Práctica 1: movimiento Browniano

1. Introducción

En esta práctica se observa el movimiento que realiza una partícula a travez de diferentes planos de movimiento, dicho movimiento lo realiza con una aleatoriedad, lo cual se conoce como movimiento Browniano, la intención será en primer momento conocer la probabilidad con la cual la partícula regresará a su punto de salida, tomando éste como el origen, así mismo los retos adicionales nos ayudarán a contabilizar los tiempos que tarda el programa en realizar una caminata, así como los tiempos que tardaría si no tuviéramos una versión paralela, intentando visualizar la ventaja o desventaja que la paralelización representa en esta práctica, todas las experimentaciónes se llevarán a cabo comparando dos formas de medición de distancias, la forma Euclideana y la conocida como Manhattan.

2. Parámetros de trabajo

La experimentación se realizó en un HP Z230 Tower Workstation con procesador Intel(R) Xenon(R) CPU E3-1240 v3 y 3.40 GHz de memoria ram 16 GB y un sistema operativo de 64 bits con Windows 7 Home Premium.

Se simulan caminatas de una partícula a través de 8 dimensiones, con un largo de pasos en las caminatas de $p \in \{50, 80, 150, 200, 400\}$ y una repetición de $\{100, 300\}$. La cantidad de pasos se aumentó con el fin de observar cambios en los tiempos de ejecución ya que los mismos parámetros para la práctica original no causaban efecto significativo, los pasos que se usaron para los retos fueron $p \in \{500, 800, 2000, 10000\}$, las cantidad de repeticiones se mantuvieron en 100 para ambos retos.

3. Modificaciones del código

Se agregaron para la práctica un ciclo `for` que nos ayudara a variar de forma automática la cantidad de pasos que realiza en cada caminata, así mismo fue agregado un `vector` el cual guarda las cantidades a variar. Para generar las imágenes fue necesario incluir la librería `ggplot`.

```
library('ggplot2')
...
d <- c(500,800,2000,10000)
...
for (l in 1:length(d)){
duracion <- d[l]
}
```

Para la generación de las imágenes que comparan tanto los tiempos como la cantidad de veces que la partícula llega al origen se utilizó una linea similar.

```
png(paste("DimEuN.png", sep=""), width=700, height=700)
ggplot(data=datos,aes(x=Dimension,y=Tiempo, fill=Pasos))+geom_boxplot()
+xlab("Dimensi\u{F3}n")+ ylab("Tiempo de ejecuci\u{F3}n (s)")
graphics.off()
```

4. Resultados y conclusiones

En la figura 1 se puede observar que la cantidad de veces que una partícula llega al origen sin importar la cantidad de pasos es sumamente superior cuando solo se tiene una dimensión. Al ir incrementando la cantidad de dimensiones este porcentaje va disminuyendo drásticamente.

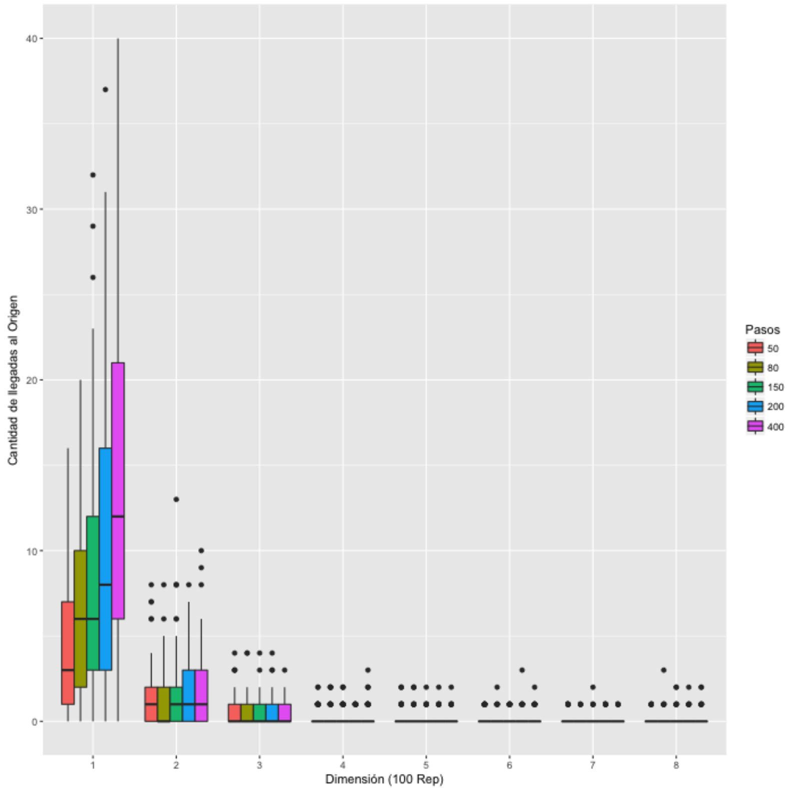

Figura 1: Porcentaje de llegada al origen referente a la cantidad de dimensiones de movimiento de la partícula.

De igual forma al aumentar la cantidad de repeticiones en la experimentación los gráficos muestran un comportamiento prácticamente idéntico, es decir es muy evidente en la figura 2 observar que el número de repeticiones no tienen un efecto significativo en la experimentación, así como es evidente que el factor que más aumenta la probabilidad de regresar al origen al estar en una sola dimensión.

5. Reto 1

El reto uno consiste en medir los tiempos de ejecución de las caminatas variando la cantidad de pasos, así como la dimensión en el movimiento de la partícula

6. Modificación del código y parámetros de R1

Fue necesario agregar las funciones que te ayudan a medir el tiempo de ejecución así como agregar el comando que te permite observar dichos tiempos en segundos. Se agregaron los gráficos de los tiempos, así comopara que variaran por largo de caminata y por dimensión.

```
c <- Sys.time()
d <- Sys.time()
ti <- c(c,d)
tie <- diff(ti,units="secs")
todo <- c(duracion,dimension,mayor,tie)
      ....
```

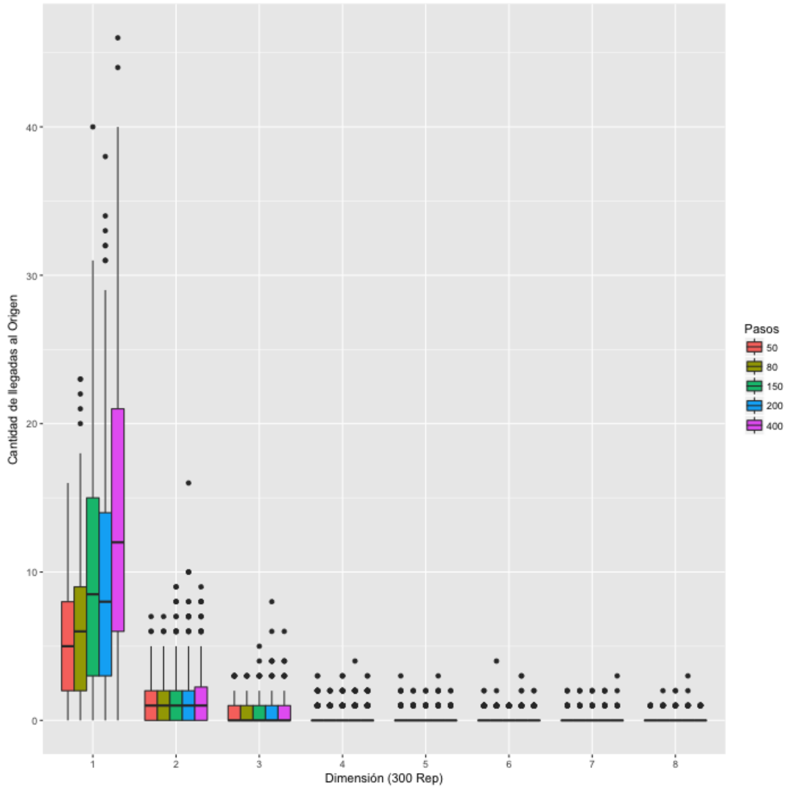

Figura 2: Cantidad de veces que la partícula llegó al origen con 300 repeticiones.

```
png(paste("LargoEu.png", sep=""), width=700, height=700)
ggplot(data=graficas, aes(x=Pasos, y=Tiempo, fill=Dimension))+
geom_boxplot()+xlab("Largo de la Caminata")+
ylab("Tiempo de ejecuci\u{F3}n (s)")
graphics.off()
```

7. Resultados y conclusiones de R1

Acorde a las figuras 3 y 4, no existe duda que en los tiempos de ejecución de las caminatas con 10000 pasos existe una diferencia significativa de pasos, es decir, acorde al incremento de pasos es de igual forma en tiempos sin inmportar la forma de medición de los pasos.

Lo interesante es al observar los cambios de núcleos, ya que no es tan evidente si existe un cambio significativo respecto a ellos, como se podrá observar en la figura 5 y en la figura es muy difícil a simple vista diferenciar si existe una diferencia significativa de ellos, tras una prueba, se llega a la conclusión que no existe diferencia significativa en las medias, por tanto se podría decir que los tiempos de ejecución respecto a las diferentes dimensiones no es significativo, para ambas formas de medición.

8. Reto 2

El reto dos consiste en correr la experimentación ahora sin aplicar la paralelización, esto con el fin de comenzar a hacer conciencia en que casos la paralelización de algún código será necesaria, y que existen casos que no lo será, depende de el tipo de acciones que se realicen.

9. Modificación de código y parámetros R2

En este reto, más que agregar se tuvo que quitar los comandos, así como la librería de la paralelización.

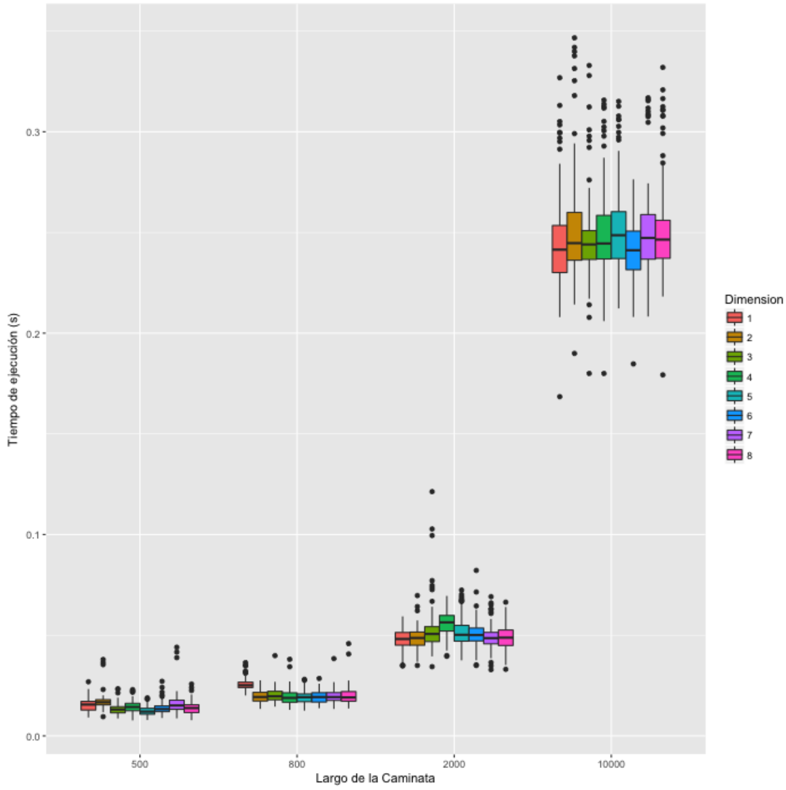

Figura 3: Tiempos de ejecución respecto a la cantidad de pasos realizados en la caminata, con medición de forma Euclideana.

10. Resultados y conclusiones de R2

Observemos que los tiempos de ejecución varían en el codigo paralelizado contra el que no lo está, observemos que en ambos casos, tanto en la forma de medición Euclideana en la figura 7, así como en la forma Manhattan en la figura 8, se observa que es más conveniente ejecutar el código paralelizado, pero lo que es interesante ver es el comportamiento de la medición Manhattan, sin duda tiene una variación muy notoria, así como

Para esta práctica está claro que es conveniente la paralelización, para cualquiera de las dos formas de medición, así como sin importar lo largo de la caminata.

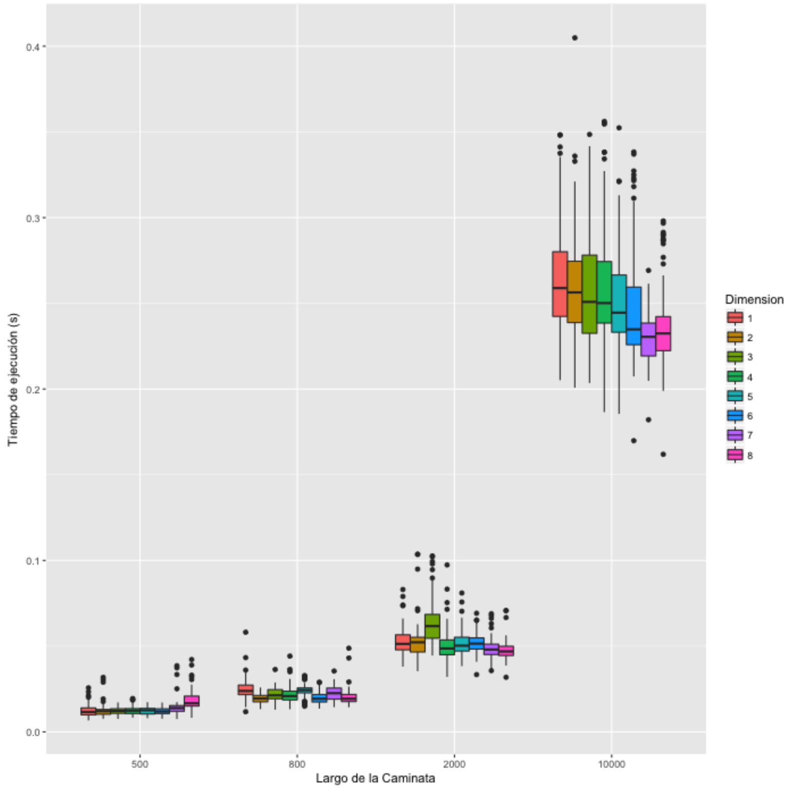

Figura 4: Tiempos de ejecución respecto a la cantidad de pasos realizados en la caminata, con medición de forma Manhattan.

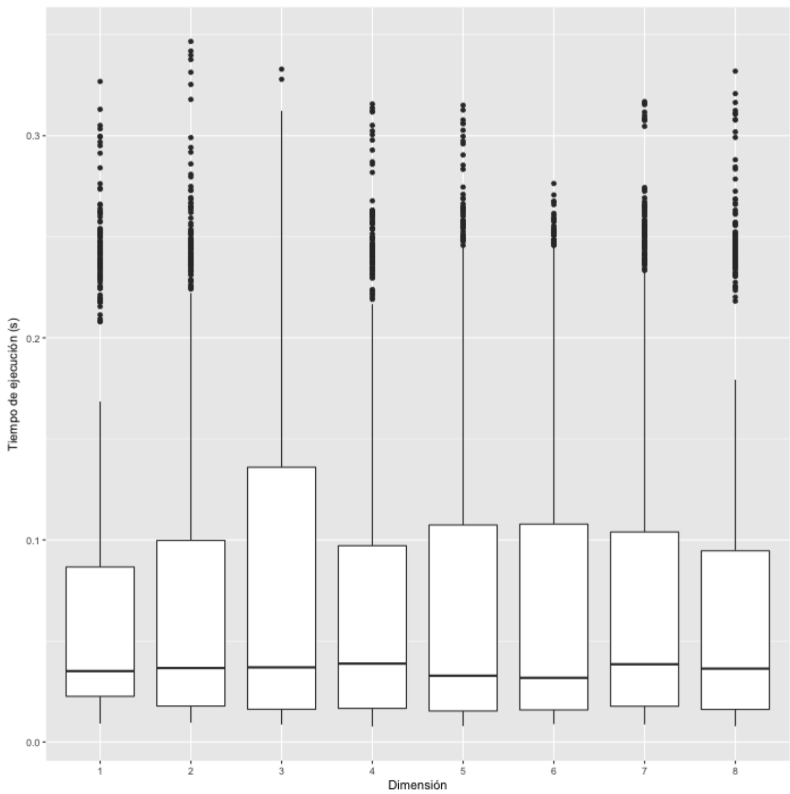

Figura 5: Tiempos de ejecución con respecto a las dimensiones de las caminatas en forma de medición Euclideana.

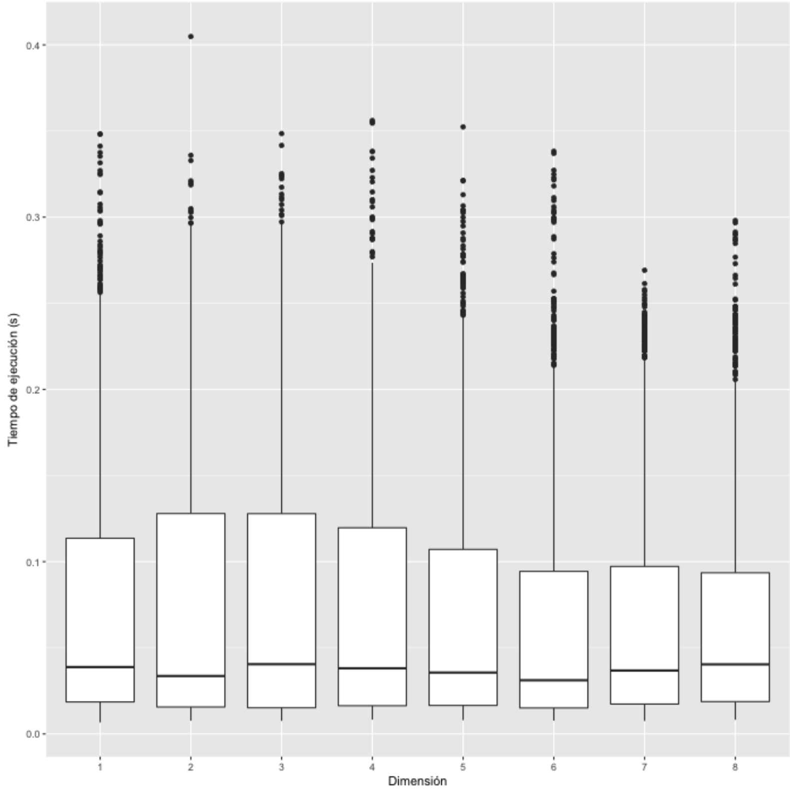

Figura 6: Tiempos de ejecución con respecto a las dimensiones de las caminatas en forma de medición Manhattan.

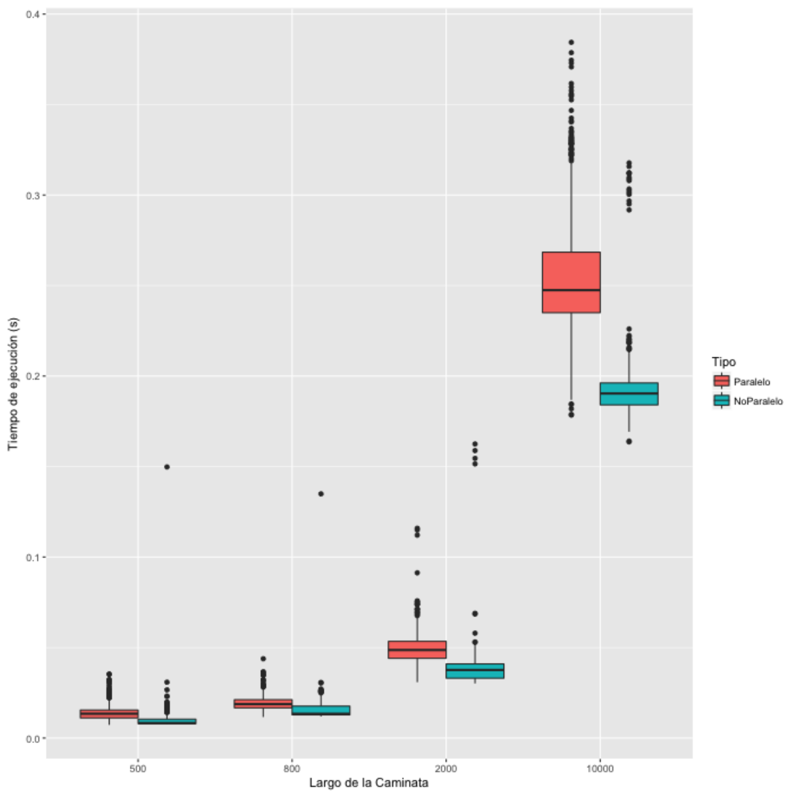

Figura 7: Tiempos de ejecución de código paralelo contra secuancial, con medida Euclideana, variando los largos de las caminatas.

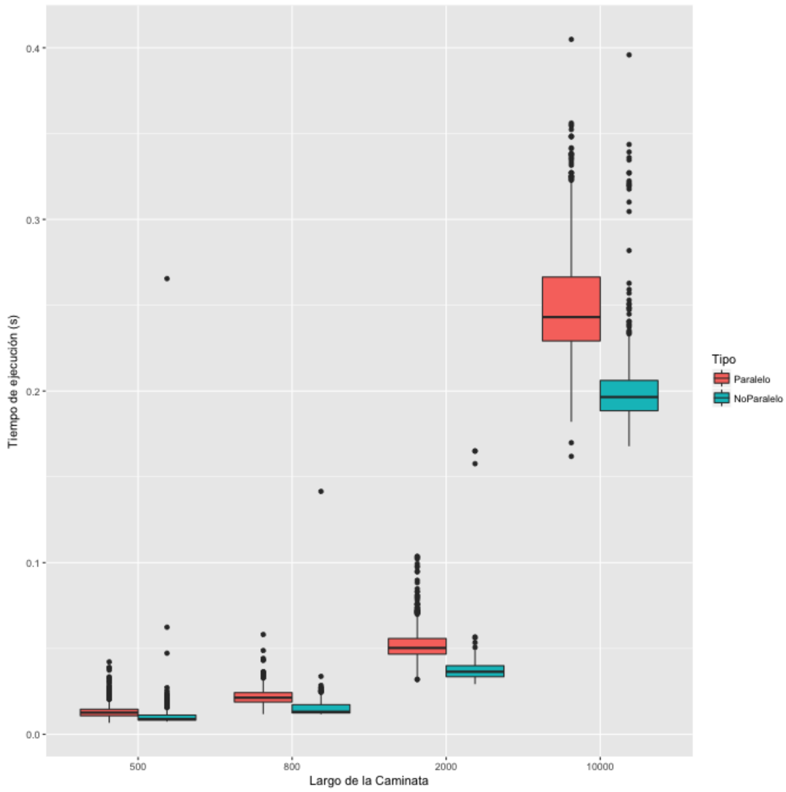

Figura 8: Tiempos de ejecución de código paralelo contra secuancial, con medida Manhattan, variando los largos de las caminatas.