



TECNOLÓGICO DE MONTERREY  
Campus Querétaro

PROFESORES ENCARGADOS:

Pedro Oscar Pérez Murueta  
Alejandro Fernández Vilchis  
Denisse Lizbeth Maldonado Flores

Modelación de sistemas multiagentes con  
gráficas computacionales

### *M1. Actividad*

Daniel Gutiérrez Gómez A01068056

FECHA DE ENTREGA:  
12/11/2023

Informe acerca de modelación de multiagentes. En esta actividad, se planteó una problemática en la que se busca modelar y analizar con el uso de librerías de programación que ayudan a la implementación de modelación de agentes. En este caso, es nuestra librería de apoyo para poder implementar este tipo de problemáticas. Ahora, se nos ha propuesto el caso en el que se habla de "aspiradoras" que limpian un suelo. Se nos proporciona un tiempo máximo de ejecución, dimensiones del espacio del suelo, número de agentes "limpiadoras", porcentaje de celdas inicialmente sucias. A partir de estos parámetros se busca obtener la modelación, no sin antes haber pasado por la etapa del diseño del algoritmo, y diseño de implementación para llegar a una solución válida. Cabe mencionar que, aunque no hay una solución como tal a este problema, podemos decir que hay aproximaciones y proponer mejoras en un ambiente que cambia dinámicamente como es el caso de esta actividad.

Una vez obtenido el diseño y la implementación, se han obtenido los siguientes resultados:

Para el primer análisis se tomará en cuenta que se tiene un espacio de piso de tamaño 100 x 100, se tiene un solo agente y 90% de celdas sucias.

Para esto, lo que se realizó fue ejecutar una vez primero con un agente y el 90% de celdas sucias, con un máximo de pasos de 10000, se eligieron esta cantidad de pasos para ver una proporción de uno por uno, lo cual es imposible, pero veamos qué es lo que nos demuestra esta simulación:

```

# Creación de instancia del modelo a ejecutar
MAX_STEPS = 10000
WIDTH = 100
HEIGHT = 100
PERCENTAGE_DIRTY = 0.90
NUM_AGENTS = 1
MAX_TIME = 1000
MODEL = CleaningModel(NUM_AGENTS, WIDTH, HEIGHT, MAX_STEPS, PERCENTAGE_DIRTY, MAX_TIME)
START_TIME = time.time()
TOTAL_STEPS = 0
IS_CLEANED = False
TOTAL_PERCENTAGE_CLEANED = 0.0
CELLS_COVERED_AGENT = 0

for i in range(MAX_STEPS):
    MODEL.step()
    if determine_clean(MODEL):
        TOTAL_STEPS += 1
        is_cleaned = True
        break
    TOTAL_STEPS += 1

```

No se ha limpiado el espacio completo :(, celdas limpiadas contando el porcentaje inicial: 3246

```

# Calculo de celdas limpiadas por el agente
CELLS_COVERED_AGENT = MODEL.cleaned_cells - MODEL.cells_to_cover

# Calcular porcentaje de celdas limpias
TOTAL_PERCENTAGE_CLEANED = (CELLS_COVERED_AGENT * 100) / (WIDTH * HEIGHT)
print("Porcentaje de celdas limpiadas: ", TOTAL_PERCENTAGE_CLEANED, " de un 100%")

```

✓ 0.0s

Porcentaje de celdas limpiadas: 22.47 de un 100%

Parece ser una buena cantidad, sin embargo viendo la proporción una celda limpia por cada paso que se da, vemos que no es muy buena proporción.

Incrementemos el número de pasos, esta vez a 10000.

```

MAX_STEPS = 100000
WIDTH = 100
HEIGHT = 100
PERCENTAGE_DIRTY = 0.90
NUM_AGENTS = 1
MAX_TIME = 1000
MODEL = CleaningModel(NUM_AGENTS, WIDTH, HEIGHT, MAX_STEPS, PERCENTAGE_DIRTY, MAX_TIME)
START_TIME = time.time()
TOTAL_STEPS = 0
IS_CLEANED = False
TOTAL_PERCENTAGE_CLEANED = 0.0
CELLS_COVERED_AGENT = 0

✓ for i in range(MAX_STEPS):
    ... MODEL.step()
✓ ... if determine_clean(MODEL):
    ...     TOTAL_STEPS += 1
    ...     is_cleaned = True
    ...     break
    ... TOTAL_STEPS += 1

# Impresión del total de pasos de ejecución
print('Total de pasos de ejecución para limpiar celdas:', TOTAL_STEPS if IS_CLEANED else 'No se ha limpiado el

# Imprimimos el tiempo que le tomó correr al modelo.
print('Tiempo de ejecución:', str(datetime.timedelta(seconds=(time.time() - START_TIME))))

✓ 12.1s Python
limpiar celdas: No se ha limpiado el espacio completo :(, celdas limpiadas contando el porcentaje inicial: 9530

```

En la parte inferior derecha veremos que se han limpiado 9530 celdas, lo cual es una mejora, pero veamos el tiempo de ejecución, este es de 12.1 segundos, bastante tiempo. Ahora, veamos el porcentaje que fue cubierto por el agente sin tomar en cuenta el 10% de celdas limpias inicialmente:

```

# Calculo de celdas limpiadas por el agente
CELLS_COVERED_AGENT = MODEL.cleaned_cells - MODEL.cells_to_cover

# Calcular porcentaje de celdas limpias
TOTAL_PERCENTAGE_CLEANED = (CELLS_COVERED_AGENT * 100) / (WIDTH * HEIGHT)
print("Porcentaje de celdas limpiadas: ", TOTAL_PERCENTAGE_CLEANED, " de un 100%")

✓ 0.0s Python
Porcentaje de celdas limpiadas: 85.31 de un 100%

```

Podemos incrementar a un número bastante grande, esto nos ayudará a determinar la cantidad de pasos que se necesitan para limpiar todo el piso, pongamos 300000:

```
MAX_STEPS = 300000
WIDTH = 100
HEIGHT = 100
PERCENTAGE_DIRTY = 0.90
NUM_AGENTS = 1
MAX_TIME = 1000
MODEL = CleaningModel(NUM_AGENTS, WIDTH, HEIGHT, MAX_STEPS, PERCENTAGE_DIRTY, MAX_TIME)
START_TIME = time.time()
TOTAL_STEPS = 0
IS_CLEANED = False
TOTAL_PERCENTAGE_CLEANED = 0.0
CELLS_COVERED_AGENT = 0

for i in range(MAX_STEPS):
    ...MODEL.step()
    ...if determine_clean(MODEL):
    ...    TOTAL_STEPS += 1
    ...    is_cleaned = True
    ...    break
    ...TOTAL_STEPS += 1

# Impresión del total de pasos de ejecución
print('Total de pasos de ejecución para limpiar celdas:', TOTAL_STEPS if IS_CLEANED else 'No se ha limpiado el

# Imprimimos el tiempo que le tomó correr al modelo.
print('Tiempo de ejecución:', str(datetime.timedelta(seconds=(time.time() - START_TIME))))

39.1s Python
```

ara limpiar celdas: No se ha limpiado el espacio completo :(, celdas limpiadas contando el porcentaje inicial: 9991  
139158

Como sabemos, será un porcentaje de casi un 90%:

```
# Calculo de celdas limpiadas por el agente
CELLS_COVERED_AGENT = MODEL.cleaned_cells - MODEL.cells_to_cover

# Calcular porcentaje de celdas limpias
TOTAL_PERCENTAGE_CLEANED = (CELLS_COVERED_AGENT * 100) / (WIDTH * HEIGHT)
print("Porcentaje de celdas limpiadas: ", TOTAL_PERCENTAGE_CLEANED, " de un 100%")

0.0s Python
```

Porcentaje de celdas limpiadas: 89.92 de un 100%

Para que se limpien todas, pondremos 400000, que con este número nos bastará para saber la cantidad de pasos con un agente para limpiar el espacio de 100 x 100:

```
# Creación de instancia del modelo a ejecutar
MAX_STEPS = 400000
WIDTH = 100
HEIGHT = 100
PERCENTAGE_DIRTY = 0.90
NUM_AGENTS = 1
MAX_TIME = 1000
MODEL = CleaningModel(NUM_AGENTS, WIDTH, HEIGHT, MAX_STEPS, PERCENTAGE_DIRTY, MAX_TIME)
START_TIME = time.time()
TOTAL_STEPS = 0
IS_CLEANED = False
TOTAL_PERCENTAGE_CLEANED = 0.0
CELLS_COVERED_AGENT = 0
```

```
Total de pasos de ejecución para limpiar celdas: 399078 10000
Tiempo de ejecución: 0:00:48.636269
```

Vemos finalmente que, como es un movimiento aleatorio el cual realiza el agente, hay un punto en el que quizás está en un espacio lleno de celdas limpias y gira en círculos por ese entorno, mientras no cambie de ese entorno, no se cumplirá la meta, o no se le añadan objetivos al agente, este estará "paseando por la vida".

Bien ahora el escenario donde hay dos agentes....

Empecemos con una cantidad de 20000 pasos:

```
# Creación de instancia del modelo a ejecutar
MAX_STEPS = 200000
WIDTH = 100
HEIGHT = 100
PERCENTAGE_DIRTY = 0.90
NUM_AGENTS = 2
MAX_TIME = 1000
MODEL = CleaningModel(NUM_AGENTS, WIDTH, HEIGHT, MAX_STEPS, PERCENTAGE_DIRTY, MAX_TIME)
START_TIME = time.time()
TOTAL_STEPS = 0
IS_CLEANED = False
TOTAL_PERCENTAGE_CLEANED = 0.0
CELLS_COVERED_AGENT = 0
```

```
ara limpiar celdas: No se ha limpiado el espacio completo :(, celdas limpiadas contando el porcentaje inicial: 9996
```

```
# Calculo de celdas limpiadas por el agente
CELLS_COVERED_AGENT = MODEL.cleaned_cells - MODEL.cells_to_cover

# Calcular porcentaje de celdas limpias
TOTAL_PERCENTAGE_CLEANED = (CELLS_COVERED_AGENT * 100) / (WIDTH * HEIGHT)
print("Porcentaje de celdas limpiadas: ", TOTAL_PERCENTAGE_CLEANED, " de un 100%")
```

✓ 0.0s

Porcentaje de celdas limpiadas: 89.97 de un 100%

Vemos que es una aproximación a la respuesta, subamos ahora a 250000 pasos de simulación:

Como vemos, ahora se ha limpiado todo el espacio con un número menor de pasos. Esto ha ocurrido por pura probabilidad y suerte.

De igual manera a continuación haremos con la cantidad de pasos que hemos llegado a limpiar las celdas en los dos escenarios pero esta vez lo haremos 7 veces y haremos un promedio para poder determinar la respuesta que más se acerca para cada uno de los escenarios.

Como vimos, no hay una respuesta definitiva a los pasos que le tomó limpiar toda la celda, sin embargo, podemos hacer un análisis con 5-10 pruebas en cada escenario para llegar a un número promedio de pasos para limpiar todo el espacio.

Para agilizar las cosas, decidí realizar un Pandas Data Frame, en el cual instancio un Modelo cada vez que se realiza un ciclo y voy guardando la información que necesito en un diccionario que contiene de key un entero y de valor una lista con el contenido necesario para el análisis. Claramente el tiempo de ejecución será más largo pero hará más sencillo el análisis.

```

MAX_STEPS = 250000
WIDTH = 100
HEIGHT = 100
PERCENTAGE_DIRTY = 0.90
NUM_AGENTS = 2
MAX_TIME = 1000
START_TIME = time.time()

PD_DATA_FRAME = {}
TIMES_REPEAT_ALGO = 0

while TIMES_REPEAT_ALGO < 7:
    ... TOTAL_STEPS = 0
    ... IS_CLEANED = False
    ... TOTAL_PERCENTAGE_CLEANED = 0.0
    ... CELLS_COVERED_AGENT = 0
    ... MODEL = CleaningModel(NUM_AGENTS, WIDTH, HEIGHT, MAX_STEPS, PERCENTAGE_DIRTY, MAX_TIME)
    ... for i in range(MAX_STEPS):
    ...     MODEL.step()
    ...     if determine_clean(MODEL):
    ...         TOTAL_STEPS += 1
    ...         IS_CLEANED = True
    ...         break
    ...     TOTAL_STEPS += 1
    ... MAX_STEPS = 250000
    ... PD_DATA_FRAME[TIMES_REPEAT_ALGO] = [{ 'Pasos totales: ': TOTAL_STEPS,
    ...                                         'Se limpió el espacio: ': 'Sí' if IS_CLEANED else 'No',
    ...                                         'Celdas limpiadas: ' : MODEL.cleaned_cells,
    ...                                         'Porcentaje de celdas limpiadas por agentes: ': ((MODEL.cleaned_cells / CELLS_COVERED_AGENT) * 100)},
    ...                                         ]
    ... TIMES_REPEAT_ALGO += 1

```

### Escenario 1:

En este caso haremos un agente con 400,000 pasos, que es el tope máximo que analizamos anteriormente, veamos el DataFrame arrojado y el promedio de pasos para limpiar el espacio de 100 x 100 con 1 agente:

```

# Creación de dataframe para visualizar resultados de las 7 corridas:
SUM_TOTAL_STEPS = sum(item['Pasos totales: '])
... for item in PD_DATA_FRAME.values() if item['Limpio el espacio'] == 'Sí')
COUNT_IF_SI = sum(1 for item in PD_DATA_FRAME.values() if item['Limpio el espacio'] == 'Sí')

AVERAGE_STEPS_TO_CLEAN = SUM_TOTAL_STEPS / COUNT_IF_SI
print(AVERAGE_STEPS_TO_CLEAN)

DATA_FRAME = pd.DataFrame.from_dict(PD_DATA_FRAME)
DATA_FRAME

```

✓ 0.1s

320401.0

	0	1	2	3	4	5	6
Pasos totales:	340921	372380	328531	400000	241082	400000	319091
Limpio el espacio	Sí	Sí	Sí	No	Sí	No	Sí
Celdas limpiadas	10000	10000	10000	9999	10000	9999	10000
Porcentaje limpiado por agentes	90.01	90.01	90.01	90.0	90.01	90.0	90.01



Como vemos, el promedio de pasos que se dieron para terminar de limpiar el espacio de 100 x 100 con 7 corridas fue de 320,401 pasos con 1 agente. Lo cual podríamos decir que se puede acercar a la respuesta que buscamos, que es de que le tomaría alrededor 320,401 pasos a dos agentes en limpiar un espacio de 10,000 casillas con un 90% sucias (9,000).

Escenario 2:

Veamos el DataFrame arrojado y el promedio de pasos para limpiar el espacio de 100 x 100 con 2 agentes y un máximo de pasos de 250,000:

```
# Creación de dataframe para visualizar resultados de las 7 corridas:
SUM_TOTAL_STEPS = sum(item['Pasos totales:'])
for item in PD_DATA_FRAME.values(): if item['Limpio el espacio'] == 'Sí')
COUNT_IF_SI = sum(1 for item in PD_DATA_FRAME.values() if item['Limpio el espacio'] == 'Sí')

AVERAGE_STEPS_TO_CLEAN = SUM_TOTAL_STEPS / COUNT_IF_SI
print(AVERAGE_STEPS_TO_CLEAN)

DATA_FRAME = pd.DataFrame.from_dict(PD_DATA_FRAME)
DATA_FRAME
```

✓ 0.0s

156543.57142857142

	0	1	2	3	4	5	6
Pasos totales:	206549	128047	195545	133953	146482	150702	134527
Limpio el espacio	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Celdas limpiadas	10000	10000	10000	10000	10000	10000	10000
Porcentaje limpiado por agentes	90.01	90.01	90.01	90.01	90.01	90.01	90.01

Como vemos, el promedio de pasos que se dieron para terminar de limpiar el espacio de 100 x 100 con 7 corridas fue de 156,543 pasos con 2 agentes. Lo cual podríamos decir que se puede acercar a la respuesta que buscamos, que es de que le tomaría alrededor de 156,543 pasos a dos agentes en limpiar un espacio de 10,000 casillas con un 90% sucias (9,000).

Ahora, lo mismo pasa para la siguiente pregunta: ¿Qué porcentaje de celdas sucias queda con los siguientes pasos de simulación: 100, 1000, 10000?

Debemos de realizar varias pruebas para saber qué cantidad de porcentaje de celdas sucias queda con cada uno de los pasos dados. De igual forma haremos el mismo análisis con 7 corridas para cada uno de los escenarios (1 y 2 agentes).

**Empecemos con 100 pasos**

Para un agente:

89.69428571428571							
	0	1	2	3	4	5	6
Pasos totales:	100	100	100	100	100	100	100
Limpio el espacio	No	No	No	No	No	No	No
Celdas limpiadas	1023	1026	1034	1038	1031	1022	1033
Porcentaje limpiado por agentes	0.24	0.27	0.35	0.39	0.32	0.23	0.34
Porcentaje no limpiado	89.76	89.73	89.65	89.61	89.68	89.77	89.66

Obtenemos que en promedio un porcentaje no limpiado del 89.7% de las celdas no fueron limpiadas con un agente y 100 pasos.

Para dos agentes:

89.54571428571428							
	0	1	2	3	4	5	6
Pasos totales:	100	100	100	100	100	100	100
Limpio el espacio	No	No	No	No	No	No	No
Celdas limpiadas	1038	1048	1034	1047	1049	1058	1037
Porcentaje limpiado por agentes	0.39	0.49	0.35	0.48	0.5	0.59	0.38
Porcentaje no limpiado	89.61	89.51	89.65	89.52	89.5	89.41	89.62

Obtenemos que en promedio un porcentaje no limpiado del 89.54% de las celdas no fueron limpiadas con dos agentes y 100 pasos.

### Ahora con 1000 pasos

Para un agente:

87.57857142857142							
	0	1	2	3	4	5	6
Pasos totales:	1000	1000	1000	1000	1000	1000	1000
Limpio el espacio	No	No	No	No	No	No	No
Celdas limpiadas	1240	1268	1287	1275	1275	1213	1130
Porcentaje limpiado por agentes	2.41	2.69	2.88	2.76	2.76	2.14	1.31
Porcentaje no limpiado	87.59	87.31	87.12	87.24	87.24	87.86	88.69

Obtenemos que en promedio un porcentaje no limpiado del 87.58% de las celdas no fueron limpiadas con un agente y 1000 pasos.

Para dos agentes:

86.02857142857144							
	0	1	2	3	4	5	6
Pasos totales:	1000	1000	1000	1000	1000	1000	1000
Limpio el espacio	No	No	No	No	No	No	No
Celdas limpiadas	1428	1421	1444	1341	1217	1456	1466
Porcentaje limpiado por agentes	4.29	4.22	4.45	3.42	2.18	4.57	4.67
Porcentaje no limpiado	85.71	85.78	85.55	86.58	87.82	85.43	85.33

Obtenemos que en promedio un porcentaje no limpiado del 86.02% de las celdas no fueron limpiadas con dos agentes y 1000 pasos.

### Finalmente con 10000 pasos

Para un agente:

72.01							
	0	1	2	3	4	5	6
Pasos totales:	10000	10000	10000	10000	10000	10000	10000
Limpio el espacio	No	No	No	No	No	No	No
Celdas limpiadas	3385	2348	2747	2505	2527	3020	3054
Porcentaje limpiado por agentes	23.86	13.49	17.48	15.06	15.28	20.21	20.55
Porcentaje no limpiado	66.14	76.51	72.52	74.94	74.72	69.79	69.45

Obtenemos que en promedio un porcentaje no limpiado del 72.01% de las celdas no fueron limpiadas con un agente y 10000 pasos.

Para dos agentes:

52.815714285714286							
	0	1	2	3	4	5	6
Pasos totales:	10000	10000	10000	10000	10000	10000	10000
Limpio el espacio	No	No	No	No	No	No	No
Celdas limpiadas	4734	4781	4751	4770	3815	4978	5193
Porcentaje limpiado por agentes	37.35	37.82	37.52	37.71	28.16	39.79	41.94
Porcentaje no limpiado	52.65	52.18	52.48	52.29	61.84	50.21	48.06

Obtenemos que en promedio un porcentaje no limpiado del 52.81% de las celdas no fueron limpiadas con dos agentes y 10000 pasos.

A continuación, determina cuál es la cantidad óptima de aspiradoras que debe de tener para realizar la limpieza en el menor tiempo posible. Considera que tenemos un máximo de 10 aspiradoras disponibles.

Para esto realizaremos el mismo análisis con el DataFrame pero aumentaremos el número de agentes, primero con 5:

Obtenemos el siguiente DataFrame con 5 agentes:

Promedio de pasos para limpiar: 33924.25							
Promedio del tiempo para limpiar: 8.401724714285715							
	0	1	2	3	4	5	6
Pasos totales:	50979	91031	100000	58269	56405	85642	64765
Limpio el espacio	Sí	Sí	No	Sí	Sí	Sí	Sí
Celdas limpiadas	10000	10000	9999	10000	10000	10000	10000
Porcentaje limpiado por agentes	90.01	90.01	90.0	90.01	90.01	90.01	90.01
Porcentaje no limpiado	-0.01	-0.01	0.0	-0.01	-0.01	-0.01	-0.01
Tiempo tomado	0:00:06.920893	0:00:14.374620	0:00:16.820818	0:00:09.349100	0:00:09.290712	0:00:10.216162	0:00:08.660586
Num agentes	5	5	5	5	5	5	5

Vemos que el promedio de tiempo en segundos ha sido de 8.40 segundos, lo cual es una respuesta bastante buena, un tiempo óptimo.

Ahora hagámoslo con 7 agentes:

Promedio de pasos para limpiar: 26983.64285714286							
Promedio del tiempo para limpiar: 8.663916							
	0	1	2	3	4	5	6
Pasos totales:	48573	77565	43641	39539	74659	60250	33544
Limpio el espacio	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Celdas limpiadas	10000	10000	10000	10000	10000	10000	10000
Porcentaje limpiado por agentes	90.01	90.01	90.01	90.01	90.01	90.01	90.01
Porcentaje no limpiado	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
Tiempo tomado	0:00:08.107306	0:00:12.074276	0:00:07.999326	0:00:04.423035	0:00:10.901683	0:00:11.679015	0:00:05.462771
Num agentes	7	7	7	7	7	7	7

Vemos que el promedio del tiempo ha subido por **.26 segundos**, lo cual nos puede indicar que entre más agentes, se puede llegar a entorpecer, debido a que no hay comunicación ni zonas que cubrir por agente.

Intentemos con 9 agentes, si vemos que la cantidad de segundos es mayor a la de 7 agentes, haremos la prueba con 6 agentes.

### Con 9 agentes:

Promedio de pasos para limpiar: 17544.35714285714							
Promedio del tiempo para limpiar: 5.909764857142856							
	0	1	2	3	4	5	6
Pasos totales:	35581	30072	54438	40731	26647	26919	31233
Limpio el espacio	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Celdas limpiadas	10000	10000	10000	10000	10000	10000	10000
Porcentaje limpiado por agentes	90.01	90.01	90.01	90.01	90.01	90.01	90.01
Porcentaje no limpiado	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
Tiempo tomado	0:00:07.750970	0:00:03.887338	0:00:08.155053	0:00:07.010438	0:00:04.877369	0:00:05.532956	0:00:04.154230
Num agentes	9	9	9	9	9	9	9

Vemos que sí ha disminuido el tiempo bastante con **9 agentes** con tiempo de **5.98 segundos**, por lo cual, nos habla de que el algoritmo de la limpieza del suelo es realmente y puramente por probabilidades.

De todas maneras, con **6 agentes** obtenemos:

Promedio de pasos para limpiar: 28135.666666666668							
Promedio del tiempo para limpiar: 7.055038285714287							
	0	1	2	3	4	5	6
Pasos totales:	56179	58080	49205	37379	64842	100000	71943
Limpio el espacio	Sí	Sí	Sí	Sí	Sí	No	Sí
Celdas limpiadas	10000	10000	10000	10000	10000	9999	10000
Porcentaje limpiado por agentes	90.01	90.01	90.01	90.01	90.01	90.0	90.01
Porcentaje no limpiado	-0.01	-0.01	-0.01	-0.01	-0.01	0.0	-0.01
Tiempo tomado	0:00:07.736147	0:00:09.080322	0:00:08.045778	0:00:05.980320	0:00:07.350177	0:00:15.020312	0:00:11.192524
Num agentes	6	6	6	6	6	6	6

Un promedio de **7.05 segundos**, lo cual vemos que sí es más eficiente mientras más agentes tenemos, sin embargo, hay mucho probabilidad de por medio.

### Con 10 agentes:

Promedio de pasos para limpiar: 17358.071428571428							
Promedio del tiempo para limpiar: 5.886012857142857							
	0	1	2	3	4	5	6
Pasos totales:	39097	30133	31744	35049	32012	43040	31938
Limpio el espacio	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Celdas limpiadas	10000	10000	10000	10000	10000	10000	10000
Porcentaje limpiado por agentes	90.01	90.01	90.01	90.01	90.01	90.01	90.01
Porcentaje no limpiado	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
Tiempo tomado	0:00:07.238765	0:00:06.754761	0:00:03.904865	0:00:04.445689	0:00:05.056149	0:00:07.205944	0:00:06.595917
Num agentes	10	10	10	10	10	10	10

Vemos que el promedio de segundos es de **5.88 segundos**, lo cual vemos que ya no ha bajado bastante el tiempo entre 9 y el máximo de agentes que en este caso es 10. Podemos decir entonces que el número óptimo es entre 9 y 10 aspiradoras(agentes) para limpiar un espacio de 100 x 100.

Después de haber realizado esta actividad, como hemos de esperar de los resultados obtenidos, hay mucha probabilidad de por medio, lo cual hace que no sea una buena opción para programación desde mi punto de vista, pues pueden haber mejoras en cuanto a los objetivos que pueden llegar a tener los agentes dentro de un ambiente dinámico, en este caso, cuando es un agente, no hay tanto problema, pero cuando hay más agentes el problema deriva con la comunicación y que, quizás de cierta manera puedan estar rondando por un espacio que ya ha sido limpiado, entonces yo propondría buscar por particiones para hacer más eficiente la búsqueda de celdas no limpiadas, que a cada agente le pertenezca una partición y se encargue únicamente de limpiar esa partición. No obstante, por las restricciones no se podía lograr un algoritmo que no dependiera puramente de probabilidad con las restricciones propuestas.