



UNIVERSIDAD DE GUADALAJARA

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES**

MATERIA:

SIMULACION POR COMPUTADORA

MAESTRO:

DAVID ALEJANDRO GOMEZ ANAYA

TITULO DE INVESTIGACIÓN:

SISTEMAS DE COORDENADAS

FECHA ENTREGA:

DOMINGO 3 DE FEBRERO 2019

ALUMNO:

FELIPE DE JESUS RUIZ GARCIA

CODIGO:

214522077

CARRERA: INGENIERIA INFORMATICA (INNI)

SECCION: D03

CALIFICACIÓN Y OBSERVACIONES:

Indice:

Objetivo.2

Marco teórico.2

Desarrollo.....2

Pruebas y resultados.....4

Conclusiones.....6

Apéndice(s).....6

Objetivo

Desarrollar un programa capaz de representar y detectar graficamente puntos en un sistema de coordenadas 2D, mostrando su posicion absoluta y relativa, permitiendo importar y exportar los puntos del sistema de coordenadas a un archivo de texto con formato CSV.

Marco Teorico

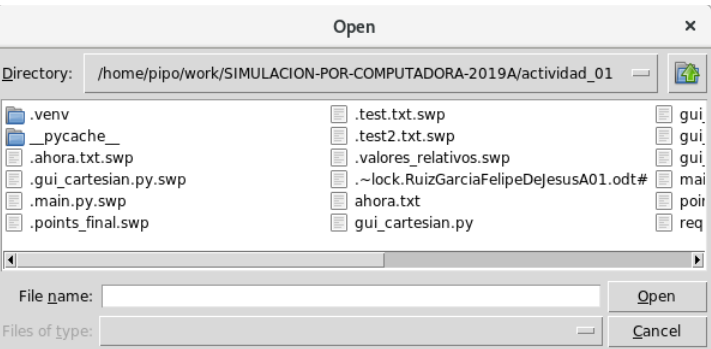
Se realizo una investigacion de librerias, herramientas y plataformas capaces de manipular en tiempo real sistemas 2D, capaz de detectar y manipular la interaccion eventos como pulsacion de teclas y pulsacion de mouse/touchpad.

Se opto por usar [matplotlib](#) una biblioteca de trazado 2D de Python que produce cifras de calidad de publicación en una variedad de formatos de papel y entornos interactivos en todas las plataformas.

Una vez definida la libreria principal y el lenguaje de programacion, se investigo de otra herramienta grafica de Python capaz de mostrar un menu grafico para seleccionar el archivo CSV a importar ; se eligio a tkinter para ello.

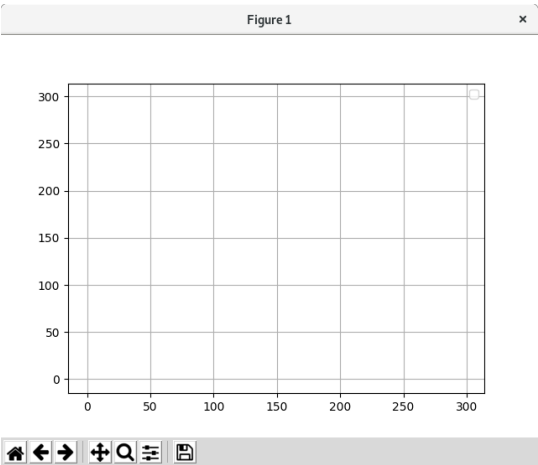
Desarrollo

Se desarrollo una solucion en el lenguaje python en 120 lineas de codigo que inicia mostrando una interfaz grafica para cargar un archivo CSV :

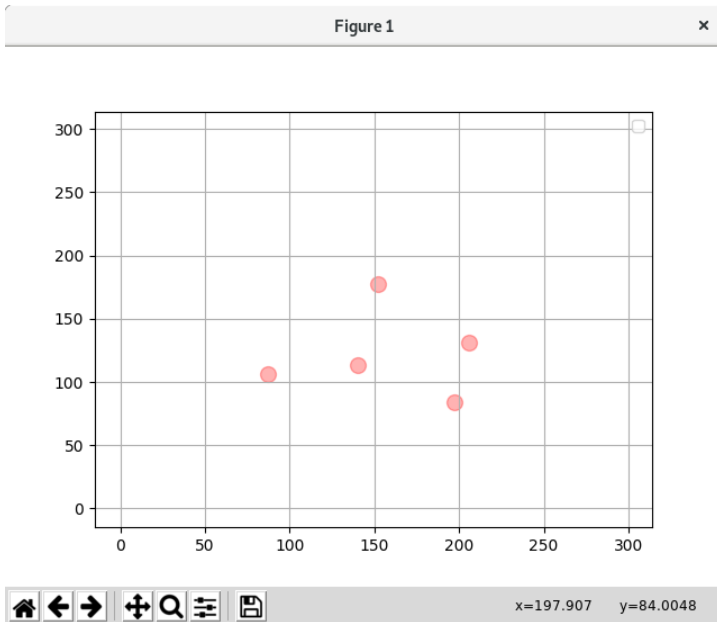


Si se selecciona un archivo se muestra un sistema de coordenadas 2D con los puntos definidos en el archivo CSV seleccionado ;

Si no se selecciona un archivo CSV, se muestra un sistema de coordenadas sin puntos.



Una vez se muestra el sistema 2D, se pueden mostrar puntos en el haciendo click derecho en la ubicacion dentro del plano.



Se muestran las coordenadas de la ubicacion del mouse dentro del sistema 2D, en la esquina inferior derecha.

Los puntos tienen un diametro visible sea cual sea el punto de zoom.

```
140 valor 113 con respecto a [140, 113]
VALORES RELATIVOS: [[140, 113]]
3 87.16814516129033 106.25178571428572
87 valor 106 con respecto a [87, 106]
VALORES RELATIVOS: [[140, 113], [-53, -7]]
3 152.8155241935484 177.44226190476192
152 valor 177 con respecto a [152, 177]
VALORES RELATIVOS: [[140, 113], [-53, -7], [65, 71]]
3 206.52701612903223 131.1684523809524
206 valor 131 con respecto a [206, 131]
VALORES RELATIVOS: [[140, 113], [-53, -7], [65, 71], [54, -46]]
3 197.90665322580645 84.00476190476192
197 valor 84 con respecto a [197, 84]
VALORES RELATIVOS: [[140, 113], [-53, -7], [65, 71], [54, -46], [-9, -47]]
```

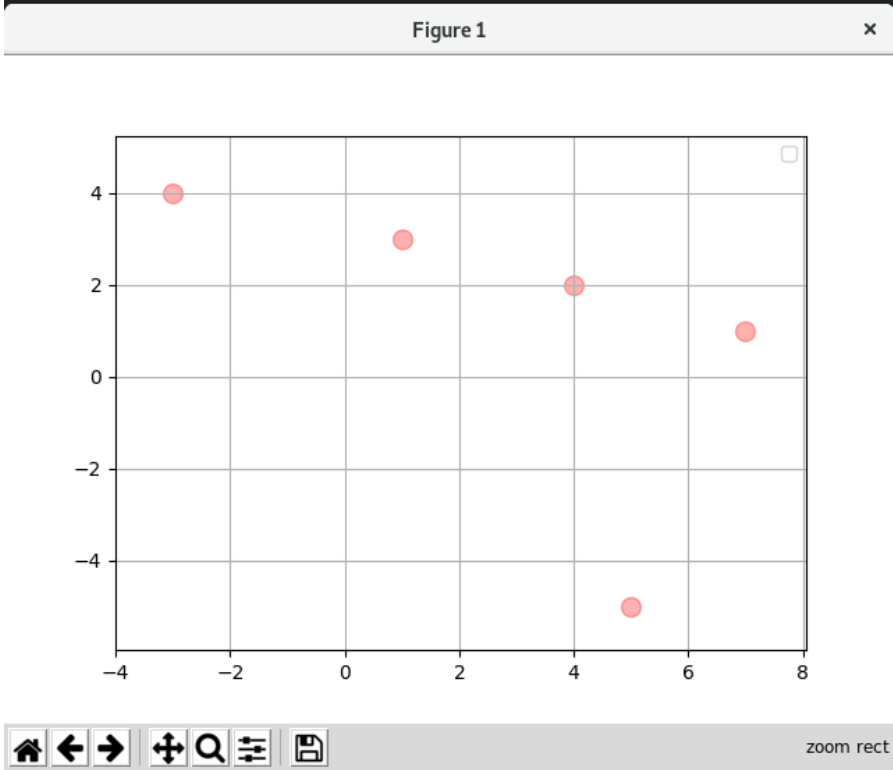
En la terminal se muestran los puntos relativos de cada punto en orden de insercion, cada vez que se dibuja o importa un punto.

Una vez que se cierra el programa, se guarda un archivo CSV con las coordenadas relativas de todos los puntos en el sistema de coordenadas.
Asi tambien se guarda otro archivo CSV con todas las coordenadas absolutas de los puntos.
Ambos archivos en orden de inserccion.

Pruebas y resultados

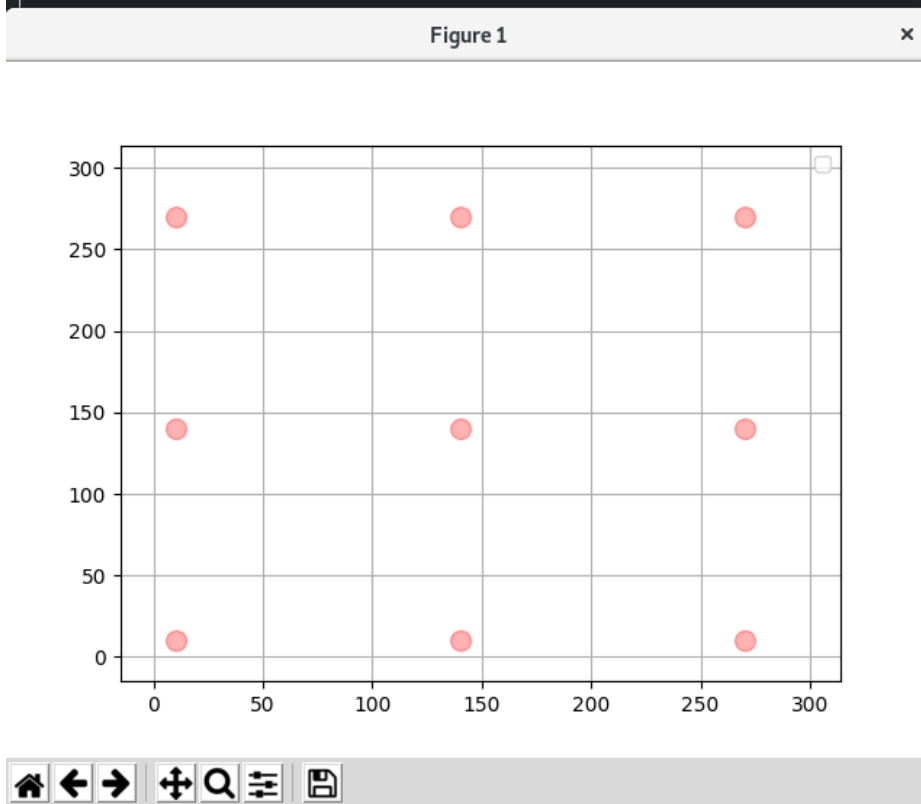
```
(.venv) [pipo@localhost actividad_01]$ python main.py
starting
/home/pipo/work/SIMULACION-POR-COMPUTADORA-2019A/actividad_01/test.txt
from the main [['1', '3'], ['4', '2'], ['7', '1'], ['-3', '4'], ['5', '-5']]
No handles with labels found to put in legend.
1 valor 3 con respecto a ['1', '3']
VALORES RELATIVOS: [[1, 3]]
4 valor 2 con respecto a ['4', '2']
VALORES RELATIVOS: [[1, 3], [3, -1]]
7 valor 1 con respecto a ['7', '1']
VALORES RELATIVOS: [[1, 3], [3, -1], [3, -1]]
-3 valor 4 con respecto a ['-3', '4']
VALORES RELATIVOS: [[1, 3], [3, -1], [3, -1], [-10, 3]]
5 valor -5 con respecto a ['5', '-5']
VALORES RELATIVOS: [[1, 3], [3, -1], [3, -1], [-10, 3], [8, -9]]
```

Se realizaron varias pruebas, en las cuales se incluyo un ejercicio realizado en clase, donde se dieron las coordenadas absolutas y se estudiaron las coordenadas relativas.



Asi tambien se probo con distancias relativas y absolutas proporcionales en el eje X y Y, donde la distancia de un punto a otro en cualquier eje, era la misma, y la distancia era

```
(.venv) [pipo@localhost actividad_01]$ python main.py
starting
/home/pipo/work/SIMULACION-POR-COMPUTADORA-2019A/actividad_01/ahora.txt
from the main [['10', '10'], ['10', '140'], ['10', '270'], ['140', '10'], ['140', '140'], ['140', '270'], ['270', '10'], ['270', '140'], ['270', '270']]
No handles with labels found to put in legend.
10 valor 10 con respecto a ['10', '10']
VALORES RELATIVOS: [[10, 10]]
10 valor 140 con respecto a ['10', '140']
VALORES RELATIVOS: [[10, 10], [0, 130]]
10 valor 270 con respecto a ['10', '270']
VALORES RELATIVOS: [[10, 10], [0, 130], [0, 130]]
140 valor 10 con respecto a ['140', '10']
VALORES RELATIVOS: [[10, 10], [0, 130], [0, 130], [130, -260]]
140 valor 140 con respecto a ['140', '140']
VALORES RELATIVOS: [[10, 10], [0, 130], [0, 130], [130, -260], [0, 130]]
140 valor 270 con respecto a ['140', '270']
VALORES RELATIVOS: [[10, 10], [0, 130], [0, 130], [130, -260], [0, 130], [0, 130]]
270 valor 10 con respecto a ['270', '10']
VALORES RELATIVOS: [[10, 10], [0, 130], [0, 130], [130, -260], [0, 130], [0, 130], [130, -260]]
270 valor 140 con respecto a ['270', '140']
VALORES RELATIVOS: [[10, 10], [0, 130], [0, 130], [130, -260], [0, 130], [0, 130], [130, -260], [0, 130]]
270 valor 270 con respecto a ['270', '270']
VALORES RELATIVOS: [[10, 10], [0, 130], [0, 130], [130, -260], [0, 130], [0, 130], [130, -260], [0, 130], [0, 130]]
```



Conclusiones

Se desarrollo un sistema en la tecnologia python con el uso de librerias graficas para mostrar en un sistema 2D puntos mediante coordenadas absolutas y eventos de mouse. Tambien permite mostrar puntos mediante definidos en un archivo CSV, asi tambien puede guardar archivos con los puntos de coordenadas absolutas y relativas mostradas en el plano 2D.

Apendice

LOS CODIGOS ESTAN DISPONIBLES EN MI REPOSITORIO:

<https://github.com/CUCEI-TAREAS/SIMULACION-POR-COMPUTADORA-2019A>

main.py
gui_loadfile.py
gui_cartesian.py

main.py

```
#!/usr/bin/env python
```

```
import sys
```

```
import gui_cartesian as plane  
import gui_loadfile as files
```

```
if __name__ == "__main__":
```

```
    print(" starting ")
```

```
    file_selected = files.selectFile()
```

```
    points = list()
```

```
    if isinstance(file_selected, (str)):  
        points = files.getListFromCSV(file_selected)
```

```
    print("from the main", points)  
    plane.setupPlane(points)
```

gui_loadfile.py

```
#!/usr/bin/env python
```

```
import csv
```

```
from tkinter import Tk
```

```
from tkinter.filedialog import askopenfilename
```

```
def selectFile():
```

```
    root = Tk()
```

```
    root.withdraw() # we don't want a full GUI, so keep the root window from appearing
```

```
    filename = askopenfilename() # show an "Open" dialog box and return the path to the  
selected file
```

```
    root.destroy()
```

```
    return filename
```

```
def getListFromCSV(path):
```

```
    print(path)
```

```
    points = list()
```

```
    with open(path, "r") as csvfile:
```

```
        reader = csv.reader(csvfile)
```

```
        for row in reader:
```

```
            points.append(row)
```

```
    return points
```

```
def writeCSVFrom(points, namefile):
```

```
    with open(namefile, 'w') as f:
```

```
        writer = csv.writer(f)
```

```
        writer.writerows(points)
```

gui_cartesian.py

```

import gui_loadfile as files

import matplotlib.pyplot as plt
import numpy as np

fig, ax = plt.subplots()

final_points = list()

def onclick(event):
    print(event.button, event.xdata, event.ydata)

    if event.button == 3:
        drawPoint([[int(event.xdata), int(event.ydata)]])
        relativePoints(final_points)

def setupPlane(points=[]):

    plt.plot(range(300), linewidth=0)
    plt.grid()

    fig.canvas.mpl_connect('button_press_event', onclick)

    ax.legend()
    drawPoint(points)
    relativePoints(final_points)

    plt.show()

    files.writeCSVFrom(final_points, "points_final")
    files.writeCSVFrom(relativePoints(final_points), "valores_relativos")

def drawPoint(points=[]):
    color=['red', 'green', 'blue']
    scale = 100
    for point in points:
        x = point[0]
        y = point[1]
        print(x, " valor ", y, " con respecto a ", point )
        final_points.append([int(x), int(y)])
        relativePoints(final_points)

        ax.scatter(int(x), int(y), c=color[0], s=scale, alpha=0.3, edgecolors='face')

    plt.show()

def relativePoints(final_points):

```



```
aux_x = final_points[0][0]
aux_y = final_points[0][1]

aux2_x = aux_x * -1
aux2_y = aux_y * -1

rel = list()
rel.append(final_points[0])

for point in final_points[1:]:
    aux_x = point[0] + aux2_x
    aux_y = point[1] + aux2_y

    rel.append([aux_x, aux_y])

    aux2_x -= aux_x
    aux2_y -= aux_y

print("VALORES RELATIVOS: ", rel)
return rel
```

