



UNIVERSIDAD DE COLIMA



PROGRAMA QUE CALCULA EL DESNIVEL ENTRE DOS PUNTOS REFERIDOS A UNA SUPERFICIE

Autores de proyecto:

Avalos Rocha Andrea

Jiménez Vargas Mariel del Rosario

Uribe García Leonardo Miguel

Institución:

Facultad de Ingeniería Civil, Universidad de Colima, Campus Coquimatlán

Carrera:

Ingeniero Topógrafo Geomático

Grado y grupo:

3ºB

Profesor titular:

Sebastián González Zepeda

Lugar:

Campus Coquimatlán; Coquimatlán, Colima

Fecha:

03 de diciembre de 2022



Programa que calcula el desnivel entre dos puntos referidos a una superficie.

Program that calculates the unevenness between two points referred to a surface.

Andrea Avalos Rocha (1), Mariel del Rosario Jiménez Vargas (2), Leonardo Miguel Uribe García (3)

(1) Facultad de Ingeniería Civil, 1604, Colima-Coquimatlán Kilómetro 9, Jardines del Llano, 28400, Coquimatlán, Col.(1)20186380,aavalos9@ucol.mx, (2) 20186524 mjimenez@ucol.mx (3) 20186673 luri-be0@ucol.mx

Resumen

Se generará un programa en colab el cual tiene como propósito leer un archivo de Excel, en el que se encuentran datos de una nivelación previamente realizada del campus Coquimatlán en la Facultad de Ingeniería Civil, para obtener como resultado que el programa genere de manera automática el desnivel obtenido, también para un mayor entendimiento visual graficar los resultados.

Palabras clave: Programa, Nivelación y Desniveles.

1. Introducción

En el presente reporte se habla acerca de la obtención de desniveles, llevado a cabo a través de una nivelación previamente realizada con los datos necesarios para su desarrollo. Con la finalidad de generar un programa en Colab que nos arroje de manera automática los resultados del desnivel del terreno. Para entrar en conceptualización, cuando hablamos de desnivel, hacemos referencia a la diferencia de altura entre dos o más puntos. El desnivel también se puede definir como la diferencia de elevación o cota entre ambos puntos.

Dentro de este reporte se mencionan diversos apartados, tales como objetivo general, objetivos específicos, estado del arte; en el que se incluye marco histórico, marco conceptual, marco teórico, marco metodológico y marco de referencia, así mismo, se mencionan las herramientas a utilizar, la metodología empleada, resultados, conclusiones y finalmente, la bibliografía, todos los mencionados con la finalidad de que tanto como el reporte como el trabajo realizado en campo, sea elaborado de la mejor manera.

La realización de este proyecto principalmente, tiene como propósito generar un programa que ayude a respaldar a las personas enfocadas al

Abstrac

A program will be generated in colab whose purpose is to read an Excel file, in which there is data from a previously carried out leveling of the Coquimatlán campus in the Faculty of Civil Engineering, to obtain as a result that the program automatically generates the unevenness obtained, also for a better visual understanding graph the results.

Keywords: Program, Leveling and Slopes.

campo de la topografía, ya que este programa ayudará a ahorrar tiempo al momento de realizar cálculos de este tipo. De igual forma, el desarrollo de este proyecto permite aplicar los conocimientos adquiridos en la materia de Proyecto Integrador, así como la de Topografía General II.

Dicho todo esto, daremos inicio al reporte comenzando con el objetivo general de este proyecto.

2. Desarrollo

Materiales

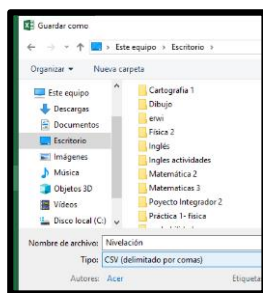
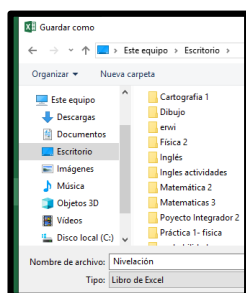
Para poder llevar a cabo este proyecto, se hizo uso de los siguientes materiales:

- Computadora con acceso a Internet
- Google Colab (Google Colaboratory)
- Excel
- Datos obtenidos de una nivelación

Primeramente, se pasan los archivos obtenidos de una nivelación topográfica de campo a un archivo Excel.

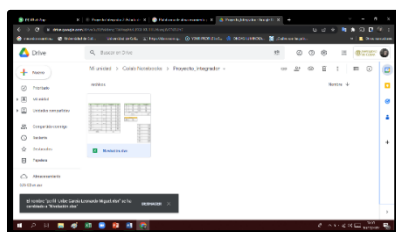
Figura 20. Datos de nivelación a Excel

En seguida de ello, se guarda el archivo en su equipo. Se recomienda que este sea guardado en formato .csv o .xlsx.



Figuras 21 y 22. Guardar en formato csv o xlsx

Una vez teniendo estos, se exportan a Google Drive. De tal manera que se guardó en la carpeta especial de Google Colab para así facilitar los procesos de búsqueda del archivo al momento de querer abrirlo en Colab.



Figuras 23. Archivo en Google Drive

Una vez teniendo nuestro archivo Excel en Google Drive con los datos de la nivelación, se comienza a programar en Colab de tal manera que abra y lea los archivos de Excel. Para ello, se emplearon comandos como **“from google.colab import drive”**, **“from google.colab import data_table”** y **“import xlrd”**.

```
1 from google.colab import drive
2 from google.colab import data_table
3 from vega_datasets import data
4 import pandas as pd
5 import numpy as np
6 import xlrd
7 import matplotlib.pyplot as plt
8 from matplotlib.ticker import PercentFormatter
9 import seaborn as sb
10
11 data_table.enable_dataframe_formatter()
12 data.supports()
13 drive.mount('/gdrive')
14 df = pd.read_excel('/gdrive/MyDrive/Colab Notebooks/Proyecto Integrador/Nivelación.xlsx', sheet_name = "Hoja1")
15 print(df.head())
16
```

Figura 24. Comandos para abrir y leer el dataframe

Con el código hecho hasta ese momento nos debe de mostrar la tabla.

	P.V	Lect(+)	A.I	Lect(-)	Cota
0	DN1	1.765	358.5248	0.000	356.7598
1	PL1	0.749	357.2008	2.073	356.4518
2	PL2	1.347	356.9028	1.645	355.5558
3	PL3	1.107	355.4188	2.591	354.3118
4	BN2	0.000	NaN	1.532	353.8868

Figura 25. Tabla mostrada

Una vez ya leído y mostrado la tabla, se obtiene la suma de las columnas de lecturas positivas (**‘Lect(+)’**) y de lecturas negativas (**‘Lect(-)’**), lo cual, para poder obtener estas primeramente se accede a las columnas específicas que se desea hacer la operación del *dataframe*.

Luego de ello aplicamos el comando *sum()* a cada una de las columnas a las que accedimos. Este comando calcula la suma de todos los elementos de un *array* sobre el eje especificado.

```
16
17 columna_lectpositivas = df['Lect(+)]
18 sumlecturaspositivas= sum(columna_lectpositivas)
19 print('La suma de las lecturas positivas es: ', sumlecturaspositivas)
20
21 columna_lectnegativas = df['Lect(-)']
22 sumlecturasnegativas= sum(columna_lectnegativas)
23 print('La suma de las lecturas negativas es: ', sumlecturasnegativas)
```

Figura 26. Suma de columnas

De esta manera obtenemos la suma de la columna de lecturas positivas y la columna de lecturas negativas.

La suma de las lecturas positivas es: 4.968
La suma de las lecturas negativas es: 7.841

Figura 27. Resultados de la suma

Una vez teniendo el resultado de la suma de cada columna, se obtiene el desnivel, el cual simplemente se realiza una resta de las lecturas positivas con las lecturas negativas.

```
5 desnivel= sumlecturaspositivas-sumlecturasnegativas
6 print('El desnivel de esta nivelación es: ', desnivel)
7
```

Figura 28. Desnivel de las lecturas

Con el comando **print()**, se muestra el desnivel.

```
El desnivel de esta nivelación es: -2.873
```

Figura 29. Resultado del desnivel

Ahora bien, con todo lo anterior ya hecho, se realiza la gráfica de cómo es la diferencia de desnivel en dicho terreno. Para ello se hace empleo de librería **matplotlib**.

Primeramente, se definió los valores del **dataframe** (archivo Excel) que se desean graficar. En este caso, las cotas y los P.V. En toda nivelación, para poder graficar y mostrar el perfil de cómo es la pendiente durante todo un tramo del terreno, se debe de establecer las cotas y los puntos que poseen dichas cotas.

```
29 valores= df[['P.V','Cota']]
```

Figura 30. Definiendo valores para graficar

Luego de ello, se especifica cual irá en el eje X y eje Y. En el eje X irá los P.V y en el eje Y las cotas.

```
30 ax= valores.plot(x='P.V',y='Cota',rot= 0)
31 plt.xlabel("P.V")
32 plt.ylabel("Cota")
33 plt.title("Nivelación Geométrica- Perfil del tramo")
34 plt.show()
35
```

Figura 31. Proceso de graficación del desnivel

De esta manera se obtiene una representación gráfica de cómo es el desnivel de los datos de la nivelación.

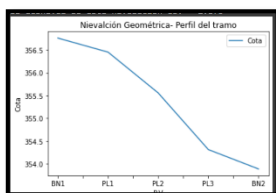


Figura 32. Representación gráfica del desnivel

Ahora bien, se utilizó la condición **If** para determinar si la pendiente del desnivel es creciente, decreciente o si el terreno es plano.

```
12
13 if desnivel > 0:
14     print('El desnivel va de forma creciente')
15 if desnivel < 0:
16     print('El desnivel va de forma decreciente')
17 if desnivel == 0:
18     print('No hay desnivel, el terreno es plano')
19
```

Figura 33. Introducción de la condición If

Con las condiciones establecidas, cuando se ejecuta el programa se imprime que la pendiente es decreciente, pues el desnivel que hubo es menor a 0 (-2.873).

```
La pendiente va de forma decreciente
```

Figura 34. Uso de la condición If

Ahora bien, luego se usó el módulo **time**, **math**, **geopy** y **seaborn** para poder obtener la ubicación en tiempo real con sus coordenadas de dónde se realizó la nivelación topográfica.

```
60 geolocator = Nominatim(user_agent="AppMap")
61 lugar= input("Lugar donde se hizo la nivelación: ")
62 location = geolocator.geocode(lugar)
63 print(location)
64 print((location.latitude, location.longitude))
65
```

Figura 35. Código para especificar el lugar

```
La pendiente va de forma decreciente
Lugar donde se hizo la nivelación: Coquimatlán
```

Figura 35. Especificación del lugar

```
Lugar donde se hizo la nivelación: Coquimatlán
Coquimatlán, Colima, México
(19.1759779, -103.92391335718723)
```

Figura 36. Datos de la ubicación

Finalmente se usó el módulo **basemap** para mostrar un mapa de escala global en donde marca en que parte del mundo se encuentra realizada la nivelación topográfica.

```
65 geo = Nominatim(user_agent="AppMap", timeout=2)
66 zona = geo.geocode(lugar)
67 plt.figure(figsize=(16,12))
68 my_map=Basemap(projection="robin", lon_0=0,lat_0=0)
69 my_map.drawcoastlines()
70 my_map.drawcountries()
71 my_map.fillcontinents(color='green')
72 x,y = my_map(zona.longitude,zona.latitude)
73 my_map.plot(x,y,color='red', marker="o",markersize="15")
74 plt.title("Ubicación donde se hizo la nivelación")
75
```

Figura 37. Uso de basemap

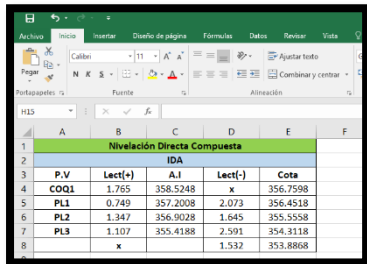
Y así el código nos muestra lo siguiente:



Figura 38. Mapa

3.- Manejo de datos

Por parte de los datos que se emplearon se encuentran toda la información de Excel obtenida de una nivelación topográfica, en donde se ordenaron por Punto Visado (P.V), Lecturas Positivas (Lect (+)), Lecturas Negativas (Lect (-)), Altura del instrumento (A.I.) y Cotas.



	P.V	Lect(+)	A.I	Lect(-)	Cota
COQ1	1.765	358.5248	x	356.7598	
PL1	0.749	357.2008	2.073	356.4518	
PL2	1.347	356.9028	1.645	355.5558	
PL3	1.107	355.4188	2.591	354.3118	
	x		1.532	353.8868	

Figura 39. Datos de Nivelación Topográfica

Ahora bien, por parte de los datos para programar, se emplearon librerías y módulos tales como **pandas**, **numpy**, **xrld** y **matplotlib**, en donde:

Pandas: Nos ayudó en el manejo y análisis de estructuras de datos. Con ella pudimos manipular nuestros datos del dataframe (Excel) para poder calcular lo esperado, así como poder trabajar con los datos a manera de tabla.

Numpy: Nos ayudó en el cálculo de operaciones matemáticas de nuestros datos a partir de arreglos o matrices. Cabe destacar que con esta librería nos permitió usar funciones con las cuales pudimos ejecutar operaciones aritméticas.

Xrld: Nos permitió analizar y trabajar con datos de archivos Excel a manera de tablas.

Matplotlib: Esta librería nos permitió crear y personalizar gráficos a partir de los datos que se tienen de nuestro archivo Excel

From google.colab import drive: Este código nos permitió acceder a Google Drive para así poder abrir la información que guardamos en nuestra nube y poder usarla en Google Colab, en este caso, nuestro archivo Excel.

Seaborn: Aunque sea una biblioteca similar a Matplotlib, esta nos sirvió para poder graficar a detalles datos e información geoespacial.

Time: Este módulo nos sirvió para trabajar con información geoespacial a escalar actual.

Math: Este módulo nos proporcionó acceso a las funciones matemáticas definidas por el estándar C.

Basemap: No sirvió para poder proyectar mapas

Como recomendación, sugerimos que al utilizar el programa realizado por nosotros se debe de tener antes muy bien ordenada la información de nuestro dataframe (Excel) pues solo se podrá ejecutar de la manera correcta si se cuenta con la información ordenada en columnas y cada una de ellas con los nombres correspondientes (P.V, Lect(+), A.I., Lect (-) y cotas).

Cabe destacar que es importante tener instalado la librería **basemap** con **pip install** pues si esta no está instalada, el programa nos marcará error.

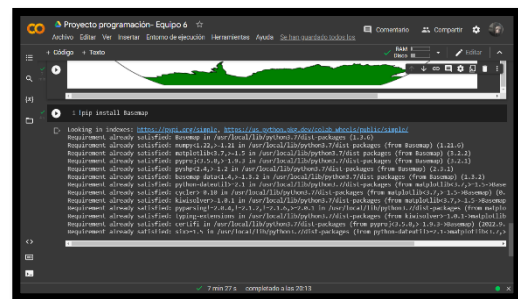


Figura 40. Basemap instalado

4.- Resultados

Respecto a los resultados, podemos decir que, si se cumplieron con los objetivos establecidos en nuestro proyecto, es decir, se pudo programar y se obtuvo el desnivel que hay entre dos bancos de nivel al igual que también se logró representar el desnivel a manera de una gráfica de barras, en donde se especifica a detalle como es el comportamiento de los datos en cada estación o punto visado. Así mismo, el código elaborado muestra el lugar, las coordenadas y un mapa con la zona donde se realizó la nivelación.

De igual manera, con el desarrollo de este proyecto se pudo comprobar que la nivelación realizada en campo fue realizada correctamente, pues con el análisis de los datos obtenidos a la hora de programar pudimos colaborar nuestras hipótesis sobre cómo es el desnivel que existe desde un banco de nivel a otro.

El código realizado fue el siguiente:

```
#Universidad de Colima- Facultad de In
#Carrera: Ingeniero Topógrafo Geomático
#Elaborado por: Avalos Rocha Andrea, J
#Jiménez Vargas Mariel del Rosario y Uri
#García Leonardo Miguel
#Grado y grupo: 3ºB
#Programación de computadoras II
```

```
%matplotlib inline
from google.colab import drive
from google.colab import data_table
from vega_datasets import data
import pandas as pd
import numpy as np
import xlrd
import matplotlib.pyplot as plt
from matplotlib.ticker import PercentFormatter
import seaborn as sb
from mpl_toolkits.basemap.test import Basemap
from geopy.geocoders import Nominatim
import time
import math

print('--
Programa que calcula el desnivel entre dos puntos referidos a una superficie--')
print('Bienvenido a nuestro programa, este te ayudará a conocer el desnivel de una nivelación topográfica y te mostrará de forma gráfica como es esta')
print('IMPORTANTE: Para que el programa te funcione debes tener tu información en Excel separada con columnas y cada columna con los nombres: P.V, Lect(+), (A.I), Lect(-) y Cotas')
print('P.V significa Punto Visado')
print('A.I significa Altura del instrumento')

data_table.enable_dataframe_formatter()
data.airports()
drive.mount('/gdrive')
df= pd.read_excel('/gdrive/MyDrive/Colab Notebooks/Proyecto Integrador/Nivelación.xlsx', sheet_name = "Hoja2")
print(df.head())

columna_lectpostivas = df['Lect(+)']
```

```
sumlecturaspositivas= sum(columna_lectpostivas)
print('La suma de las lecturas positivas es: ', sumlecturaspositivas)

columna_lectnegativas = df['Lect(-)']
sumlecturasnegativas= sum(columna_lectnegativas)
print('La suma de las lecturas negativas es: ', sumlecturasnegativas)

desnivel= sumlecturaspositivas-sumlecturasnegativas
print('El desnivel de esta nivelación es: ', desnivel)

valores= df[['P.V','Cota']]
ax= valores.plot(x='P.V',y='Cota',rot=0)
plt.xlabel("P.V")
plt.ylabel("Cota")
plt.title("Nivelación Geométrica- Perfil del tramo")
plt.show()

if desnivel > 0:
    print('La pendiente va de forma creciente')
if desnivel < 0:
    print('La pendiente va de forma decreciente')
if desnivel == 0:
    print('No hay pendiente, el terreno es plano')

geolocator = Nominatim(user_agent="AppMap")
lugar= input("Lugar donde se hizo la nivelación: ")
location = geolocator.geocode(lugar)
print(location)
print((location.latitude, location.longitude))
geo = Nominatim(user_agent="AppMap", timeout=2)
zona = geo.geocode(lugar)
plt.figure(figsize=(16,12))
my_map=Basemap(projection="robin", lon_0=0,lat_0=0)
```



```
my_map.drawcoastlines()
my_map.drawcountries()
my_map.fillcontinents(color='green'
)
x,y = my_map(zona.longitude,zona.la
titude)
my_map.plot(x,y,color='red', marker
="o",markersize="15")
plt.title("Ubicación donde se hizo
la nivelación")
```

Link del código en colab:

https://colab.research.google.com/drive/1C6E2Dd3u2N2XMu5XNNS4_6UzthRyFEGr?usp=sharing

5.- Conclusiones

A manera de conclusión, podemos decir que nuestro proyecto concluyó de la manera en que esperábamos, el código que desarrollamos aplicado al campo de la Topografía y Geomática funciona, arrojándonos el valor del desnivel entre dos bancos de nivel. De igual forma, el código nos muestra información geoespacial, la cual es muy importante en la ingeniería debido a que los datos geoespaciales ayudan a describir eventos con ciertas características con una ubicación en la superficie terrestre, por lo general, estos datos geoespaciales combinan información de ubicación (en este caso coordenadas del lugar donde se realizó la nivelación) e información que posee las características del evento que se realizó.

Consideramos que la programación podría llegar a ser muy útil y necesaria en nuestra vida laboral como futuros ingenieros, ya que este es un gran tutor cuando se trata de aprender a solucionar problemas, principalmente situaciones de lógica e ingenio. Es el arte de crear soluciones desde cero a paradigmas complejos a base de un pensamiento estructurado, lógico y creativo.

Por ello, no cabe duda alguna que la programación juega un papel muy útil en la vida humana, y en las ingenierías no es la excepción, pues con ella podemos realizar varios procesos u operaciones las cuales manualmente nos tomaría bastante tiempo realizarlas, por lo que con el uso de programas informáticos podemos efectuar y ejecutar procesos de manera más rápida, precisa y eficaz, tal y como lo pudimos demostrar en este proyecto, pues a hacer uso de lenguajes informáticos de programación, pudimos facilitar operaciones y procesos con los que tardamos más y nos ayudó a ahorrarnos mucho tiempo de trabajo.

Para finalizar, el realizar este código nos deja una gran enseñanza, pues ejemplifica cómo es que podemos emplear la programación o los lenguajes de programación para resolver cuestiones de la vida diaria de cualquier tipo, principalmente enfocados en la vida estudiantil, aplicando los conocimientos adquiridos durante el curso de Programación de Computadoras II.

6. Referencias

Aws(2022). "¿Qué es Python?", Amazon.com. Recuperado el 31 de octubre de 2022, de <https://aws.amazon.com/es/what-is/python/>

Brinker, W. (1997). Topografía. Ed. Alfaomega. 9ª Edición. Colombia.

Cientec instrumentos. (2022). ¿Cómo usar un Nivel Topográfico? 01 de septiembre de 2022, de Cientec instrumentos Sitio web: <https://cientecinstrumentos.cl/como-usar-un-nivel-topografico/>

Gabri,P(17 de octubre de 2018). ArcGeek, recuperado de <https://acolita.com/evolucion-de-las-tecnicas-topograficas/>

Torres N., Villate B. (2001). Topografía. Ed. Escuela Colombiana de Ingeniería. 4a edición. Colombia

Apéndice

Utilice esta página y las siguientes para situar tablas y figuras de tamaño superior al ancho de la columna, u otros materiales. Si no la utiliza, borre esta sección.