



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

Materia:

Inteligencia Artificial

Trabajo:

Proyecto de la Aspiradora

Grupo:

802-A

Equipo:

Aldaz Contreras Daniela

Garcia Garcia Andrea Catalina

Maciel Lopez Nydia Beatriz

Velazquez Zamora Samuel

13 de abril del 2021

Introducción

Un agente debe de tratar de hacer lo correcto según lo que perciba y las acciones que pueda realizar pero también hay que tener en cuenta las medidas del rendimiento.

Para este proyecto se realizó un programa en el lenguaje de programación Python, va a presentar un agente, una aspiradora, cuyo objetivo es recoger toda la basura colocada de manera aleatoria dentro del área designada, el cual será un tablero de 10x10.

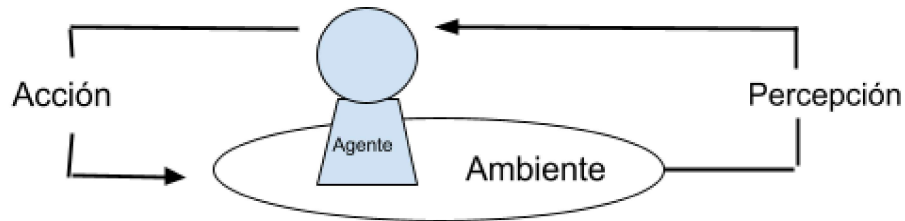
A continuación presentaremos el objetivo general del proyecto, un marco teórico con la información básica y necesaria para el planteamiento del problema, el desarrollo del código con una breve explicación, los resultados obtenidos y una conclusión al respecto.

Objetivo general

Aplicar los conocimientos que se obtuvieron en el primer parcial para poder concluir de manera satisfactoria el proyecto. Reconocer ¿Que es un agente? ¿Qué percibe un agente? ¿Cuáles son sus acciones? por medio de la simulación de una aspiradora de basura.

Marco teórico

Un **agente inteligente**, es una entidad capaz de percibir su entorno, procesar tales peticiones y responder o actuar en su entorno de manera racional, es decir, de manera correcta y teniendo a maximizar un resultado esperado.



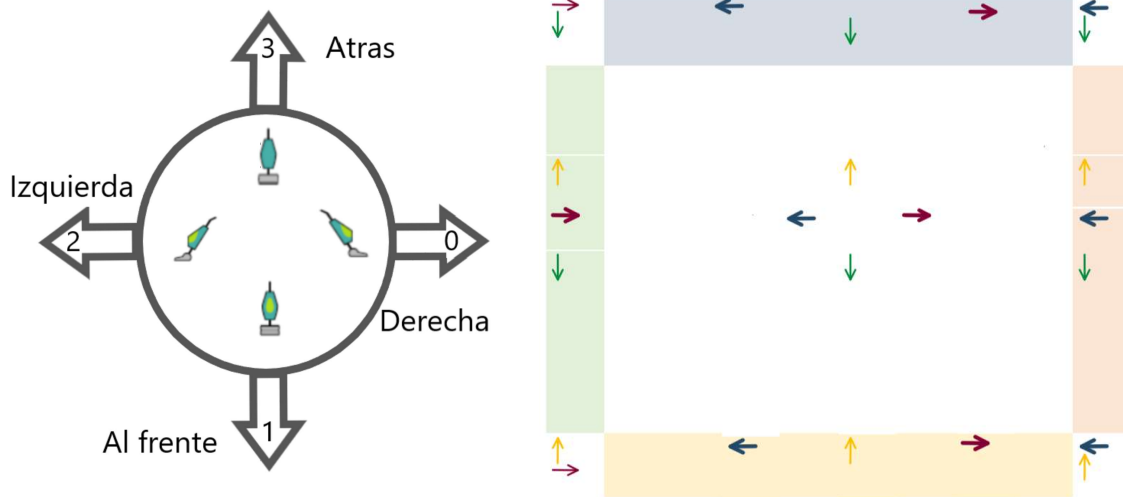
Un agente haciendo interacción con el medio ambiente.

La aspiradora debe percibir: posición y contenido, y si ha recogido basura. Puede realizar movimientos a la izquierda, derecha, arriba, abajo, acción de recoger o no hacer nada. La aspiradora debe saber si en el lugar que se encuentra está sucio, entonces debe aspirar y debe cambiar a otro lugar.

Desarrollo

El juego se desarrolló en el lenguaje python usando pydev y eclipse. Se usó una matriz de dos dimensiones para colocar las trece basuras que el agente recogerá, para encontrar las basuras implementamos un algoritmo de búsqueda lineal por renglón columna dependiendo la dirección. A partir de la posición del agente primero se asigna un vector de posibles direcciones para buscar un objetivo (basura), si encuentra un objetivo alcanzable entonces retorna el número de pasos para llegar al objetivo. De no encontrar basura entonces se asigna una dirección y un número de casillas a recorrer aleatorias.

Dependiendo de las direcciones las imágenes de la aspiradora se cambiaban para generar una transición más agradable. Las direcciones empleadas fueron las siguientes:



La dirección a la cual podía dirigirse la aspiradora se generaba dependiendo de la posición en la que se encontrará, conforme la tabla de arriba.

Para la implementación del algoritmo de búsqueda, se usó la función `where` de `numpy`, que retorna el index del vector dada una condición. por lo que dependiendo de la dirección generada, se seleccionaba la matriz en vectores de fila o columna dependiendo el caso, para hacer uso de la función y retorna el número de casillas a recorrer para encontrar la basura.

Código

```
#-----INTERFAZ-----#
```

```
raiz = Tk()
raiz.title("Aspiradora")
raiz.resizable(0,0)
miFrame = Frame(raiz)
miFrame.pack(pady=5)
```

Se define la ventana y el frame

Se definen los tres botones en un grid, el cual se comporta como una tabla en Html.

```
#-----BOTONES-----#
```

```
BasuraBtn = Button(miFrame,text="Colocar Basura",width=10,command=lambda:generateGarbage())
BasuraBtn.grid(row=1,column=1,padx=10)
```

```
IniciarBtn = Button(miFrame,text="Iniciar",width=10,command=lambda:iniciarRec())
IniciarBtn.grid(row=1,column=2,padx=10)
IniciarBtn["state"] = "disable"
```

```
NuevoBtn = Button(miFrame,text="Nuevo",width=10,command=lambda:nuevoMap())
NuevoBtn.grid(row=1,column=3,padx=10)
NuevoBtn["state"] = "disable"
```

```
#-----CANVAS-----#
```

```
mapa = Canvas(raiz,width=501,height=601,bg="white")
mapa.pack()
drawGrid()
```

Se define el canvas y sus atributos
Luego de eso, se genera la matriz de 10x10

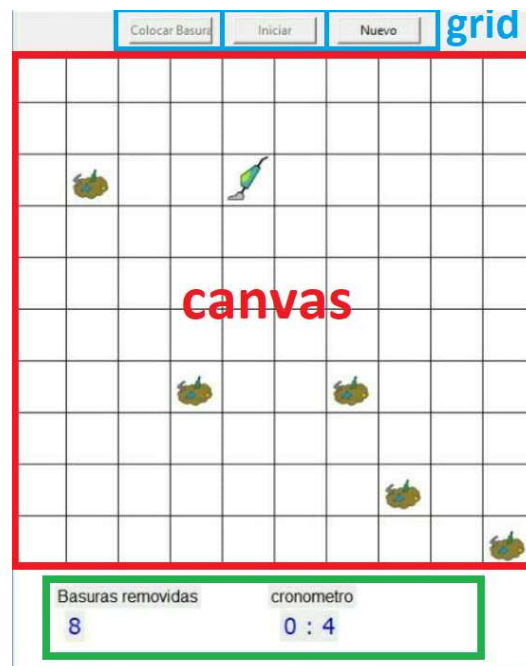
```
img = PhotoImage(file="50px/basura_50.png")
```

```
#vector de imagenes a usar#
```

```
asp.append(PhotoImage(file="50px/Asp_Der_50.png"))
asp.append(PhotoImage(file="50px/Asp_Fro_50.png"))
asp.append(PhotoImage(file="50px/Asp_Izq_50.png"))
asp.append(PhotoImage(file="50px/Asp_Bac_50.png"))
```

Se define las imagenes que iran cambiando
Dando la ilusion de que la aspiradora se mueve

```
aspiradora.append(mapa.create_image(pos[0][0]*50,pos[0][1]*50,anchor=NW,image=asp[0]))
aspiradora.append(mapa.create_image(pos[1][0]*50,pos[1][1]*50,anchor=NW,image=asp[1]))
aspiradora.append(mapa.create_image(pos[2][0]*50,pos[2][1]*50,anchor=NW,image=asp[2]))
aspiradora.append(mapa.create_image(pos[3][0]*50,pos[3][1]*50,anchor=NW,image=asp[3]))
```



```

def iniciarRec():
    global iActual
    global NV
    global mMapa
    global tiempoInicio
    tiempoInicio=datetime.datetime.now()
    IniciarBtn["state"] = "disable"
    i=0
    while( i<nBasuras):
        pasos=[]
        direc=definirDireccion(iActual)
        pasos=numeroPasos(direc)
        direccionFinal=direc[int(pasos[0])]

        j=0
        while((j<pasos[1]) & valoracion(direccionFinal)):
            basu.set(str(i))
            basur.config(textvariable=basu)
            if(NVO==True): break
            if (direccionFinal==0): #          hacia la aderecha
                cambialmage(0)
                pos[iActual][1]+=1
                mapa.move(aspiradora[iActual],50,0)
            elif(direccionFinal==1):          #al frente
                cambialmage(1)
                pos[iActual][0]+=1

            elif (direccionFinal==2): #          izquierda
                cambialmage(2)
                pos[iActual][1]-=1
                mapa.move(aspiradora[iActual],-50,0)
            else: #          atras
                cambialmage(3)
                pos[iActual][0]-=1
                mapa.move(aspiradora[iActual],0,-50)
            if (NVO==True):
                break
            sleep(.1)
            if mMapa[pos[iActual][0]][pos[iActual][1]]==1:
                quitarBasura()
                i+=1
            raiz.update()
            j+=1
        if mMapa[pos[iActual][0]][pos[iActual][1]]==1:
            quitarBasura()
            i+=1

```

Dependiendo de la posición del tablero se asigna un vector de direcciones a las cuales es posible dirigirse sin salir de los límites [0,1,2,3]

Retorna el número de casillas o pasos, en la que se encuentra la basura, así como el index de la dirección en la que se encuentra. Si no hay basura alcanzable, entonces envía numero de pasos aleatorio y una dirección aleatoria

Después de asignar la dirección se itera el número de casillas para eliminar la basura, en la direccion indicada.

Moviendo la aspiradora de casilla en casilla para simular la animación.


```

def generateGarbage():
    global mMapa
    global NVO

    x=0
    y=0
    for i in range(0,nBasuras):
        while True:
            x = randrange(10)
            y = randrange(10)
            if ((mMapa[x][y]==0) and (x+y>0) ):
                break
            mMapa[x][y] = 1
            vBasura.append([x,y])

        mapa.move(iBasura[i],(y-10)*50,(x-10)*50)
    BasuraBtn["state"] = "disable"
    IniciarBtn["state"] = "normal"
    NuevoBtn["state"] = "normal"
    NVO=False

```

Se genera posiciones aleatorias según el número de basuras generadas con anterioridad. Se coloca la posición en el la matriz de enteros mMapa con el valor 1 y se mueven las basuras que están fuera del canvas. Finalmente se desactiva en botón "Colocar basura" y se activan los botones "Iniciar" y "Nuevo"

En la función IniciarRec(), cuando la aspiradora cambia de posición comprueba si en la nueva posición hay una basura, en caso afirmativo, se manda a llamar a la función quitarBasura()

```

def quitarBasura():
    global mapa
    #Modifique que fueran solo 11 basuras#
    for i in range(0,nBasuras):
        if (len(vBasura)==0):
            break
        if (vBasura[i][0]==pos[iActual][0] and vBasura[i][1]==pos[iActual][1]):
            #sale
            mapa.move(iBasura[i],(10-vBasura[i][1])*50,(10-vBasura[i][0])*50)
            vBasura[i][0]=10
            vBasura[i][1]=10
            mMapa[pos[iActual][0]][pos[iActual][1]]=0
            mapa.update()

```

Esta función busca en el vector de posiciones de las basuras a la basura que tenga la misma posición de la aspiradora y la coloca fuera del canvas de manera que aparenta que desaparece.

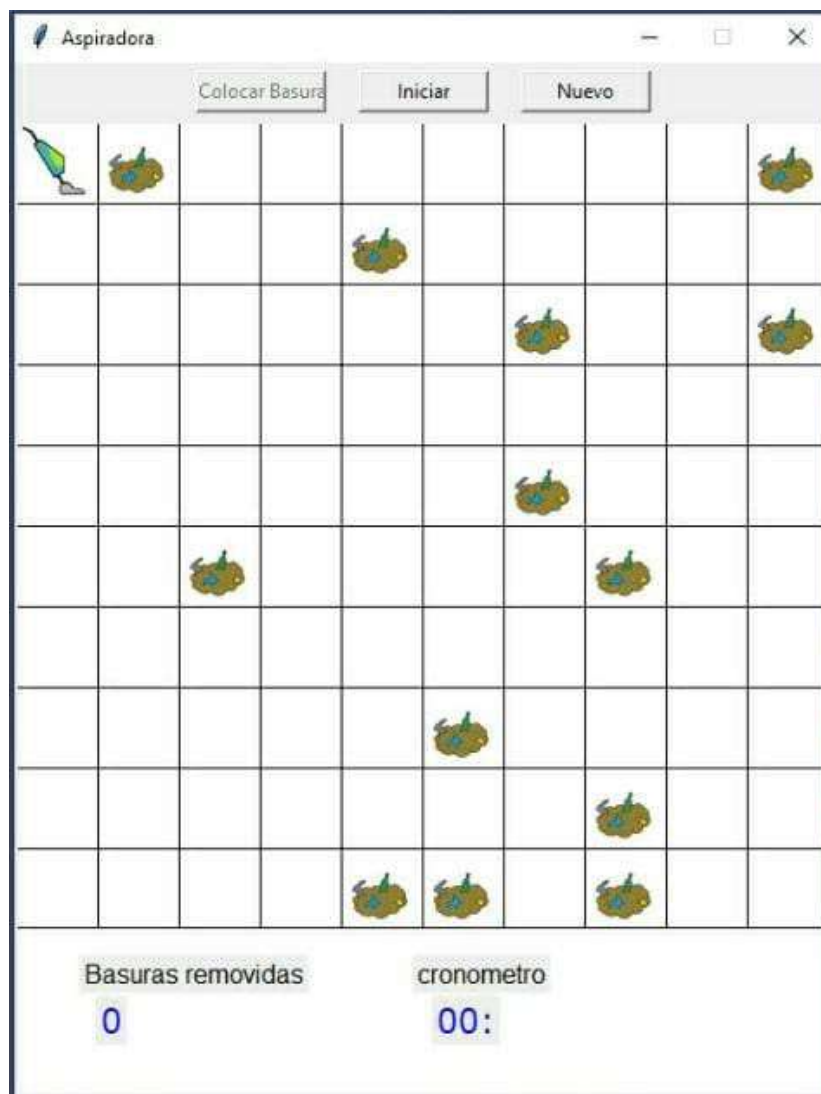
Se quita la posición de la matriz mMapa y se actualiza el canvas.

Resultados

A continuación mostraremos los resultados de la ejecución del programa.

Programa en funcionamiento

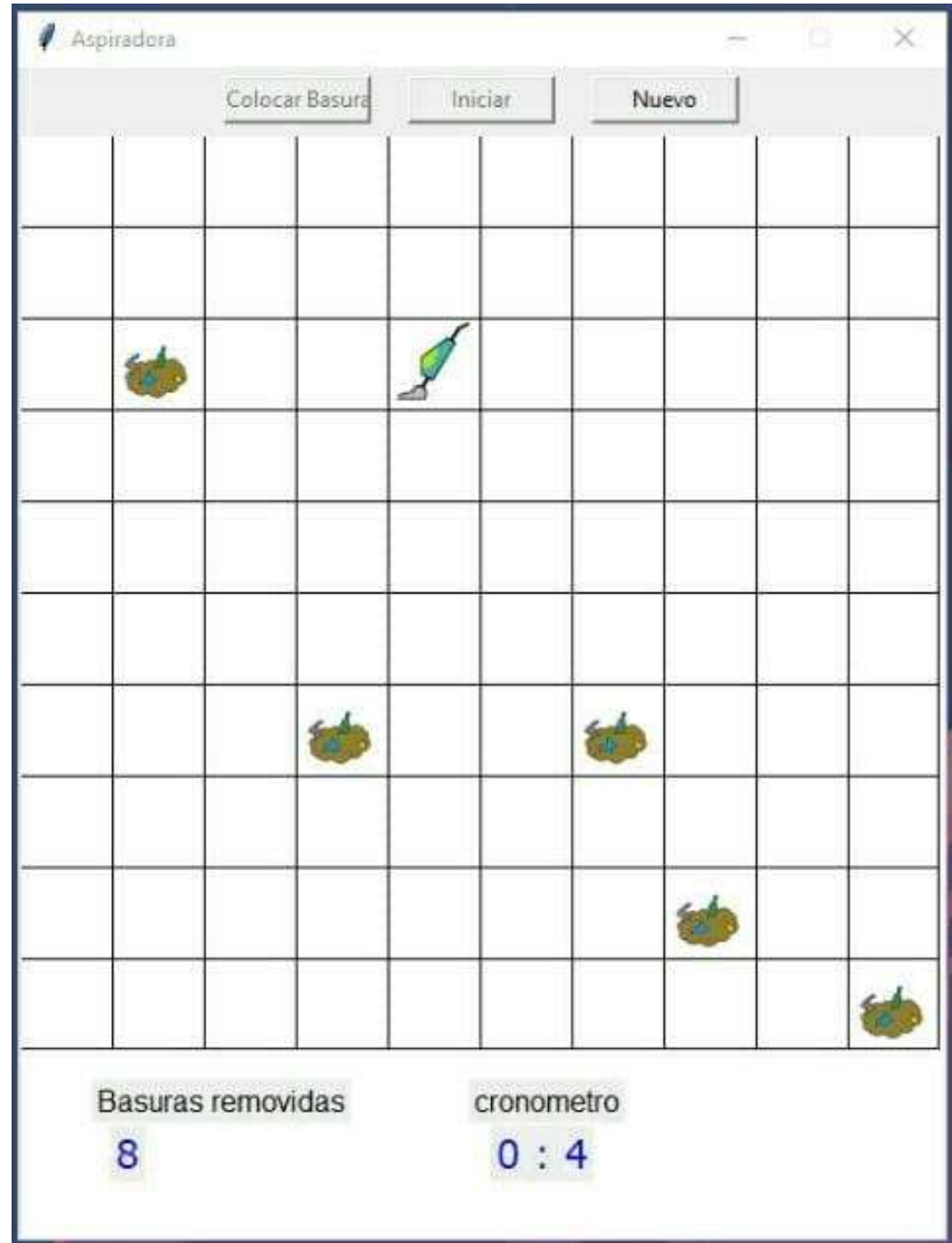
Nuestra interfaz cuenta con tres botones en la parte superior del programa; Colocar Basura, Iniciar, Nuevo. En la parte inferior se visualiza el número de basuras recogidas y un cronómetro del tiempo en que tarda en lograr el objetivo.



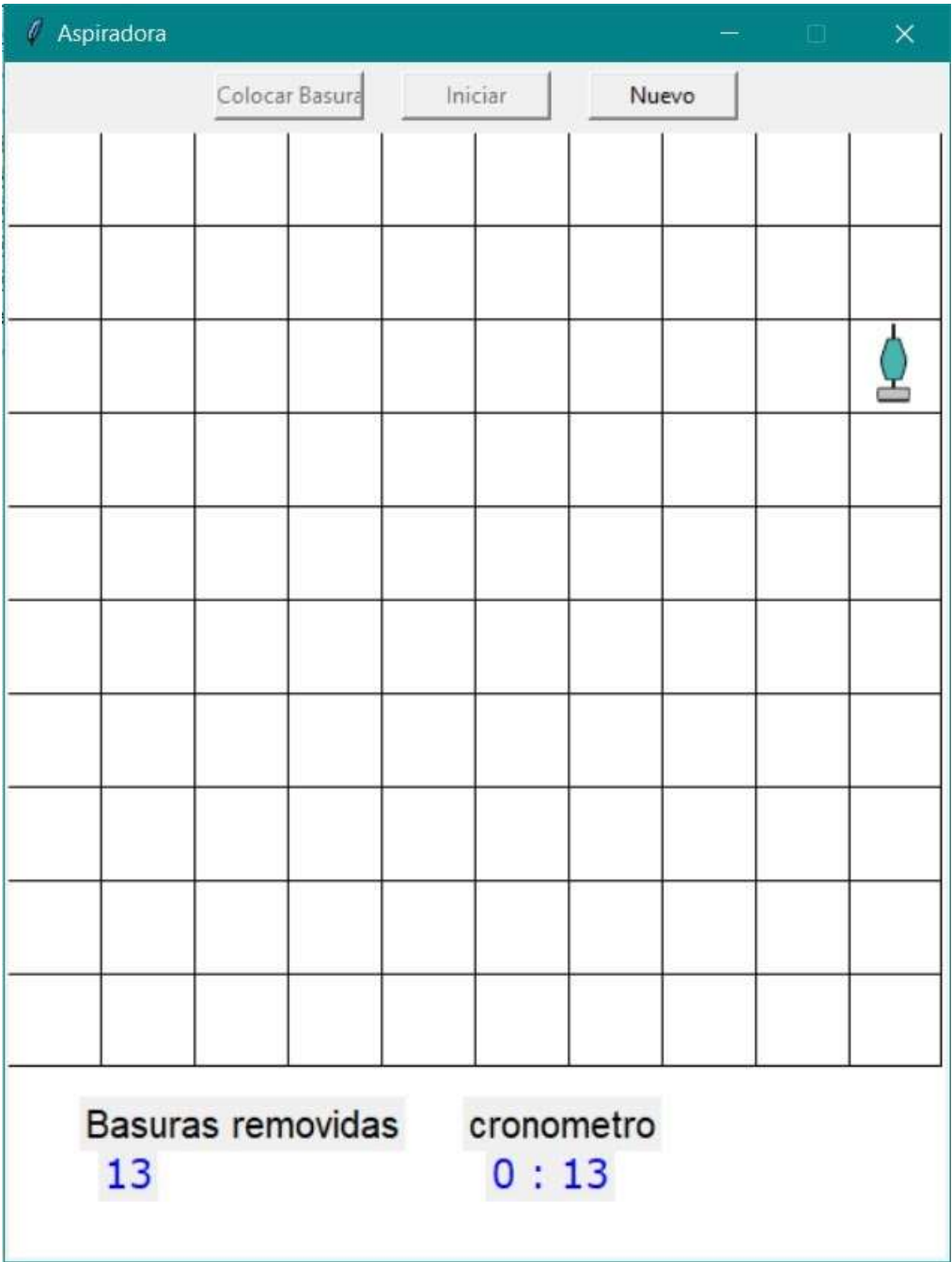
Primero tenemos que presionar el botón de **Colocar Basura**, una vez que nos muestra la basura aleatoriamente en el tablero procedemos a dar click en **Iniciar**.

Podemos observar que el contador de basura y el cronómetro están en ceros.

Cuando la aspiradora comienza a moverse por el tablero, se guía por la columna o fila en la que se encuentre buscando alguna basura, al ubicarla procede a avanzar para recogerla. El contador va aumentando conforme se vaya limpiando el área.



Una vez que la aspiradora haya terminado de limpiar toda el área del tablero, se queda en el último lugar que recogió la basura. El contador marca el total de basuras y el cronómetro el tiempo total en completar la tarea.



Para iniciar una nueva partida tendremos que presionar el botón **Nuevo** entonces la aspiradora se va a regresar a la posición (0,0) del tablero y estará listo para una nueva ejecución.

Conclusiones

Al finalizar con este proyecto podemos concluir que a pesar de no tener un algoritmo muy óptimo de búsqueda la aspiradora logró su objetivo en un tiempo considerable. Así como el funcionamiento de la aspiradora, en la vida cotidiana existen múltiples procesos y algoritmos que el ser humano adapta para la realización de tareas satisfactoriamente y que pueden sustituir al mismo.

El uso de la IA en la vida cotidiana facilita la vida de las personas en tareas rutinarias y día con día se van actualizando de manera que tienen un gran impacto en la sociedad. No podemos dejar de lado que conforme se van teniendo avances tecnológicos, se obtienen mejoras en la calidad de vida y también posibles riesgos al usar máquinas, como crear una dependencia hacia ellas, no obstante, podemos decir que éstas no pueden reemplazar por completo al ser humano ya que carecen de consciencia propia.