

Ejemplo del proceso de automatización para un indicador del SIESDE

Elaborado por: Dr. Fernando Rebollar Castelan.

Introducción

El SIESDE (Sistema de Información Estratégica en Salud, Dependencia Funcional y Envejecimiento) es un proyecto de ciencia de datos especializado que permite procesar y analizar las fuentes del Sistema de Información en Salud (SIS) y el Sistema Nacional de Información Estadística y Geográfica (SNIEG) para generar estadísticas e indicadores, escenarios prospectivos e información geoespacial a nivel municipal, entidad federativa y nacional.

El objetivo del SIESDE es generar información y extracción de conocimiento, a partir de los datos actualmente disponibles, para el procesamiento, el análisis, la administración, el almacenamiento y la traslación de la información que se requiere para el mejoramiento de la salud de la población adulta y adulta mayor mexicana.

Al recopilar y analizar información de distintas fuentes es necesario automatizar los procesos de ordenamiento y análisis de información de diferentes indicadores, esto también porque constantemente se van obteniendo nuevos datos conforme pasa el tiempo y es necesario cuando se obtenga nueva información agregarla al SIESDE, de allí la importancia de contar con mecanismos automatizados que permitan agregar nuevos datos al SIESDE de manera rápida.

El proceso de automatización de indicadores consiste en la construcción de un código genérico que permita calcular, filtrar, ordenar y exportar los datos de un determinado indicador, de manera que se pueda subir el resultado obtenido a la base de datos sin la necesidad de realizar adecuaciones. El objetivo principal de construir el proceso automatizado para determinado indicador consiste en reutilizar el código para obtener resultados de otros años del mismo indicador cuando exista información que se encuentre en una estructura idéntica o muy similar que permita la reutilización del código construido y genere resultados de manera más rápida comparado con realizar el cálculo, filtrado, ordenamiento y exportación a la base de datos de manera manual.

Descarga e instalación del software de programación

Primero es necesario instalar el software requerido, en el caso de la automatización de los indicadores del SIESDE se utiliza Python, se recomienda instalar el software ANACONDA, pero se pueden utilizar otros.

Python es un lenguaje sencillo, rápido y liviano y es ideal para aprender, experimentar, practicar y trabajar con machine learning, ciencias de datos, redes neuronales y aprendizaje profundo entre otros.

Utilizaremos la Suite de Anaconda que nos facilitará la tarea de instalar el ambiente e incluirá las Jupyter Notebooks, esta Suite es multiplataforma y se puede utilizar para Windows, Linux y Macintosh.

1 – Descargar Anaconda

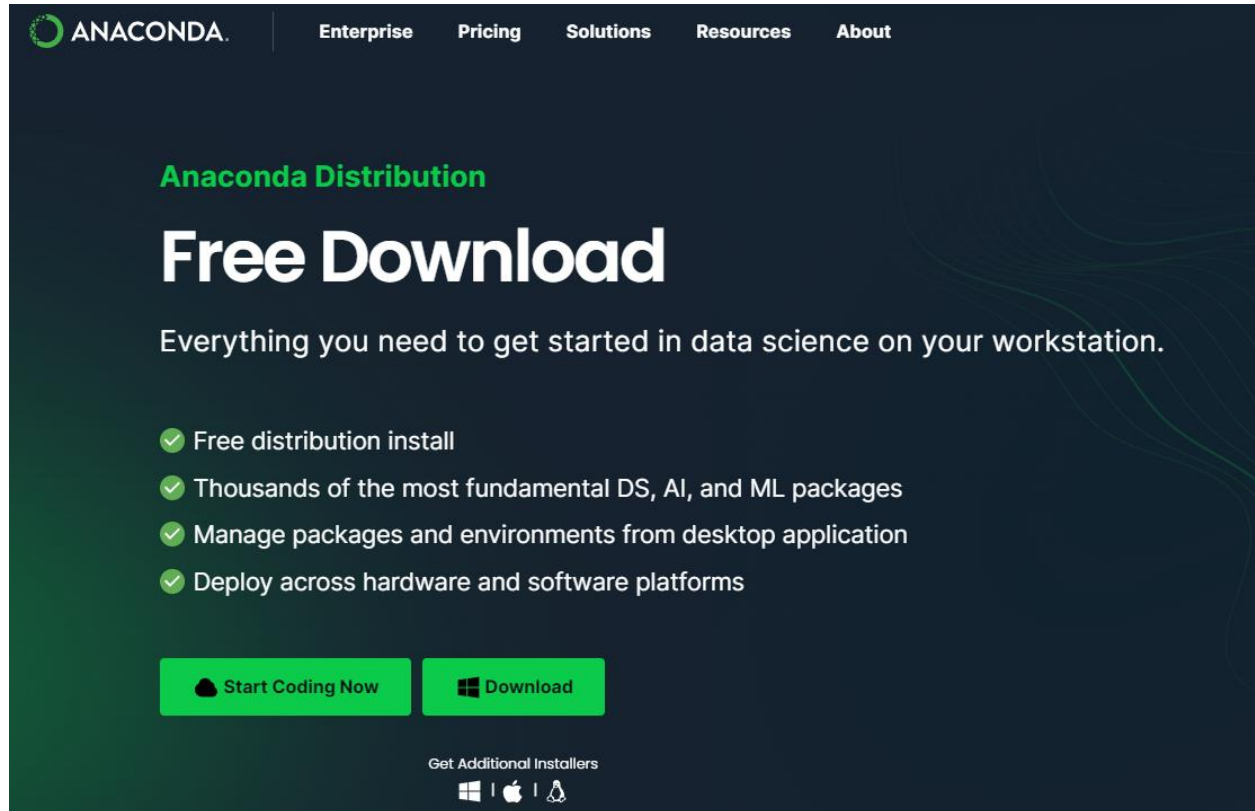
En este paso veremos como descargar anaconda a nuestro disco y obtener esta suite científica de Python



Nos dirigimos a la Home de Anaconda: <https://www.anaconda.com/>

e iremos a la sección de Download (descargas)

Elegimos nuestra plataforma: Windows, Mac o Linux



Con esto guardaremos en nuestro disco duro unos 620MB (según sistema operativo) del programa que se deberá instalar.

2 – Instalar Anaconda

En este paso instalaremos la app en nuestro sistema. (Deberá tener permisos de Administrador si instala para todos los usuarios).

Ejecutamos el archivo que descargamos haciendo doble click.

Se abrirá un “Típico Wizard” de instalación.

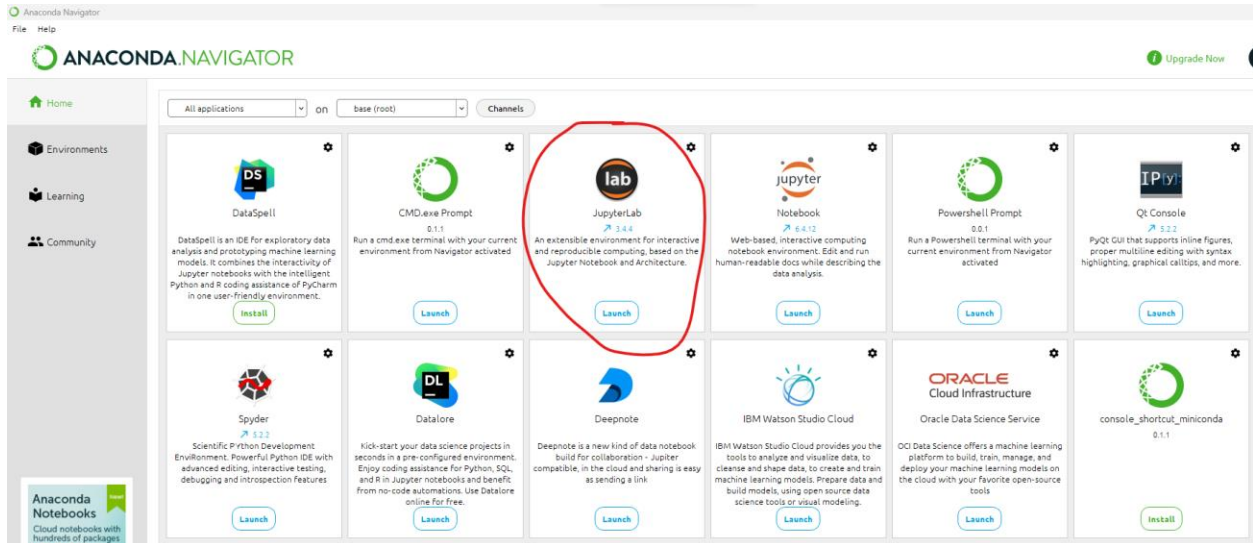
Seguiremos los pasos, podemos seleccionar instalación sólo para nuestro usuario, seleccionar la ruta en disco donde instalaremos y listo.

Al instalarse el tamaño total podrá superar 1Gb en disco.

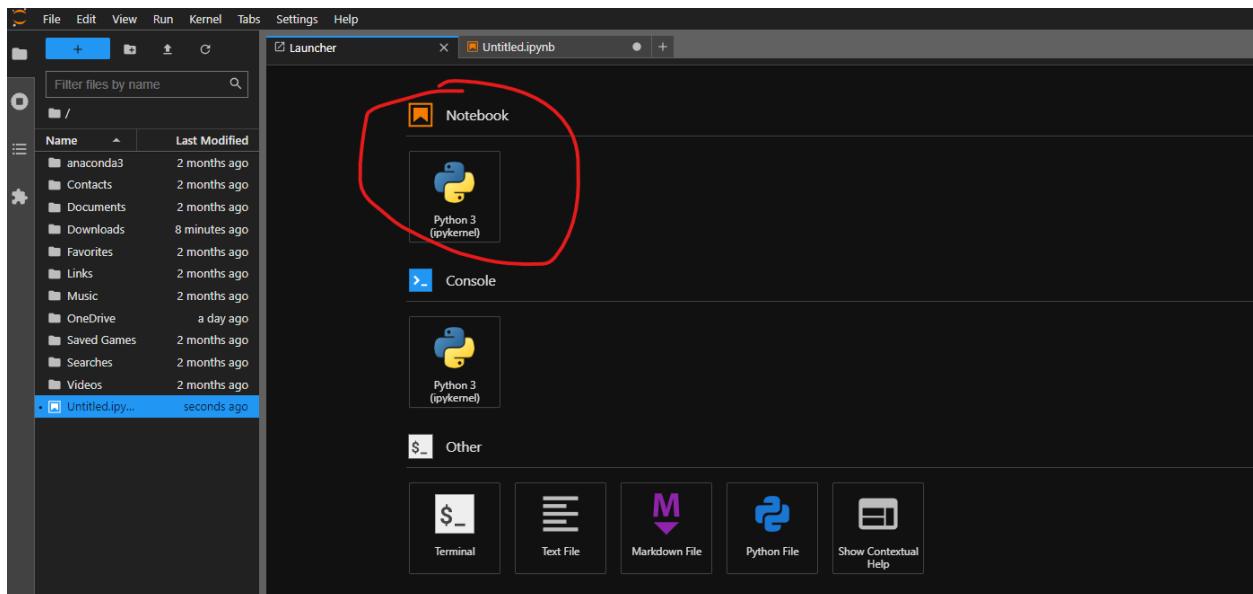
3 – Iniciar y Actualizar Anaconda

En este paso comprobaremos que se haya instalado correctamente y verificar tener la versión más reciente, es posible que pide actualizaciones.

Anaconda viene con una suite de herramientas gráficas llamada “Anaconda Navigator”. Iniciemos la aplicación y veremos una pantalla como esta:







Entre otros vemos que podemos lanzar las JupyterLab, el cual debemos iniciar. Posteriormente se abrirá en nuestro navegador el ambiente donde podemos agregar nuestro código en python.



EJEMPLO DE AUTOMATIZACION DEL INDICADOR 1.1.2

Para realizar la automatización es necesario descargar los datos en el caso del indicador 1.1.2 se descargan de las bases de datos del INEGI, (se omite el proceso de descarga de los datos puesto que para cada indicador es un proceso diferente y depende de la fuente de datos).

Una vez descargados los datos necesarios para generar el indicador (por año) en este caso se guardó un archivo correspondiente a los datos del 2020 y se guarda como "INEGI_exporta_23_3_2022_2020.xls".

Nombre	Fecha de modificación	Tipo	Tamaño
 INEGI_exporta_23_3_2022_2000.xls	23/03/2022 09:29 a. m.	Hoja de cálculo d...	1,590 KB
 INEGI_exporta_23_3_2022_2005.xls	23/03/2022 09:48 a. m.	Hoja de cálculo d...	1,599 KB
 INEGI_exporta_23_3_2022_2010.xls	23/03/2022 08:55 a. m.	Hoja de cálculo d...	856 KB
 INEGI_exporta_23_3_2022_2020.xls	23/03/2022 09:15 a. m.	Hoja de cálculo d...	862 KB

Lo siguiente es revisar los datos y definir el formato de salida que tendrán los datos para darles la estructura para que se puedan subir a la base de datos del INGER.

Población total																																						
Consulta de Población total Por: Entidad y municipio Según Sexo y Edad																																						
	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total	Total										
	De 0 a 4 años	De 5 a 9 años	De 10 a 14 años	De 15 a 19 años	De 20 a 24 años	De 25 a 29 años	De 30 a 34 años	De 35 a 39 años	De 40 a 44 años	De 45 a 49 años	De 50 a 54 años	De 55 a 59 años	De 60 a 64 años	De 65 a 69 años	De 70 a 74 años	De 75 a 79 años	De 80 a 84 años																					
1	Población total																																					
2																																						
3	Consulta de Población total Por: Entidad y municipio Según Sexo y Edad																																					
4																																						
5																																						
6																																						
7																																						
8	01	Agua Calientes	5,423,607	524,430	511,048	529,717	531,967	527,020	518,426	506,825	509,257	52,378	54,669	54,121	58,865	48,306	55,823	25,586	56,581	51,186																		
9	01	Agua Calientes	948,990	76,863	81,713	82,007	85,161	85,225	80,077	71,850	66,892	60,077	58,264	52,400	42,458	35,046	55,677	18,157	51,458	6,872																		
10	01	Ajontzen	51,536	5,421	5,473	5,372	5,282	4,817	5,972	5,170	5,288	5,008	2,720	2,799	1,797	3,166	5,048	891	578	427																		
11	01	Cavillo	58,250	5,524	5,638	5,558	5,363	4,792	5,348	5,933	5,496	5,103	5,884	5,850	5,288	5,188	4,831	5,463	5,025	584																		
12	01	Cosío	17,000	1,700	1,777	1,680	1,719	1,304	1,381	1,181	1,085	921	772	636	495	571	506	395	504	567																		
13	01	El Llano	20,853	2,112	2,070	2,087	2,029	1,961	1,709	1,479	1,322	1,314	1,188	969	810	596	469	282	559	242																		
14	01	Jesus María	529,929	52,219	53,214	52,870	52,851	51,469	50,188	50,882	50,049	50,049	50,086	50,244	50,370	50,277	50,556	50,244	50,044	50,553																		
15	01	Palmitón de Antega	47,446	4,744	4,768	4,645	4,743	4,279	4,928	4,476	4,894	4,163	4,740	4,504	4,254	4,740	4,504	4,066	4,733	504	567																	
16	01	Rincón de Romos	57,349	5,734	5,897	5,830	5,729	5,114	5,744	5,322	5,767	5,522	2,944	2,539	1,970	1,885	5,244	893	619	468																		
17	01	San Francisco de los Romo	61,897	6,189	6,257	6,545	6,864	6,473	6,444	6,225	4,788	4,048	5,229	4,391	4,604	5,382	593	622	573	515																		
18	01	San José de Gracia	5,532	1,068	1,031	954	920	817	812	655	587	522	486	534	500	375	500	342	502	502																		
19	01	Tepehual	22,485	2,248	2,219	2,219	2,208	2,019	1,883	1,572	1,389	1,321	1,058	945	812	582	570	582	521	485																		
20	01	Baja California	5,789,020	566,301	599,654	512,948	515,839	560,359	528,244	501,889	520,069	478,611	460,844	521,388	468,653	534,022	54,704	55,425	60,874	75,204																		
21	02	Ensenada	443,807	53,035	54,788	55,853	56,228	58,920	56,764	54,752	52,516	28,786	26,271	51,504	58,243	50,491	21,202	23,640	30,709	36,800																		
22	02	Mexicali	1,049,792	97,286	90,976	86,125	87,742	91,917	86,787	82,139	78,709	75,478	74,330	63,688	50,950	42,347	50,491	21,202	23,640	30,709																		
23	02	Payas de Rosamón	126,890	6,188	50,391	10,947	10,984	11,326	10,387	9,402	9,238	8,865	8,859	7,753	5,875	4,762	3,728	2,705	1,572	909																		
24	02	San Quintín	117,568	12,824	12,008	11,200	11,246	11,126	11,077	9,407	8,185	8,903	9,992	9,793	9,835	9,917	1,257	889	484																			
25	02	Tecate	108,440	8,302	8,481	8,814	8,714	8,939	8,821	8,375	8,384	8,991	7,949	8,437	8,200	8,869	8,869	2,039	1,237	886																		
26	02	Tijuana	5,923,233	536,586	553,212	559,349	560,753	578,751	574,408	568,134	559,043	441,858	458,328	512,412	511,331	511,676	511,331	495,179	514,039	514,039																		
27	03	Baja California Sur	798,447	62,455	66,933	67,665	65,469	66,743	68,779	64,927	69,389	51,771	63,337	53,731	57,095	58,274	52,545	6,166	6,449																			
28	03	Comondú	75,021	5,541	5,026	5,445	5,347	5,386	5,101	5,189	5,157	5,267	4,912	5,353	5,260	4,858	4,633	4,532	4,532	4,532																		
29	03	La Paz	292,241	29,241	29,098	29,334	29,651	29,652	29,009	29,950	29,371	31,520	34,828	27,797	34,855	31,158	4,495	6,254	6,254	6,254																		
30	03	Loreto	18,052	1,805	1,556	1,623	1,477	1,346	1,303	1,189	1,044	1,279	1,140	961	770	510	554	222	115																			
31	03	Los Cabos	351,111	30,814	31,581	31,118	28,831	31,513	34,085	30,057	31,160	27,020	21,829	16,380	11,063	7,886	3,327	3,144	1,954	978																		
32	03	Murietta	64,022	6,402	6,492	6,465	6,361	6,466	6,260	6,190	6,466	6,344	6,033	6,467	6,019	6,281	1,846	1,180	819	471																		
33	04	Campeche	928,363	78,835	82,015	80,168	76,140	75,556	72,798	70,275	70,275	54,170	52,628	50,254	40,805	52,778	53,796	17,559	11,916	5,536																		
34	04	Cárdenas	31,714	3,171	3,144	3,144	3,059	2,982	2,462	2,255	2,391	1,761	1,510	1,000	756	647	431	346	197																			
35	04	Culmún	59,212	5,921	5,268	5,129	4,924	4,883	4,772	4,687	4,542	4,045	3,148	2,853	2,408	1,314	1,892	1,257	512	652																		
36	04	Campeche	294,077	24,077	22,769	22,655	23,133	24,306	22,443	23,153	22,443	21,073	19,774	17,882	14,586	12,024	9,229	6,466	3,303	1,722																		
37	04	Camelaria	66,913	6,691	6,390	6,830	6,372	6,399	6,437	6,291	6,039	2,801	2,662	2,252	1,835	1,490	1,228	810	570	368																		
38	04	Comen	248,845	20,885	20,488	20,187	19,216	18,782	20,068	20,745	19,018	18,574	16,421	15,803	10,893	8,442	5,022	3,862	1,431	1,582																		
39	04	Champón	78,170	7,817	7,809	7,375	6,899	6,412	6,099	5,792	5,406	5,018	4,615	4,332	3,241	2,374	1,080	1,499	1,012	591																		
40	04	Escárcega	39,923	3,992	3,467	3,923	3,271	3,933	4,593	4,793	4,501	4,027	3,772	3,284	2,845	2,468	1,825	1,509	999	733	461																	
41	04	Hecicobá	51,817	5,181	4,888	4,954	4,821	4,811	4,509	4,511	4,202	4,070	3,504	3,599	3,040	2,652	2,068	1,578	668	269																		
42	04	Hopichén	62,140	6,214	6,375	6,373	6,938	6,833	6,208	5,084	4,618	4,784	4,824	4,175	3,116	2,152	1,799	999	563	552																		
43	04	Palizada	6,683	780	790	872	892	862	901	577	573	573	569	524	406	302	287	228	108	108																		
44	04	Seyé	15,297	1,529	1,275	1,274	1,399	1,119	1,119	1,119	1,076	1,076	1,018	923	632	638	637	448	265	144																		
45	04	Tenabo	11,452	1,145	1,052	1,122	916	904	901	901	904	904	904	904	904	904	904	904	904	904	904																	
46	05	Coahuila de Zaragoza	5,146,771	779,912	776,224	788,144	789,284	784,313	745,917	724,329	719,634	618,687	505,302	418,214	342,215	218,950	164,422	91,684	39,801	24,851																		
hoja																																						

Los datos se descargan como los guarda la fuente de datos y se define el orden de los datos de salida y los cálculos que se deben hacer para llegar a ellos, (consultar en los metadatos del indicador 1.1.2 como se calculan los datos de este indicador).

	A	B	C	D	E	F	G	H	I	J
1		id_ent	id_mun	d_gpoedad	entidad	municipio	gpoedad	valor	periodo	
2	0	0	0	0	Estados U	Total	Total	95.24758	2020	
3	1	1	0	0	Aguascalie	Aguascalie	Total	95.57691	2020	
4	2	1	1	0	Aguascalie	Aguascalie	Total	94.89769	2020	
5	3	1	2	0	Aguascalie	Asientos	Total	96.14082	2020	
6	4	1	3	0	Aguascalie	Calvillo	Total	96.21383	2020	
7	5	1	4	0	Aguascalie	Cosío	Total	95.22278	2020	
8	6	1	10	0	Aguascalie	El Llano	Total	100.3747	2020	
9	7	1	5	0	Aguascalie	Jesús Mar	Total	97.73094	2020	
10	8	1	6	0	Aguascalie	Pabellón	Total	96.32453	2020	
11	9	1	7	0	Aguascalie	Rincón de	Total	96.01271	2020	
12	10	1	11	0	Aguascalie	San Franci	Total	98.12412	2020	
13	11	1	8	0	Aguascalie	San José d	Total	90.27888	2020	
14	12	1	9	0	Aguascalie	Tepezalá	Total	97.73986	2020	

Después de calcular una muestra pequeña se procede a programar en Python el cálculo de todos los demás datos.

Regresando al navegador para programar el manejo, calculo y ordenamiento del indicador, es necesario como requisito previo instalar las dos librerías que se utilizan: 1. Pandas 2. Numpy

```

File Edit View Run Kernel Tabs Settings Help
Untitled2.ipynb x Untitled1.ipynb x +
Code
[1]: pip install pandas
Requirement already satisfied: pandas in c:\users\fc\anaconda3\lib\site-packages (1.4.4)
Requirement already satisfied: numpy>=1.18.5 in c:\users\fc\anaconda3\lib\site-packages (from pandas) (1.21.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\fc\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\fc\anaconda3\lib\site-packages (from pandas) (2022.1)
Requirement already satisfied: six>=1.5 in c:\users\fc\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

[2]: pip install numpy
Requirement already satisfied: numpy in c:\users\fc\anaconda3\lib\site-packages (1.21.5)
Note: you may need to restart the kernel to use updated packages.

[ ]:

```

Para instalar pandas se introduce el comando “pip install pandas” y se ejecuta.

Para instalar numpy se introduce el comando “pip install numpy” y se ejecuta.

Lo siguiente es programar las instrucciones para abrir el archivo descargado y comenzar con los cálculos, modificaciones, ordenamiento y posterior exportación de resultados.

A continuación, se pone el código en Python comentado para la generación del indicador 1.1.2:

```

import pandas as pd
import numpy as np

#-----
#-----
# MODIFICA EL AÑO DE LA VARIABLE periodo Y EL NOMBRE DEL ARCHIVO A CARGAR
periodo=2020 #QUE AÑO
#nombreArchivo="Población total 2020.xlsx" #archivo para generar el indicador
1.1.2
#nombreArchivo="INEGI_exporta_23_3_2022_2020.xls" #archivo para generar el
indicador 1.1.2
nombreArchivo="INEGI_exporta_23_3_2022_2020.xls" #archivo para generar el
indicador 1.1.2
#-----
#-----

df1= pd.read_excel(nombreArchivo, header=None) #lee el excel sin poner
nombres de columnas
df1=df1.dropna(subset=[1]) # elimina las filas donde la columna 1 sea vacio
df1.rename(columns={0:"entidad"}, inplace=True) # renombrar columna 0
df1["entidad"].fillna("00 000", inplace=True)# agregar en el total 00
df=df1["entidad"].str.split(' ', expand=True) # dividir columna 0 estado
municipio en dos columnas
df.rename(columns={0:"id_ent"}, inplace=True) # renombrar columna 0
df.rename(columns={1:"id_mun"}, inplace=True) # renombrar columna 1
df1=pd.concat([df,df1], axis=1) #Unir las nuevas columnas
df1["id_ent"].fillna(df1["entidad"], inplace=True) #ponemos el valor de
id_lugar en los espacios NaN (vacios)
df1["id_mun"].fillna("0", inplace=True) #ponemos 000 de id_municipio en los
espacios NaN (vacios)
df1.rename(columns={1:"municipio"}, inplace=True) # renombrar columna 1
df1["id_ent"] = df1["id_ent"].astype(int)
df1["id_mun"] = df1["id_mun"].astype(int)

df1["entidad"]=df1["id_ent"]# PASAMOS EL ID ESTADO A ESTADO

df1["entidad"]=df1["entidad"].map({0:"Estados Unidos Mexicanos",
1:"Aguascalientes", 2:"Baja California", 3:"Baja California Sur",
4:"Campeche",5:"Coahuila de Zaragoza",6:"Colima",7:"Chiapas",
8:"Chihuahua",9:"Ciudad de
México",10:"Durango",11:"Guanajuato",12:"Guerrero",13:"Hidalgo",14:"Jalisco",
15:"México",16:"Michoacán de Ocampo",17:"Morelos",
18:"Nayarit",19:"Nuevo
León",20:"Oaxaca",21:"Puebla",22:"Querétaro",23:"Quintana Roo",24:"San Luis
Potosí",25:"Sinaloa",26:"Sonora",27:"Tabasco",28:"Tamaulipas",
29:"Tlaxcala",30:"Veracruz de Ignacio de
la Llave",31:"Yucatán",32:"Zacatecas"
}, na_action=None) ## CAMBIAMOS EL ID
POR EL NOMBRE DEL ESTADO

df1.reset_index(inplace=True, drop=True)
df1=df1.fillna("0") # reemplazar los valores vacios con 0 para evitar las
sumas vacias

```

```

### HAY QUE GENERAR GRUPOS QUE FALTAN PARA 2005 y 2000 ---REQUISITO---
DESCARGAR LAS BASE DE DATOS DESPLEGANDO DE 75 EN ADELANTE LOS GRUPOS
QUINQUENALES QUEDADO AÑO POR AÑO QUEDANDO EL EXCEL HASTA LA COLUMNA ED LA
ULTIMA
#SI DE GENENRAR GRUPOS QUE FALTAN PARA 2005 y 2000
if (periodo==2005 or periodo==2000):
    for i in range (2,134): # convertir columnas seleccionadas de string a
int
        df1[i] = pd.to_numeric(df1[i].str.replace(',',' '),errors='ignore')
# GENERANDO LOS GRUPOS QUINQUENALES QUE LE FALTAN
#TOTALES
col=18
df1[col]=0
for i in range (19,24): # De 75 a 79
    df1[col]=df1[col]+df1[i]

col=19
df1[col]=0
for i in range (24,29): # De 80 a 84 años
    df1[col]=df1[col]+df1[i]

col=20
df1[col]=0
for i in range (29,45): # 85 años y más
    df1[col]=df1[col]+df1[i]

for i in range (21,45): # borra las demas columnas que no se ocupan
del(df1[i])

for i in range (45,134):
    df1.rename(columns={i:i-24}, inplace=True) # renombrar columns
# HOMBRES
col=38
df1[col]=0
for i in range (39,44): # De 75 a 79
    df1[col]=df1[col]+df1[i]

col=39
df1[col]=0
for i in range (44,49): # De 80 a 84 años
    df1[col]=df1[col]+df1[i]

col=40
df1[col]=0
for i in range (49,65): # 85 años y más
    df1[col]=df1[col]+df1[i]

for i in range (41,65): # borra las demas columnas que no se ocupan
del(df1[i])

for i in range (65,110):
    df1.rename(columns={i:i-24}, inplace=True) # renombrar columns
# MUJERES

```

```

col=58
df1[col]=0
for i in range (59,64): # De 75 a 79
    df1[col]=df1[col]+df1[i]

col=59
df1[col]=0
for i in range (64,69): # De 80 a 84 años
    df1[col]=df1[col]+df1[i]

col=60
df1[col]=0
for i in range (69,85): # 85 años y más
    df1[col]=df1[col]+df1[i]

for i in range (61,85): # borra las demas columnas que no se ocupan
    del(df1[i])

df1.rename(columns={85:61}, inplace=True) # renombrar columnns

#FIN_SI DE GENENRAR GRUPOS QUE FALTAN PARA 2005
# si es 2020 o 2010 no hay que hacer mas
else:
    for i in range (2,62): # convertir columnas seleccionadas de string a int
        df1[i] = pd.to_numeric(df1[i].str.replace(',',''),errors='ignore')
# fin_sino que es 2020 o 2010 no hay que hacer mas
#-----
# Hasta aqui se limpian los datos para ser usados para el calculo de los
indicadores
df2 =pd.concat([df1], axis=1) #variable para indicador
df3 =pd.concat([df1], axis=1) ##variable para indicador
#####
#####

## IDICADOR 1.1.2
for i in range (0,20): # hace los calculos
100*hombres[rangoedad]/mujeres[rangoedad]  indicador 1.1.2
    df1[62+i]= 100*df1[22+i]/df1[42+i]
#GRANDES GRUPOS
df1[2]=0
df1[3]=0
for i in range (23,27): # 0-19
    df1[2] = df1[2]+df1[i]
    df1[3] = df1[3]+df1[i+20]

df1[82]= 100*df1[2]/df1[3] # 0-19

df1[2]=0
df1[3]=0
for i in range (27,41): #20+
    df1[2] = df1[2]+df1[i]
    df1[3] = df1[3]+df1[i+20]

```



```

df1[83]= 100*df1[2]/df1[3] #20+

df1[2]=0
df1[3]=0
for i in range (27,35): #20-59
    df1[2] = df1[2]+df1[i]
    df1[3] = df1[3]+df1[i+20]

df1[84]= 100*df1[2]/df1[3] #20-59

df1[2]=0
df1[3]=0
for i in range (35,41): #60+
    df1[2] = df1[2]+df1[i]
    df1[3] = df1[3]+df1[i+20]

df1[85]= 100*df1[2]/df1[3] #60+

df1[2]=0
df1[3]=0
for i in range (38,41): #75+
    df1[2] = df1[2]+df1[i]
    df1[3] = df1[3]+df1[i+20]

df1[86]= 100*df1[2]/df1[3] #75+

df1[2]=0
df1[3]=0
for i in range (39,41): #80+
    df1[2] = df1[2]+df1[i]
    df1[3] = df1[3]+df1[i+20]

df1[87]= 100*df1[2]/df1[3] #80+

for i in range (2,62): # borra las demas columnas que no se ocupan indicador
1.1.2
    del(df1[i])

for i in range (62,88):
    df1.rename(columns={i:i-62}, inplace=True) # renombrar columns

df1= df1.melt(id_vars = ["id_ent","id_mun","entidad","municipio"], # Columns
to keep (not gathered)
            value_vars =
[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25], #
Columns to gather
            var_name = "id_gpoedad", # Name of new
gathered column

```

```

        value_name = "valor")                                # Name of new value
column

df1["gpoedad"]=df1["id_gpoedad"]# PASAMOS EL ID grupo edad a dpoedad










df1["gpoedad"]=df1["gpoedad"].map({0:"Total", 1:"00-04 años", 2:"05-09 años",
3:"10-14 años", 4:"15-19 años",5:"20-24 años",6:"25-29 años",7:"30-34 años",
8:"35-39 años",9:"40-44 años",10:"45-49
años",11:"50-54 años",12:"55-59 años",13:"60-64 años",14:"65-69 años",15:"70-
74 años",16:"75-79 años",17:"80-84 años",
18:"85 años y más",19:"No
especificado",20:"00-19 años",21:"20 años y más",22:"20-59 años",23:"60 años
y más",24:"75 años y más",25:"80 años y más"
}, na_action=None) ## CAMBIAMOS EL ID POR
EL NOMBRE DEL grupoedad

df1 =
df1.reindex(columns=["id_ent","id_mun","id_gpoedad","entidad","municipio","gp
oedad","valor"])

df1["periodo"]=periodo
df1.to_excel("auto"+str(periodo)+".1.1.2.xlsx")# Guardamos la tabla en excel
indicador 1.1.2
df1.to_csv("auto"+str(periodo)+".1.1.2.csv",sep=";")# Guardamos la tabla en
csv indicador 1.1.2
df1

```

Una vez construido el código de la automatización, (para cada indicador es diferente), se ejecuta y va a generar los archivos de salida (generalmente en csv y si es posible en xlsx tambien).

Nombre	Fecha de modificación	Tipo	Tamaño
 auto112.py	19/04/2022 11:15 a. m.	Archivo PY	9 KB
 auto2000.1.1.2.csv	08/08/2022 04:32 p. m.	Archivo de valores...	4,426 KB
 auto2000.1.1.2.xlsx	19/04/2022 11:11 a. m.	Hoja de cálculo d...	3,008 KB
 auto2005.1.1.2.csv	08/08/2022 04:33 p. m.	Archivo de valores...	4,446 KB
 auto2005.1.1.2.xlsx	19/04/2022 11:12 a. m.	Hoja de cálculo d...	3,032 KB
 auto2010.1.1.2.csv	08/08/2022 04:33 p. m.	Archivo de valores...	4,452 KB
 auto2010.1.1.2.xlsx	19/04/2022 11:12 a. m.	Hoja de cálculo d...	3,034 KB
 auto2020.1.1.2.csv	08/08/2022 04:34 p. m.	Archivo de valores...	4,471 KB
 auto2020.1.1.2.xlsx	19/04/2022 11:13 a. m.	Hoja de cálculo d...	3,048 KB

Posteriormente se comparan los resultados automatizados con los resultados que ya se tenían (generalmente del último año). Una vez comparados los resultados se procede a calcular los datos del indicador para otros años obteniendo los resultados mucho más rápido.

Finalmente, los nuevos resultados se suben a la base de datos del SIESDE a la tabla correspondiente al indicador que ya se tenía y se utiliza el archivo en csv para ello, con eso se finaliza el proceso de automatizar un indicador del indicador 1.1.2 del SIESDE.

Para otros indicadores el proceso es similar aunque en el caso de algunos se requieren pasos intermedios para obtener los resultados.