UNIVERSIDAD DE GUADALAJARA

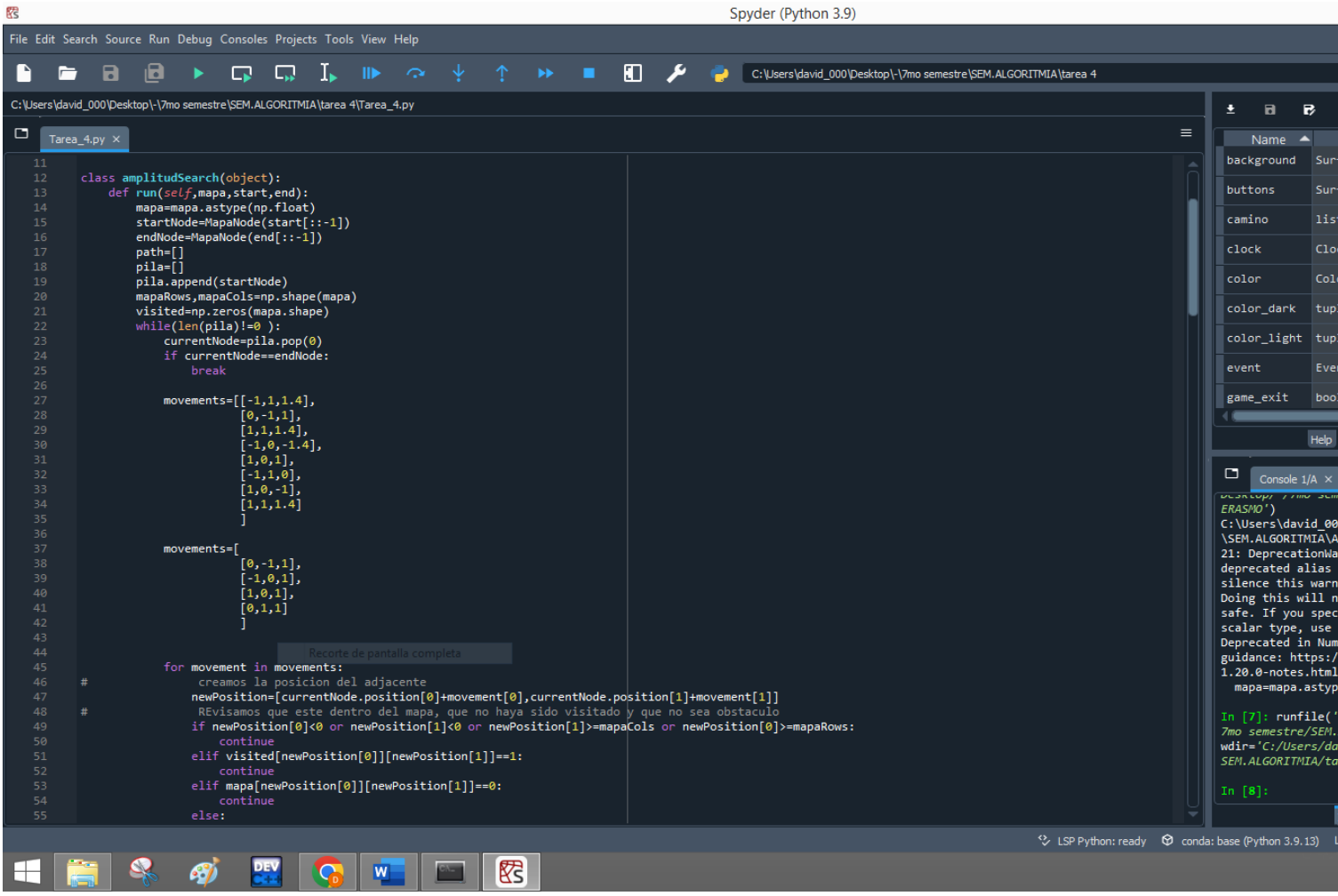
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

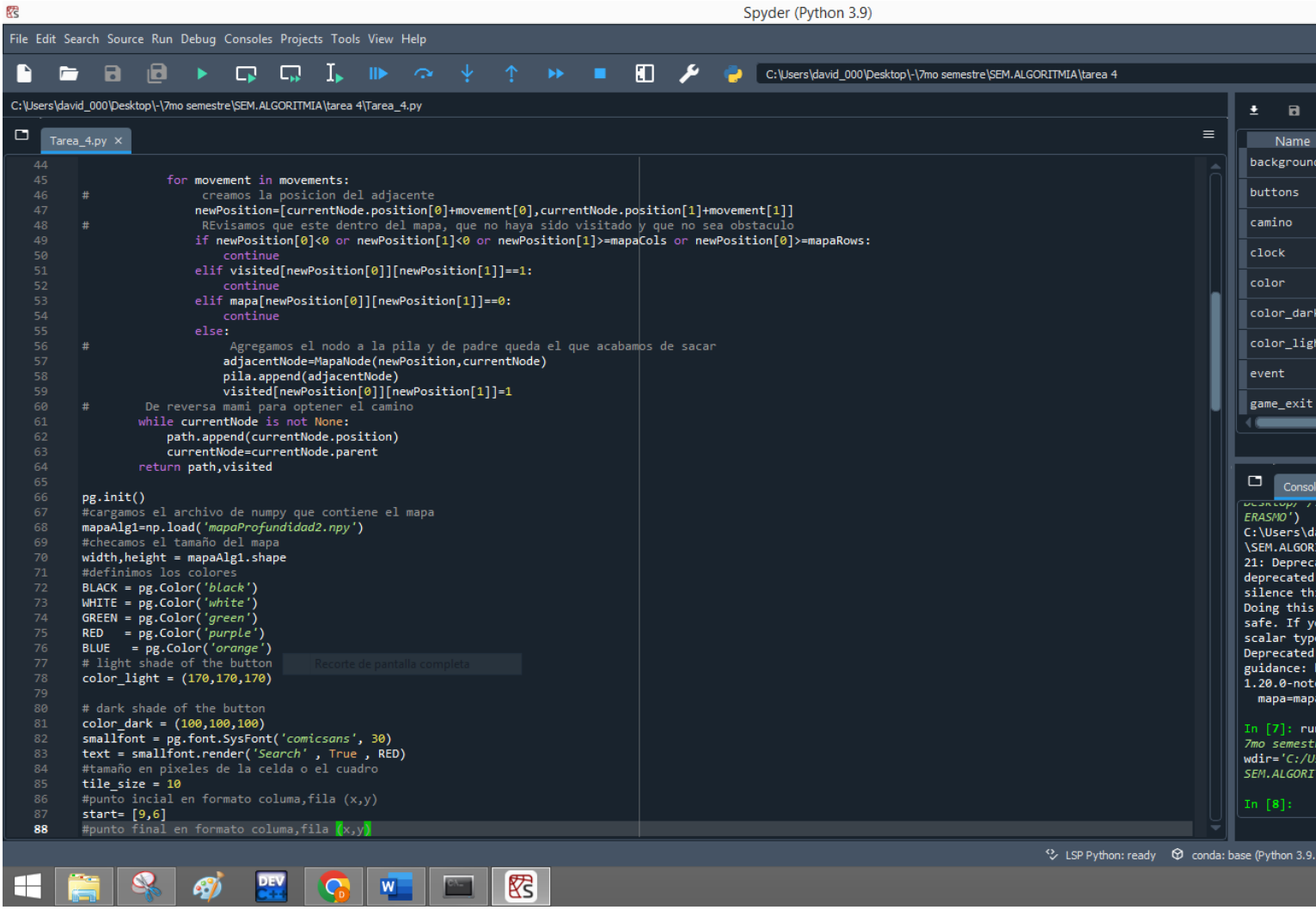
Seminario de Algoritmia

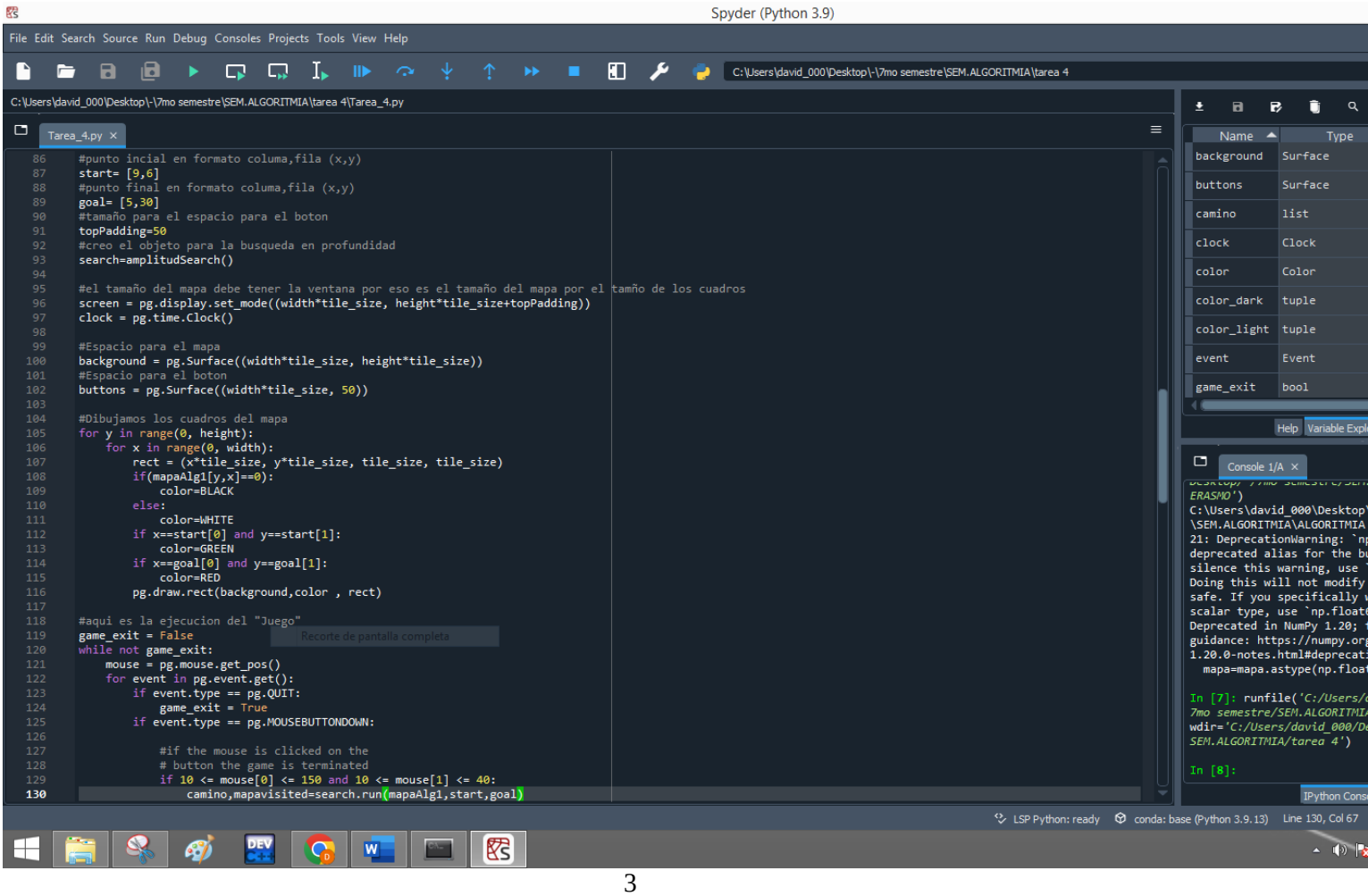
Reporte de práctica

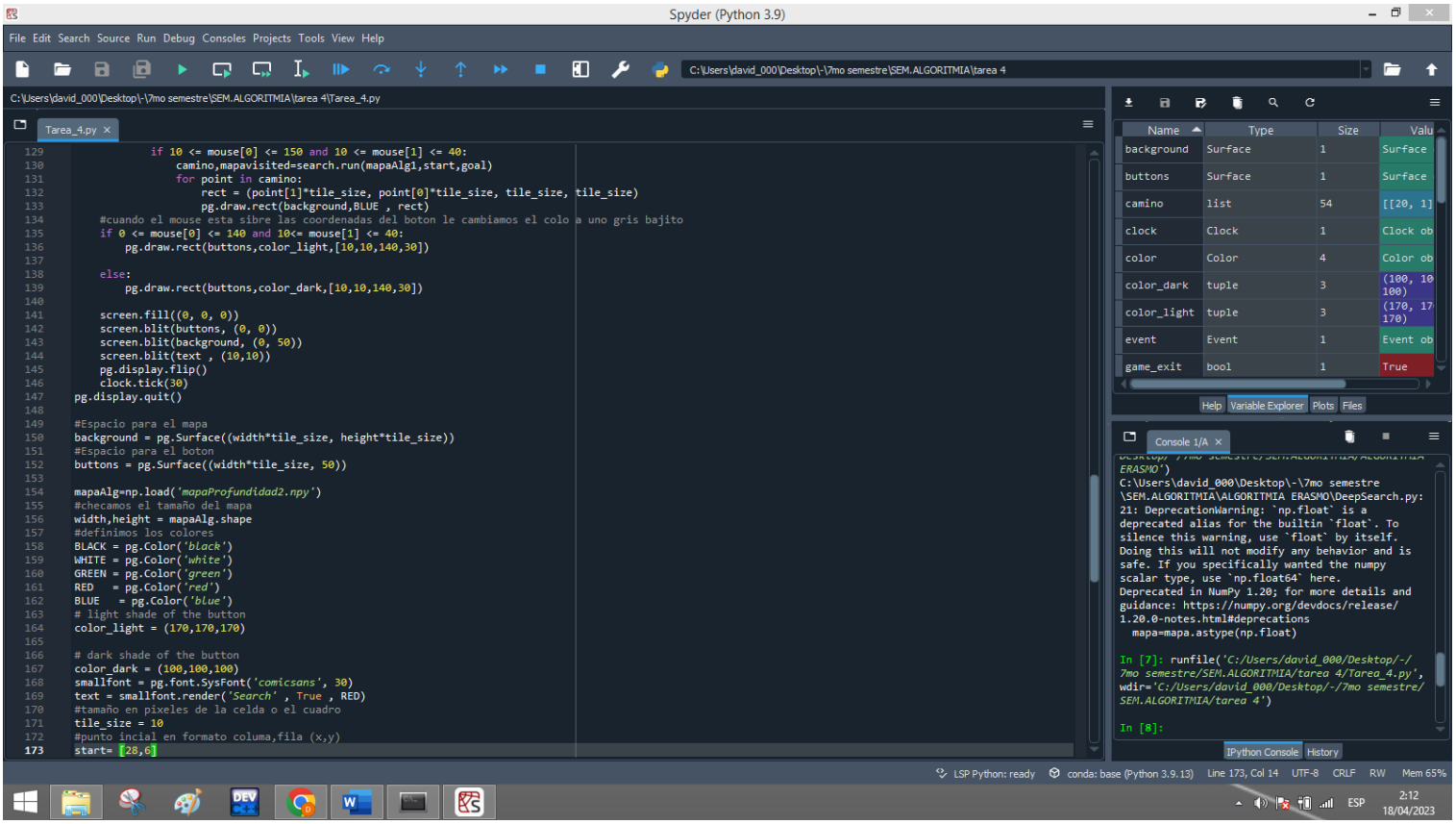
Nombre del alumno:	Ricardo David López Arellano
Profesor:	Erasmo Gabriel Martínez Soltero
Título de la práctica:	Tarea 4. Busquedas
Fecha:	23 febrero del 2023

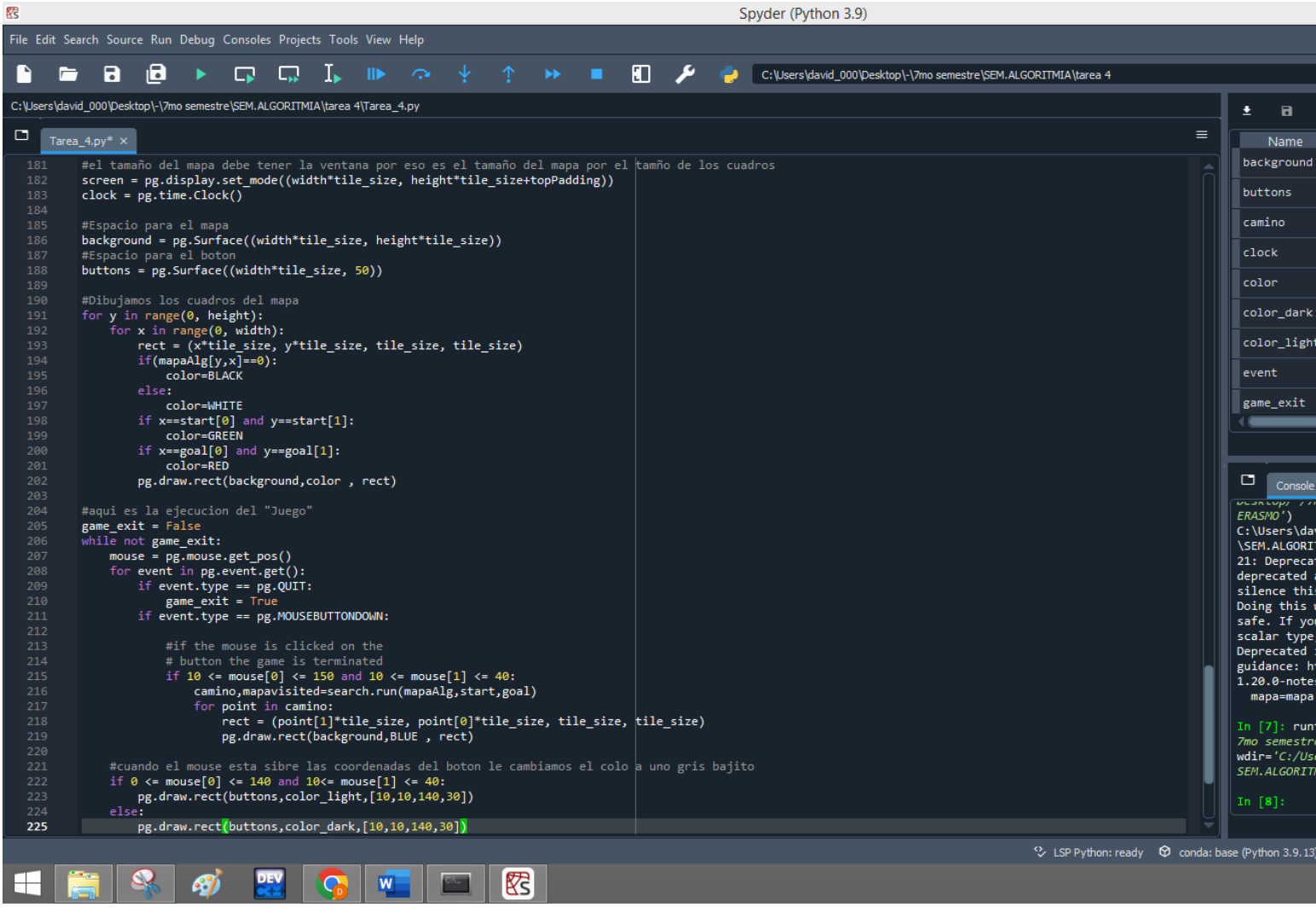
Capturas de pantalla del código



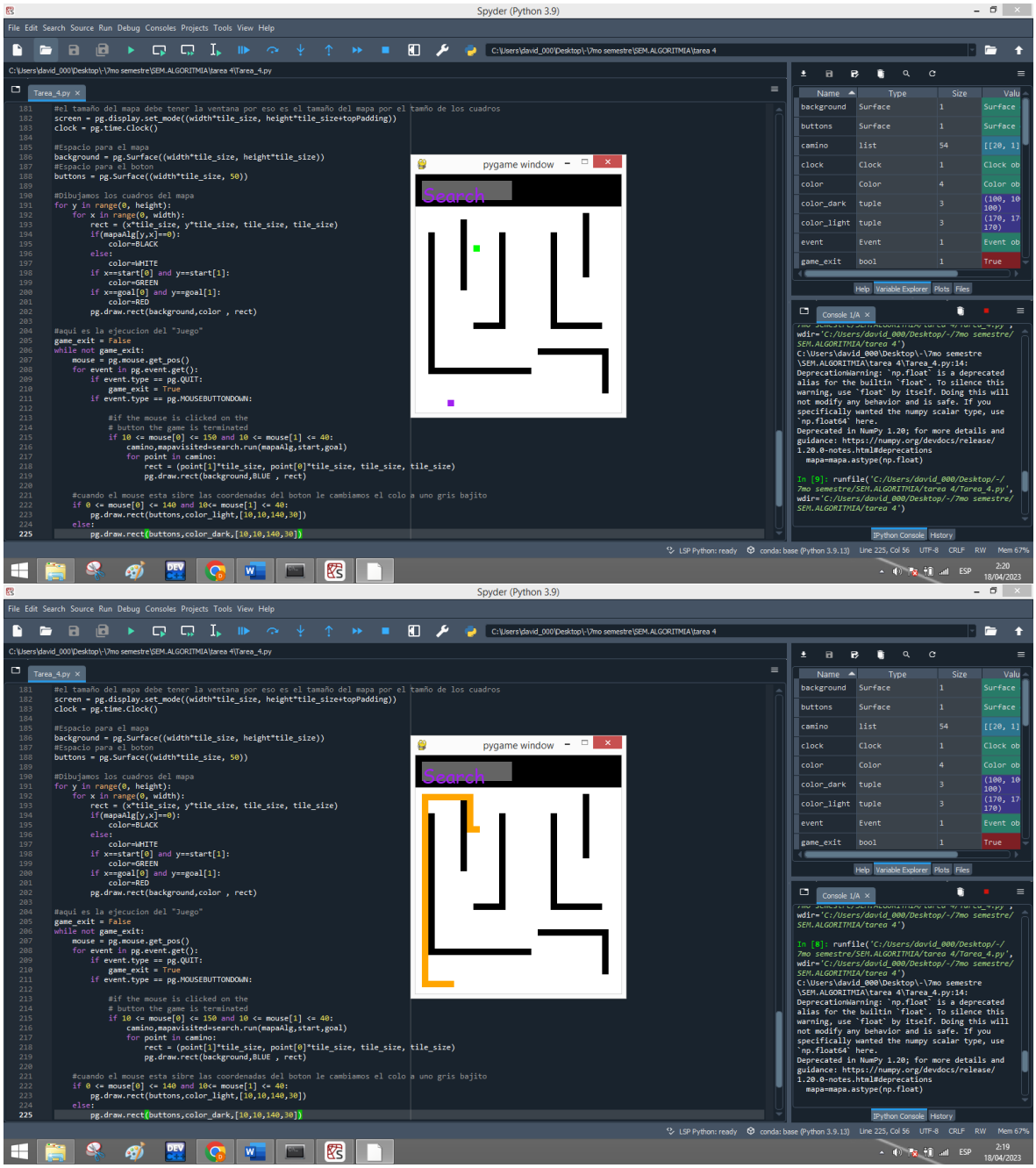

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\david_000\Desktop\-\7mo semestre\SEM.ALGORITMIA\tarea 4
C:\Users\david_000\Desktop\-\7mo semestre\SEM.ALGORITMIA\tarea 4\Tarea_4.py
Tarea_4.py
for movement in movements:
# creamos la posicion del adjacente
newPosition=[currentNode.position[0]+movement[0],currentNode.position[1]+movement[1]]
# REvisamos que este dentro del mapa, que no haya sido visitado y que no sea obstaculo
if newPosition[0]<0 or newPosition[1]<0 or newPosition[1]>=mapaCols or newPosition[0]>=mapaRows:
continue
elif visited[newPosition[0]][newPosition[1]]==1:
continue
elif mapa[newPosition[0]][newPosition[1]]==0:
continue
else:
# Agregamos el nodo a la pila y de padre queda el que acabamos de sacar
adjacentNode=MapaNode(newPosition,currentNode)
pila.append(adjacentNode)
visited[newPosition[0]][newPosition[1]]=1
# De reversa mami para optener el camino
while currentNode is not None:
path.append(currentNode.position)
currentNode=currentNode.parent
return path,visited
pg.init()
#cargamos el archivo de numpy que contiene el mapa
mapaAlg1=np.load('mapaProfundidad2.npy')
#checamos el tamaño del mapa
width,height = mapaAlg1.shape
#definimos los colores
BLACK = pg.Color('black')
WHITE = pg.Color('white')
GREEN = pg.Color('green')
RED = pg.Color('purple')
BLUE = pg.Color('orange')
# light shade of the button
color_light = (170,170,170)
# dark shade of the button
color_dark = (100,100,100)
smallfont = pg.font.SysFont('comicsans', 30)
text = smallfont.render('Search' , True , RED)
#tamaño en pixeles de la celda o el cuadro
tile_size = 10
#punto incial en formato columa,fila (x,y)
start= [9,6]
#punto final en formato columa,fila (x,y)
Recorte de pantalla completa
Name
background
buttons
camino
clock
color
color_dark
color_light
event
game_exit
In [7]: ru
In [8]:
LSP Python: ready
conda: base (Python 3.9.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\david_000\Desktop\-\7mo semestre\SEM.ALGORITMIA\tarea 4
C:\Users\david_000\Desktop\-\7mo semestre\SEM.ALGORITMIA\tarea 4\Tarea_4.py
Tarea_4.py
#punto incial en formato columa,fila (x,y)
start= [9,6]
#punto final en formato columa,fila (x,y)
goal= [5,30]
#tamaño para el espacio para el boton
topPadding=50
#creo el objeto para la busqueda en profundidad
search=amplitudSearch()
#el tamaño del mapa debe tener la ventana por eso es el tamaño del mapa por el tamño de los cuadros
screen = pg.display.set_mode((width*tile_size, height*tile_size+topPadding))
clock = pg.time.Clock()
#Espacio para el mapa
background = pg.Surface((width*tile_size, height*tile_size))
#Espacio para el boton
buttons = pg.Surface((width*tile_size, 50))
#Dibujamos los cuadros del mapa
for y in range(0, height):
for x in range(0, width):
rect = (x*tile_size, y*tile_size, tile_size, tile_size)
if(mapaAlg1[y,x]==0):
color=BLACK
else:
color=WHITE
if x==start[0] and y==start[1]:
color=GREEN
if x==goal[0] and y==goal[1]:
color=RED
pg.draw.rect(background,color , rect)
#aqui es la ejecucion del "Juego"
game_exit = False
Recorte de pantalla completa
while not game_exit:
mouse = pg.mouse.get_pos()
for event in pg.event.get():
if event.type == pg.QUIT:
game_exit = True
if event.type == pg.MOUSEBUTTONDOWN:
#if the mouse is clicked on the
# button the game is terminated
if 10 <= mouse[0] <= 150 and 10 <= mouse[1] <= 40:
camino,mapavisited=search.run(mapaAlg1,start,goal)
Name
Type
background
Surface
buttons
Surface
camino
list
clock
Clock
color
Color
color_dark
tuple
color_light
tuple
event
Event
game_exit
bool
In [7]: runfile('C:/Users/
In [8]:
LSP Python: ready
conda: base (Python 3.9.13)
Line 130, Col 67

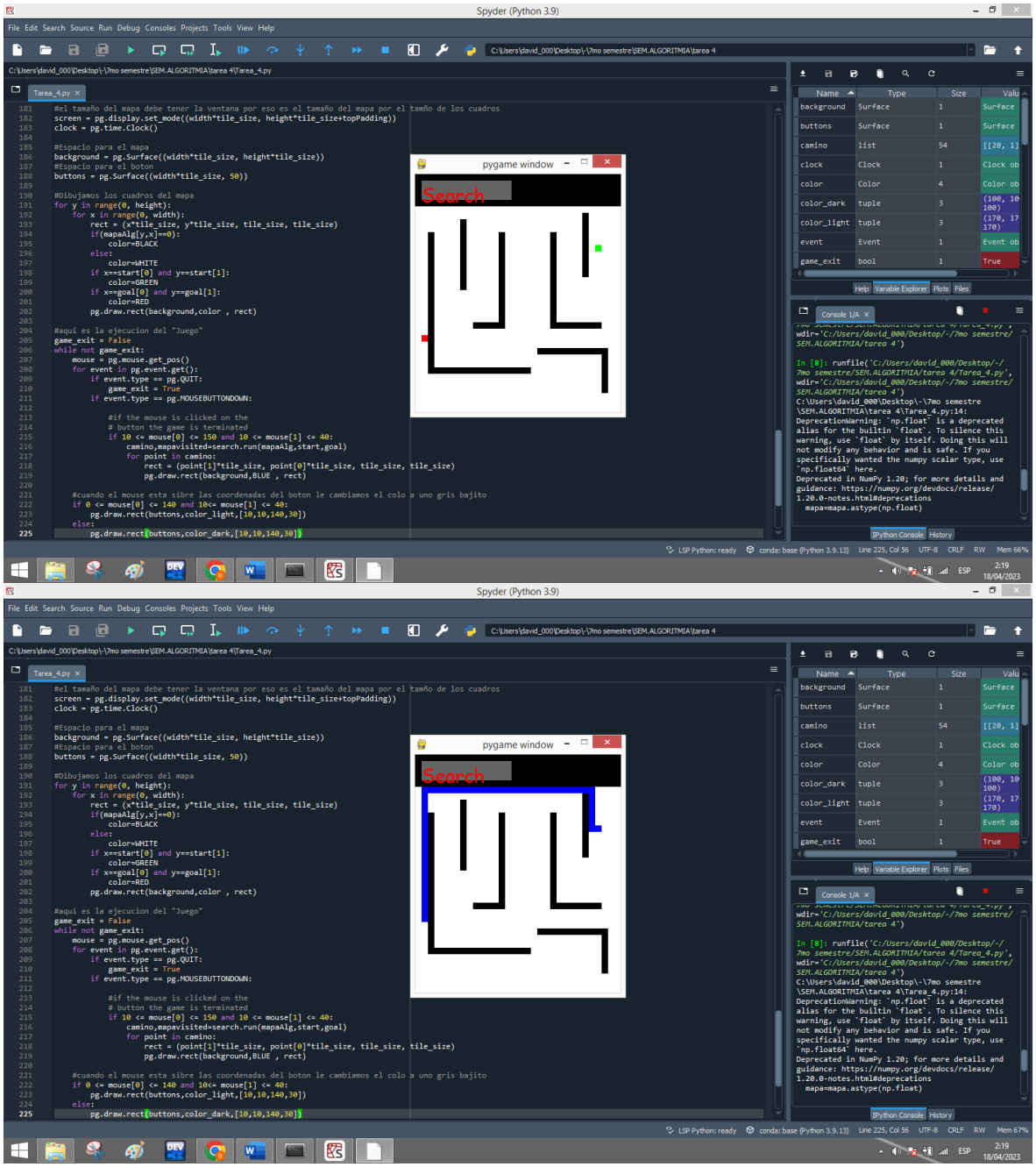

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\david_000\Desktop\-\7mo semestre\SEM.ALGORITMIA\tarea 4
C:\Users\david_000\Desktop\-\7mo semestre\SEM.ALGORITMIA\tarea 4\Tarea_4.py
Tarea_4.py
if 10 <= mouse[0] <= 150 and 10 <= mouse[1] <= 40:
camino,mapavisited=search.run(mapaAlg1,start,goal)
for point in camino:
rect = (point[1]*tile_size, point[0]*tile_size, tile_size, tile_size)
pg.draw.rect(background,BLUE , rect)
#cuando el mouse esta sibre las coordenadas del boton le cambiamos el colo a uno gris bajito
if 0 <= mouse[0] <= 140 and 10<= mouse[1] <= 40:
pg.draw.rect(buttons,color_light,[10,10,140,30])
else:
pg.draw.rect(buttons,color_dark,[10,10,140,30])
screen.fill((0, 0, 0))
screen.blit(buttons, (0, 0))
screen.blit(background, (0, 50))
screen.blit(text , (10,10))
pg.display.flip()
clock.tick(30)
pg.display.quit()
#Espacio para el mapa
background = pg.Surface((width*tile_size, height*tile_size))
#Espacio para el boton
buttons = pg.Surface((width*tile_size, 50))
mapaAlg=np.load('mapaProfundidad2.npy')
#checamos el tamaño del mapa
width,height = mapaAlg.shape
#definimos los colores
BLACK = pg.Color('black')
WHITE = pg.Color('white')
GREEN = pg.Color('green')
RED = pg.Color('red')
BLUE = pg.Color('blue')
# light shade of the button
color_light = (170,170,170)
# dark shade of the button
color_dark = (100,100,100)
smallfont = pg.font.SysFont('comicsans', 30)
text = smallfont.render('Search' , True , RED)
#tamaño en pixeles de la celda o el cuadro
tile_size = 10
#punto incial en formato columa,fila (x,y)
start= [28,6]
Name
Type
Size
Value
background
Surface
1
Surface
buttons
Surface
1
Surface
camino
list
54
[[20, 1]
clock
Clock
1
Clock ob
color
Color
4
Color ob
color_dark
tuple
3
(100, 10 100)
color_light
tuple
3
(170, 17 170)
event
Event
1
Event ob
game_exit
bool
1
True
Help
Variable Explorer
Plots
Files
Console 1/A
ERASMO')
C:\Users\david_000\Desktop\-\7mo semestre \SEM.ALGORITMIA\ALGORITMIA ERASMO\DeepSearch.py: 21: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
mapa=mapa.astype(np.float)
In [7]: runfile('C:/Users/david_000/Desktop/-/7mo semestre/SEM.ALGORITMIA/tarea 4/Tarea_4.py', wdir='C:/Users/david_000/Desktop/-/7mo semestre/SEM.ALGORITMIA/tarea 4')
In [8]:
IPython Console
History
LSP Python: ready
conda: base (Python 3.9.13)
Line 173, Col 14
UTF-8
CRLF
RW
Mem 65%
ESP
2:12
18/04/2023


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\david_000\Desktop\-\7mo semestre\SEM.ALGORITMIA\tarea 4
C:\Users\david_000\Desktop\-\7mo semestre\SEM.ALGORITMIA\tarea 4\Tarea_4.py
Tarea_4.py*
#el tamaño del mapa debe tener la ventana por eso es el tamaño del mapa por el tamño de los cuadros
screen = pg.display.set_mode((width*tile_size, height*tile_size+topPadding))
clock = pg.time.Clock()
#Espacio para el mapa
background = pg.Surface((width*tile_size, height*tile_size))
#Espacio para el boton
buttons = pg.Surface((width*tile_size, 50))
#Dibujamos los cuadros del mapa
for y in range(0, height):
for x in range(0, width):
rect = (x*tile_size, y*tile_size, tile_size, tile_size)
if(mapaAlg[y,x]==0):
color=BLACK
else:
color=WHITE
if x==start[0] and y==start[1]:
color=GREEN
if x==goal[0] and y==goal[1]:
color=RED
pg.draw.rect(background,color , rect)
#aqui es la ejecucion del "Juego"
game_exit = False
while not game_exit:
mouse = pg.mouse.get_pos()
for event in pg.event.get():
if event.type == pg.QUIT:
game_exit = True
if event.type == pg.MOUSEBUTTONDOWN:
#if the mouse is clicked on the
# button the game is terminated
if 10 <= mouse[0] <= 150 and 10 <= mouse[1] <= 40:
camino,mapavisited=search.run(mapaAlg,start,goal)
for point in camino:
rect = (point[1]*tile_size, point[0]*tile_size, tile_size, tile_size)
pg.draw.rect(background,BLUE , rect)
#cuando el mouse esta sibre las coordenadas del boton le cambiamos el colo a uno gris bajito
if 0 <= mouse[0] <= 140 and 10<= mouse[1] <= 40:
pg.draw.rect(buttons,color_light,[10,10,140,30])
else:
pg.draw.rect(buttons,color_dark,[10,10,140,30])
Name
background
buttons
camino
clock
color
color_dark
color_light
event
game_exit
Console
LSP Python: ready
conda: base (Python 3.9.13)

Resultados



MAPA 1



MAPA 2

Conclusiones

Gracias a esta practica me sirvio para ver como funciona la libreria pygame y comograficar en ella y que sirve para mapear y graficar grafos, y el manejo de ellos y como esel uso de la busqueda profunda y busqueda en amplitud y ver la diferencia entre una y otra, se logro la practica y fue bastante interesante de hacer y entender la realizacion de grafos y el uso de estas librerías.