

Algoritmo para obtener nodos

DISTANCIA_NODOS = 10 #esta en pix, ojo con eso

DEMANDA_NODO = 20

Tomando n como el id del nodo:

diccionario_nodo = {} # La idea es que sea del tipo diccionario_nodo[n] = [x_pix, y_pix, demanda]

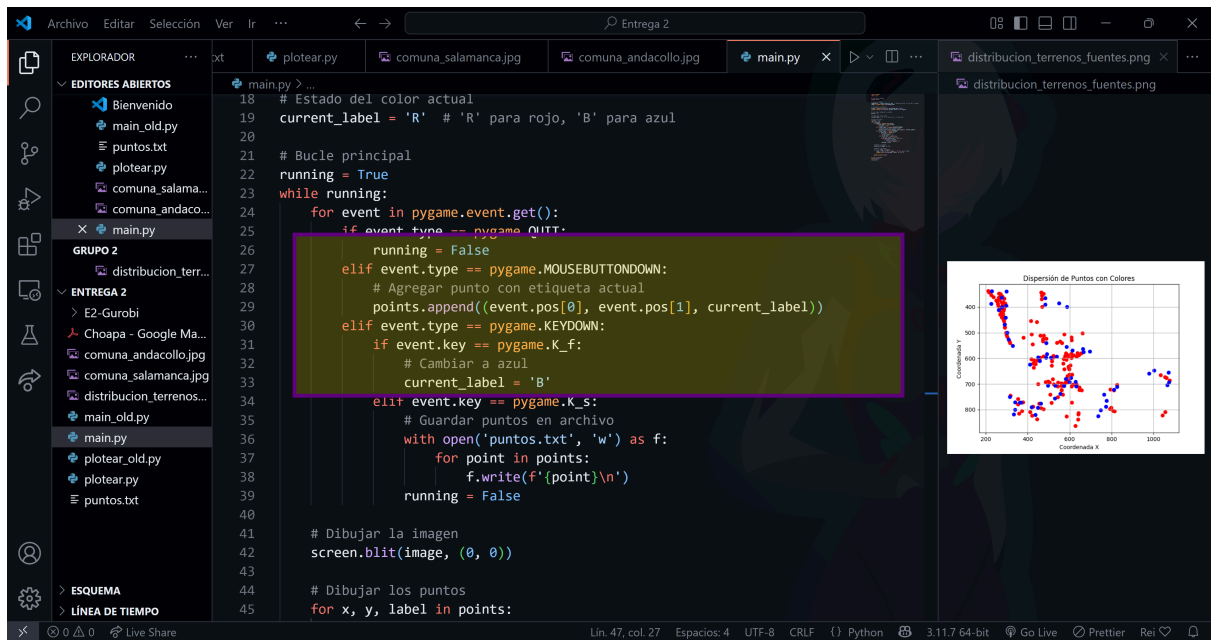
Contador_nodos = 0

#Lo de abajo es una vez que ingresamos al manejador de eventos de pygame.

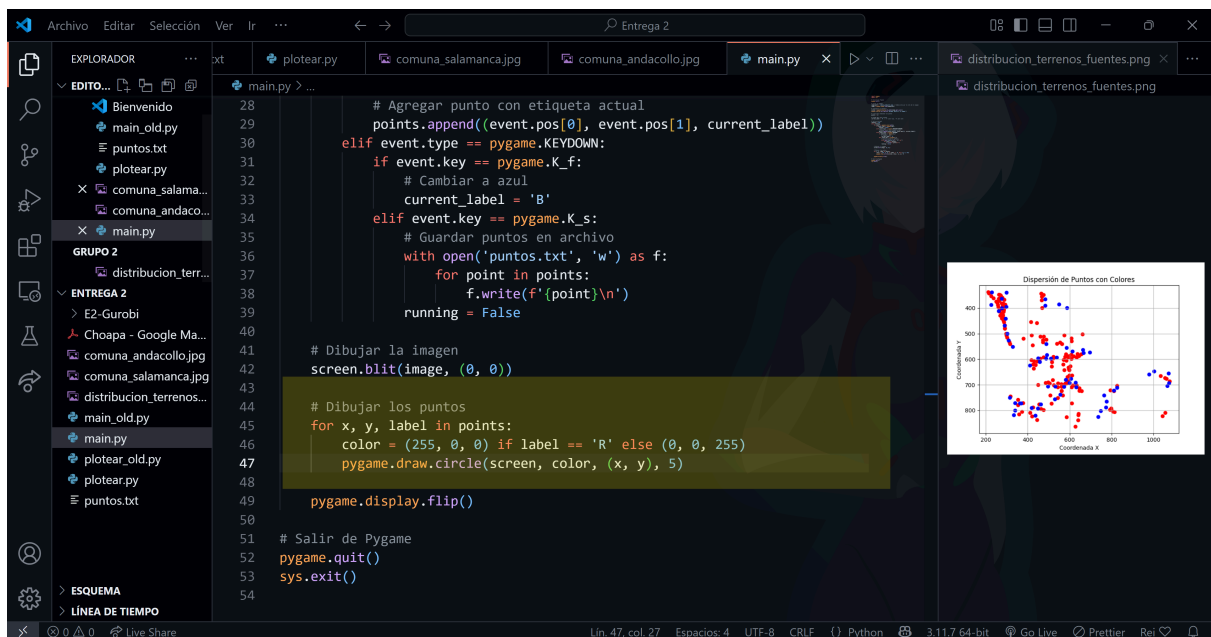
1. Detectamos el punto en la pantalla (x_new, y_new)
2. Iteramos por todas las llaves del diccionario.
3. Si la distancia euclideana entre el punto actual y el punto a ingresar es menor a DISTANCIA_NODO, Pasamos a 4, si no, pasamos a 5
4. La distancia es corta entre los puntos. Como tenemos un diccionario, le sumamos al tercer valor de la lista una demanda mas.
5. Es un nuevo punto, sumamos uno al contador y guardamos
diccionario_nodos[n] = [x_new, y_new, DEMANDA]

Luego podríamos ver graficamente esto iterando por el diccionario y graficando los puntos presionados y la intensidad del color puede estar en proporcion con la demanda guardada de ese nodo.

Una vez que recibimos un punto, debemos trabajar la lógica. Aquí podemos ver que en la línea 36 se guardan los puntos al presionar K y al presionar F cambiamos a ubicar las fuentes.



Aquí se realiza la parte gráfica (sólo en pygame).



Luego para poder formar un gráfico y sólo conservar los nodos, nos la imagen de fondo podemos plotear en matplotlib. Notar que debemos invertir el eje y (línea 24)

