

# UNIVERSIDADE FEDERAL DO PAMPA ENGENHARIA CARTOGRÁFICA E DE AGRIMENSURA

TRABALHANDO COM DADOS GEOESPACIAIS UTILIZANDO A BIBLIOTECA  
FOLIUM EM PYTHON

Autor: Israel da Rosa Passos neto  
Orientador: Prof. Dr. Rogério Rodrigues Vargas

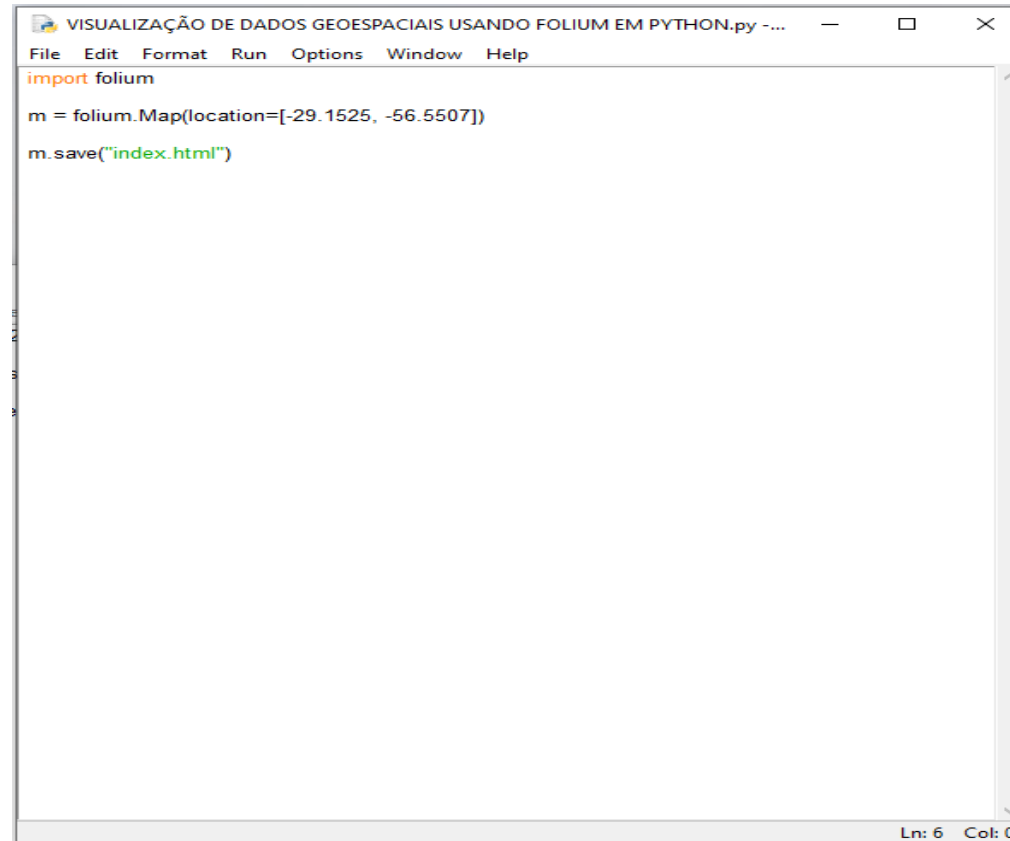
# 1. Introdução

- A biblioteca Folium é um importantes ferramenta para quem trabalha com dados Geoespaciais.
- Pois é possível visualizar dados em Python com contesto geográficos.
- Tais como localização físicas coordenadas e mapas.
- Neste tutorial iremos aprender a plotar mapas a partir e dados físicos através de coordenada geográficas no formato decima, com certas ferramentas que permitem a visualização de marcadores.
- Será criado através de uma base de dados com 15 locais aleatórios no município de Itaqui, Rio grande do sul, onde será identificado através de marcadores.

## 2. Métodos

- Primeiramente devemos importar a biblioteca folium.
- Digitando `import folium` o que irá ativar os dados da biblioteca.
- 
- `“import folium”`
- É preciso identificar a área de interesse a ser feito o trabalho, nesse caso a cidade de Itaqui, para tal basta escrever o seguinte código.
- `“mapa = folium.Map (location=[ -29.125000, -56.552778],`
- Para salva-lo em um arquivo html basta escrevermos o seguinte código:
- `m.save(“index.html”)`

## 2. Métodos



A screenshot of a Python IDE window titled "VISUALIZAÇÃO DE DADOS GEOESPACIAIS USANDO FOLIUM EM PYTHON.py -...". The window contains the following Python code:

```
import folium  
  
m = folium.Map(location=[-29.1525, -56.5507])  
m.save("index.html")
```

The status bar at the bottom right of the window indicates "Ln: 6 Col: 0".

## 2. Métodos



- O código criou um mapa base, muito grande e torna difícil a visualização da área de interesse.

## 2. Métodos

- Através da ferramenta zoom podemos definir o mínimo e o máximo que necessitamos visualizar no mapa.
- Para isso basta incrementar o seguinte código.
- `“mapa = folium.Map (location=[ -29.125000, -56.552778], zoom_start=12`

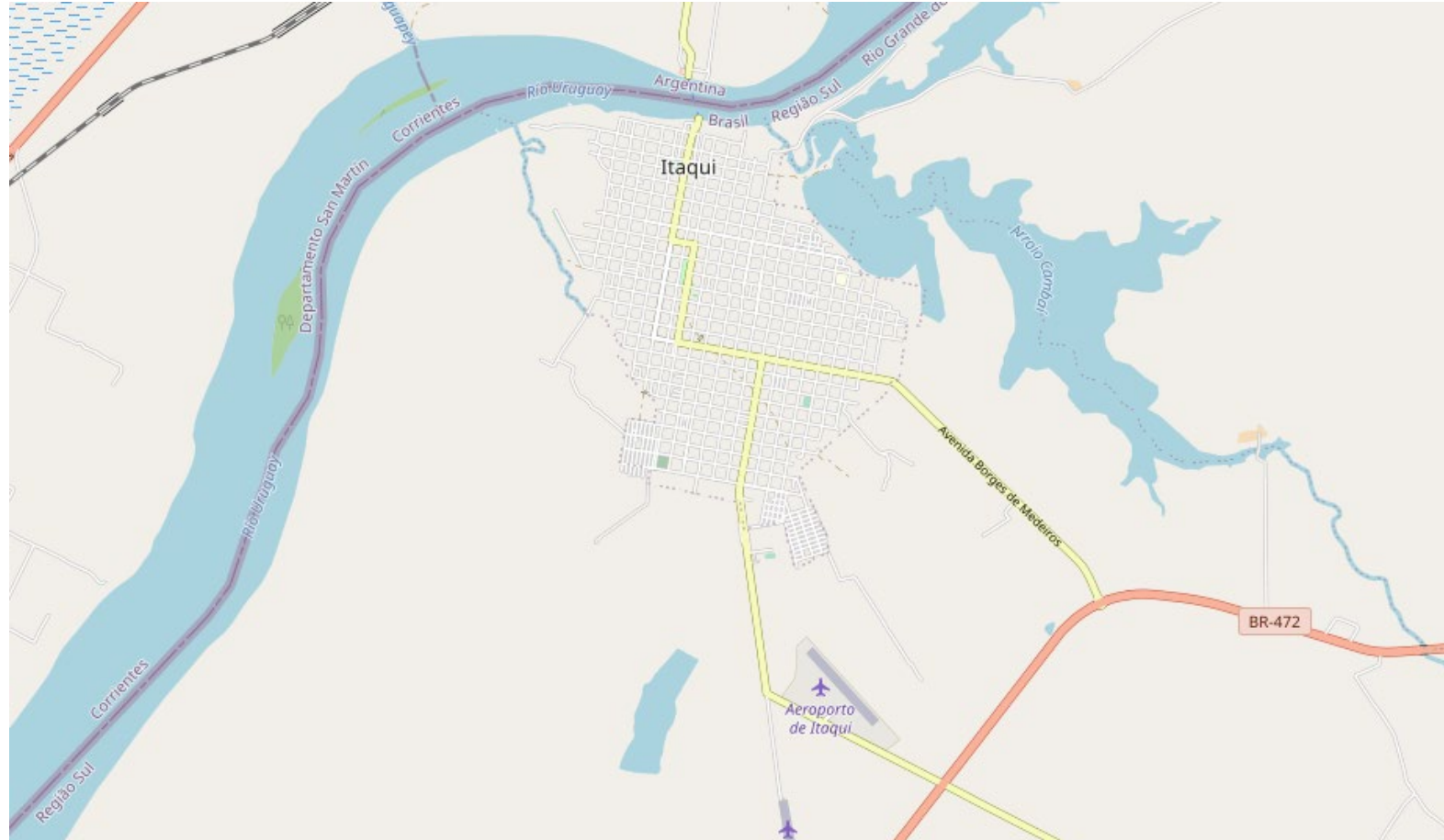
## 2. Métodos

```
VISUALIZAÇÃO DE DADOS GEOESPACIAIS USANDO FOLIUM EM PYTHON.py - F:/Israel/Documents/PROGII/VISUALIZAÇÃO DE DADOS GEOESPACIAIS USANDO FOLIUM EM PYTHON.py (3.10.4)
File Edit Format Run Options Window Help
import folium

m = folium.Map(location=[-29.1525, -56.5507], zoom_start=13)

m.save("index.html")
```

## 2. Métodos





## 2. Métodos

- Agora o mapa base esta bem visível nos dando uma boa visualização na área de interesse.
- Marcadores
- Para inserir marcadores nos mapas bases iremos utilizar o seguinte código:
- `folium.Marker([Latitude,Longitude],popup="Acesso_este_p1",  
tooltip=tooltip).add_to(m)`

## 2. Métodos

- `folium.Marker`, insere o marcador no mapa base através das coordenadas inseridas pelo usuário.
- `popup="<i>Nome do ponto</i>"`, o comando (popup abre a chamada para o nome que o usuário quer dar ao marcador.
- `<i>`, `</i>` da formato itálico para a chamada, assim como `<b>`, `</b>` da formato negrito ao texto.
- `tooltip=tooltip`, é uma moldura flutuante que abra ao passar o mouse sobre o marcador, mostrando uma mensagem que o usuário deseja.

## 2. Método

📄 VISUALIZAÇÃO DE DADOS GEOESPACIAIS USANDO FOLIUM EM PYTHON.py - F:/Israel/Documents/PROGII/VISUALIZAÇÃO DE DADOS GEOESPACIAIS USANDO FOLIUM EM PYTHON.py (3.10.4)

File Edit Format Run Options Window Help

---

```
import folium
```

```
m = folium.Map(location=[-29.1525, -56.5507], zoom_start=13)
```

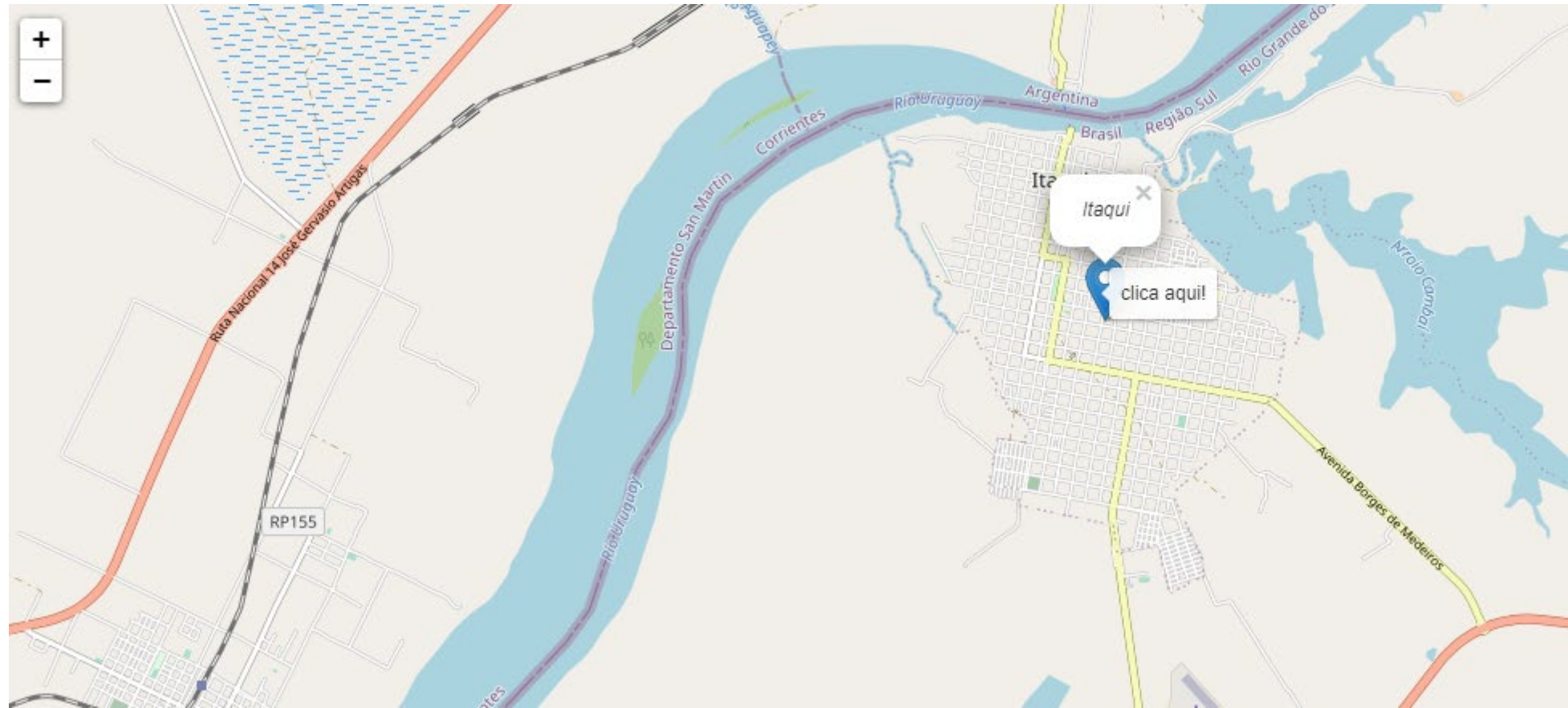
```
tooltip = "clic me!"
```

```
folium.Marker(  
    [-29.135556, -56.550397], popup="<i>Itaqui</i>", tooltip=tooltip  
).add_to(m)
```

```
m.save("index.html")
```

```
|
```

## 2. Métodos



## 3. Exemplo

Neste exemplo iremos criar o código do mapa base com os 15 marcadores na cidade de Itaquí.

# 3. Exemplo

\*teste1.py - F:\Israel\Documents\PROGII\teste1.py (3.10.4)\*

File Edit Format Run Options Window Help

```
import folium
```

```
m = folium.Map(location=[-29.1525, -56.5507], zoom_start=12, tiles="Stamen Terrain")
```

```
tooltip = "Clica aqui!"
```

```
folium.Marker(  
    [-29.163746, -56.510072], popup="<i>Acesso Leste p1</i>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.181217, -56.527449], popup="<b>Acesso_Sul_p2</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.175404, -56.540162], popup="<b>Aeroporto_p3</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.158362, -56.548507], popup="<b>1°RCmeo_p4</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.158362, -56.548507], popup="<b>Unipampa_p5</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.132942, -56.538948], popup="<b>Hospital_p6</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.132127, -56.555592], popup="<b>Parcão_p7</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.135316, -56.555373], popup="<b>Bombeiros_p8</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.132870, -56.5557137], popup="<b>Brigada_Militar_p9</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.121578, -56.557137], popup="<b>Prefeitura Municipal_p10</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.122368, -56.556354], popup="<b>Praça Centra_p11</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.121299, -56.556722], popup="<b>Camara_Municipal_p12</b>", tooltip=tooltip  
)add_to(m)
```

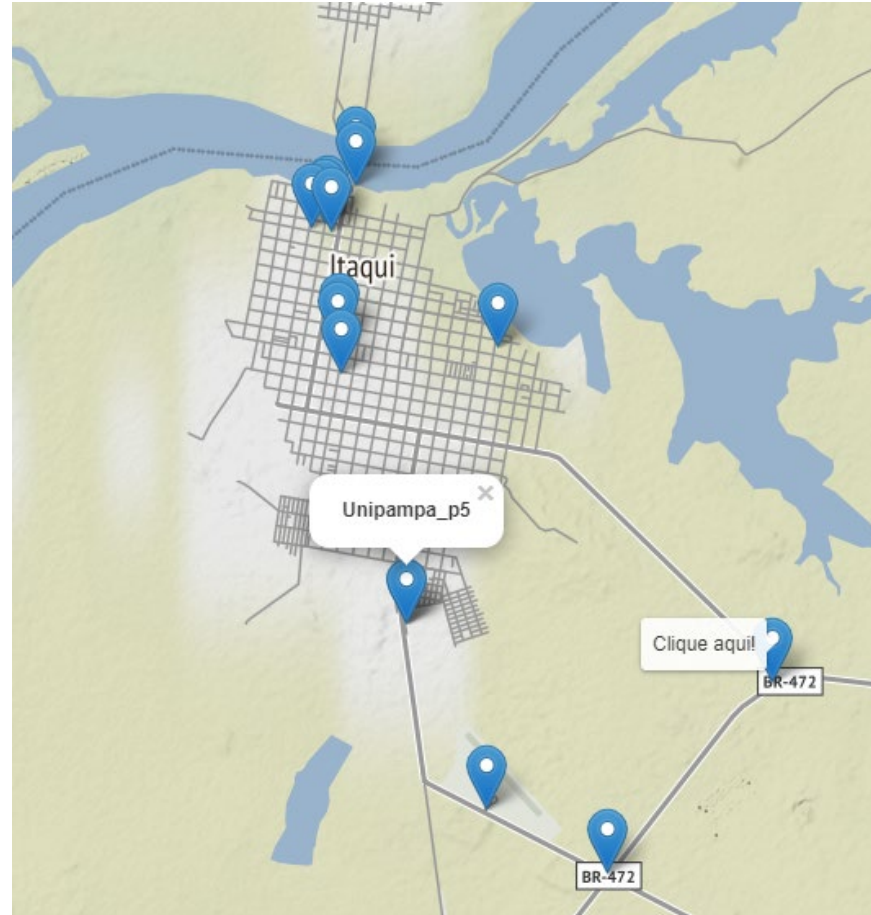
```
folium.Marker(  
    [-29.118197, -56.553793], popup="<b>Praça_do_Porto_13</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.116842, -56.553793], popup="<b>Porto_de_Itaqui_14</b>", tooltip=tooltip  
)add_to(m)
```

```
folium.Marker(  
    [-29.181217, -56.527449], popup="<b>Cartório_Eleitoral_p15</b>", tooltip=tooltip  
)add_to(m)
```

```
m.save("index.html")
```

### 3. Exemplo



# Calculo de Área Pelo Método de Gauss

O método de Gauss baseia-se no calculo da dupla área da poligonal.

Programas bem conhecidos em especial o AutoCAD, utilizam-se desse método, que baseia-se no sistema de coordenadas planas X e Y.

Devido o alto custo desse programa, é possível achar outros meios que possibilitem a verificação em campo das poligonais com baixo custos.

Uma forma de diminuir os custos é a criação desse método em Python.



# Método de Gauss

Formula do calculo da área de Gauss

	X	Y	
	X 0	Y 0	
	X 1	Y 1	
Y0*X1	X 2	Y 2	X0 * Y1
Y1*X2	X 3	Y 3	X1 * Y2
Y2*X3	X 0	Y 0	X2* Y3
Y3*X0			X3 * Y0
$\Sigma \text{ ESQ.} = X0*Y1 + X1*Y2+X2*Y3+X3*Y0$			
$\Sigma \text{ DIR.} = Y0*X1+ Y1*X2+Y2*X3+Y3*X0$			
$\text{Área} = \Sigma (\text{ESQ.}+\text{DIR.})/2$			
$\text{Área} =  \Sigma (\text{ESQ.} + \text{DIR.})/2 $			

# Método de Gauss

O exemplo a seguir nos mostram como é possível a criação de um código no sistema Python.

- ✓ Declara listas ou tuplas

```
coordenadas = [], []
```

- ✓ Criação de variáveis e valor das mesmas

```
somaesquerda = somaedireita = area = 0
```

- ✓ Solicita ao usuário o número de vértices da poligonal

```
vertices = int(input('Digite o número de vértices: '))
```

- ✓ Vai do valor zero de entrada ao número de vértices que o usuário informou.

```
for n in range(0, vertices):
```

- ✓ Passará somente uma única vez pelos vértices.

```
for c in range(1):
```

# Método de Gauss

- ✓ Solicita as coordenadas x ao usuário ( lembrando que f' é uma string Que armazenara {} um valor).

```
coordenadas[0].append(float(input(f'Digite o ponto x de {n}: ')))
```

- ✓ Solicita as coordenadas Y ao usuário

```
coordenadas[1].append(float(input(f'Digite o ponto y de {n}: ')))
```

- ✓ Printa em tela as coordenadas digitadas.

```
print(f'Coordenadas X: {coordenadas[0]}\nCoordenadas Y: {coordenadas[1]}')
```

- ✓ for vai de 0 ate o penúltimo vértice calcular com o primeiro novamente.

```
for linha in range(0, vertices-1):
```

- ✓ Repete somente uma vez o for anterior.

```
for coluna in range(1):
```

# Método de Gauss

✓ Calculo:

✓ soma os elementos da multiplicação cruzada até a penúltima coordenada.

```
somaesquerda = somaesquerda + coordenadas[0][linha] * coordenadas[1][linha+1]
```

```
somaedireita = somaedireita + coordenadas[1][linha] * coordenadas[0][linha+1]
```

✓ Acumula o valor do comando anterior e soma com a multiplicação da penúltima com a primeira de cada elemento.

```
somaesquerda = somaesquerda + coordenadas[0][vertices-1] * coordenadas[1][0]
```

```
somaedireita = somaedireita + coordenadas[1][vertices-1] * coordenadas[0][0]
```

```
print(f'Somatório da Esquerda := {somaesquerda}\nSomatório da Direita := {somaedireita}')
```

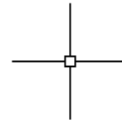
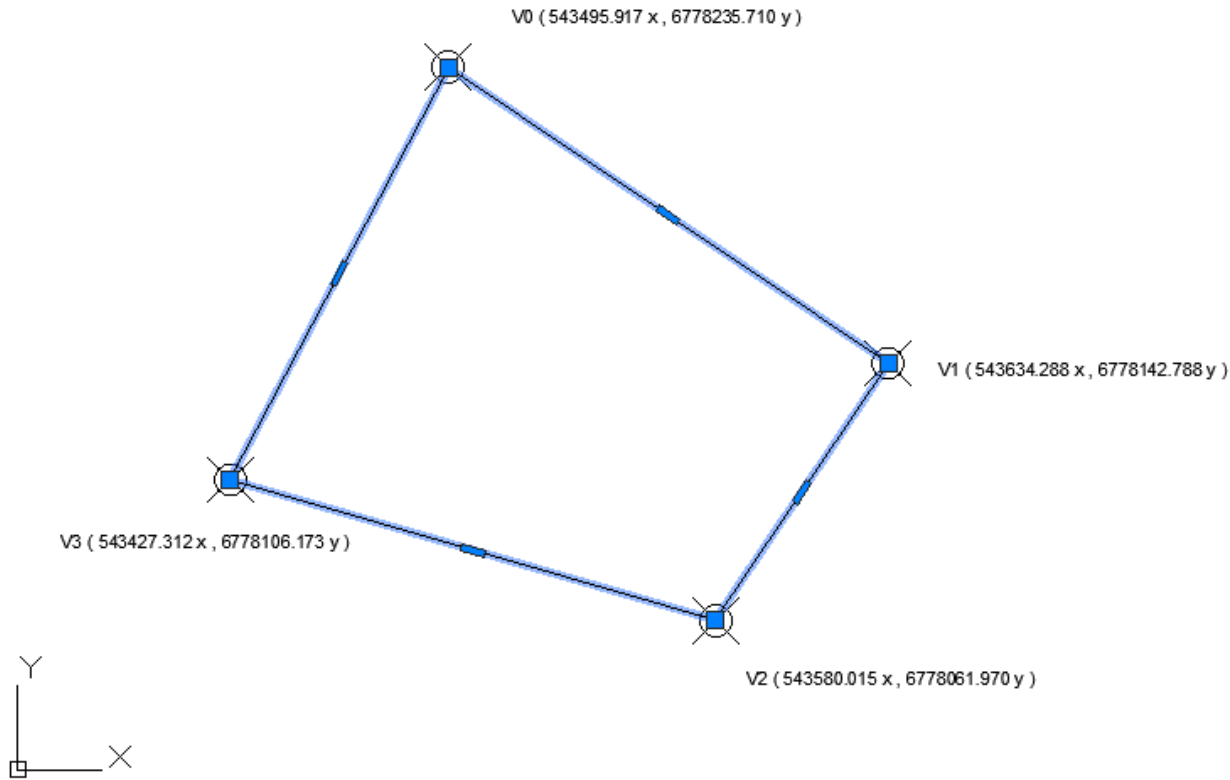
✓ Calculo do modulo dos somatório das áreas

```
area = (somaesquerda + -somaedireita)/2
```

```
print(f'A área do poligono fechado é de : {abs(area)} Metros²\nSeu número de Pontos é de {vertices}')
```

# Área no AutoCAD

[-][Superior][Estrutura de arame 2D]



PROPRIEDADES

Polilinha

**Geral**

Cor	PorBloco
Camada	0
Tipo de linha	Contínua
Escala do ti...	1.000
Estilo de pl...	PorCor
Espessura...	PorCa...
Transparên...	PorCamada
Hiperlink	
Espessura	0.000

**Visualização 3D**

Material	PorCamada
----------	-----------

**Geometria**

Vértice atual	1
Vértice X	543427.312
Vértice Y	6778106.173
Largura do...	0.000
Largura do...	0.000
Largura glo...	0.000
Elevação	0.000
Área	19519.598
Comprimen...	560.594

# Área por meto de Gauss em Python

```
calculador area.py - F:\Israel\Desktop\prog_2\calculador area.py (3.10.4)
File Edit Format Run Options Window Help
coordenadas = [], []
somaesquerda = somaedireita = area = 0
vertices = int(input("Digite o numero de vértices: "))
for n in range(0, vertices):
    for c in range(1):
        coordenadas[0].append(float(input(f"Digite o ponto x de {n}: ")))
        coordenadas[1].append(float(input(f"Digite o ponto y de {n}: ")))
print(f"Coordenadas X: {coordenadas[0]}\nCoordenadas Y: {coordenadas[1]}")
for linha in range(0, vertices-1):
    for coluna in range(1):
        somaesquerda = somaesquerda + coordenadas[0][linha] * coordenadas[1][linha+1]
        somaedireita = somaedireita + coordenadas[1][linha] * coordenadas[0][linha+1]
somaesquerda = somaesquerda + coordenadas[0][vertices-1] * coordenadas[1][0]
somaedireita = somaedireita + coordenadas[1][vertices-1] * coordenadas[0][0]
print(f"Somatorio da Esquerda : = {somaesquerda}\nSomatório da Direita : = {somaedireita}")
area = (somaesquerda + -somaedireita)/2
print(f"A área do poligono fechado é de : {abs(area)} Metros²\nSeu número de Pontos é de {vertices}")

IDLE Shell 3.10.4
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on w
in32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\Israel\Desktop\prog_2\calculador area.py =====
Digite o numero de vértices: 4
Digite o ponto x de 0: 543495.917
Digite o ponto y de 0: 6778235.710
Digite o ponto x de 1: 543634.288
Digite o ponto y de 1: 6778142.788
Digite o ponto x de 2: 543580.015
Digite o ponto y de 2: 6778061.970
Digite o ponto x de 3: 543427.312
Digite o ponto y de 3: 6778106.173
Coordenadas X: [543495.917, 543634.288, 543580.015, 543427.312]
Coordenadas Y: [6778235.71, 6778142.788, 6778061.97, 6778106.173]
Somatorio da Esquerda : = 14736601290380.469
Somatório da Direita : = 14736601329419.727
A área do poligono fechado é de : 19519.62890625 Metros²
Seu número de Pontos é de 4
>>>
```