# **Programa para la obtención de rumbos en base a dos coordenadas geográficas (x, y) dadas.**

Article template and editorial instructions for practices

Cielo Jazziris Campos García, Brayan Alexis Aguilar Gutierrez

Universidad de Colima, Campus Coquimatlán, Facultad de Ingeniería Civil, 28400, baguilar12@ucol.mx,  
ccampos@ucol.mx

Resumen

Se creará un programa computacional en base al lenguaje Python que nos ayude a obtener los rumbos directos de un levantamiento topográfico en base a Coordenadas Geográficas, para agilizar de esta manera en primera instancia la realización de las prácticas y a futuro nuestro trabajo.

**Palabras clave**: Coordenadas, Coordenadas  
Geográficas, Rumbos, Sistemas de coordenadas.

Abstract

A computer program based on the Python language will be created to help us obtain the geographic directions of a topographic survey based on geographical coordinates to accelerate the realization of the practices and future work in the first instance.

**Keywords**: Coordinates, Geographical coordinates, Directions, Coordinate systems.

## Introducción

El uso de las tecnologías de información es de  
gran ayuda para el Ingeniero Topógrafo  
Geomático, por ende, es necesario crear un  
programa que agilice algunos procesos que  
utilizan dichos ingenieros. La Geomática aplica  
las tecnologías de la información y la  
comunicación en las geociencias, lo cual hace  
posible la captura, procesamiento y análisis de  
información digital geoespacial.  
En ello intervienen las coordenadas geográficas  
y el análisis de estas, gracias a las coordenadas  
podemos obtener mucha información que a un  
Ingeniero Topógrafo Geomático le es de mucha  
utilidad, como por ejemplo los rumbos de  
construcción de un polígono, los azimuts, las  
distancias entre distintos puntos, áreas de  
polígonos, entre otros.  
De ahí se genera el objetivo principal del  
proyecto que es el crear un programa que ayude  
al Ingeniero Topógrafo Geomático, y no solo a  
él, a todo el que lo necesite, a obtener  
automatizadamente los rumbos directos con l  
introducción de coordenadas geográficas  
cartesianas.

## Función del Programa

El funcionamiento de nuestro programa se basa en la lectura de coordenadas geográficas (x,y) para posteriormente con una fórmula matemática que involucra la tangente inversa (arctan).

### Funcionamiento Teórico

El programa abre y lee el archivo de texto en extensión csv (archivo delimitado por comas) con los datos registraos en un orden especifico; Punto o vértice, las coordenadas en X y las coordenadas en Y, en ese orden. Cada uno de los datos debe registrase en forma de columna, en la que cada coma define donde termina una columna y donde comienza la otra, ejemplo:

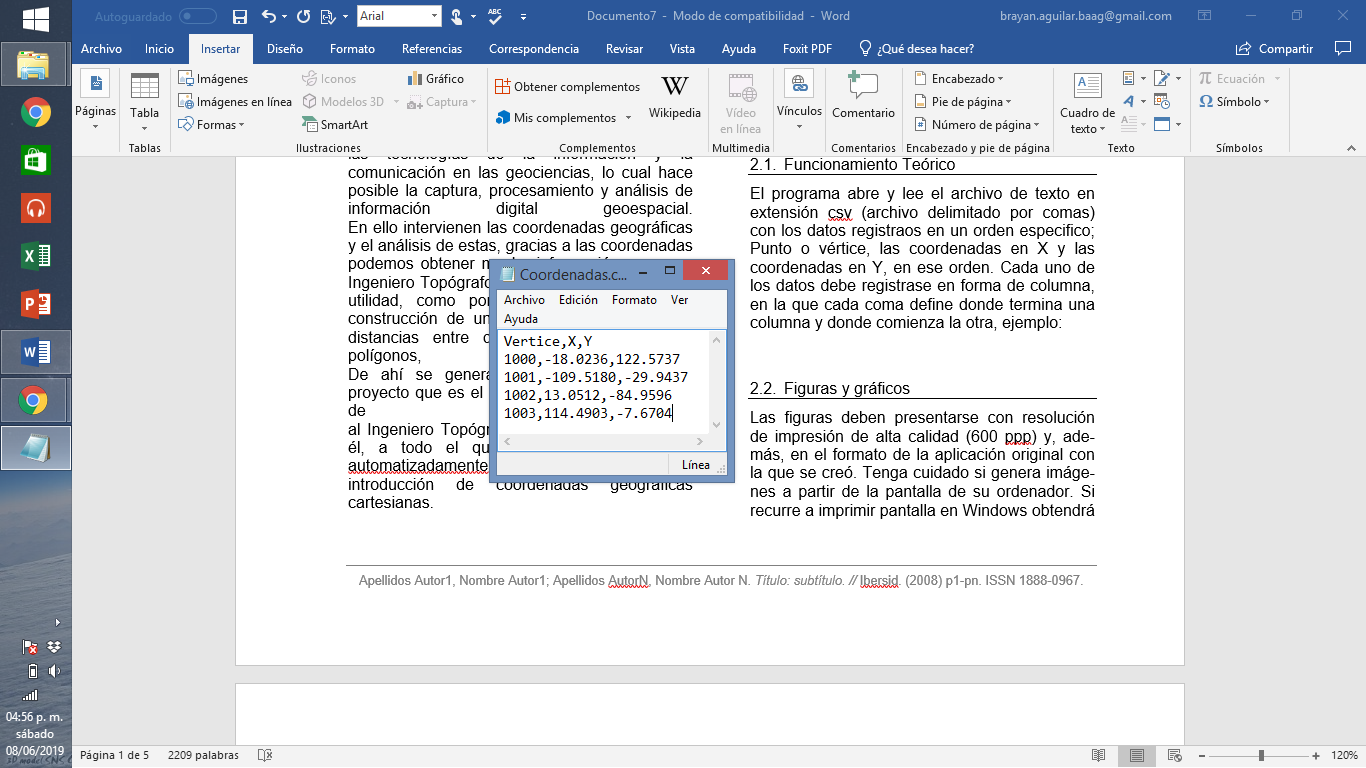


Figura 1. Registro de datos

El programa lee línea por línea y guarda la información de cada columna en una lista. Con esa información guarda en la variable, trabajará el programa para realizar las operaciones necesarias.

### Funcionamiento Gráfico

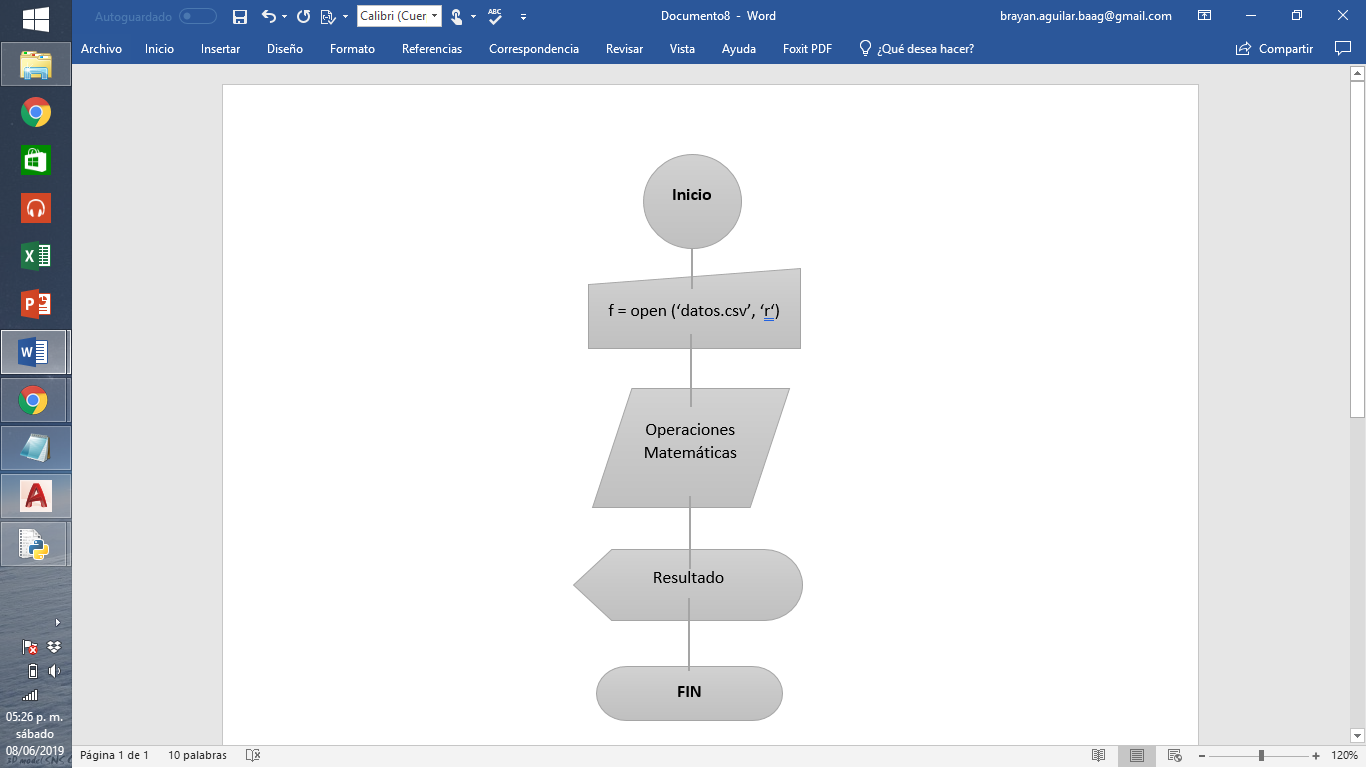


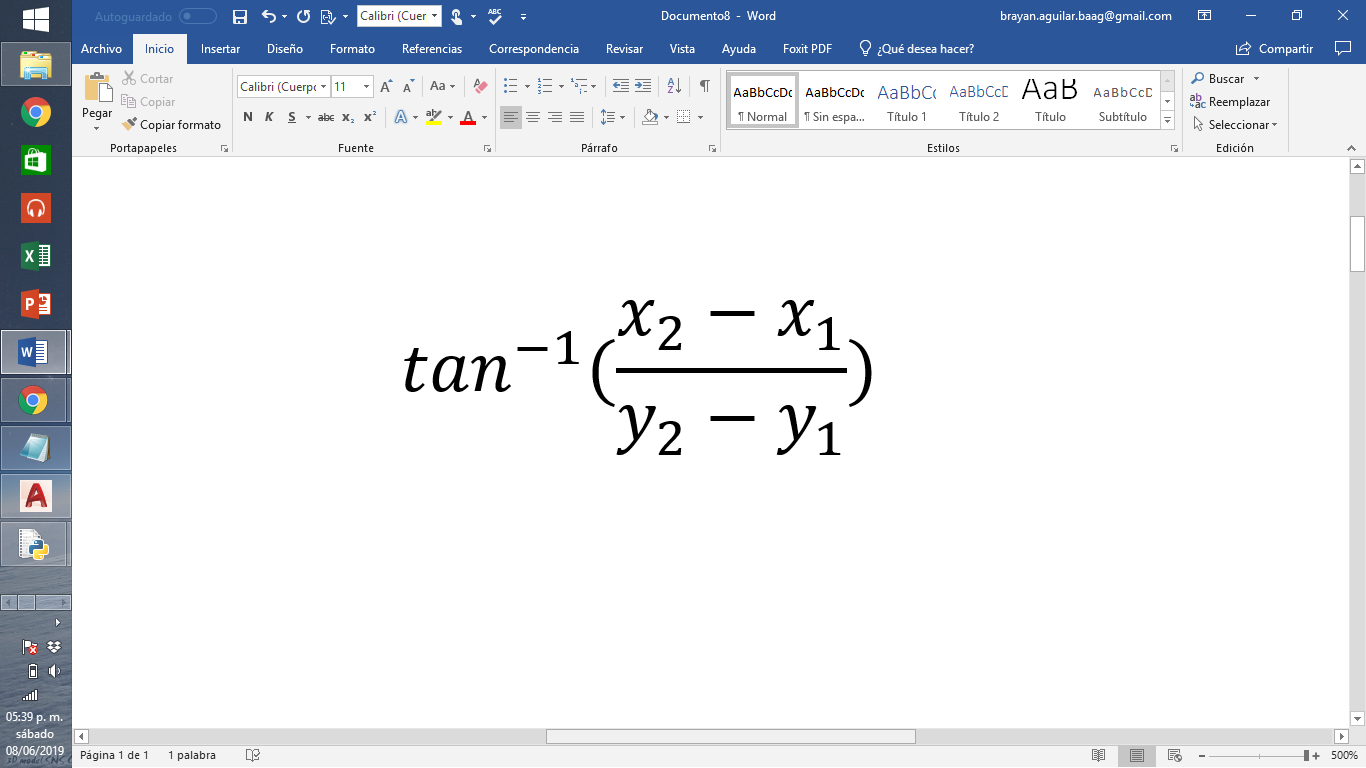
Figura 2. Diagrama de Flujo del proceso general del programa.

El diagrama de flujo anterior muestra de una manera general el funcionamiento de nuestro programa.

### Cálculos matemáticos

Los cálculos matemáticos que se necesitaron para desarrollar el programa fueron los siguientes, cabe señalar que fue indispensable ir guardando los datos en variables para realizar operaciones independientes.

**Fórmula:**



**Proceso:**

* *r = ((x2 – x1)/(y2-y1))*

(Aquí estamos guardando la operación en una variable)

* *tg = math.atan(r1)*

(En esta variable obtenemos la tangente inversa de la operación)

* *grados1 = math.degrees(tg1)*

*(*Cabe señalar que si lo dejamos como está anteriormente, el resultado estará en radianes, así que es necesario convertirlo a grados*)*

### Estructura

En la Figura 3 se muestran los códigos para la importanción de las librerias que necesita el programa para funcionar.

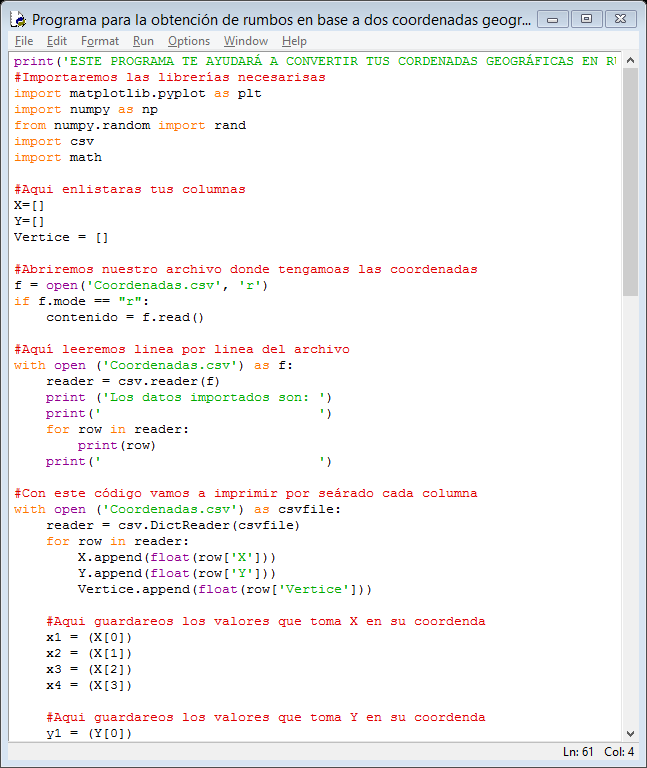


Figura 3. Importación de las librerías.

La librería de Matplotlib nos sirve para graficar los datos que se introducirán, la librería Numpy, es necesaria para el manejo y aplicación de las forulas matemáticas, math se necesita para integrar las funciones trigonométricas que se manejarán.

El siguiente código es para definir las variables como listas con la ayuda de los corchetes para que enliste columna por columna.

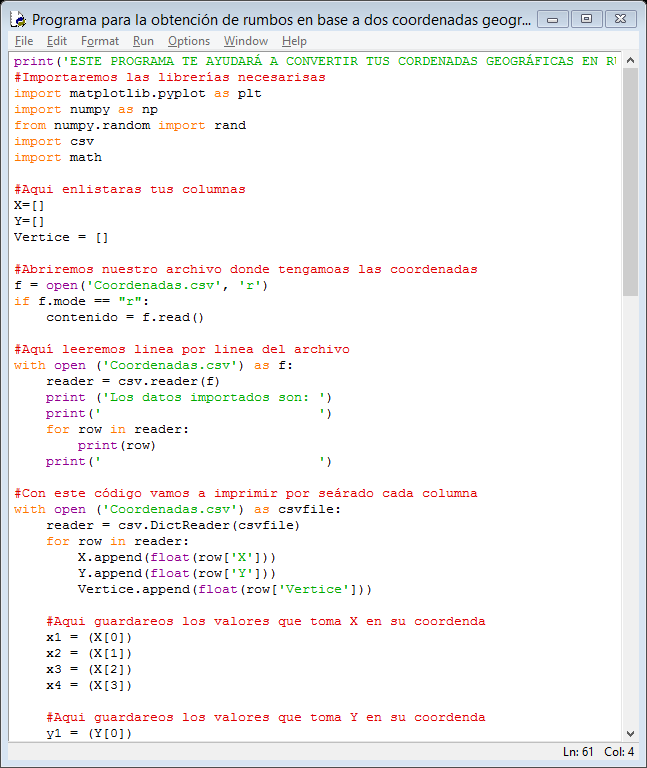


Figura 4. Definir las variables como listas.

En este código el programa está pidiendo al usuario que digite el nombre de su programa y a continuación lo delimita como csv, la función de input se utiliza para que se introduzca un lenguaje.

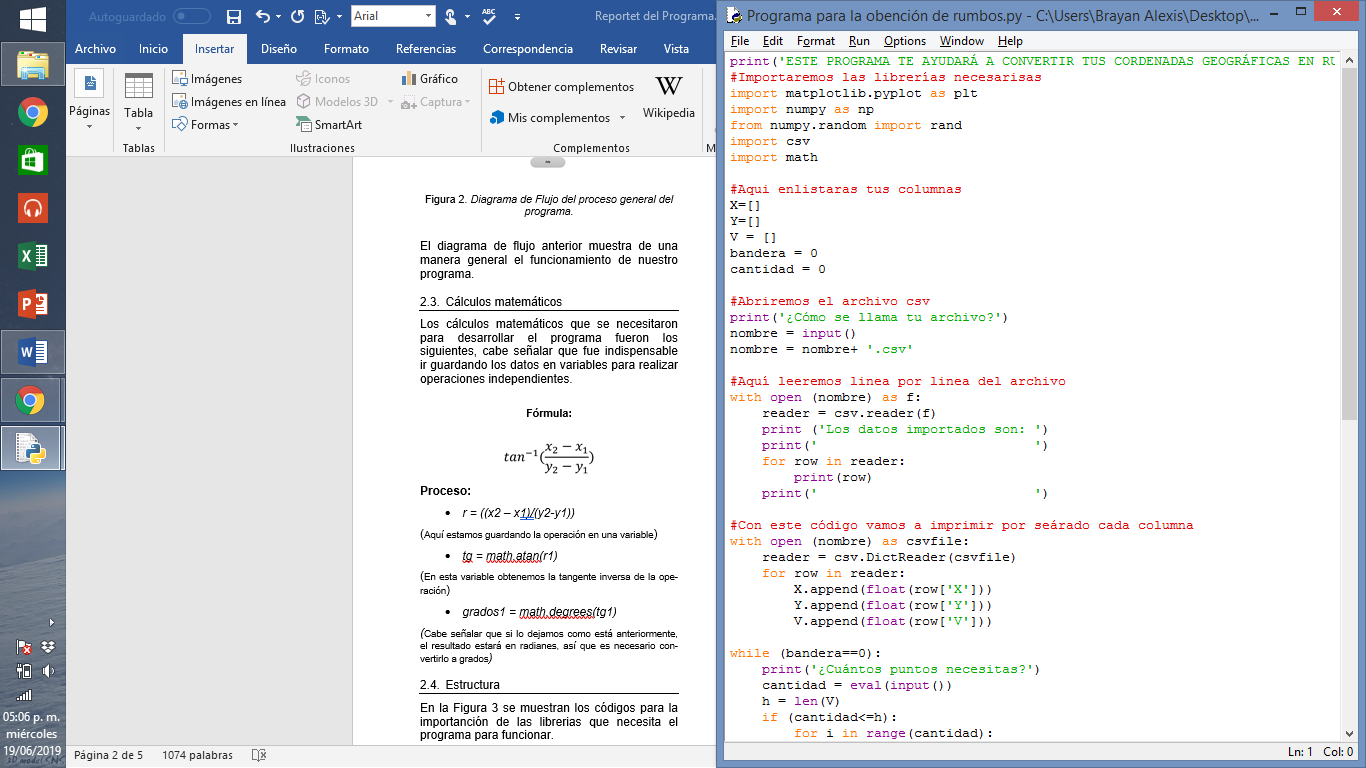


Figura 5. Identificación del archivo

Ahora se incluye un ciclo para que el programa lea el archivo ya abierto línea por línea:

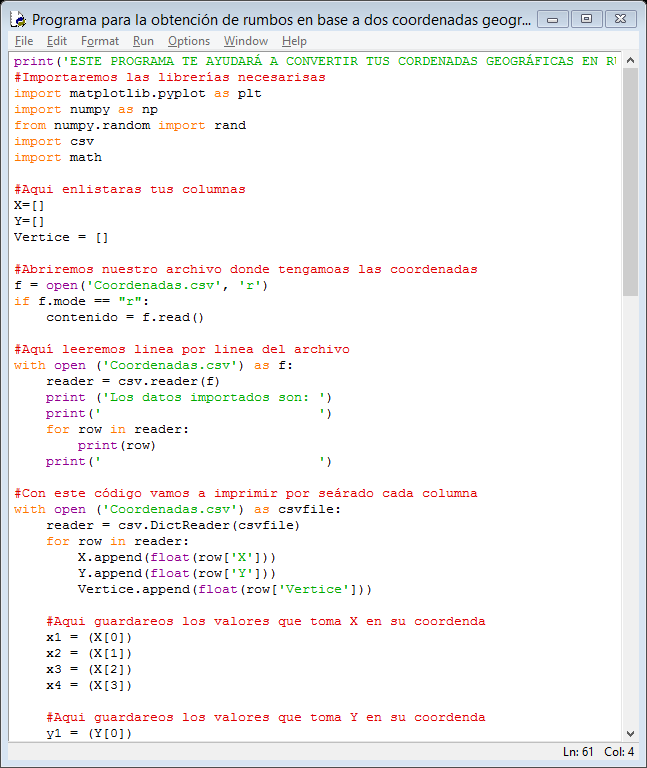


Figura 6. Lectura del archivo línea por línea

Con el siguiente ciclo, vamos a guardadar la columna que le pertenece a cada variable (X, Y, Vertice) para posteriormente se maneje como lista para poder allicar las operaciones.

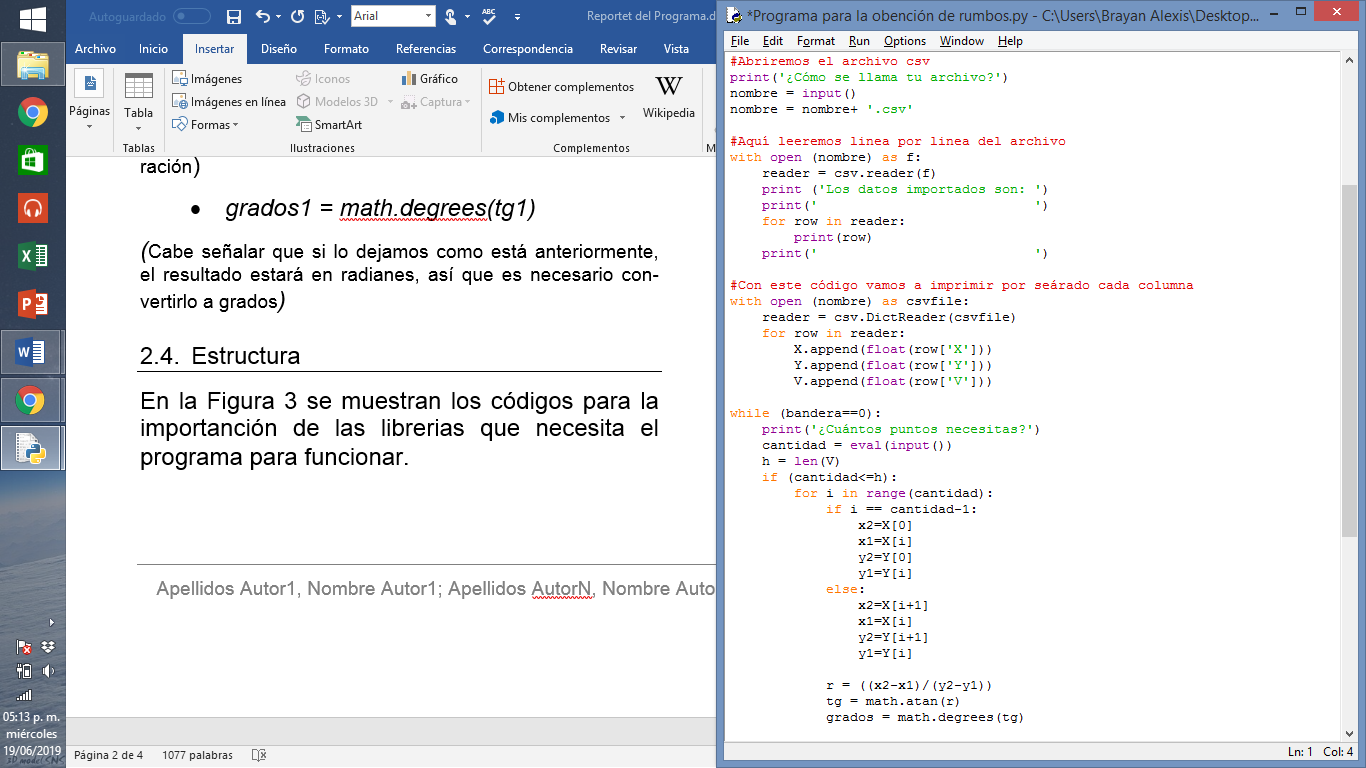


Figura 7: Guardado de las columnas en variables.

A continuación asiganaremos variables para guardar las columnas y definir sus posiciones en la lista.

La funcion del ciclo for nos ayuda a ingresar y maipular una cantidad n de variables

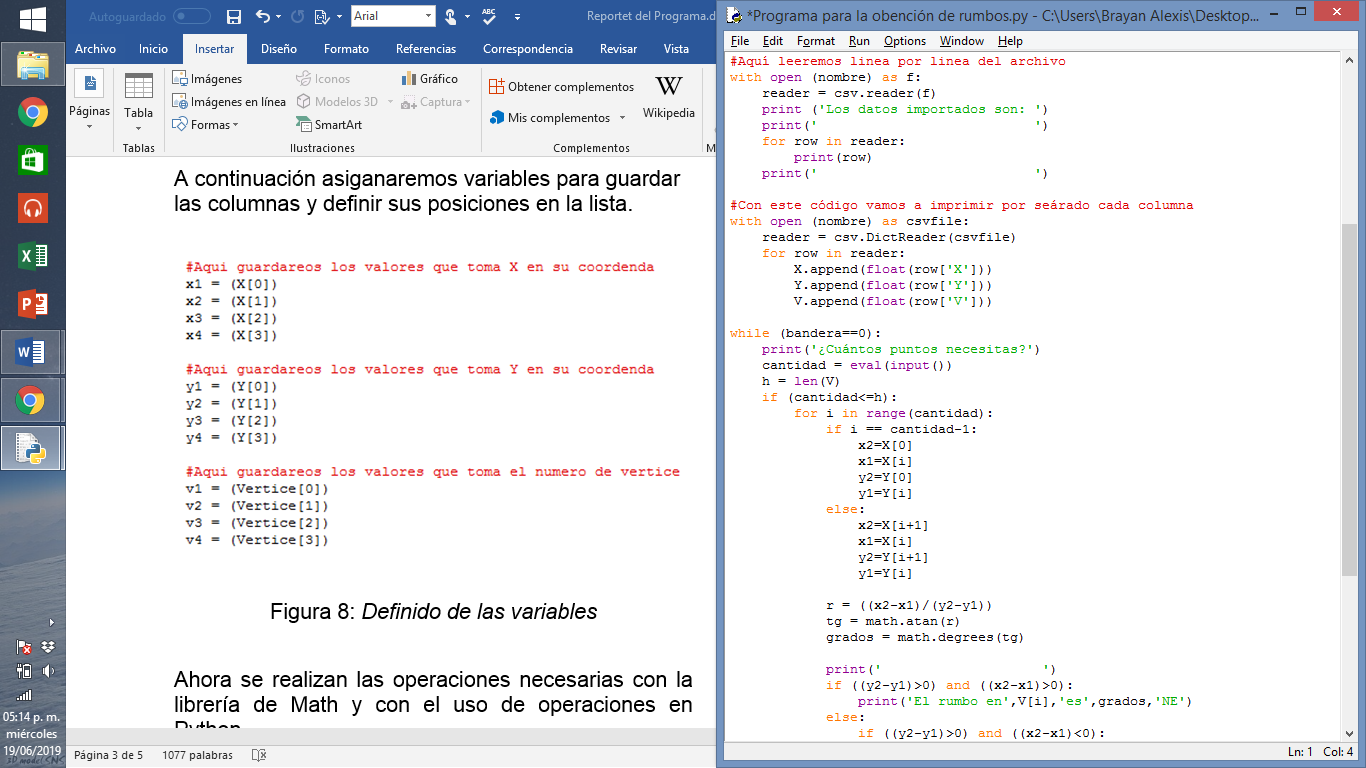


Figura 8: Definido de las variables

Ahora se realizan las operaciones necesarias con la librería de Math y con el uso de operaciones en Python.

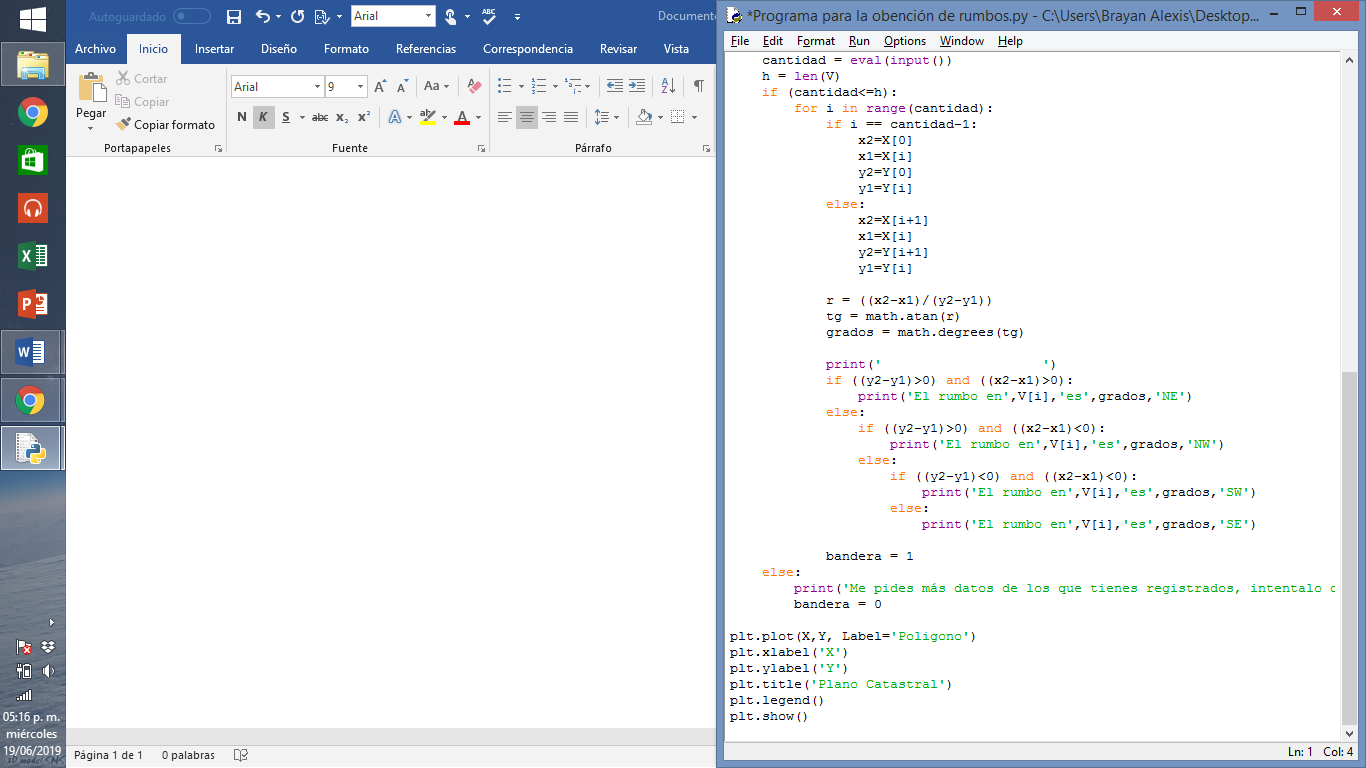


Figura 9: Operaciones con las variables

Los siguientes códigos de ciclo nos ayudan para podar darle la dirección al rumbo (NW, NE, SW, SE), para ello se necesita evluar las restas entre las variables que funcionen como x2 y x1, por ejemplo, en caso de el resultado entre y2 – y1 sea negativo es con rumbo Sur (S), por el contrario sería Norte (N), por otro lado al evaluar x2 – x1 sea negativo sería con rumbo Oeste (W), por el conttrario sería Este (E).

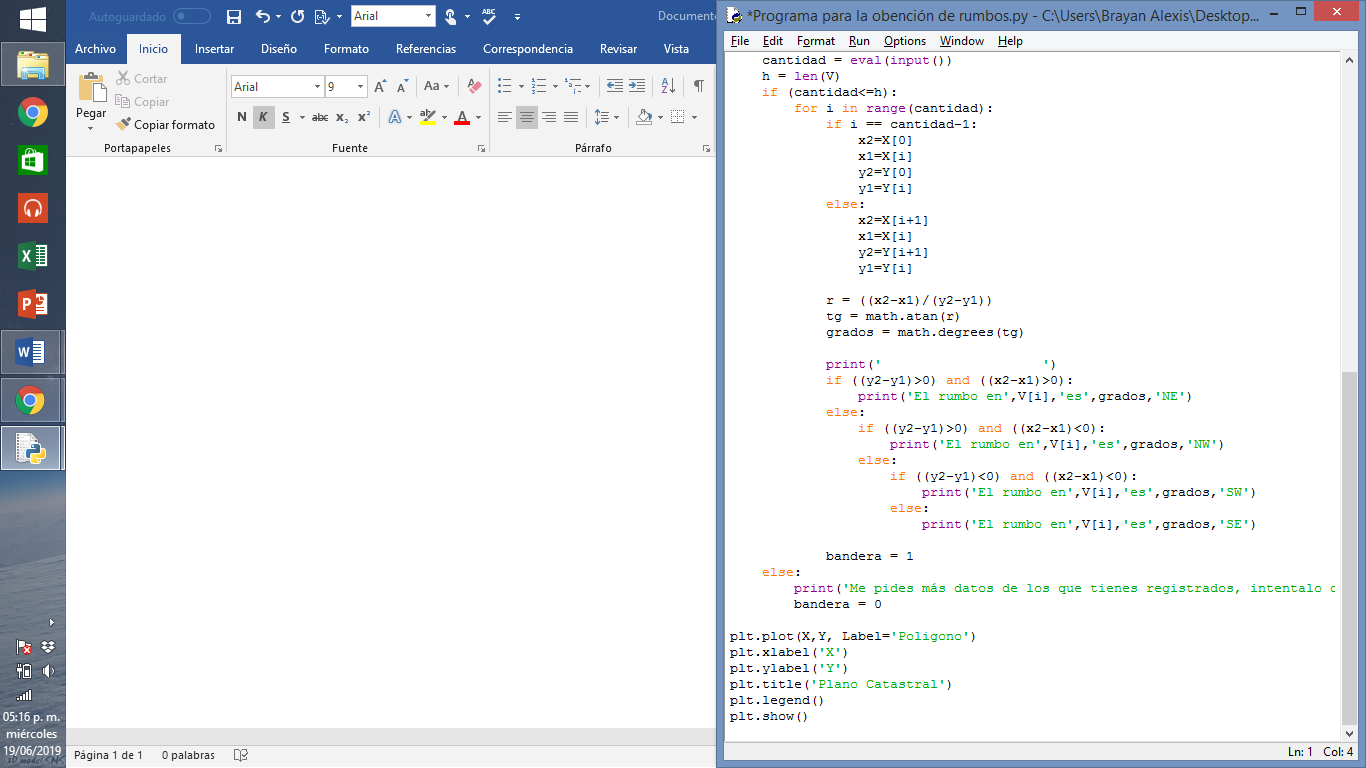


Figura 10: Evaluación del rumbo

Por último vamos a poner en marcha la librería de Matplot para realizar la grafica de los datos, el código es el siguiente:

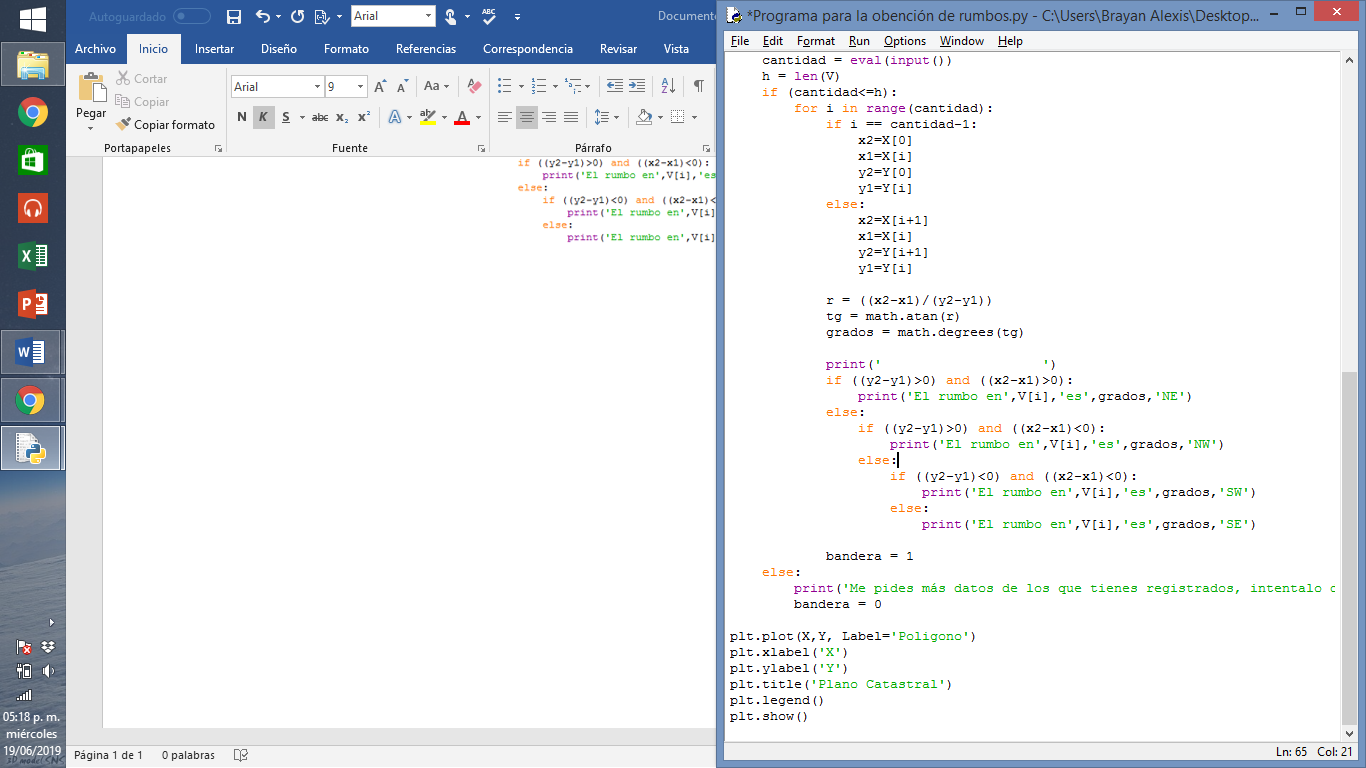


Figura 11: Ploteo de los datos

La función plt.plot() nos dice qué columnas o listas va a graficar, con plt.xlabel, indicamos las osiciones de dichas columnas y con plt.title(), brindamos el titulo del gráfico.

## Resultados

A continuación, en esta sección vamos a poner en practica el programa con un problema topográfico Real

### Cálculo de los azimuts de un polígono real

El polígono que integrará con nuestro programa es el pedido del padre de nuestro compañero Daniel Peralta Landín, que se ubica en la comunidad del Poblado, Municipio de Coquimatlán.



Figura 12: Definición de los vértices del polígono

A continuación se muestran los datos reales del levantamiento obtenidas con Estación Total modelo Leica:

Cuadro 1: Datos reales del predio

Con esos datos en gabinete se realizan procesos y calculoss para obtener las coordenadas de cada uno de los vértices:



Cuadro 2: Coordenadas de los vértices.

A continuación se procede a pasar esos datos a un archivo csv, y nos quedará de la siguiente manera:

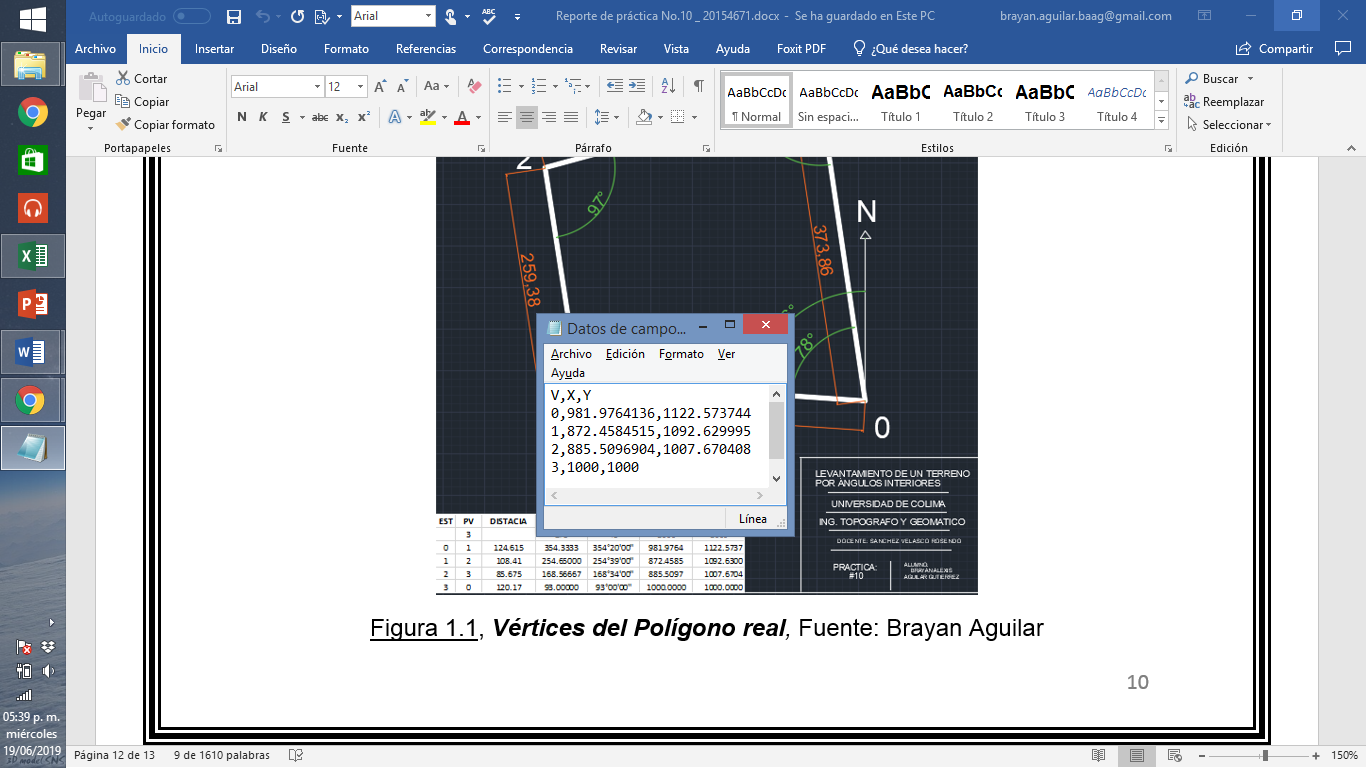
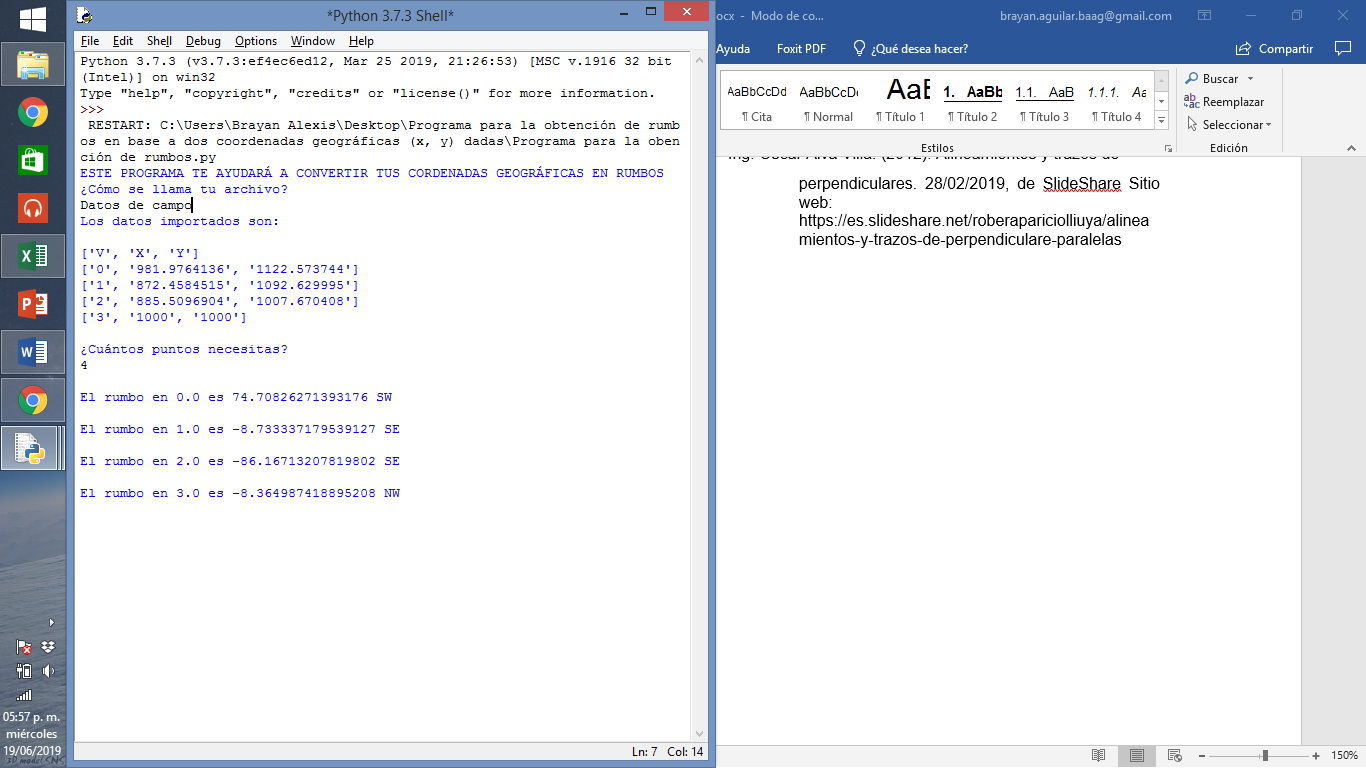


Figura 13: Datos csv

Para finalizar pondremos en práctica nuestro progrma usando el csv de los datos reales y los compararemos con los resultados obtenidos en la hoja dálculo microsoft excel:

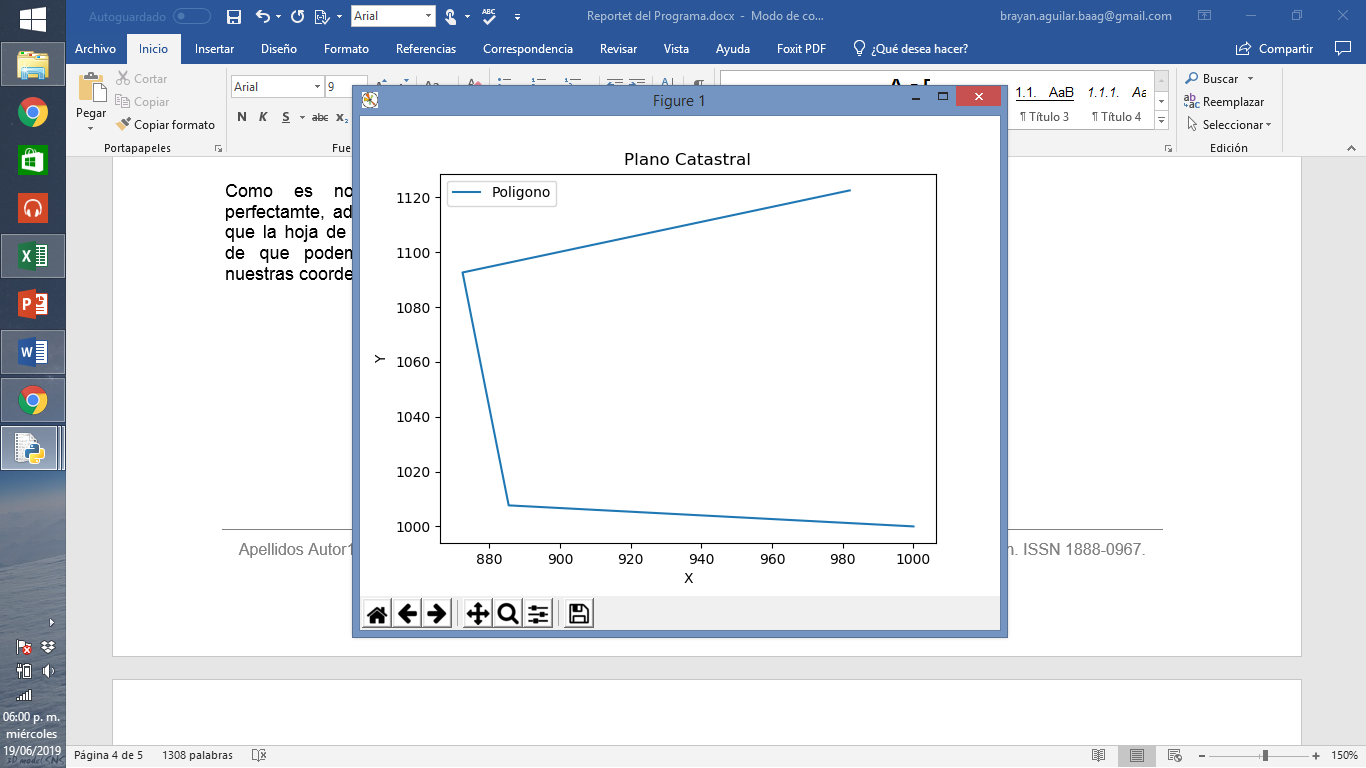


Cuadro 3: Rumbos obtenidos con la hoja de calculo de Microsoft Excel.



Cuadro 4: Rumbos obtenidos con nuestro pograma.

Como es notorio nuestro programa funciona perfectamte, además de ello es mucho más preciso que la hoja de cálculo, y eso es excelente, además de que podemos graficar con ayuda de pyplot nuestras coordendas.



Cuadro 4: Rumbos obtenidos con nuestro pograma.

## Conclusiones

El programa funcionó de la mejor manera, se comprobó manualmente los resultados arrojados por el programa, al analizar se observó que fueron muy precisos y por tanto logramos nuestro objetivo. Se presentaron una serie de obstáculos en el desarrollo del trabajo, pero afortunadamente se logró resolver todos y cada uno de ellos con la ayuda de los consejos del profesor y páginas web en línea. El principal de nuestros problemas fue que en un inicio el programa solo funcionaba con 4 datos, pero por fortuna se logró resolver y finalmente nuestro programa funciona para n cantidad de datos.

Agradecemos a nuestros concejales por ayudarnos a resolver nuestras dudas durante el desarrollo del programa.

## Referencias

Gonzalo Vázquez Vela . (2018). Métodos de levantamientos

planimétricos. 08/03/2019, de Repositorio Digital Sitio web: <https://www.repositoriodigital.ipn.mx/bitstream/123456789/3493/1/COMPETENCIA%20No2.pdf>

Ing. Oscar Alva Villa. (2012). Alineamientos y trazos de

perpendiculares. 28/02/2019, de SlideShare Sitio web: https://es.slideshare.net/roberapariciolliuya/alineamientos-y-trazos-de-perpendiculare-paralelas