

# Reporte práctica 6

## Sistema multiagente

### 1. INTRODUCCIÓN

Para la presente práctica se trabajó con un sistema multiagentes, similar al de autómatas celulares, en el que se tienen varias entidades o agentes, cada uno con un estado asociado que a su vez puede observar el estado de los otros; cambiando su propio estado, variando su posición y con la capacidad de cambiar el estado de sus vecinos.

Cuando se tienen simulaciones en donde son requeridas múltiples tareas se ha optado por utilizar la herramienta de paralelizado de procesos para minimizar los tiempos de ejecución. El objetivo de la práctica es comparar los tiempos de ejecución para un código paralelizado y uno no paralelizado de la simulación.

### 2. SIMULACIÓN

Se utilizó este sistema en una simulación de epidemiología, en la que los agentes pueden tomar tres estados distintos: susceptible, infectado o recuperado. En donde el estado infectado tiene cierta probabilidad de recuperarse ( $p_r$ ) y una vez ocurrido esto ya no es posible volver a infectarse.

Se fija el número de agentes con la variable  $n$  y una probabilidad de infección ( $p_i$ ). La probabilidad de que un agente infectado contagie a uno susceptible está en función de la distancia euclidiana entre ambos, en donde se define un umbral ( $r$ ) de distancia para el contagio.

Para definir los cambios de posición se tienen las coordenadas  $x$  y  $y$  para cada agente, así como una dirección y una velocidad. Al iniciar la simulación los agentes están distribuidos uniformemente al azar posteriormente se moverán en función de su dirección en cada paso.

Lo primero es crear los agentes con distintos estados susceptible e infectado, dependiendo de la probabilidad de cada uno. Después se crea el vector *epidemia* que guardará la cantidad de infectados a lo largo del tiempo (número de pasos).

Cada estado de cada agente es comparado con el estado infectado. Y se evalúan las posiciones de sus vecinos para descubrir posibles contagios. En el siguiente paso se contagian aquellos vecinos que quedaron dentro del umbral de un agente infectado.

Luego de los contagios se mueve cada agente de acuerdo con la dirección que tiene y al final se grafica cada agente y su estado en una matriz de puntos en cada paso.

Para algunos de los procesos que se realizan en la simulación es posible paralelizarlos para agilizar la ejecución del código. Al comparar cada estado, cada posición y posible contagio se utilizan bucles tipo *for* para repetir el número de veces necesario, pero algunas de estas funciones no dependen una de la otra. Como es el caso de contagios a agentes susceptibles debido a que en este paso de la función determina a los posibles contagios de agentes vecinos, pero un posible contagiado de un agente no depende de otro agente por lo tanto es posible hacerlo al tiempo (en paralelo).

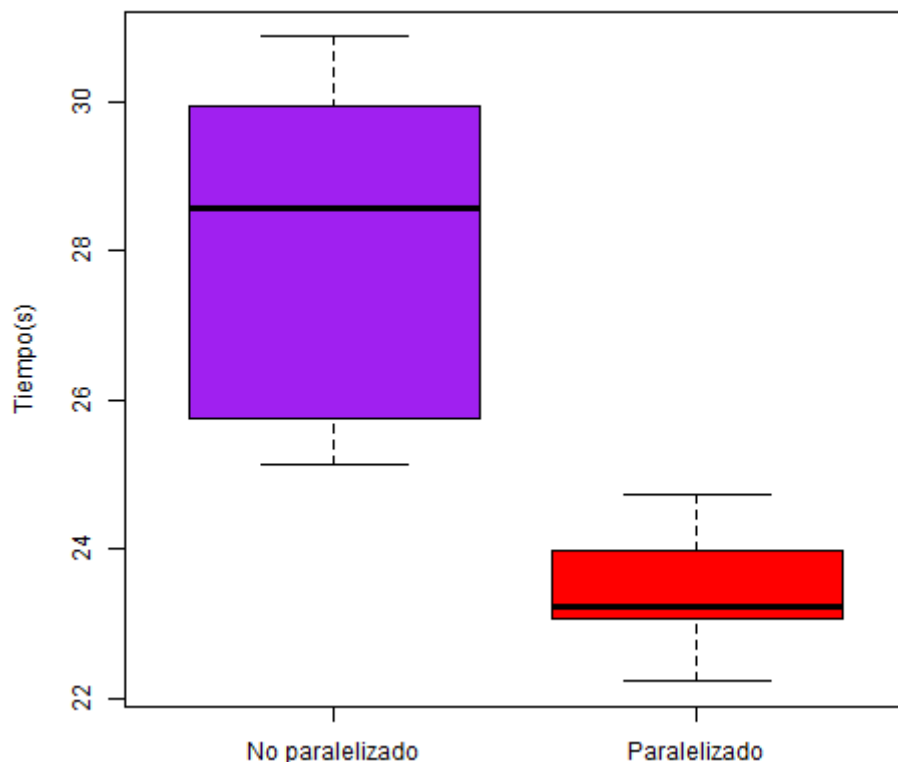
El paralelizado se realizó mediante la creación de clusters con el paquete *doParallel* los cuales después ejecutó una función (*f1*) y repitiéndola con un *foreach* por el número de agentes puestos al inicio.

Cómo se observaba un cambio mínimo en los tiempos de ejecución se buscó que variable era posible modificar para observar mayor diferencia, encontrando que al variar el número de agentes el tiempo tiene mayor variación por lo que se fijó en cien el número de agentes al iniciar.

Para realizar la comparación de los tiempos entre el paralelizado y el original se realizó otro código en donde se repite por diez corridas cada código, mediante *for*, ejecutados mediante la opción *source* el cual llama al archivo que contiene cada código.

### 3. RESULTADOS

**Tiempos comparados**



**Figura 1.** Gráfica caja-bigote de tiempos de ejecución con diez réplicas de cada código.

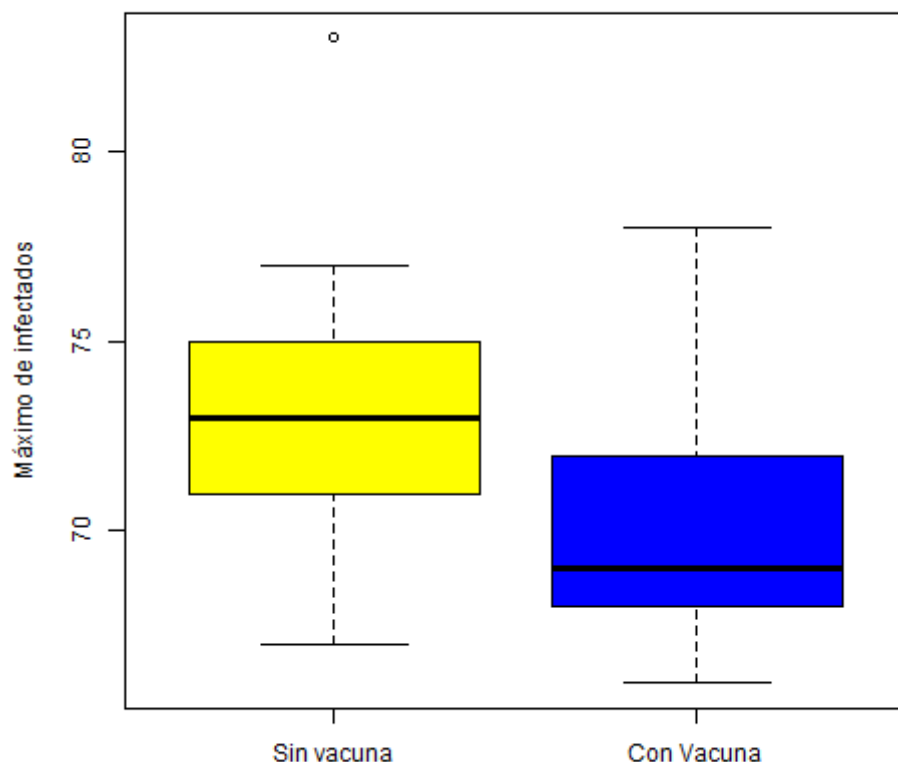
La figura 1 ilustra la diferencia en tiempo de ejecución entre el código paralelizado y el original. Se puede observar que la diferencia en el tiempo de ejecución del código original es alrededor del doble que el paralelizado. Además, la distribución en el tiempo es más amplia en la ejecución del código original.

## 4. RETOS

- Reto 1

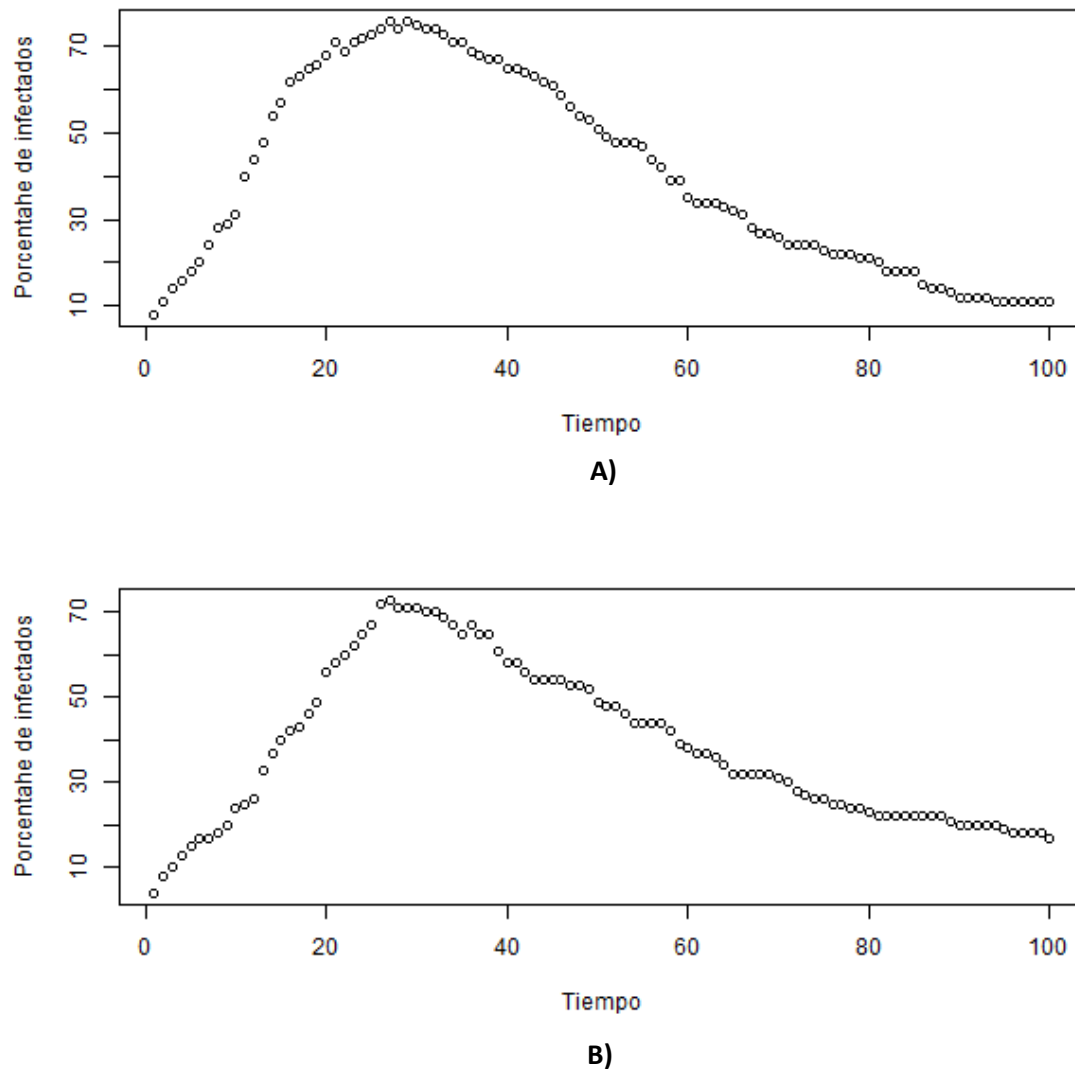
Se modificó el código para saber si existe un cambio en el comportamiento de la epidemia si se le aplicase una vacuna desde el inicio de la simulación en dónde aparezcan agentes con estado de recuperado con una probabilidad fija en  $p_v$  de 0.03; los que ya no van a poder ser infectados. Esto se realizó en dónde se crean los posibles agentes que para este caso podrán ser: susceptibles, infectados y recuperados.

Cómo se puede observar en la figura 2 la cantidad máxima de infectados es mayor en la epidemia sin vacuna previa que en la que se aplicó una vacuna con probabilidad de 0.03.



**Figura 2.** Gráfica caja-bigote con diez réplicas comparando la cantidad máxima de infectados con y sin vacuna al inicio con una  $p_v=0.03$ .

Para comparar la cantidad de infectados en el tiempo con y sin vacuna se realizaron gráficas de puntos para visualizar los máximos de infectados. Como se muestra en la figura 3, la cantidad máxima de infectados decae más rápido en la población con vacuna (figura 3B) que la que no fue vacunada (figura 3A).



**Figura 3.** Gráfica puntos de cantidad máxima de infectados en el tiempo sin vacuna (A) comparando la cantidad máxima de infectados con vacuna  $p_v=0.03$  (B).

## 5. CONCLUSIONES

Se concluye que el código tiene pasos paralelizables con lo que se puede disminuir el tiempo de ejecución, el más eficiente es el paso de contagios.

Se observa mayor cambio en el tiempo de ejecución al aumentar la cantidad de agentes.

Al paralelizar el tiempo de ejecución disminuye considerablemente, llegando cerca de la mitad del tiempo en comparación con el código original.

Si se administra una vacuna antes de que inicie la infección el número máximo de infectados disminuirá. Además, que el máximo de infectados decae más rápido con la vacuna.