

“Obtención de un plano a partir de datos obtenidos en un levantamiento planimétrico mediante un programa en python”.

Autores: Felipe de Jesús Duran Luna, Sánchez Cabellos Arturo
Facultad de Ingeniería Civil - ITG
km. 9 carretera Colima-Coquimatlán
Coquimatlán, Colima, México.
C.P. 28400
Programación II
fduran@ucol.mx, arturo_sanchez@ucol.mx



UNIVERSIDAD DE COLIMA



Objetivos:

Realizar un programa en Python que facilite un poco mas el trabajo de gabinete que se realiza después de un levantamiento planimétrico. Esto con la finalidad de obtener la representación grafica del polígono levantado junto con su cuadro de construcción.

Código fuente

Métodos:

Para la elaboración del programa fue necesario importar las librerías correspondientes como lo son numpy que contiene las funciones matemáticas; pandas que tiene herramientas para el manejo de la información, además de leer y crear archivos; dxfwrite para la elaboración de archivos dxf. “Se abrevia cada módulo para que el código del programa sea más simple.”

```
import numpy as np
import pandas as pd
import dxfwrite as dxf
```

Librerías utilizadas

```
def cal_az(num):
    if num > 360: return(num-360)
    else: return(num)
```

Función que define si un ángulo es mayor a 360 grados.

```
def cal_pr(d,az):
    Px = d*np.sin(np.radians(az))
    Py = d*np.cos(np.radians(az))
    return(Px,Py)
```

Creamos una función que calcule la proyección en X y en Y.

```
def cal_rb(gr):
    rm = []
    for i in gr:
        grd = int(i),int((i-int(i))*60),int(((i-int(i))*60-int((i-int(i))*60))*60)
        if i < 90: rm.append("%i %i %i NE" % (grd[0],grd[1],grd[2]))
        if i >= 90 and i < 180: rm.append("%i %i %i SE" % (grd[0]-90,grd[1],grd[2]))
        if i >= 180 and i < 270: rm.append("%i %i %i SW" % (grd[0]-180,grd[1],grd[2]))
        if i >= 270 and i < 360: rm.append("%i %i %i NW" % (grd[0]-270,grd[1],grd[2]))
    return pd.DataFrame(rm,columns=["Rumbo"])
```

```
def cal_er(grd):
    err_A = (len(grd)-2)*180 - sum(grd)
    err_A = int(err_A*3600)
    comp = grd + err_A/len(grd)
    return comp,err_A
```

Definimos una función que obtenga la condición angular.

Elaboramos dos funciones que obtengan el rumbo a partir de cierta información.

```
def cal_cd(xy,rd=None):
    plano = dxf.drawing("plano.dxf")
    plano.add_layer("PA", color=3, linetype="DASHED")
    plano.add(dxf.polyline(xy,layer="pa"))
    if rd == None:
        ar = np.array(xy).T
        cx = (max(ar[0]) + min(ar[0]))/2
        cy = (max(ar[1]) + min(ar[1]))/2
        dx = (max(ar[0]) - cx)/0.0025
        dy = (max(ar[1]) - cy)/0.100
    else:
        ar = np.array(rd).T
        plano.add_layer("PR", color=2)
        plano.add(dxf.polyline(rd,layer="PR"))
        cx = (max(ar[0]) + min(ar[0]))/2
        cy = (max(ar[1]) + min(ar[1]))/2
        dx = (max(ar[0]) - cx)/0.0025
        dy = (max(ar[1]) - cy)/0.100
    med = np.array((0.0025,0.008,0.032,0.100))*cal_es(max(dx,dy))
    plano.add(dxf.rectangle((cx-med[0],cy-med[3]),(cx+med[0],cy+med[3]*2))
    plano.add(dxf.rectangle((cx+med[1],cy-med[3]),(cx+med[1],cy+med[3]*2))
    plano.save()
```

Para la elaboración del plano se definió una función donde recibe como parámetros las coordenadas de la poligonal de apoyo.

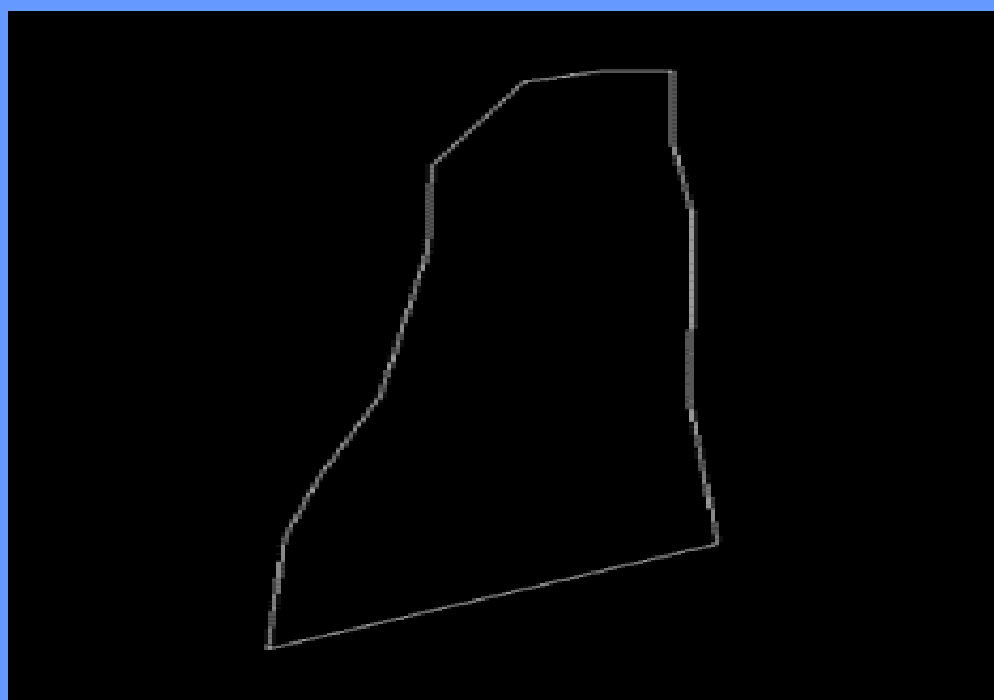
Resultados:



Polígono creado por el programa:

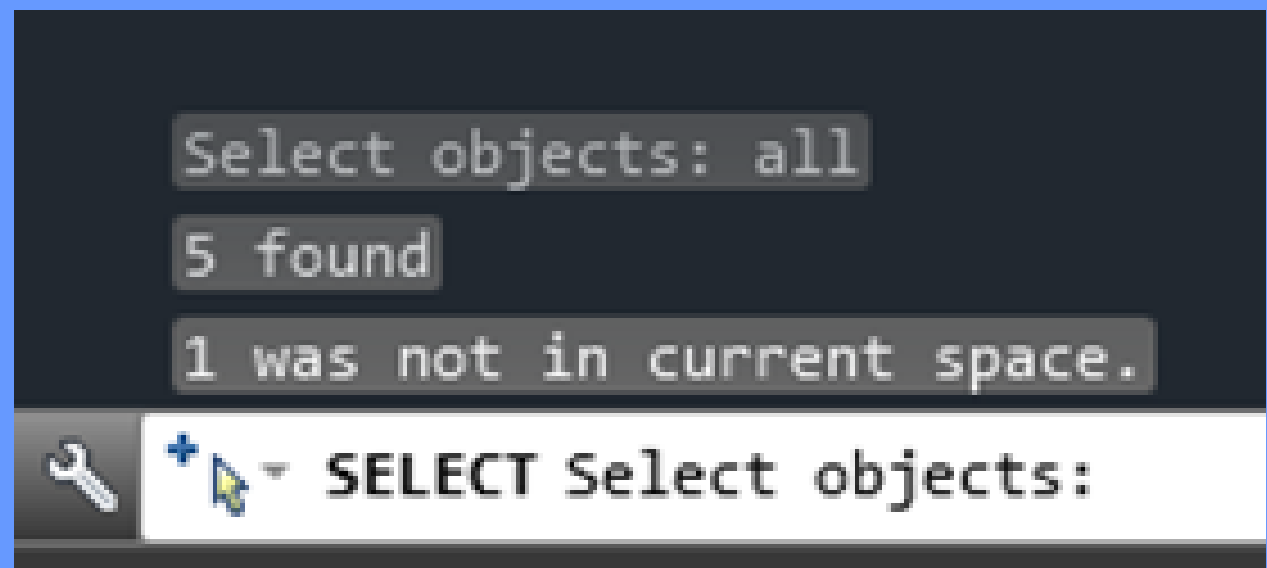


Polígono original:



Como resultado se obtuvo una parte de lo que contiene el plano de escrituración, los cuales son: el dibujo del polígono ya sea de forma directa o por radiaciones, el dibujo de la poligonal de apoyo (si es por radiaciones), el cuadro que encierra el dibujo del polígono y el cuadro donde se muestra la información del polígono. Cada uno se tiene una capa definida para tener un mejor control en el manejo de las propiedades de cada uno.

Como conclusión del presente proyecto, se podría decir que se cumplió con el objetivo que era elaborar el polígono y el cuadro de construcción, el único detalle fue que en AutoCAD el polígono se crea y se ve a simple vista, sin embargo el cuadro de construcción no aparece, ya que al digitar el comando “select” y posteriormente digitar “all”, el programa arroja el siguiente mensaje: Como mejora se tendría que revisar en donde hay error del por qué no arroja el cuadro de construcción como un objeto dentro de ese espacio. A pesar de eso el programa si crea un cuadro de construcción solamente que en un formato csv.



Conclusiones:

Referencias:

<http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>
<https://www.campusmvp.es/recursos/post/los-diez-lenguajes-de-programacion-de-2017.aspx>
<https://lolap.wordpress.com/2015/01/14/que-es-un-csv-como-se-hace-y-para-que-sirve/>
<http://www.ictea.com/cs/knowledgebase.php?action=displayarticle&id=8817>
<https://github.com/burissanz/Ordinario->