# Determinar el área y perímetro de un polígono irregular usando las coordenadas de un script

Brenda Briselle Orozco Fabila, Ángel Rogelio Galván Rodríguez

Universidad de Colima, Facultad de Ingeniería Civil, Ing. Topógrafo Geomático, CP. 28400 Kilometro 4 (1) [borozco4@ucol.mx](mailto:borozco4@ucol.mx) (2) agalvan8@ucol.mx

Resumen

Se va a realizar un programa en Python y sus librerías con el fin de determinar el área y perímetro de un polígono irregular a partir de las coordenadas de un script.

**Palabras clave**: Área, perímetro, polígono, programa.

Abstract

A program in Python and its librarians will be carried out in order to determine the area and the perimeter of a polygon.

**Keywords**: Area, perimeter, polygon, program.

## Introducción

El perímetro y el área son dos elementos fundamentales en matemáticas y uno de los principales objetivos de un levantamiento topográfico.

El perímetro es una medida de la distancia alrededor de una figura y el área nos da una idea de qué tanta superficie cubre dicha figura. La importancia de ello es para ayudar a cuantificar el espacio físico y para proveer las bases de matemáticas más avanzadas como en el álgebra, trigonometría, y cálculo.

El conocimiento del área y el perímetro lo aplican muchas personas día con día, como los arquitectos, ingenieros, y diseñadores gráficos, y es muy útil también para la gente en general. Entender cuánto espacio tienes y aprender cómo conjuntar figuras te ayudará cuando pintas tu cuarto, compras una casa, remodelas la cocina, o construyes un escritorio.

La finalidad de este programa es simplificar el procedimiento que rutinariamente se debe realizar a la hora de hacer un levantamiento topográfico; con ello nos ahorraremos tiempo y rendimiento en nuestras tareas.

## Desarrollo

Una función en Python es una herramienta de modularización, el cual consiste en un fragmento de código que se comporta de manera independiente dentro de un programa, y puede ser invocada varias veces desde otras partes del mismo.

Las funciones reciben parámetros y retornan resultados.

Paso 1: Primeramente, importamos las coordenadas que se encuentran en Excel en terminación .csv de la siguiente manera:

**>>> from os import system**

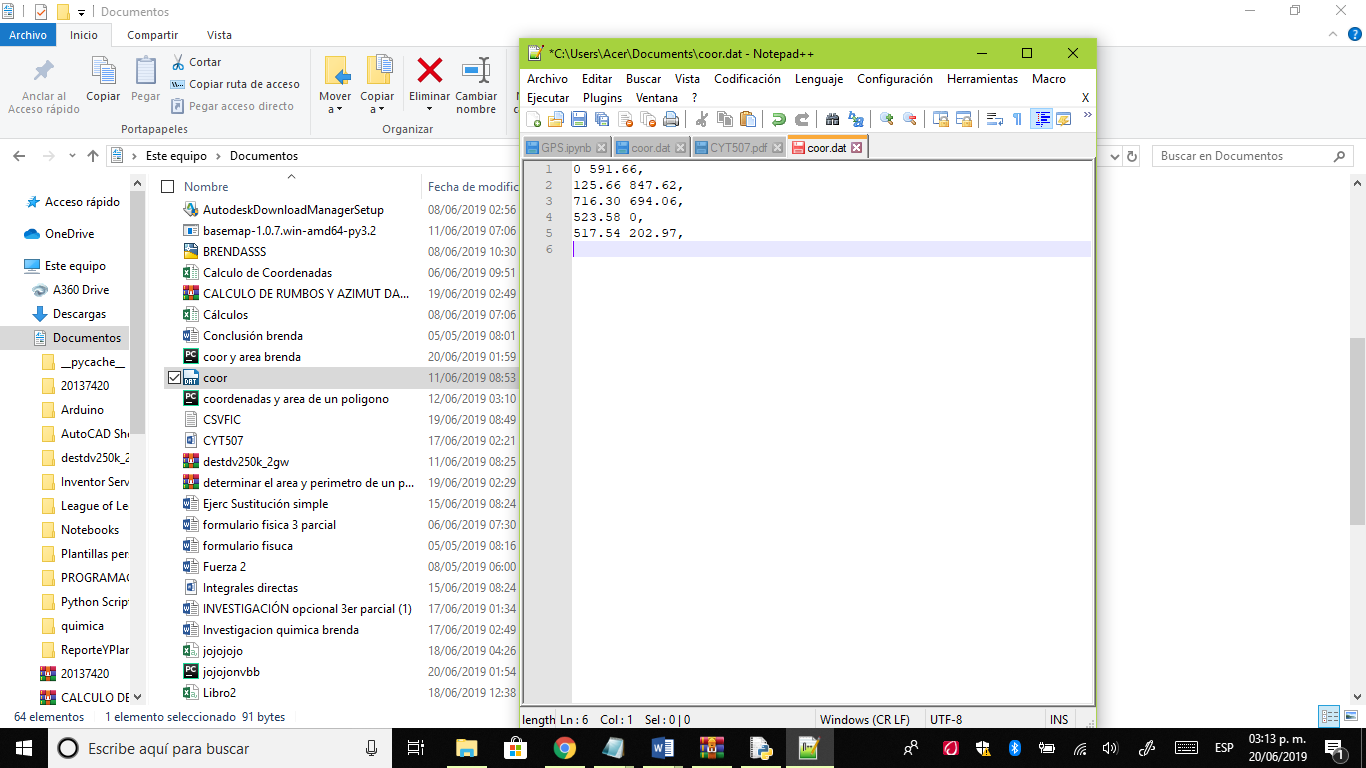
**>>> import csv**

Paso 2: También importamos la librería math en la cual Este módulo proporciona acceso a las funciones matemáticas definidas por el estándar C.

**>>> import math**

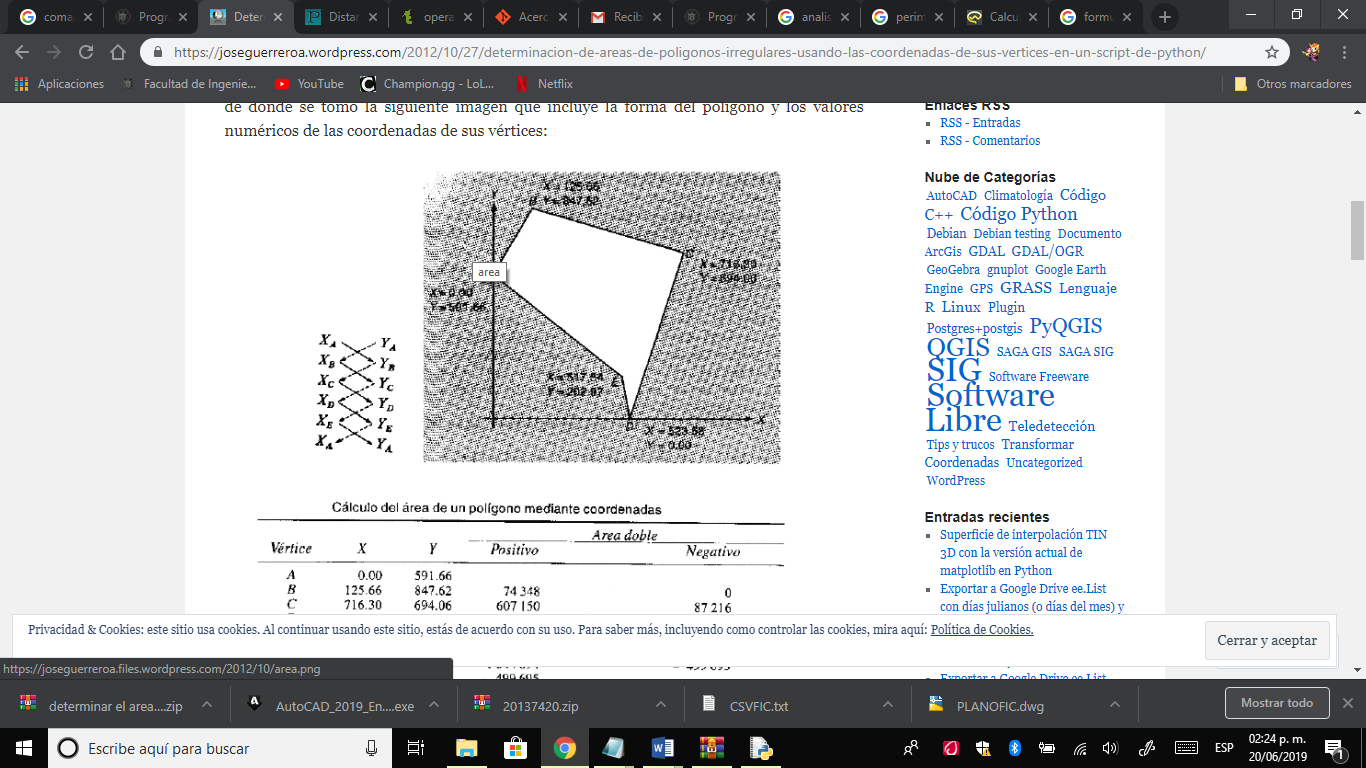
Paso 3: Abrimos el archivo en donde tenemos nuestras coordenadas.

**>>> with open('C:\\Users\\Acer\\Documents\\Notebooks\\coor.csv', 'r') as csvfile**:



**Figura** 1. “Coordenadas en un archivo .csv delimitado por comas”.

La determinación de áreas de polígonos irregulares, usando las coordenadas de sus vértices, emplea el algoritmo conocido como “producto en cruz”.



**Figura 2.** “Ejemplo de un polígono irregular por el método del producto en cruz”

Paso 4: Se lee el archivo y con el comando append se agrega un solo elemento a la lista existente. No devuelve una nueva lista; más bien modifica la lista original, y le agregamos el punto flotante para que nos arroje los decimales.

**>>> x.append (float(row['Easting']))**

Paso 5: Imprimimos las coordenadas para que nos aparezcan en pantalla.

**>>> print ("coordenadas x,y (archivo coor.dat)")**

Paso 6: Con el ciclo for y la fórmula de producto en cruz.

**>>> for i in range(n-1):**

**>>> sum1=x[i]\*y[i+1]+sum1**

**>>> sum2=y[i]\*x[i+1]+sum2**

Paso 7: Imprimimos la sum1 y sum2

**>>> print (sum1, sum2)**

Paso 8: Se desarrolló un algoritmo para la determinación del área

**>>> sub\_area = ((x[1]\*y[n-1])-(x[n-1]\*y[0]))\*(-1)**

**>>> area = sub\_area/2**

Paso 9: Imprimimos el área

**>>> print (x, y, "\narea = %.2f" % area)**

Paso 10: Ingresamos las coordenadas

**>>> N = int(input("Ingresar el número de vértices: "))**

Paso 11: Creamos un ciclo para y determinamos el rango dependiendo de los vértices ingresados en el paso anterior.

**>>> for i in range(0,N):**

**>>> xP1 = float (input("Favor de ingresar la abscisa del primer punto: "))**

**>>>yP1 = float (input("Favor de ingresar la ordenada del primer punto: "))**

**>>> xP2 = float (input("Favor de ingresar la abscisa del segundo punto: "))**

**>>> yP2 = float (input("Favor de ingresar la abscisa del segundo punto: "))**

Paso 11: Imprimimos las distancias y se saca con la siguiente fórmula:

*RAIZ ( (x2-x1)^2 + (y2-y1)^2 )*

**>>> print("La distancia entre los 2 puntos es de:",math.sqrt(abs(((xP2-xP1)\*(xP2-xP1)+((yP2-yP1)\*(yP2-yP1))))))**

Paso 12: Para determinar el perímetro lo obtuvimos sumando las distancias obtenidas de cada 2 coordenadas.

**>>>Suma=math.sqrt(abs(((xP2-xP1)\*(xP2-xP1)+((yP2-yP1)\*(yP2-yP1)))+(N)))**

**>>> perimetro=Suma**

Paso 13: Imprimimos en pantalla el perímetro obtenido.

**>>> print ('El perímetro es: ',perimetro)**

### **Manejo de datos**

El equipo que se utilizo para realizar este programa es una computadora ACER que cuenta con la versión de Windows 10. El programa que empleamos es Python, 3.7. descargado para 64 bits.

En dicho programa pudimos hacer que con la ayuda de las medidas de una figura nos de área y perímetro. Mas aparte que con las coordenadas de un polígono nos de el área y perímetro. Y así mismo poder facilitar trabajos y esto se pueda usar posterior mente para diferentes situaciones.

### **Conclusiones**

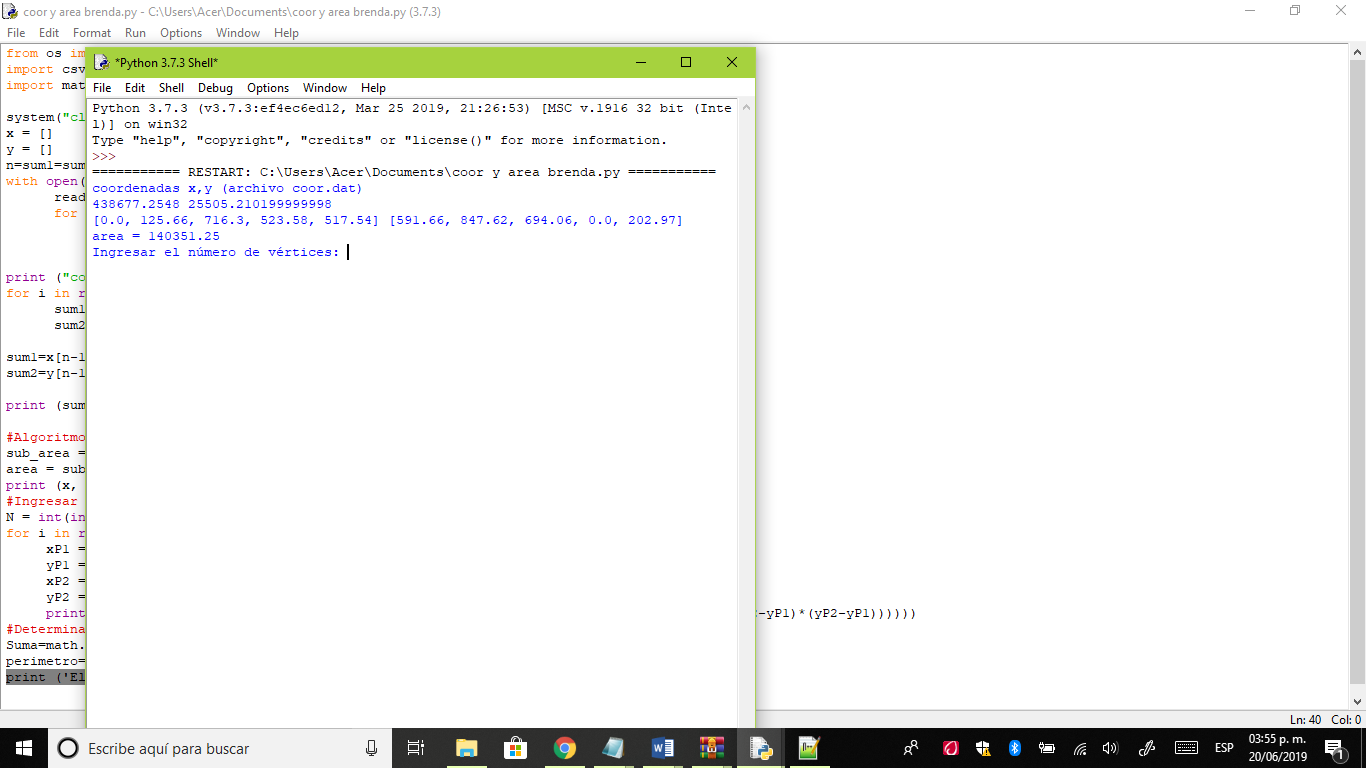
Al realizar este programa nos encontramos con diversas problemáticas que fuimos resolviendo con consultas de diversas páginas y con el profesor con el fin de lograr obtener nuestro propósito inicial, aunque lo fuimos modificando cada vez más.

Consideramos que este programa nos será muy útil para los levantamientos que realicemos. Este programa funcionó de la manera esperada ya que pudimos obtener los resultados correctos

Los problemas que se ocasionaron fueron solucionados en su momento mediante ayuda de diferentes fuentes de información y de consulta.

Este trabajo servirá para aquellos que requieren obtener el área y perímetro usando las coordenadas así se podrá facilitar el trabajo y se realizará más rápido.

### **2.4 Resultados**



### **2.5 Bibliografía**

Sin autor . (21/09/2017). OPERADORES – PYTHON (DISTANCIA ENTRE DOS PUNTOS). 19/06/2019, de Tutorias Colombia Sitio web: https://tutorias.co/operadores-python-distancia-entre-dos-puntos/

Jose Guerrero. (27 octubre, 2012). Determinación de áreas de polígonos irregulares usando las coordenadas de sus vértices en un script de Python. 19/06/2019, de El Blog de José Guerrero Sitio web: https://joseguerreroa.wordpress.com/2012/10/27/determinacion-de-areas-de-poligonos-irregulares-usando-las-coordenadas-de-sus-vertices-en-un-script-de-python/

### **2.5 Apéndice**

