



**PROCOLOMBIA**

EXPORTACIONES TURISMO INVERSIÓN MARCA PAÍS

---

# MANUAL AUTOMATIZACIONES

## PYTHON

---

JUAN ESTEBAN RINCÓN POVEDA

PRACTICANTE

VICEPRESIDENCIA PLANEACIÓN EXPORTACIONES

CUALQUIER DUDA PUEDEN PONERSE EN CONTACTO CONMIGO AL CORREO

[JUAN.RINCONP@HOTMAIL.COM](mailto:JUAN.RINCONP@HOTMAIL.COM)

# Índice

<b>1</b>	<b>Introducción</b>	<b>2</b>
1.1	Como abrir Jupyter Notebook . . . . .	2
1.2	Como encontrar los archivos . . . . .	3
<b>2</b>	<b>Automatización Tablero Como Vamos</b>	<b>5</b>
2.1	TCV Exportaciones . . . . .	6
2.2	TCV Turismo . . . . .	7
2.3	TCV Inversión . . . . .	9
<b>3</b>	<b>Automatización Informe Semanal (Exportaciones)</b>	<b>11</b>
3.1	Cargar bases e información importante . . . . .	11
3.2	Resultados . . . . .	12
<b>4</b>	<b>Automatización Planes Sectoriales</b>	<b>14</b>
4.1	Cargar bases y cambio de nombre de variables . . . . .	14
4.2	Resultados . . . . .	15
<b>5</b>	<b>Ayuda</b>	<b>16</b>
5.1	Errores Frecuentes . . . . .	16

## Sobre este documento

Este documento fue creado para explicar el uso de las automatizaciones en Python. No se busca enseñar la programación en Python sino a usar los códigos ya existentes.

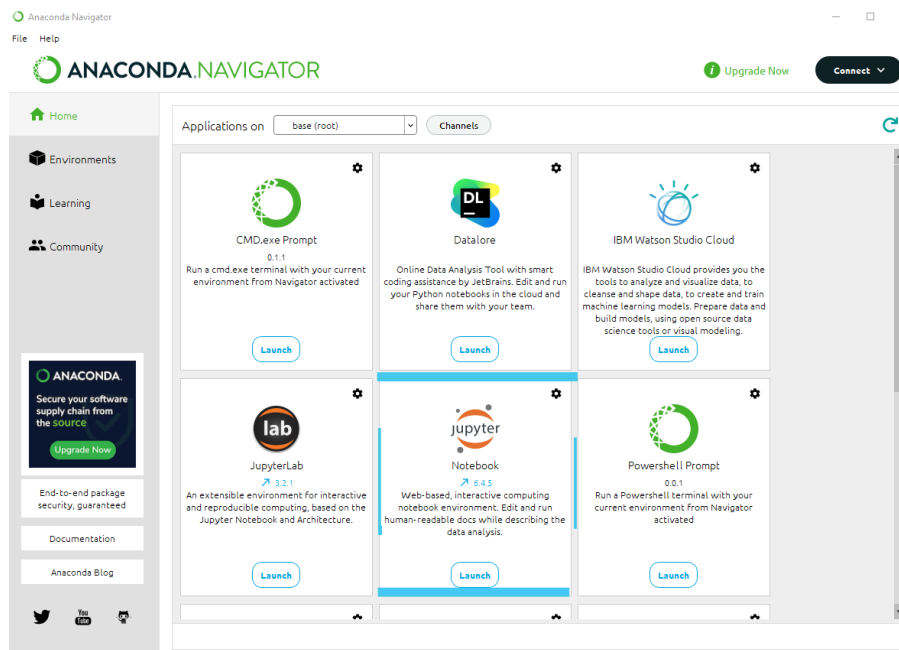
# Introducción

Este manual busca explicar como se usan las automatizaciones para obtener los insumos de los tableros como vamos (exportaciones, inversión y turismo), el informe semanal y el informe de planes sectoriales. Inicialmente se explicara como entrar a Python para después explicar como correr los códigos.

## 1.1 Como abrir Jupyter Notebook

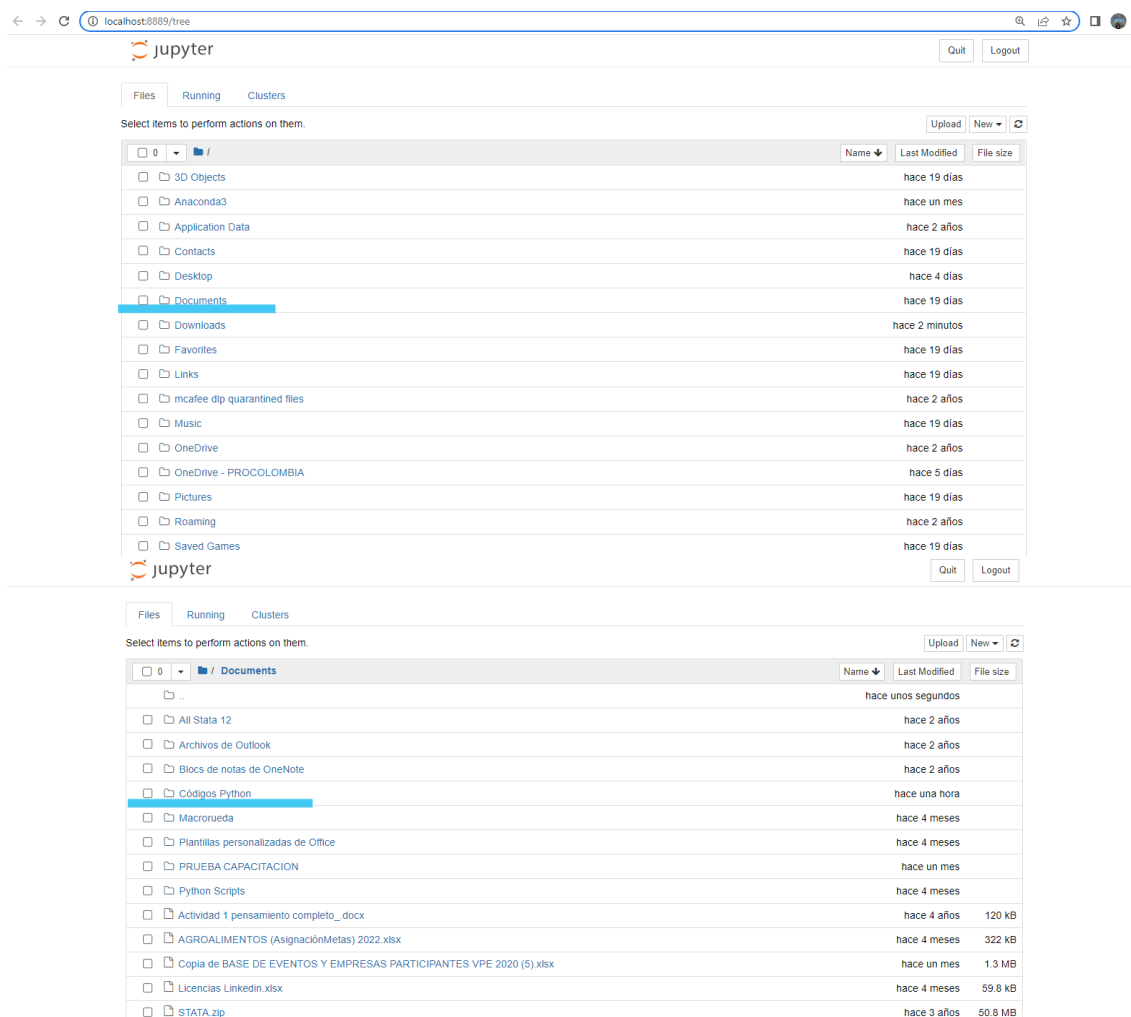
Para usar Python vamos a emplear un programa llamado Jupyter Notebook. En Jupyter podemos abrir los archivos de Python, editarlos y correrlos. Este programa ya fue instalado en el computador pero si por alguna razón se requiere volver a instalar, este link contiene una guía paso a paso de como hacerlo tanto en mac como en windows <https://econgrowth.github.io/pages/Computation.html>

Para abrir Jupyter notebook vamos a abrir el programa Anaconda en el computador y luego hacer clic en el boton **launch** de Jupyter notebook como se muestra en la imagen (es la casilla encerrada en azul):



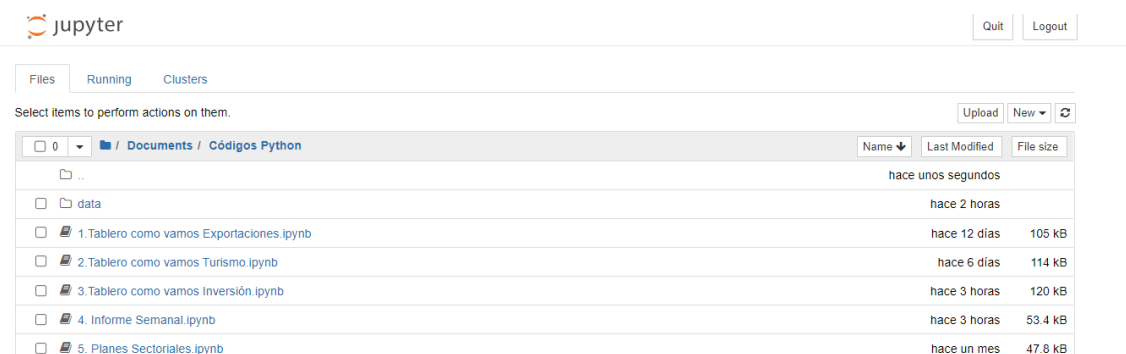
## 1.2 Como encontrar los archivos

Después de abrir Jupyter, vamos a buscar los códigos. Para esto abriremos la carpeta documentos y luego buscaremos la carpeta Códigos Python.



Acá encontrarán los siguientes archivos:

- 1. Tablero como vamos Exportaciones
- 2. Tablero como vamos Turismo
- 3. Tablero como vamos Inversión\*
- 4. Informe Semanal
- 5. Planes Sectoriales



The screenshot shows the JupyterLab interface. At the top, there's a 'jupyter' logo and 'Quit' and 'Logout' buttons. Below that, there are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' with 'Upload', 'New', and a refresh icon. The main area shows a file browser for the path '/ Documents / Códigos Python'. It lists five files, each with a checkbox, a file icon, the filename, and its last modified time and size.

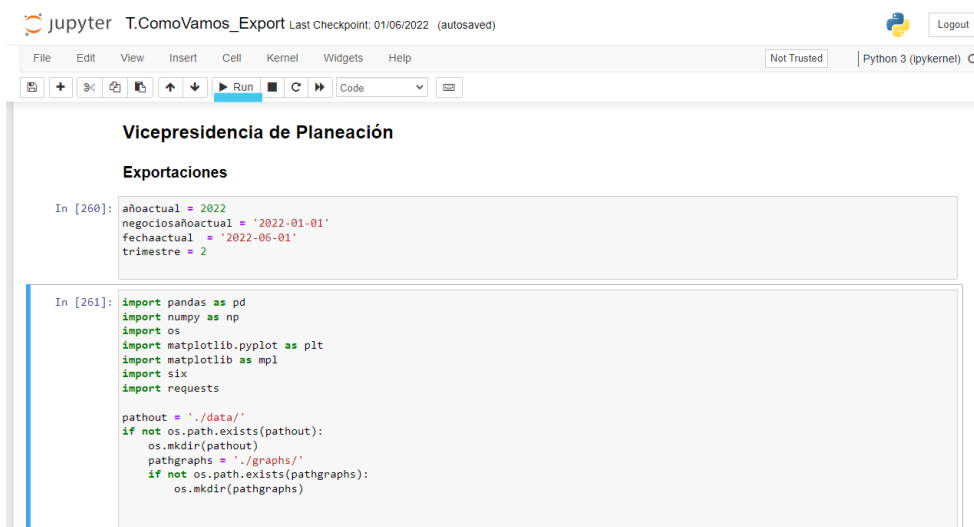
	Name	Last Modified	File size
<input type="checkbox"/>	..	hace unos segundos	
<input type="checkbox"/>	data	hace 2 horas	
<input type="checkbox"/>	1. Tablero como vamos Exportaciones.ipynb	hace 12 días	105 kB
<input type="checkbox"/>	2. Tablero como vamos Turismo.ipynb	hace 6 días	114 kB
<input type="checkbox"/>	3. Tablero como vamos Inversión.ipynb	hace 3 horas	120 kB
<input type="checkbox"/>	4. Informe Semanal.ipynb	hace 3 horas	53.4 kB
<input type="checkbox"/>	5. Planes Sectoriales.ipynb	hace un mes	47.8 kB

En las siguientes partes del documento encontrarán una explicación más detallada sobre como funciona cada automatización. Aunque los códigos están guardados en el PC del practicante, una copia de estos puede encontrarse en github en el siguiente link: <https://github.com/juanrinconp/Procolombia>.

# Automatización Tablero

## Como Vamos

Los archivos 1, 2 y 3 contienen el código para obtener los insumos del tablero como vamos de cada subdirección. Cada código requiere de bases distintas para calcular los insumos, los cuales deben estar guardados en la carpeta llamada **data**. Esta carpeta ya debe estar creada, pero en el caso de que no este, esta se crea automáticamente al correr la primera linea de cualquiera de los códigos. Para correr una casilla de código debemos hacer clic en esta y luego hacer clic en run. La siguiente imagen muestra como estamos haciendo clic en la casilla que importa las librerías y crea las carpetas así como el botón de run.



The screenshot shows a Jupyter Notebook window titled "T.ComoVamos\_Export" with a last checkpoint of "01/06/2022 (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The notebook content is divided into two sections: "Vicepresidencia de Planeación" and "Exportaciones". The "Exportaciones" section contains two code cells. The first cell, labeled "In [260]:", defines variables for the current year, business date, report date, and quarter. The second cell, labeled "In [261]:", imports necessary libraries (pandas, numpy, os, matplotlib.pyplot, matplotlib, six, requests) and creates two directories, "data" and "graphs", if they do not already exist.

```
In [260]: añoactual = 2022
negociosañoactual = '2022-01-01'
fechaactual = '2022-06-01'
trimestre = 2

In [261]: import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import matplotlib as mpl
import six
import requests

pathout = './data/'
if not os.path.exists(pathout):
    os.mkdir(pathout)
pathgraphs = './graphs/'
if not os.path.exists(pathgraphs):
    os.mkdir(pathgraphs)
```

Ahora vamos a ver como funciona cada automatización.

## 2.1 TCV Exportaciones

Para el tablero como vamos de exportaciones necesitamos cargar las siguientes bases de datos:

- Negocios
- Metas
- Oportunidades
- Empresas Nacionales (Tableau 12a)

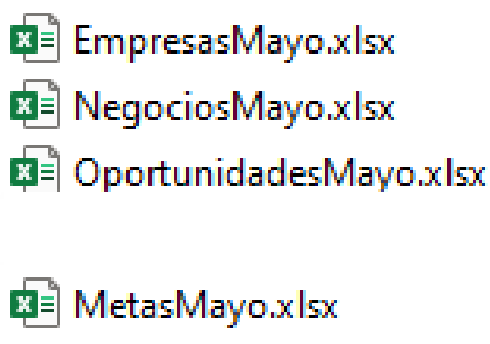
Lo primero es que al cargar estas bases en la carpeta **data** debemos dejarlas con el mismo nombre que esta en la programación. Por ejemplo el nombre original de la base de negocios es *(Histórico -Estandarizado) Negocios 3 Ejes 2009 a Cierre de Mayo*. Lo que debemos hacer es por ejemplo cambiarle el nombre por *NegociosMayo* tanto en el archivo como en la línea de Python que lo carga. La siguiente imagen muestra como se cambiaron los nombres a las 4 bases.

```
Metas = pd.read_excel(pathout + 'MetasMayo.xlsx')
```

```
Negocios = pd.read_excel(pathout + 'NegociosMayo.xlsx')
```

```
Oportunidades = pd.read_excel(pathout + 'OportunidadesMayo.xlsx')
```

```
ENacionales = pd.read_excel(pathout + 'EmpresasMayo.xlsx')
```



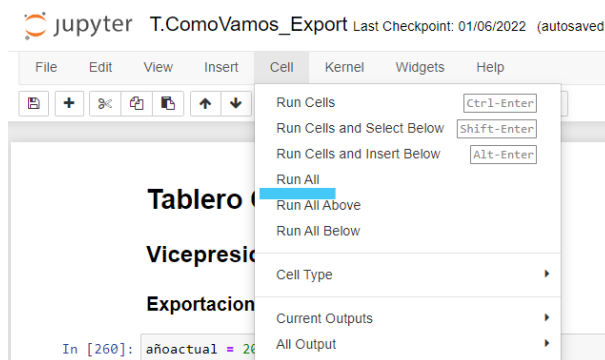
Después de cargar las bases debemos darle a Python 4 datos en unos formatos específicos. La siguiente imagen muestra las variables en las que guardaremos esta información:

```
añoactual = 2022  
negociosañoactual = '2022-01-01'  
fechaactual = '2022-06-01'  
trimestre = 2
```

- **añoactual:** Año actual en formato yyyy
- **negociosactual:** Enero del año actual en formato yyyy-mm-dd
- **fechaactual:** Fecha actual o del corte al cual se desea calcular los insumos en formato yyyy-mm-dd
- **trimestre:** El trimestre actual en formato t

En el caso de la imagen anterior se muestra como estamos calculado el tablero como vamos de mayo 2022 (en fecha actual ponemos 1 de junio de 2022).

Luego de cargar las bases y de darle a Python la información de las fechas vamos a poner el cursor sobre el botón **cell** para que despliegue una lista de opciones, y después haremos clic en run all. La siguiente imagen muestra un ejemplo de esto:



Aquí Python iniciara a cargar las bases para después realizar lo cálculos de los insumos. Para dar un contexto de lo que el código hace es; primero calcular tablas con las metas (diferenciando por HUB, Oficina regional, etc) para después calcular los resultados y añadirlos a cada tabla. El código se demora aproximadamente 15 minutos en dar los insumos. La mayoría de este tiempo se usa en cargar las bases ya que estas son relativamente pesadas.

Una vez el programa termina de cargar, iremos al final del código y haremos clic en el link que se llama **Insumos Tablero Como Vamos.xlsx**. Esta va a descargar un archivo en excel que contiene los insumos necesarios para construir el tablero. Estos insumos deben ser enviados al asesor encargado de terminar la construcción del informe.

```
In [320]: from IPython.display import FileLink
          FileLink('Insumos Tablero Como Vamos.xlsx')

Out[320]: Insumos Tablero Como Vamos.xlsx
```

Por favor hacer clic en el Link para descargar el archivo, gracias.

## 2.2 TCV Turismo

Para el tablero como vamos de turismo necesitamos cargar las siguientes bases de datos:

- Negocios
- Metas



- Oportunidades
- Empresas Nacionales (Tableau 12a)
- Proyectos e Hitos
- Asistente PFE
- Asistentes PFE (año anterior)

Al igual que en la automatización de exportaciones, los nombres de las bases deben ser iguales a los escritos en el código.

Adicionalmente debemos darle al programa la siguiente información:

- **añoactual:** Año actual en formato yyyy
- **añoanterior:** Año anterior en formato yyyy
- **trimestre:** El trimestre actual en formato t
- **negociosactual:** Enero del año actual en formato yyyy-mm-dd
- **fechaactual:** Fecha actual o del corte al cual se desea calcular los insumos en formato yyyy-mm-dd
- **fechaactual2:** Enero del año actual en formato en formato d/mm/yyyy
- **fechaanterior2:** Enero del año anterior en formato en formato d/mm/yyyy
- **fechaactual-añoanterior:** Fecha actual del año anterior en formato en formato yyyy-mm-dd
- **diciembreanterior:** Diciembre del año anterior en formato en formato yyyy-mm-dd'

Adicionalmente se le deben solicitar al asesor los siguientes valores:

- **frecuencias:** Resultados de frecuencias
- **nsillas:** Número de sillas
- **nVisitantes:** Número de visitantes
- **mVisitantes:** Meta de visitantes
- **mqVisitantes:** Meta del Q de visitantes
- **mVisitanterior:** Meta de visitantes del año anterior

```

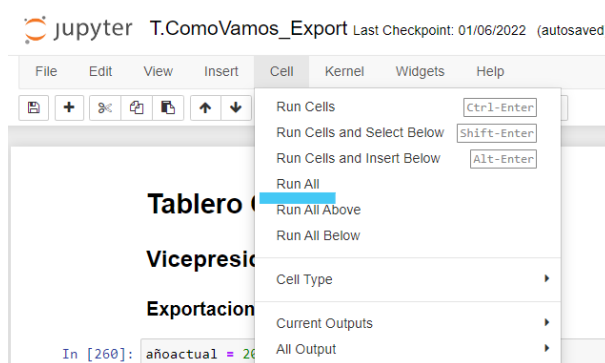
añoactual = 2022
añoanterior = 2021
trimestre = 2

#Fecha en Formato aaaa/mm/dd para la base de Tableau 12a
negociosañoactual = '2022-01-01'
fechaactual = '2022-06-01'
fechaactual2 = '1/01/2022'
fechaanterior2 = '1/01/2021'
eneroañoanterior = '2021-01-01'
fechaactual_añoanterior = '2021-06-01'
diciembreañoanterior = '2021-12-31'

#Valores
frecuencias = 1160
nsillas = 199579
nVisitantes = 914882
mVisitantes = 3100000
mqVisitantes = 89900
mVisitanteanterior = 2157504

```

Luego de cargar las bases y de darle a Python la información de las fechas vamos a poner el cursor sobre el botón **cell** para que despliegue una lista de opciones, y después haremos clic en run all. La siguiente imagen muestra un ejemplo de esto:



Al igual que en el caso de exportaciones, debemos hacer clic en el archivo que se nos muestra al final, para así descargar el excel con los insumos que posteriormente debe ser enviado al asesor responsable.

## 2.3 TCV Inversión

El tablero como vamos de inversión no se logra terminar al 100%. Este solo entrega resultados diferenciando por HUB, Gerencia y oficinas regionales para las metas de Monto de Inversión, Proyectos de Inversión y Movimiento de oportunidades (etapa 1, 2 y 3).

Para el tablero como vamos de Inversión necesitamos cargar las siguientes bases de datos:

- Negocios
- Metas
- Oportunidades
- Empresas Nacionales (Tableau 12a)
- Cambio de Etapa
- Oportunidades Regionales
- Portafolio de Proyectos

Al igual que en la automatización de exportaciones, los nombres de las bases deben ser iguales a los escritos en el código.

Adicionalmente debemos darle al programa la siguiente información:

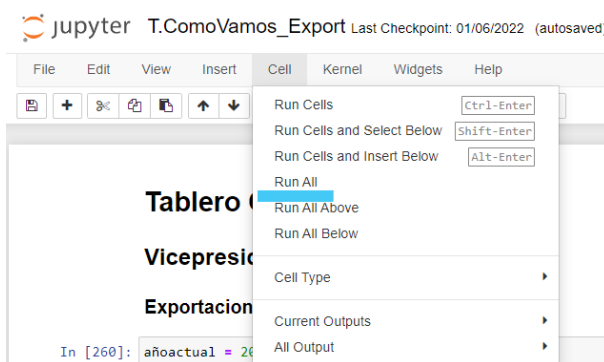
- **añoactual:** Año actual en formato yyyy
- **trimestre:** El trimestre actual en formato t
- **fechaetapa:** Diciembre del año anterior en formato dd/mm/yyyy
- **fecha etapa2:** Fecha actual o del corte al cual se desea calcular los insumos en formato dd/mm/yyyy
- **fecha2:** Diciembre del año anterior en formato dd/mm/yyyy
- **negociosañoactual:** Enero del año actual en formato en formato yyyy-mm-dd
- **fechaactual:** Fecha actual en formato yyyy-mm-dd
- **fechaactual2:** Enero del año actual en formato en formato d/mm/yyyy

```
añoactual = 2022
trimestre = 2

fechaetapa = '31/12/2021' #Esta fecha es para eliminar lo menor a 2022 en la base de cambio de etapa
fechaetapa2 = '02/05/2022'
fecha2 = '31/12/2021'

#Fecha en Formato aaaa/mm/dd para La base de Tableau 12a
negociosañoactual = '2022-01-01'
fechaactual = '2022-05-02'
fechaactual2 = '1/01/2022'
```

Luego de cargar las bases y de darle a Python la información de las fechas vamos a poner el cursor sobre el botón **cell** para que despliegue una lista de opciones, y después haremos clic en run all. La siguiente imagen muestra un ejemplo de esto:



Al igual que en el caso de exportaciones, debemos hacer clic en el archivo que se nos muestra al final, para así descargar el excel con los insumos que posteriormente debe ser enviado al asesor responsable.



PARTE

## Automatización Informe Semanal (Exportaciones)

El informe semanal es un informe que se presenta los miércoles y busca mostrar los resultados de Procolombia semanalmente. Se calculan indicadores como monto de exportación, número de compradores, número de exportadores, diferenciando entre periodos(años, o periodos de gobierno) y tipo de productos(economía naranja o servicios).

Este código sigue la misma logica que el tablero como vamos, a diferencia que los insumos deben ser usados por el practicante (no por el asesor) para llenar los cuadros del formato de excel.

### 3.1 Cargar bases e información importante

Al igual que las demás automatizaciones, para que el código funcione debemos cargar algunas bases y agregar fechas.

Para este informe necesitamos 6 bases, de las cuales 5 solo cambian una vez al año:

- Base de Negocios actualizada al corte
- Formato Resultados 2022
- Formato Resultados Cuatrienio 2018-2022
- Formato Periodo Gobierno Ago2018-2022
- Formato HUB
- Formato Regional

Los formatos ya están cargados por lo que no deben cambiar en el 2022. Estos formatos tienen los nombres de los indicadores y son aquí donde la automatización agrega los resultados.

El proceso a seguir es el de darle a Python las fechas según el corte deseado. Estas deben ser escritas en Python de la siguiente forma:

- **fechaactual:** Fecha actual en formato yyyy-mm-dd
- **fecha2021:** Fecha actual del año 2021 en formato yyyy-mm-dd
- **fecha2020:** Fecha actual del año 2021 en formato yyyy-mm-dd
- **fecha2019:** Fecha actual del año 2021 en formato yyyy-mm-dd
- **fecha2018:** Fecha actual del año 2021 en formato yyyy-mm-dd

Las demás fechas no se deben mover. La siguiente imagen muestra donde se deben reemplazar los valores mencionados anteriormente:

```
añoactual = 2022
año2021 = 2021
año2020 = 2020
año2019 = 2019
año2018 = 2018
año2017 = 2017
año2016 = 2016
año2015 = 2015
año2014 = 2014

cuatrenio1 = '2018-01-01'
Cuatrenio = '2018 - 2022'
Cuatrenio2 = '2014 - 2018'

FInicioGobierno = '2018-08-01'
FInicioGobiernoAnterior = '2014-08-01'

fechaactual = '2022-06-08'
fecha2021 = '2021-06-08'
fecha2020 = '2020-06-08'
fecha2019 = '2019-06-08'
fecha2018 = '2018-06-08'
```

## 3.2 Resultados

Después de correr el código y descargar los insumos, obtendremos un archivo de excel con 6 pestañas las cuales contienen la información que debemos pegar al archivo de resultados del informe semanal.

Las siguientes 2 imágenes muestran (1) la primera tabla del informe y (2) la primera tabla de la automatización. Como podemos ver en la segunda imagen, las columnas D, E, F y G muestran los resultados que debemos pegar en la tabla de resultados.

Exportaciones	Meta 2022	8-jun-22	Avance Meta	8-jun-19	8-jun-20	8-jun-21	Inc. 2022/2021	Inc. 2022/2020	Inc. 2022/2019
Empresas Col con negocios	2.890	1552	54%	1232	911	1124	38%	70%	20%
Compradores con negocios	8.820	4.014	46%	3.276	3.372	3.690	3%	18%	23%
<b>Negocios Exportaciones US\$</b>	<b>6.400</b>	<b>2.740</b>	<b>43%</b>	<b>2.388</b>	<b>2.306</b>	<b>2.144</b>	<b>28%</b>	<b>19%</b>	<b>15%</b>
# Departamentos	22	22		20	23		-4%	10%	0%
Países de los negocios	126	126		127	122		3%	-1%	3%
No. Mipymes	837	791		549	646		30%	52%	8%
No. Medianas	330	281		255	256		29%	29%	17%
<b>Monto de negocios a mercados con acuerdo</b>	<b>2.474,86</b>			<b>2.173,13</b>	<b>2.053,40</b>	<b>1.853,10</b>	<b>34%</b>	<b>21%</b>	<b>14%</b>
% negocios a mercados con acuerdo	90%			91%	89%	86%	4%	1%	-1%
Empresas Col con negocios en mercados con Acuerdo	1421			1121	797	1021	33%	78%	27%
Empresas Col con negocios (Economía Naranja)	180			107	78	77	134%	131%	68%
Compradores con negocios (Economía Naranja)	298			282	245	193	54%	22%	6%
<b>Negocios Exportaciones US\$ millones (Economía Naranja)</b>	<b>300</b>	<b>331</b>	<b>110%</b>	<b>87</b>	<b>78</b>	<b>136,62</b>	<b>143%</b>	<b>325%</b>	<b>282%</b>
# Departamentos	15			13	10	12	25%	50%	15%
Países de los negocios	36			33	37	30	20%	-3%	-8%
Empresas Col con negocios (Servicios)	214			139	107	108	56%	100%	54%
Compradores con negocios (Servicios)	370			417	385	399	-7%	-4%	-11%
<b>Negocios Exportaciones US\$ millones (Servicios)</b>	<b>1039</b>	<b>465</b>	<b>45%</b>	<b>307,9</b>	<b>201,3</b>	<b>313</b>	<b>48%</b>	<b>131%</b>	<b>51%</b>
# Departamentos	16			13	10	12	33%	60%	23%
Países de los negocios	48			79	75	57	-16%	-38%	-33%

	A	B	C	D	E	F	G	H
1		Indicador	Meta 2022	2022-06-08	2021-06-08	2020-06-08	2019-06-08	
2	0	Empresas	2890	1552	1124	911	1292	
3	1	Compradores	8820	4014	3690	3372	3276	
4	2	Negocios	6400	2740,23	2143,7	2305,66	2388,16	
5	3	# Departamentos		22	23	20	22	
6	4	Países de los negocios		126	122	127	122	
7	5	No. Mipymes		837	646	549	791	
8	6	No. Medianas		330	256	255	281	
9	7	Monto de negocios a		2474,86	1853,1	2053,4	2173,13	
10	8	% negocios a mercados		90	86	89	91	
11	9	Empresas Col con ne		1421	1021	797	1121	
12	10	Empresas Col con ne		180	77	78	107	
13	11	Compradores con ne		298	193	245	282	
14	12	Negocios	300	331,37	136,62	77,98	86,66	
15	13	# Departamentos (Ec		15	12	10	13	
16	14	Países de los negoc		36	30	37	39	
17	15	Empresas Col con ne		214	108	107	139	
18	16	Compradores con ne		370	399	385	417	
19	17	Negocios	1039	465,08	313,43	201,3	307,9	
20	18	# Departamentos (Se		16	12	10	13	
21	19	Países de los negoc		48	57	75	79	
22								
		Años Individuales	Cuatrenio		Gobierno	HUB	Regional	

Después de actualizar todas las pestañas con los insumos de la automatización, debemos subir el archivo al repositorio i para que los asesores puedan usar esta información.

## Automatización Planes Sectoriales

### 4.1 Cargar bases y cambio de nombre de variables

Para este informe necesitamos 4 bases:

- Base de Planes Sectoriales
- Base del DANE
- Base de Quienes
- Base de Negocios

Al igual que en las demás automatizaciones debemos asegurarnos que los nombres de las bases sean iguales a los nombres que le ponemos en Python.

Adicionalmente necesitamos cambiar 3 nombres ya que la base del DANE contiene unas variables que van cambiando según la fecha. Por ejemplo, en marzo teníamos una variable que se llamaba **2022 USD (Ene-Mar)**. En Junio esta variable se llama **2022 USD (Ene-Jun)**, por lo que en Python debemos cambiar el nombre para que el programa pueda encontrarla. En la siguiente imagen están resaltados los 3 nombres que debemos actualizar según aparecen en la base del DANE. En otras palabras hay que ir a la base del DANE a identificar el nuevo nombre para después actualizarlo en el código de Python. La siguiente imagen muestra donde se deben actualizar los nombres (son solo las 3 resaltadas, las demás se actualizan automáticamente en la programación).

```

var1 = '2021 USD'
var2 = '2021 USD (Ene-Mar)'
var3 = '2022 USD (Ene-Mar)'

fecha2 = '2021-12-31'
fecha3 = '2022-12-31'

monto1 = 'Monto ' + var1
monto2 = 'Monto ' + var2
monto3 = 'Monto ' + var3

numero1 = 'Numero de Empresas ' + var1
numero2 = 'Numero de Empresas ' + var2
numero3 = 'Numero de Empresas ' + var3

eind1 = 'Exportaciones Indirectas ' + var1
eind2 = 'Exportaciones Indirectas ' + var2
eind3 = 'Exportaciones Indirectas ' + var3

Q1 = 'Exportaciones Quienes ' + var1
Q2 = 'Exportaciones Quienes ' + var2
Q3 = 'Exportaciones Quienes ' + var3

QP1 = 'Exportaciones Quienes Procolombia ' + var1
QP2 = 'Exportaciones Quienes Procolombia ' + var2
QP3 = 'Exportaciones Quienes Procolombia ' + var3

```

Un posible error es que los nombres de las demás variables cambien. En la imagen anterior se ve como se programa para actualizar los nombres, pero si sale algún error, lo primero que habría que revisar es que por ejemplo el nombre de QP3 encaje con el que esta en la base del DANE.

## 4.2 Resultados

Al igual en que en casos anteriores, hacemos clic en **run all** y luego descargamos el archivo haciendo clic en el link del final. Estos insumos deben ser enviados al asesor responsable.





# Ayuda

## 5.1 Errores Frecuentes

Estos son los errores más frecuentes a la hora de usar cualquiera de las automatizaciones. Por favor tenerlos en cuenta para evitar inconvenientes:

- No cambiar los nombres de las bases en Python
- Que la primera pagina del excel no sea la base de datos (a veces los asesores crean otras tablas en las mismas bases. Python toma siempre la primera pestaña del excel por lo que hay que asegurarse que esta primera pestaña sea la base y no otra cosa)
- Que alguna variable cambie de nombre. Para que las automatizaciones sigan siendo útiles a lo largo del tiempo no se deben cambiar los nombres de las variables.
- No cambiar las fechas
- No cambiar los nombres de las variables en la automatización de planes sectoriales.
- Que alguna variable tenga espacios al final. Esto también es un problema cuando se cruzan nombres por ejemplo.
- Diferencias en las mayúsculas y minúsculas. Si hay variables, nombres, etc., que tienen diferencias entre mayúsculas o minúsculas, Python lo tomara como si fuesen 2 elementos distintos.