1. IDENTIFICACIÓN DEL INSUMO

Fecha	Septiembre 2023
Ciudad	Bogotá D.C.
Esquema de presentación del insumo	Cuaderno Jupyter
Título del insumo	**Descarga de datos y unificación de fuentes**
Descripción y alcance	Notebook para la descarga y unificación de los datos que alimentan a la base unificada.
Periodicidad del insumo	único
Solicitante	No aplica
Versión del insumo	Final

2. DESTINO Y AUTORES DEL INFORME / INSUMO

Destinatario	*Nombre:* Equipo analítica EMAE *Cargo:* NA *Área:* Subdirección de estudios de Mercado y Abastecimiento Estratégico – EMAE
Autores	*Nombre:* Equipo de Datos - GAEC*Área:* Subdirección de estudios de Mercado y Abastecimiento Estratégico – EMAE.
Aprobación	*Nombre:* Maria del Pilar Suarez Sebastian *Cargo:* Subdirectora Estudios de Mercado y Abastecimiento Estratégico *Área:* Subdirección de estudios de Mercado y Abastecimiento Estratégico – EMAE.

Introducción

En el presente documento se muestra el proceso de descarga y unificación de las bases de datos del SECOP I y del SECOP II desde las fuentes "Reports" y "Sandbox" respectivamente. Tiendo como objetivo, proporcionar una guía detallada y clara sobre el proceso llevado a cabo para adquirir y consolidar la información de ambas plataformas, abarcando el período desde el año 2020 hasta el 30 de junio del 2023.

El Sistema Electrónico de Contratación Pública, conocido como SECOP, es una herramienta fundamental para el registro y seguimiento de los procesos de contratación en entidades estatales. Consta de dos versiones distintas: SECOP I y SECOP II. Cada versión maneja información relevante para la gestión de contratos públicos, y la unificación de ambas bases de datos permitirá obtener una visión más completa y detallada de los procesos de contratación en el país.

Este cuadernillo servirá como una guía paso a paso para comprender el proceso de descarga y unificación de los datos, detallando cada etapa desde la obtención de la información desde las fuentes originales hasta su integración en una base de datos consolidada. Se presentarán los

procedimientos técnicos utilizados, así como las herramientas y tecnologías empleadas para garantizar la precisión y la integridad de los datos.

Cabe destacar que el período de descarga y unificación abarca desde el 1 de enero del 2020 hasta el 30 de junio del 2023, lo que nos permitirá obtener una perspectiva histórica amplia y actualizada sobre los procesos de contratación durante este periodo.

Es importante señalar que el proceso de unificación de las bases de datos busca facilitar el análisis y la toma de decisiones informadas para las entidades estatales, los investigadores, analistas y el público en general, contribuyendo así a una mayor transparencia en el manejo de los recursos públicos y fomentando una gestión más eficiente y responsable.

Esperamos que este cuadernillo sea de gran utilidad para todos aquellos involucrados en el análisis de datos del SECOP y que contribuya a un mejor entendimiento de los procesos de contratación pública en nuestro país. Agradecemos su interés y compromiso en este importante proyecto, y estamos seguros de que juntos lograremos un proceso de descarga y unificación exitoso y enriquecedor.

```
# Decarga de librarías
import pandas as pd # Tratamiento de datos
from libreria import login # Lógica para el inicio de sesión a las
fuentes de datos

# Cargue de las conexiones a las fuentes de datos
cnxn_sandbox, cursor_sandbox =
login.login_sql(login.tipo_bodega.SANDBOX)
cnxn_reports, cursor_reports =
login.login_sql(login.tipo_bodega.REPORTS)

# Cambio de formato
pd.options.display.float_format = '{:.0f}'.format
```

Descarga de Información desde las Fuentes Originales (SECOP I - SECOP II)

En esta sección, detallaremos el código utilizado para realizar la descarga directa de la información desde las fuentes originales del SECOP I y SECOP II, específicamente desde las plataformas "Reports" y "Sandbox", respectivamente. El proceso de obtención de datos es esencial para garantizar la precisión y actualización de la información que se utilizará en la unificación posterior de las bases de datos. A continuación, se presenta el código utilizado para cada plataforma:

```
# SECOP I
DF_SECOPI_contratos_2023 = pd.read_sql_query("select * from
[CCE_Sandbox].[SECOPI].[V_SECOPI] where YEAR(FECHA_FIRMA_CONTRATO) =
2023 and MONTH(FECHA_FIRMA_CONTRATO) = 1", cnxn_sandbox)
# DF_SECOPI_contratos_2023 = pd.read_sql_query("select * from
```

```
[CCE_Sandbox].[SECOPI].[V_SECOPI] where FECHA_FIRMA_CONTRATO > '2019-12-31' and FECHA_FIRMA_CONTRATO < '2023-08-31'", cnxn_sandbox)
DF_SECOPI_contratos_2023.to_csv('../../../muestras de datos/sin procesar/SECOPI_ENERO23.csv', index=False, sep=';')

# SECOP II
DF_SECOPII_contratos_2023 = pd.read_sql_query("Select * from [CCE_Reports].[SECOPII].[V_HistoricoContratos_Depurado] where YEAR(AprovalDate) = 2023 and MONTH(AprovalDate) = 1", cnxn_reports)
# DF_SECOPII_contratos_2023 = pd.read_sql_query("Select * from [CCE_Reports].[SECOPII].[V_HistoricoContratos_Depurado] where AprovalDate > '2019-12-31' and AprovalDate < '2023-08-31'", cnxn_reports)
DF_SECOPII_contratos_2023.to_csv('../../../muestras de datos/sin procesar/SECOPII_ENERO23.csv', index=False, sep=';')

# TVEC
# DF_TVEC_Ordenes = pd.read_sql_query("select * from TVEC.Ordenes where Fecha > '2021-12-31' and Fecha < '2023-08-31'", cnxn_sandbox)
# DF_TVEC_Ordenes.to_csv('../../../muestras de datos/sin procesar/TVEC_Ordenes_2022_2023.csv', index=False, sep=';')
# DF_TVEC_Ordenes_Items.to_csv('../../../muestras de datos/sin procesar/TVEC_Ordenes_Items.csv', index=False, sep=';')
# DF_TVEC_Entidades.to_csv('../../../muestras de datos/sin procesar/TVEC_Entidades.csv', index=False, sep=';')
# DF_TVEC_Proveedores.to_csv('../../../muestras de datos/sin procesar/TVEC_Proveedores.csv', index=False, sep=';')

C:\Users\Jorge\AppData\Local\Temp\ipykernel_7220\669651293.py:2: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
  DF_SECOPI_contratos_2023 = pd.read_sql_query("select * from [CCE_Sandbox].[SECOPI].[V_SECOPI] where YEAR(FECHA_FIRMA_CONTRATO) = 2023 and MONTH(FECHA_FIRMA_CONTRATO) = 1", cnxn_sandbox)
C:\Users\Jorge\AppData\Local\Temp\ipykernel_7220\669651293.py:7: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
  DF_SECOPII_contratos_2023 = pd.read_sql_query("Select * from [CCE_Reports].[SECOPII].[V_HistoricoContratos_Depurado] where YEAR(AprovalDate) = 2023 and MONTH(AprovalDate) = 1", cnxn_reports)

DF_SECOPI_contratos_2023 = pd.read_csv('../../../muestras de datos/sin procesar/SECOPI_contratos_2022_2023.csv', sep=';')
DF_SECOPII_contratos_2023 = pd.read_csv('../../../muestras de datos/sin procesar/SECOPII_contratos_2022_2023.csv', sep=';')
```

```
C:\Users\Jorge\AppData\Local\Temp\ipykernel_17104\2609911550.py:1:
DtypeWarning: Columns (21,45,49,63,64,67,74,79,81,83,84,89) have mixed
types. Specify dtype option on import or set low_memory=False.
  DF_SECOPI_contratos_2023 = pd.read_csv('../../../muestras de
datos/sin procesar/SECOPI_contratos_2022_2023.csv', sep=';')
C:\Users\Jorge\AppData\Local\Temp\ipykernel_17104\2609911550.py:2:
DtypeWarning: Columns (26,27,30,31) have mixed types. Specify dtype
option on import or set low_memory=False.
  DF_SECOPII_contratos_2023 = pd.read_csv('../../../muestras de
datos/sin procesar/SECOPII_contratos_2022_2023.csv', sep=';')
```

Se separán los registros que no tengan un `ID_Entidad` o un `ID_Proveedor` de la base de ordenes de la tienda virtual.

```
Ordenes_Entidades_SIN_ID=DF_TVEC_Ordenes[DF_TVEC_Ordenes['ID_Entidad']
.isna()]
Ordenes_Proveedores_SIN_ID=DF_TVEC_Ordenes[DF_TVEC_Ordenes['ID_Proveed
or'].isna()]

Ordenes_Entidades_SIN_ID.to_excel("../../../muestras de
datos/procesados/Ordenes_Entidades_TVEC_SIN_ID.xlsx")
Ordenes_Proveedores_SIN_ID.to_excel("../../../muestras de
datos/procesados/Ordenes_Proveedores_TVEC_SIN_ID.xlsx")

DF_TVEC_Ordenes.dropna(subset=['ID_Entidad'],inplace=True)
DF_TVEC_Ordenes.dropna(subset=['ID_Proveedor'],inplace=True)
```

Ejecutamos un cambio de tipo para los campos que contengan id's.

```
DF_TVEC_Ordenes['ID_Entidad']=DF_TVEC_Ordenes['ID_Entidad'].astype(int
).astype(str)
DF_TVEC_Ordenes['ID_Proveedor']=DF_TVEC_Ordenes['ID_Proveedor'].astype
(int).astype(str)
DF_TVEC_Entidades['ID']=DF_TVEC_Entidades['ID'].fillna(0).astype(int).
astype(str)
DF_TVEC_Proveedores['ID']=DF_TVEC_Proveedores['ID'].fillna(0).astype(i
nt).astype(str)
```

Y por último, en lo que corresponde a la TVEC, se hace un merge para consolidar toda la información en un solo dataframe, de modo que, se pueda ejecutar la unificación con las bases del SECOP I y del SECOP II.

```
# Primero, unimos DF_TVEC_Ordenes con DF_TVEC_Entidades
df_merge_Ordenes_Entidades = pd.merge(DF_TVEC_Ordenes,
DF_TVEC_Entidades, left_on='ID_Entidad', right_on='ID',
suffixes=('_Ordenes', '_Entidades'),how='left')

# Luego, unimos el DataFrame resultante con DF_TVEC_Proveedores
```

```
df_final_TVEC = pd.merge(df_merge_Ordenes_Entidades,
DF_TVEC_Proveedores, left_on='ID_Proveedor', right_on='ID',
suffixes=('', '_Proveedores'),how='left')

df_final_TVEC['URL']='https://www.colombiacompra.gov.co/content/tienda
-virtual'
df_final_TVEC['Modalidad']='TVEC'
df_final_TVEC['Tipo']='TVEC'
df_final_TVEC['Tipo Documento Proveedor']='TVEC'
df_final_TVEC['Objeto Contrato']=df_final_TVEC['Agregacion']
df_final_TVEC['Tipo de documento proveedor']='NIT'

df_final_TVEC.columns
```

```
Index(['ID_Ordenes', 'ID_Entidad', 'Entidad', 'Solicitante',
'Fecha_Ordenes',
       'Fecha_vence', 'ID_Proveedor', 'Proveedor', 'Estado',
'Solicitud',
       'Items', 'Total', 'Agregacion', 'Cotizacion', 'Padre',
'Ciudad_Ordenes',
       'Categoria', 'RFQ', 'Paz', 'Proceso', 'UNSPSC', 'Contrato',
'email',
       'Supervisor', 'Version', 'ID_Entidades', 'Nombre', 'NIT',
'Obligada',
       'Orden', 'Rama', 'Sector', 'Departamento', 'Ciudad_Entidades',
       'Fecha_Entidades', 'Active', 'Fechatemporal', 'Telefono', 'ID',
       'Nombre_Proveedores', 'NombreComercial', 'NIT_Proveedores',
       'Estado_Proveedores', 'Contacto', 'email_Proveedores',
'Direccion',
       'Ciudad', 'Departamento_Proveedores', 'Agregacion_Proveedores',
       'ActividadEconomica', 'RegTributario', 'FechaCreacion', 'URL',
       'Modalidad', 'Tipo', 'Tipo Documento Proveedor', 'Objeto
Contrato',
       'Tipo de documento proveedor'],
      dtype='object')
```

Se crea la columna `Fuente` para identificar la información por plataforma

```
DF_SECOPI_contratos_2023['Fuente'] = 'SECOP I'
DF_SECOPII_contratos_2023['Fuente'] = 'SECOP II'
# df_final_TVEC['Fuente'] = 'TVEC'
```

A continuación, se eliminarán los registros de la base del SECOP I que no tienen un `ID_ADJUDICACION`. Adicionalmente, haremos una transformación del tipo de datos a entero.

```
DF_SECOPI_contratos_2023.dropna(subset=['ID_ADJUDICACION'],inplace=Tru
e)
DF_SECOPI_contratos_2023['ID_ADJUDICACION']=DF_SECOPI_contratos_2023['
ID_ADJUDICACION'].astype('int64').astype('str')
```

Una vez se han suprimido los contratos con `ID_ADJUDICACION` nulo, daremos inicio a la unificación de la información en una única tabla.

Proceso de Unificación de Datos

Una vez que los datos están preparados, procederemos a unificarlos en una sola tabla consolidada. Dado que las bases de datos del SECOP I y SECOP II pueden tener diferentes esquemas y campos, se realizará una adecuada identificación y mapeo de columnas para garantizar una correspondencia adecuada.

```
# Leer el archivo Excel que contiene la homologación de columnas
(Mapeo de variables)
HC = pd.read_excel('../../../muestras de
datos/auxiliar/Homologa_columnas.xlsx')

# Crear dos DataFrames vacíos para almacenar los datos unificados
new_SECOP_I = pd.DataFrame()
new_SECOP_II = pd.DataFrame()
new_TVEC = pd.DataFrame()

for ind_column in HC.index:
    # Asignar los valores de la columna del DataFrame SECOP I original
a la nueva columna unificada
    new_SECOP_I[HC['Unificado']
[ind_column]]=DF_SECOPI_contratos_2023[HC['SECOP I'][ind_column]]
    # Asignar los valores de la columna del DataFrame SECOP II
original a la nueva columna unificada
    new_SECOP_II[HC['Unificado']
[ind_column]]=DF_SECOPII_contratos_2023[HC['SECOP II'][ind_column]]
    # Asignar los valores de la columna del DataFrame TVEC original a
la nueva columna unificada
    #new_TVEC[HC['Unificado'][ind_column]]=df_final_TVEC[HC['TVEC']
[ind_column]]

# Eliminar los DataFrames originales para liberar memoria
del DF_SECOPI_contratos_2023
del DF_SECOPII_contratos_2023
#del df_final_TVEC
```

Finalmente, procedemos a unificar la información en la tabla `DF_Consulta`

```
DF_Consulta = pd.concat([new_SECOP_I,new_SECOP_II])
#DF_Consulta = pd.concat([new_SECOP_I,new_SECOP_II,new_TVEC])
DF_Consulta.reset_index(inplace=True)

DF_Consulta['Fecha_firma'] =
pd.to_datetime(DF_Consulta['Fecha_firma'])
```

```
DF_Consulta['Año_firma'] = DF_Consulta['Fecha_firma'].dt.year
DF_Consulta.head(2)
```

```
   index ID_Contrato   ID_Proceso ID_Entidad
Nombre_Entidad  \
0      0    12527776  23-22-57504  215000073  BOYACÁ  LOTERÍA DE
BOYACÁ
1      1    12527776  23-22-57504  215000073  BOYACÁ  LOTERÍA DE
BOYACÁ

  NIT_Entidad Orden_Entidad
Modalidad  \
0   891801039   TERRITORIAL  Contratos y convenios con más de dos
partes
1   891801039   TERRITORIAL  Contratos y convenios con más de dos
partes

      Estado
Descripcion_proceso  ...  \
0  Celebrado  SUMINISTRO DE COMBUSTIBLE GASOLINAACPM PARA  L...  ...

1  Celebrado  SUMINISTRO DE COMBUSTIBLE GASOLINAACPM PARA  L...  ...


  Valor_contrato Departamento_Entidad Municipio_Entidad  \
0       30000000               Boyacá             Tunja
1       30000000               Boyacá             Tunja

  Departamento_Proveedor Municipio_Proveedor Fecha_inicio_contrato  \
0                 Boyacá              Boyacá            2023-02-01
1                 Boyacá              Boyacá            2023-02-01

    Fecha_fin_contrato
Link  \
0  2024-01-01 00:00:00
https://www.contratos.gov.co/consultas/detalle...
1  2024-01-01 00:00:00
https://www.contratos.gov.co/consultas/detalle...

    Fuente Año_firma
0  SECOP I      2023
1  SECOP I      2023

[2 rows x 27 columns]
```

```
DF_Consulta[DF_Consulta['Año_firma'] ==
2020].to_csv('../../../muestras de
datos/procesados/bronce/SECOP_2020.csv', index=False, sep=';')
DF_Consulta[DF_Consulta['Año_firma'] ==
2021].to_csv('../../../muestras de
datos/procesados/bronce/SECOP_2021.csv', index=False, sep=';')
```

```
DF_Consulta[DF_Consulta['Año_firma'] ==
2022].to_csv('../../../muestras de
datos/procesados/bronce/SECOP_2022.csv', index=False, sep=';')
DF_Consulta[DF_Consulta['Año_firma'] ==
2023].to_csv('../../../muestras de
datos/procesados/bronce/SECOP_2023.csv', index=False, sep=';')

DF_Consulta.to_csv('../../../muestras de
datos/procesados/bronce/SECOP_ENERO23.csv', index=False, sep=';')
```

Metricas de la base previas al proceso de limpieza

Con la siguiente línea, se proporcionará un resumen conciso y útil sobre la estructura y contenido del DataFrame, incluyendo:

- El número total de filas en el DataFrame.
- El número de columnas en el DataFrame.
- El nombre de cada columna y su tipo de datos.
- La cantidad de valores no nulos en cada columna.
- La cantidad de memoria utilizada por el DataFrame.

Esta información será útil para verificar la integridad y la consistencia de la base consolidada y para identificar posibles problemas, como valores faltantes o tipos de datos incorrectos. También proporciona una visión general rápida de la cantidad de datos presentes en el DataFrame y ayudará a tomar decisiones informadas sobre el manejo de los datos y el análisis posterior.

```
DF_Consulta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6172969 entries, 0 to 6172968
Data columns (total 26 columns):
 #   Column                  Dtype
---  ------                  -----
 0   index                   int64
 1   ID_Contrato             object
 2   ID_Proceso              object
 3   ID_Entidad              object
 4   Nombre_Entidad          object
 5   NIT_Entidad             object
 6   Orden_Entidad           object
 7   Modalidad               object
 8   Estado                  object
 9   Descripcion_proceso     object
 10  Objeto_Contrato         object
```

```
 11  Tipo_de_contrato        object
 12  Fecha_firma             object
 13  UNSPSC                  object
 14  Nombre_Proveedor        object
 15  Documento_Proveedor     object
 16  Tipo_proveedor          object
 17  Valor_contrato          float64
 18  Departamento_Entidad    object
 19  Municipio_Entidad       object
 20  Departamento_Proveedor  object
 21  Municipio_Proveedor     object
 22  Fecha_inicio_contrato   object
 23  Fecha_fin_contrato      object
 24  Link                    object
 25  Fuente                  object
dtypes: float64(1), int64(1), object(24)
memory usage: 1.2+ GB
```

Y, por último, usaremos la función `describe()` de pandas para proporcionar información clave sobre las variables numéricas del DataFrame, incluyendo el número de observaciones (count), la media (mean), la desviación estándar (std), el valor mínimo (min), los percentiles (25%, 50% y 75%), y el valor máximo (max) de cada columna numérica.

El resumen estadístico generado por la siguiente línea de código será útil para obtener una visión general rápida de la distribución y las tendencias centrales de los datos numéricos en el DataFrame.

```
DF_Consulta.describe()
```

```
              index  Valor_contrato
count  6.172969e+06    6.172969e+06
mean   1.647730e+06    2.555750e+08
std    1.051403e+06    8.803351e+10
min    0.000000e+00    0.000000e+00
25%    7.716210e+05    5.987664e+06
50%    1.543242e+06    1.290000e+07
75%    2.346361e+06    2.791667e+07
max    3.889603e+06    1.420769e+14
```