentada', axis=1, inplace=True)
cuadro4.rename(columns={‘Fecha Presentacion Documento’: ‘Declaraciones’}, inplace=True)
cuadro4 = cuadro4.merge(c4_5, left_on='Seccional', right_on='Cod Administración Presentada', how='left')
cuadro4 .drop('Cod Administración Presentada', axis=1, inplace=True)
cuadro4 = cuadro4.merge(c4_6, left_on='Seccional', right_on='Cod Administración Presentada', how='left')
cuadro4 .drop('Cod Administración Presentada', axis=1, inplace=True)
cuadro4 = cuadro4.merge(c4_7, left_on='Seccional', right_on='Cod Administración Presentada', how='left')
cuadro4 .drop('Cod Administración Presentada', axis=1, inplace=True)
cuadro4 = cuadro4.merge(c4_8, left_on='Seccional', right_on='Cod Administración Presentada', how='left')
cuadro4 .drop('Cod Administración Presentada', axis=1, inplace=True)
cuadro4.rename(columns={‘Fecha Presentacion’: ‘DeclaracionesActual’}, inplace=True)
cuadro4 = cuadro4.merge(c4_9, left_on='Seccional', right_on='Cod Administración Presentada', how='left')
cuadro4 .drop('Cod Administración Presentada', axis=1, inplace=True)
cuadro4 = cuadro4.merge(aduanas, left_on=‘Seccional’, right_on='Cod Administración Presentada', how='left')

cuadro4= cuadro4.sort_values(by='Total Valor cif', ascending=False)
for indice, valor in enumerate(cuadro4[‘Seccional’]):
    archivo_anexo[‘Cuadro 4’][‘C’ + str(indice + 9)].value = cuadro4[‘NOMBRE ADUANA’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘D’ + str(indice + 9)].value = cuadro4[‘Total Valor Fob’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘E’ + str(indice + 9)].value = cuadro4[‘Total Vr CIF’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘F’ + str(indice + 9)].value = cuadro4[‘Total Peso Neto’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘G’ + str(indice + 9)].value = cuadro4[‘Declaraciones’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘H’ + str(indice + 9)].value = cuadro4[‘Total Valor Fob’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘I’ + str(indice + 9)].value = cuadro4[‘Total Valor cif’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘J’ + str(indice + 9)].value = cuadro4[‘Total Peso Neto’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘K’ + str(indice + 9)].value = cuadro4[‘DeclaracionesActual’].iloc[indice]
    archivo_anexo[‘Cuadro 4’][‘L’ + str(indice + 9)].value = cuadro4[‘Total Valor Cif Pesos’].iloc[indice]
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.5 Cuadro 5 - Principales subpartidas importadas

Contiene, para el periodo de análisis, los valores FOB y peso neto de las 20 principales subpartidas importadas

Para su construcción debe realizarse los siguientes pasos:

- Realice el merge de las bases con la tabla paramétrica del Totpar.

2. Genere los valores agregados por subpartida, del valor FOB y peso neto.
3. Organice la tabla de manera descendente por el valor FOB.
4. Seleccione las primeras 20 subpartidas con mayor valor FOB.
5. Ejecute el script para llenado del cuadro 5, en el que se calculan los totales y se organizan en el formato definido. El código que ejecuta este paso en Python es el siguiente:

Figura 40 Script para llenado el cuadro 5

```
###Llenado de Cuadro 5###
cuadro5 = impo_actual.groupby('Subpartida Arancelaria1')[['Total Valor Fob1', 'Total Peso Netol']].sum().reset_index()
cuadro5 = cuadro5.sort_values(by='Total Valor Fob1', ascending=False)
cuadro5 = cuadro5[0:20]
cuadro5 = cuadro5.merge(totpar, left_on='Subpartida Arancelaria1', right_on='Subpartida Arancelaria', how='left')

for indice, valor in enumerate(cuadro5['Subpartida Arancelaria1']):
    archivo_anexo['Cuadro 5']['C' + str(indice + 8)].value = cuadro5['Subpartida Arancelaria1'].iloc[indice]
    archivo_anexo['Cuadro 5']['D' + str(indice + 8)].value = cuadro5['Descripción Arancelaria'].iloc[indice]
    archivo_anexo['Cuadro 5']['E' + str(indice + 8)].value = cuadro5['Total Valor Fob1'].iloc[indice]
    archivo_anexo['Cuadro 5']['F' + str(indice + 8)].value = cuadro5['Total Peso Netol'].iloc[indice]
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.6 Cuadro 6 - Principales países de origen de las importaciones colombianas

Contiene, para el periodo de análisis, los valores CIF, FOB y peso neto de las importaciones provenientes de los cinco principales socios comerciales. De igual manera, se presentan las participaciones de estos cinco países en términos de las 3 variables analizadas.

Para su construcción debe realizarse los siguientes pasos:

1. Realice el merge de las bases con la tabla paramétrica países.
2. Genere los valores agregados por país, del valor CIF, FOB y peso neto.
3. Organice la tabla de manera descendente por el valor CIF.
4. Seleccione los primeros 5 países con mayor valor CIF agregado.
5. Ejecute el script para llenado del cuadro 6, en el que se calculan los totales y se organizan en el formato definido. El código que ejecuta este paso en Python es el siguiente:

Figura 41 Script para llenado el cuadro 6

```
###Llenado de Cuadro 6###
cuadro6 = impo_actual.groupby('Codigo País Origen1')[['Total Valor cif', 'Total Valor Fob1', 'Total Peso Netol']].sum().reset_index()
cuadro6 = cuadro6.sort_values(by='Total Valor cif', ascending=False)
cuadro6 = cuadro6[0:5]
cuadro6 = cuadro6.merge(paises, left_on='Codigo País Origen1', right_on='COD_PAIS', how='left')

for indice, valor in enumerate(cuadro6['Codigo País Origen1']):
    archivo_anexo['Cuadro 6']['B' + str(indice + 10)].value = cuadro6['DESCRIPCION'].iloc[indice]
    archivo_anexo['Cuadro 6']['C' + str(indice + 10)].value = cuadro6['Total Valor cif'].iloc[indice]
    archivo_anexo['Cuadro 6']['E' + str(indice + 10)].value = cuadro6['Total Valor Fob1'].iloc[indice]
    archivo_anexo['Cuadro 6']['G' + str(indice + 10)].value = cuadro6['Total Peso Netol'].iloc[indice]
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.7 Cuadro 7 - Importaciones según clasificación CUODE

Contiene, para el periodo de análisis y el mismo periodo del año anterior, los valores CIF y las participaciones de las importaciones, segmentadas por la clasificación CUODE a nivel de 2 dígitos.

Para su construcción debe realizarse los siguientes pasos:

1. Realice el merge de las bases con la tabla paramétrica Totpar.
2. Genere los valores agregados por CUODE 2, del valor CIF.
3. Ejecute el script para llenado del cuadro 7, en el que se calculan los totales y se organizan en el formato definido. El código que ejecuta este paso en Python es el siguiente:

Figura 42 Script para llenado el cuadro 7

```
###llenado de Cuadro 7###
impo_actual= impo_actual.merge(totpar, left_on='Subpartida Arancelaria1', right_on='Subpartida Arancelaria1', how='left')
impo_anterior= impo_anterior.merge(totpar, left_on='Subpartida Arancelaria1', right_on='Subpartida Arancelaria1', how='left')
cuadro7= cuodel(cuodel['CUODE'] > 9) | (cuodel['CUODE'] == 0)
cuadro7= cuadro7.drop_duplicates(subset=['CUODE'])
cuadro7.loc[cuadro7['CUODE'] == 0, 'CUODE'] = 10
c7_1 = impo_actual.groupby('CUODE 2')[['Total Valor cif']].sum().reset_index()
c7_2 = impo_anterior.groupby('CUODE 2')[['Total Vr CIF']].sum().reset_index()
c8_1 = impo_actual.groupby('CUODE 2')[['Total Subtotal Aranceli']].sum().reset_index()
c8_2 = impo_anterior.groupby('CUODE 2')[['Total Subtotal Ival']].sum().reset_index()
cuadro7 = cuadro7.merge(c7_1, left_on='CUODE', right_on='CUODE 2', how='left')
cuadro7 .drop(['CUODE 2', axis=1, inplace=True]
cuadro7 = cuadro7.merge(c7_2, left_on='CUODE', right_on='CUODE 2', how='left')
cuadro7 .drop(['CUODE 2', axis=1, inplace=True]
cuadro7 = cuadro7.merge(c8_1, left_on='CUODE', right_on='CUODE 2', how='left')
cuadro7 .drop(['CUODE 2', axis=1, inplace=True]
cuadro7 = cuadro7.merge(c8_2, left_on='CUODE', right_on='CUODE 2', how='left')
cuadro7 .drop(['CUODE 2', axis=1, inplace=True]
cuadro7= cuadro7.sort_values(by='CUODE', ascending=True)
cuadro7 = cuadro7.fillna(0)

for i in range(0,6):
    archivo_anexol['Cuadro 7'][['D' + str(i + 10)]].value = cuadro7['Total Vr CIF'].iloc[i+1]
    archivo_anexol['Cuadro 7'][['D' + str(i + 17)]].value = cuadro7['Total Vr CIF'].iloc[i+7]
    archivo_anexol['Cuadro 7'][['E' + str(i + 10)]].value = cuadro7['Total Valor cif'].iloc[i+1]
    archivo_anexol['Cuadro 7'][['E' + str(i + 17)]].value = cuadro7['Total Valor cif'].iloc[i+7]
for i in range(0,5):
    archivo_anexol['Cuadro 7'][['D' + str(i + 43)]].value = cuadro7['Total Vr CIF'].iloc[i+26]
    archivo_anexol['Cuadro 7'][['E' + str(i + 43)]].value = cuadro7['Total Valor cif'].iloc[i+26]
for i in range(0,4):
    archivo_anexol['Cuadro 7'][['D' + str(i + 32)]].value = cuadro7['Total Vr CIF'].iloc[i+18]
    archivo_anexol['Cuadro 7'][['E' + str(i + 32)]].value = cuadro7['Total Valor cif'].iloc[i+18]
for i in range(0,3):
    archivo_anexol['Cuadro 7'][['D' + str(i + 25)]].value = cuadro7['Total Vr CIF'].iloc[i+13]
    archivo_anexol['Cuadro 7'][['D' + str(i + 39)]].value = cuadro7['Total Vr CIF'].iloc[i+23]
    archivo_anexol['Cuadro 7'][['D' + str(i + 49)]].value = cuadro7['Total Vr CIF'].iloc[i+31]
    archivo_anexol['Cuadro 7'][['E' + str(i + 25)]].value = cuadro7['Total Valor cif'].iloc[i+13]
    archivo_anexol['Cuadro 7'][['E' + str(i + 39)]].value = cuadro7['Total Valor cif'].iloc[i+23]
    archivo_anexol['Cuadro 7'][['E' + str(i + 49)]].value = cuadro7['Total Valor cif'].iloc[i+31]
for i in range(0,2):
    archivo_anexol['Cuadro 7'][['D' + str(i + 29)]].value = cuadro7['Total Vr CIF'].iloc[i+16]
    archivo_anexol['Cuadro 7'][['E' + str(i + 29)]].value = cuadro7['Total Valor cif'].iloc[i+16]

archivo_anexo['Cuadro 7'][['D37']].value = cuadro7['Total Vr CIF'].iloc[22]
archivo_anexo['Cuadro 7'][['D52']].value = cuadro7['Total Vr CIF'].iloc[0]
archivo_anexo['Cuadro 7'][['E37']].value = cuadro7['Total Valor cif'].iloc[22]
archivo_anexo['Cuadro 7'][['E52']].value = cuadro7['Total Valor cif'].iloc[0]
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.8 Cuadro 8 - Recaudo externo autoliquidado según clasificación CUODE

Contiene, para el periodo de análisis, el valor del arancel e IVA autoliquidado en el formulario de importaciones, agregado a nivel de 2 dígitos de la clasificación CUODE.

Para su construcción debe realizarse los siguientes pasos:

1. Realice el *merge* de las bases con la tabla paramétrica Totpar.
2. Genere los valores agregados por CUODE2, del valor del arancel e IVA autoliquidado
3. Ejecute el script para llenado del cuadro 8, en el que se calculan los totales y se organizan en el formato definido. El código que ejecuta este paso en Python es el siguiente:

Figura 43 Script para llenado el cuadro 8

```
###Llenado de Cuadro 8###

for i in range(0,6):
    archivo_anexo['Cuadro 8']['D' + str(i + 9)].value = cuadro7['Total Subtotal Arancell'].iloc[i+1]
    archivo_anexo['Cuadro 8']['D' + str(i + 16)].value = cuadro7['Total Subtotal Arancell'].iloc[i+7]
    archivo_anexo['Cuadro 8']['E' + str(i + 9)].value = cuadro7['Total Subtotal Ival'].iloc[i+1]
    archivo_anexo['Cuadro 8']['E' + str(i + 16)].value = cuadro7['Total Subtotal Ival'].iloc[i+7]
for i in range(0,5):
    archivo_anexo['Cuadro 8']['D' + str(i + 42)].value = cuadro7['Total Subtotal Arancell'].iloc[i+26]
    archivo_anexo['Cuadro 8']['E' + str(i + 42)].value = cuadro7['Total Subtotal Ival'].iloc[i+26]
for i in range(0,4):
    archivo_anexo['Cuadro 8']['D' + str(i + 31)].value = cuadro7['Total Subtotal Arancell'].iloc[i+18]
    archivo_anexo['Cuadro 8']['E' + str(i + 31)].value = cuadro7['Total Subtotal Ival'].iloc[i+18]
for i in range(0,3):
    archivo_anexo['Cuadro 8']['D' + str(i + 24)].value = cuadro7['Total Subtotal Arancell'].iloc[i+13]
    archivo_anexo['Cuadro 8']['D' + str(i + 38)].value = cuadro7['Total Subtotal Arancell'].iloc[i+23]
    archivo_anexo['Cuadro 8']['D' + str(i + 48)].value = cuadro7['Total Subtotal Arancell'].iloc[i+31]
    archivo_anexo['Cuadro 8']['E' + str(i + 24)].value = cuadro7['Total Subtotal Ival'].iloc[i+13]
    archivo_anexo['Cuadro 8']['E' + str(i + 38)].value = cuadro7['Total Subtotal Ival'].iloc[i+23]
    archivo_anexo['Cuadro 8']['E' + str(i + 48)].value = cuadro7['Total Subtotal Ival'].iloc[i+31]
for i in range(0,2):
    archivo_anexo['Cuadro 8']['D' + str(i + 28)].value = cuadro7['Total Subtotal Arancell'].iloc[i+16]
    archivo_anexo['Cuadro 8']['E' + str(i + 28)].value = cuadro7['Total Subtotal Ival'].iloc[i+16]

archivo_anexo['Cuadro 8']['D36'].value = cuadro7['Total Subtotal Arancell'].iloc[22]
archivo_anexo['Cuadro 8']['D51'].value = cuadro7['Total Subtotal Arancell'].iloc[0]
archivo_anexo['Cuadro 8']['E36'].value = cuadro7['Total Subtotal Ival'].iloc[22]
archivo_anexo['Cuadro 8']['E51'].value = cuadro7['Total Subtotal Ival'].iloc[0]
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.9 Cuadro 9 - Exportaciones tramitadas en las Direcciones Seccionales con declaración electrónica (se excluye petróleo)

Contiene, tanto para el periodo de análisis como para el mismo periodo en el año anterior, los valores FOB, peso neto, número de registros y número de declaraciones por Dirección Seccional. De igual manera, se calculan las variaciones entre el periodo actual y el mismo periodo en el año anterior.

Para su construcción debe realizarse los siguientes pasos:

1. Realice el merge de las bases con la tabla paramétrica de Aduanas.
2. Genere los valores agregados, por dirección seccional, de valor FOB, peso neto y conteo en la cantidad de registros y declaraciones. Lo anterior para el periodo actual y el mismo periodo del año anterior. Tenga presente que el número de declaraciones corresponde al conteo de valores únicos de los IDE de la declaración, mientras que la cantidad de registros corresponde al recuento de filas que contiene la base.
3. Organice la tabla de manera descendente por el valor FOB para el año actual.
4. Ejecute el script para relleno del cuadro 9, en el que se calculan los totales y se organizan en el formato definido. El código que ejecuta este paso en Python es el siguiente:

Figura 44 Script para llenado el cuadro 9

```
###Llenado de Cuadro 9###
seccionales_expo_actual=expo_actual[['Cod Administracion Pres']].unique()
seccionales_expo_anterior=expo_anterior[['Codigo De Aduana']].unique()
seccionales_cuadros_expo = list(set(seccionales_expo_actual) | set(seccionales_expo_anterior))

if archivo_anexo['Cuadro 9'][['D12']].value<len(seccionales_cuadros_expo):
    print("Aduanas expo actual: " + str(expo_actual[['Cod Administracion Pres']].nunique()))
    print("Aduanas expo anterior: " + str(expo_anterior[['Codigo De Aduana']].nunique()))
    print("Aduanas para cuadro 4 : " + str(len(seccionales_cuadros_expo)))
    print("Ajuste el Cuadro 9 en el archivo Anexos antes de continuar")
    sys.exit()

cuadro9 = pd.DataFrame({'Seccional': seccionales_cuadros_expo}, columns=['Seccional'])

c9_1 = expo_anterior.groupby('Codigo De Aduana')[['Valor Fob Dolares', 'Peso Neto Kilogramos']].sum().reset_index()
c9_2 = expo_anterior.groupby('Codigo De Aduana')[['Numero De La Declaracion']].count().reset_index()
c9_2.rename(columns={'Numero De La Declaracion': 'Registros'}, inplace=True)
c9_3 = expo_anterior.groupby('Codigo De Aduana')[['Numero De La Declaracion']].nunique().reset_index()
c9_3.rename(columns={'Numero De La Declaracion': 'Declaraciones'}, inplace=True)

c9_4 = expo_actual.groupby('Cod Administracion Pres')[['Val Total Fob Us', 'Num Peso Neto']].sum().reset_index()
c9_5 = expo_actual.groupby('Cod Administracion Pres')[['Ide Declaracion Exportacion']].count().reset_index()
c9_5.rename(columns={'Ide Declaracion Exportacion': 'Registros'}, inplace=True)
c9_6 = expo_actual.groupby('Cod Administracion Pres')[['Ide Declaracion Exportacion']].nunique().reset_index()
c9_6.rename(columns={'Ide Declaracion Exportacion': 'Declaraciones'}, inplace=True)

cuadro9 = cuadro9.merge(c9_1, left_on='Seccional', right_on='Codigo De Aduana', how='left')
cuadro9 .drop('Codigo De Aduana' , axis=1, inplace=True)
cuadro9 = cuadro9.merge(c9_2, left_on='Seccional', right_on='Codigo De Aduana', how='left')
cuadro9 .drop('Codigo De Aduana' , axis=1, inplace=True)
cuadro9 = cuadro9.merge(c9_3, left_on='Seccional', right_on='Codigo De Aduana', how='left')
cuadro9 .drop('Codigo De Aduana' , axis=1, inplace=True)
cuadro9 = cuadro9.merge(c9_4, left_on='Seccional', right_on='Cod Administracion Pres', how='left')
cuadro9 .drop('Cod Administracion Pres' , axis=1, inplace=True)
cuadro9 = cuadro9.merge(c9_5, left_on='Seccional', right_on='Cod Administracion Pres', how='left')
cuadro9 .drop('Cod Administracion Pres' , axis=1, inplace=True)
cuadro9 = cuadro9.merge(c9_6, left_on='Seccional', right_on='Cod Administracion Pres', how='left')
cuadro9 .drop('Cod Administracion Pres' , axis=1, inplace=True)
cuadro9 = cuadro9.merge(aduanas, left_on='Seccional', right_on='Cod Administracion Presentada', how='left')

cuadro9= cuadro9.sort_values(by='Val Total Fob Us', ascending=False)
for indice, valor in enumerate(cuadro9['Seccional']):
    archivo_anexo['Cuadro 9'][['E' + str(indice + 12)]].value = cuadro9['NOMBRE_ADUANA'].iloc[indice]
    archivo_anexo[['Cuadro 9'][['F' + str(indice + 12)]]].value = cuadro9['Valor Fob Dolares'].iloc[indice]
    archivo_anexo[['Cuadro 9'][['G' + str(indice + 12)]]].value = cuadro9['Peso Neto Kilogramos'].iloc[indice]
    archivo_anexo[['Cuadro 9'][['H' + str(indice + 12)]]].value = cuadro9['Registros_x'].iloc[indice]
    archivo_anexo[['Cuadro 9'][['I' + str(indice + 12)]]].value = cuadro9['Declaraciones_x'].iloc[indice]
    archivo_anexo[['Cuadro 9'][['J' + str(indice + 12)]]].value = cuadro9['Val Total Fob Us'].iloc[indice]
    archivo_anexo[['Cuadro 9'][['K' + str(indice + 12)]]].value = cuadro9['Num Peso Neto'].iloc[indice]
    archivo_anexo[['Cuadro 9'][['L' + str(indice + 12)]]].value = cuadro9['Registros_y'].iloc[indice]
    archivo_anexo[['Cuadro 9'][['M' + str(indice + 12)]]].value = cuadro9['Declaraciones_y'].iloc[indice]
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.10 Cuadro 10 - Principales destinos de las exportaciones colombianas (se excluye petróleo)

Contiene, para el periodo de análisis, el valor FOB y el peso neto de las exportaciones hacia los cinco principales socios comerciales. De igual manera, se presentan las participaciones de estos países en términos de las 2 variables analizadas.

Para su construcción debe realizarse los siguientes pasos:

1. Realice el *merge* de las bases con la tabla paramétrica países.
2. Genere los valores agregados por país, del valor FOB y peso neto.
3. Organice la tabla de manera descendente por el valor FOB.
4. Seleccione los primeros 5 países con mayor valor FOB agregado.
5. Ejecute el script para llenado del cuadro 10, en el que se calculan los totales y se organizan en el formato definido. El código que ejecuta este paso en Python es el siguiente:

Figura 45 Script para llenado el cuadro 10

```
###Llenado de Cuadro 10###
cuadro10 = expo_actual.groupby('Cod País Destino')[['Val Total Fob Us', 'Num Peso Neto']].sum().reset_index()
cuadro10 = cuadro10.sort_values(by='Val Total Fob Us', ascending=False)
cuadro10 = cuadro10[0:5]
cuadro10 = cuadro10.merge(paises, left_on='Cod País Destino', right_on='COD_PAIS', how='left')

for indice, valor in enumerate(cuadro10['Cod País Destino']):
    archivo_anexo['Cuadro 10'][['E' + str(indice + 10)]].value = cuadro10['DESCRIPCION'].iloc[indice]
    archivo_anexo['Cuadro 10'][['F' + str(indice + 10)]].value = cuadro10['Val Total Fob Us'].iloc[indice]
    archivo_anexo['Cuadro 10'][['H' + str(indice + 10)]].value = cuadro10['Num Peso Neto'].iloc[indice]

###Llenado de Cuadro 11###
cuadroll = expo_actual.groupby('Cod Colombia')[['Val Total Fob Us', 'Num Peso Neto']].sum().reset_index()
cuadroll = cuadroll.sort_values(by='Val Total Fob Us', ascending=False)
cuadroll = cuadroll[0:20]
cuadroll = cuadroll.merge(totpar, left_on='Cod Colombia', right_on='Subpartida Arancelaria', how='left')

for indice, valor in enumerate(cuadroll['Cod Colombia']):
    archivo_anexo['Cuadro 11'][['E' + str(indice + 9)]].value = cuadroll['Cod Colombia'].iloc[indice]
    archivo_anexo['Cuadro 11'][['F' + str(indice + 9)]].value = cuadroll['Descripción Arancelaria'].iloc[indice]
    archivo_anexo['Cuadro 11'][['G' + str(indice + 9)]].value = cuadroll['Val Total Fob Us'].iloc[indice]
    archivo_anexo['Cuadro 11'][['H' + str(indice + 9)]].value = cuadroll['Num Peso Neto'].iloc[indice]
```

Fuente: U.A.E. DIAN. CETCE.

4.3.2.11 Cuadro 11 - Principales subpartidas exportadas (se excluye petróleo)

Contiene, para el periodo de análisis, los valores FOB y peso neto de las 20 principales subpartidas exportadas.

Para su construcción debe realizarse los siguientes pasos:

1. Realice el *merge* de las bases con la tabla paramétrica del Totpar.
2. Genere los valores agregados por subpartida, del valor FOB y peso neto.
3. Organice la tabla de manera descendente por el valor FOB.
4. Seleccione las primeras 20 subpartidas con mayor valor FOB.
5. Ejecute el script para llenado del cuadro 11, en el que se calculan los totales y se organizan en el formato definido. El código que ejecuta este paso en Python es el siguiente:

Figura 46 Script para llenado el cuadro 11

```
###Llenado de Cuadro 11###
cuadroll = expo_actual.groupby('Cod Colombia')[['Val Total Fob Us', 'Num Peso Neto']].sum().reset_index()
cuadroll = cuadroll.sort_values(by='Val Total Fob Us', ascending=False)
cuadroll = cuadroll[0:20]
cuadroll = cuadroll.merge(totpar, left_on='Cod Colombia', right_on='Subpartida Arancelaria', how='left')

for indice, valor in enumerate(cuadroll['Cod Colombia']):
    archivo_anexo['Cuadro 11'][['E' + str(indice + 9)]].value = cuadroll['Cod Colombia'].iloc[indice]
    archivo_anexo['Cuadro 11'][['F' + str(indice + 9)]].value = cuadroll['Descripción Arancelaria'].iloc[indice]
    archivo_anexo['Cuadro 11'][['G' + str(indice + 9)]].value = cuadroll['Val Total Fob Us'].iloc[indice]
    archivo_anexo['Cuadro 11'][['H' + str(indice + 9)]].value = cuadroll['Num Peso Neto'].iloc[indice]
```

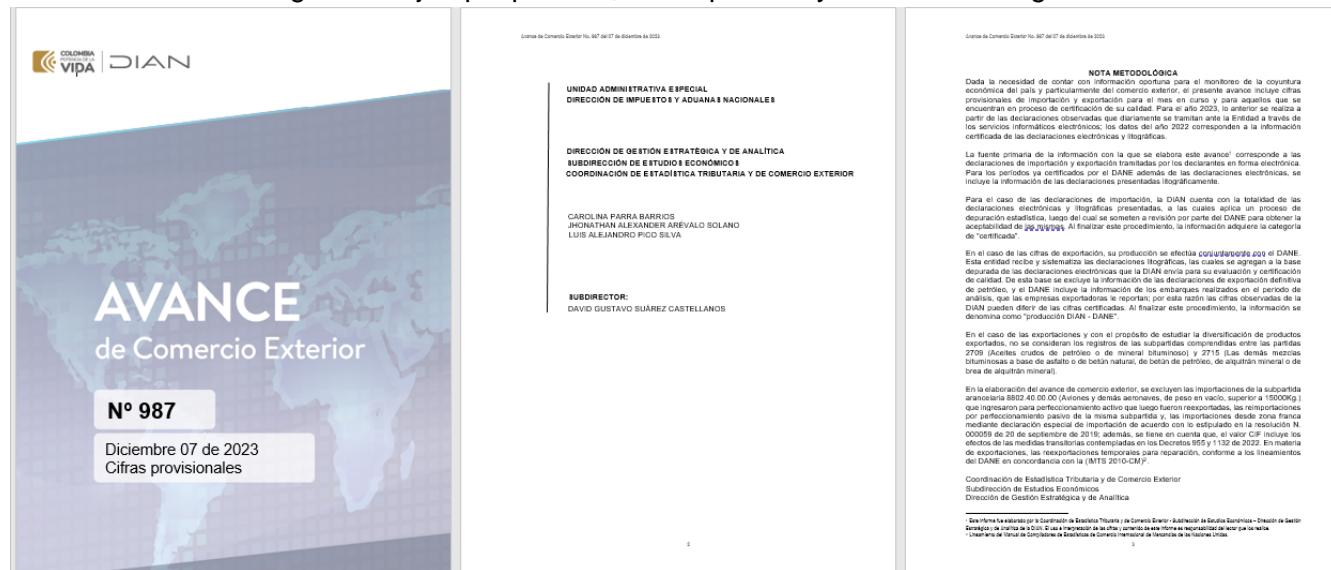
Fuente: U.A.E. DIAN. CETCE.

4.3.3 Elaboración del Informe

Una vez preparados los cuadros antes señalados, se procede a elaborar el informe en la plantilla Word definida y que se ubica en la carpeta Plantillas del OneDrive destinado para el proceso de elaboración de avances.

Proceda a pegar los cuadros en el documento, en el orden que allí se relaciona. De igual manera, ajustes los textos de análisis según corresponda al comportamiento de los datos.

Figura 47 Ejemplo portada, contraportada y ficha metodológica



Fuente: U.A.E. DIAN. CETCE.

4.3.4 Publicación

Cuando se cuente con el documento elaborado (numeral 4.3.3), se procede a solicitar la publicación en la sección de estadísticas de la página web de la DIAN.

Para lo anterior, se diligencia el formato FT-PEC-1463 (Solicitud de Publicación, Actualización o Eliminación en el Portal WEB de la Entidad) que se encuentra en el Listado Maestro de Documentos de la Diannet y, por medio del buzón de Estadísticas (estadisticas@dian.gov.co), se envía este formato diligenciado junto con el informe al correo electrónico publicacionesweb@dian.gov.co. En el mensaje de este envío se deben especificar la dirección de la página web donde se deben publicar, la cual es: <https://www.dian.gov.co/dian/cifras/Paginas/EstadisticasComEx.aspx>

Finalmente, con posterioridad a la confirmación de la publicación por parte del Publicacionesweb, se verifica que el Avance haya quedado publicado en el lugar indicado de la página web. En caso afirmativo, se procede a diligenciar la encuesta.

5. CONTROL DE CAMBIOS

Versión	Vigencia		Descripción de los cambios	Tipo de información
	Desde	Hasta		
1	29/05/2024		Versión inicial	Esta versión corresponde a Información Pública

Elaboró:	Luis Alejandro Pico Silva Luisa Fernanda Suárez León	Gestor III	Coordinación de Estadística Tributaria y de Comercio Exterior
	Elaboración técnica Néstor Raúl Sánchez Mateus	Gestor III	Coordinación de Procesos y Riesgos Operacionales
	Elaboración metodológica		
Revisó:	Alba Clemencia Avendaño Cruz	Jefe Coordinación	Coordinación de Estadística Tributaria y de Comercio Exterior
Aprobó:	David Gustavo Suárez Castellanos	Subdirector (A)	Subdirección de Estudios Económicos