Fase 1 – Representación de datos.

* Laberintos, posiciones, movimiento, galletas:

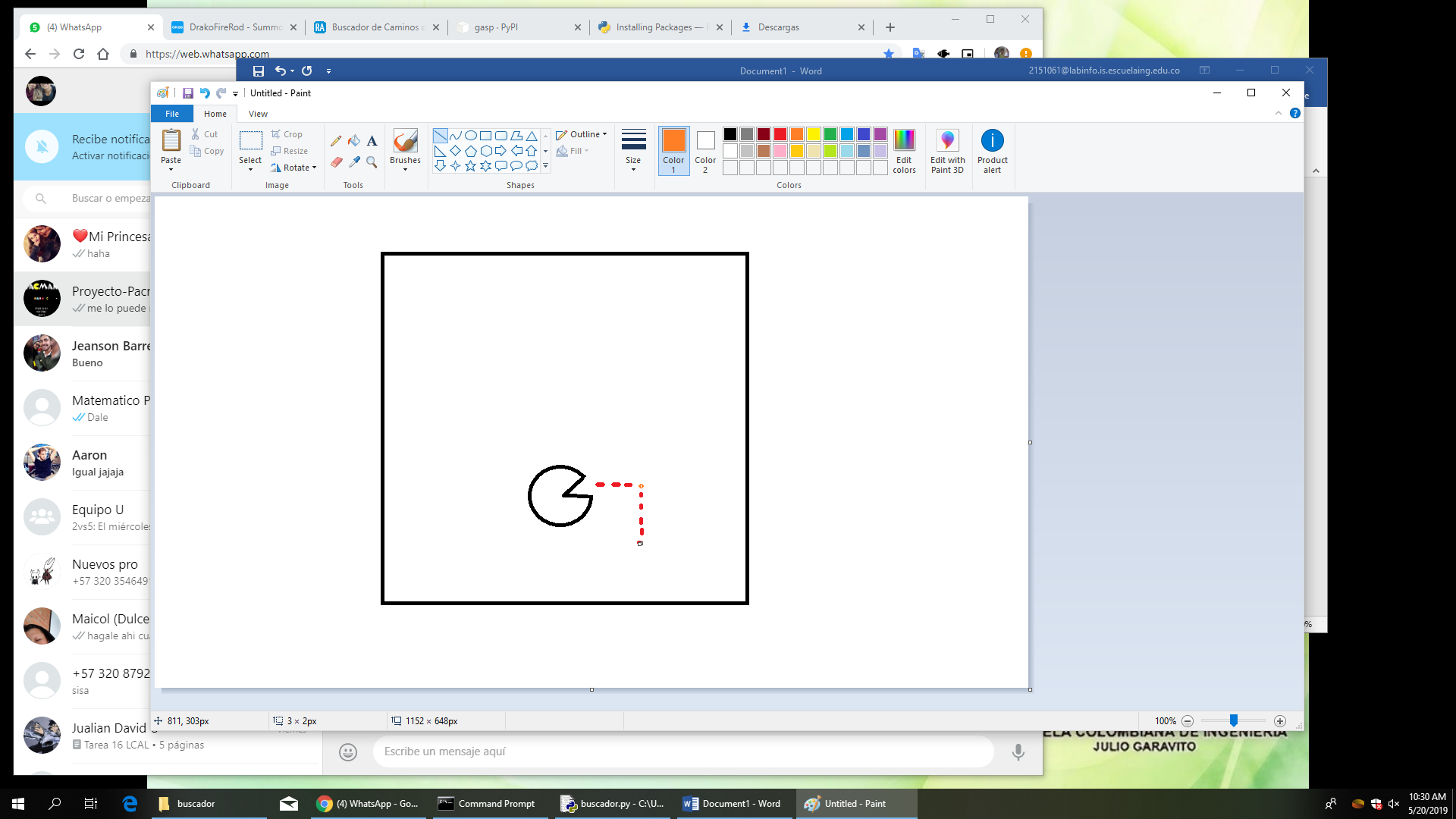
Para la representación del laberinto de usaron arreglos, donde el usuario ingresa las dimensiones (n filas x m columnas) del laberinto, así como la posición final del agente (). La matriz está representada con 1, 0, “x” y “+”, donde 1 son paredes, 0 galletas, “x” el agente y “+” fantasmas.

* Representación de salida:

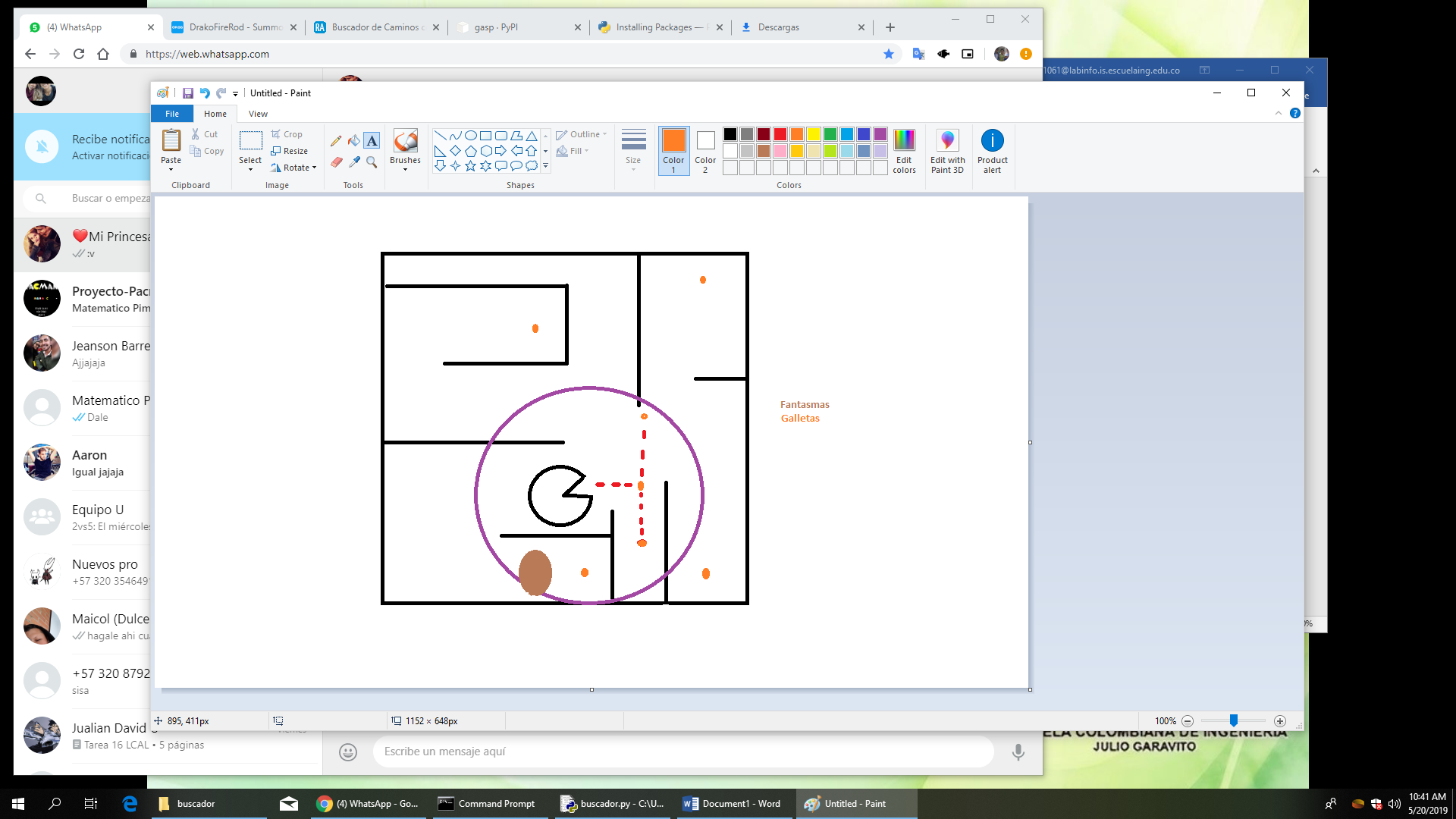
Se mostrará la matriz por cada paso del agente hasta que haya completado la ruta más óptima según se establezca (Llegar a un punto específico o comer todas las galletas).

Fase 2, Fase 3 – Ruta específica a un punto del laberinto, Comer todas las galletas.

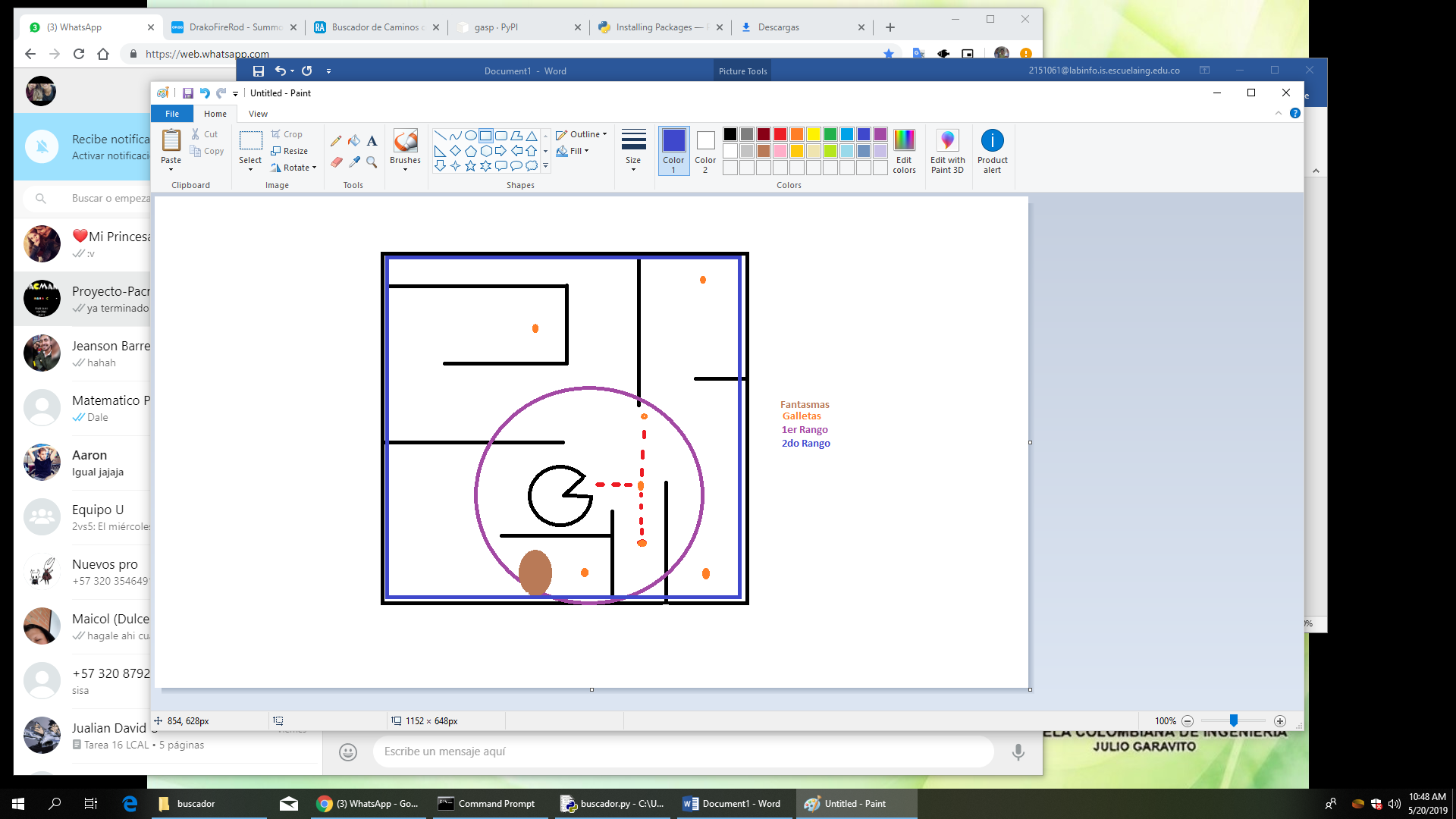
Al comenzar el Proyecto planteamos varias ideas para el movimiento de el pac-man la primera de ellas fue que cada galleta fuera un nudo de un grafo y de esta manera empezamos a plantear el programa con base en eso.



Pero al llegar a la parte de implementar los fantasmas y hacer que este fantasma corriera de el nos dimos cuenta que esta ideología no funcionaba debido a el que pac-man seguirá ciegamente hasta el fantasma y solo se daría cuenta de él tarde y provocaría su muerte entonces empezamos a plantear otra posible solución de crear un rango para que detectara las galletas y el grafo recorriera este mapa en ese rango hasta terminar las galletas y en este rango el pacman pudiera detectar los fantasmas y cambiar su trayectoria con respecto a las posiciones de los fantasmas.



Pero al crearlo de esta manera nos dimos cuenta que al crear este rango hay la posibilidad de que el pac-man considere que ya ha terminado de comer todas las galletas si en este rango no hay lo cual no nos funcionara así que probamos hacer dos tipos de rango uno que sea el que detecta los fantasmas y el Segundo que sea todo el mapa para que cuando el primer rango deje de funcionar el Segundo pueda entrar y llevar el pac-man a la siguiente galleta y cuando la galleta entre en el rango pequeño este tome nuevamente el mando para poder detectar los fantasmas y las galletas.



**¿Como pensábamos realizar esta idea?**

Llevariamos una matriz para el rango mayor que pues seria una matriz de todo el mapa y una submatriz para el rango más pequeño.

Creamos el código para sacar una submatriz de una matriz con respecto a unas dimensiones específicas y realizar los grafos para recorrer todas las galletas desde la primera implementación pero al mezclar los dos codigo generaba varios errores como atravesar paredes crear submatrices vacías.

Y empezamos a replantearnos la forma de realizarlo hasta que se nos ocurrió utilizar backtracking pero como podriamos implementar esta técnica para que recorra todo el mapa pensamos en que en mes de recorrer de un punto a otro en específico quitar este punto y que el pac-man recorriera en busca de las galletas.

Fase 4 – Fantasmas.

a) El algoritmo que genera el laberinto está diseñado de tal manera que en las cuatro esquinas del laberinto no haya obstáculo para dar espacio para colocar un fantasma, así que la posición inicial de estos cuatro será en las cuatro esquinas del laberinto.

b) Se tienen posiciones siempre disponibles como iniciales como fantasmas, no dando pie a sobreposiciones. La solución se limita a cuatro únicas posiciones iniciales para fantasmas.

c) El usuario podría elegir las posiciones donde los fantasmas inician.

Fase 5 – Fantasmas móviles.

a) Cada fantasma se moverá de manera aleatoria a través del laberinto teniendo en cuenta los obstáculos.

b) El sistema se limita a un único tipo de movimiento sin tomar en cuenta la presencia del agente.

c) Un sistema dirigido tomando en cuenta al agente y su posición, donde cada fantasma actúe de manera distinta como en el juego original.

d) Un único algoritmo .