

# Taller #2 Métodos Computacionales

Daniel Lozano Gómez

1 de septiembre de 2018

## 1. Listas, escritura de archivos y plots en Python

En el siguiente taller se debe realizar un programa que contenga la clase *punto()* la cual describe una partícula en una cierta posición inicial. Haciendo uso de esta clase, se generaran caminatas aleatorias y se probaran diferentes aptitudes relevantes.

Su script debe tener lo siguiente

1. Dentro de la clase *punto()*, se debe tener un método de inicialización que contenga atributos de posición inicial en  $x$ , en  $y$  y el radio,  $r$ , donde la partícula está ubicada. En caso de que la clase se inicialice sin parámetros, la partícula debe iniciar en el origen.
2. Dentro de la clase *punto()*, genere un método llamado *move()* que, dados un paso en las coordenadas, actualice la posición de la partícula.
3. Dentro de la clase *punto()*, agregue un método que imprima la posición actual  $(x, y, r)$ .
4. Genere un archivo llamado *caminata.txt* y creelo en modo escritura.
5. Haciendo uso de la clase *punto()*, realice una caminata aleatoria donde el paso este dado por un número aleatorio entre -1 y 1, para ello use la función *random.rand()* de *numpy*. La caminata debe darse hasta que la partícula alcance un radio mayor a 10.
6. Prueba del teorema de límite central (TLC): Usando la clase *punto()* se realizará la prueba del TLC. Para ello realice  $N$  simulaciones de partículas que dan  $N_p$  pasos. En cada simulación, el radio final de la partícula debe ser guardado en un arreglo e impreso en el archivo *caminata.txt* junto al numero de la simulación.
7. Pruebe los siguientes valores para  $N$  y  $N_p$  ( $(N = 100, N_p = 50)$ ,  $(N = 100, N_p = 100)$ ,  $(N = 1000, N_p = 050)$ ). Ponga un comentario en su programa que comente los valores que funcionarían mejor.

8. Realice un histograma de los valores de alcance, suponiendo que los pasos anteriores fueron realizados correctamente. El histograma debe asemejarse a una función gaussiana.
1. Dentro de la clase *punto()*, se debe tener un método de inicialización que contenga atributos de posición inicial en  $x$ , en  $y$  y el radio,  $r$ , donde la partícula está ubicada. En caso de que la clase se inicialice sin parámetros, la partícula debe iniciar en el origen.
2. Dentro de la clase *punto()*, genere un método llamado *move()* que, dados un paso en las coordenadas, actualice la posición de la partícula.
3. Dentro de la clase *punto()*, agregue un método que imprima la posición actual  $(x, y, r)$ .
4. Genere un archivo llamado *caminata.txt* y creelo en modo escritura. Este archivo servirá para guardar las coordenadas históricas de la partícula en el tiempo.
5. Haciendo uso de la clase *punto()*, programe una función que realice una caminata aleatoria donde el paso esté dado por un número aleatorio entre -1 y 1, para ello use la función *random.rand()* de *numpy*. La caminata debe darse hasta que la partícula alcance un radio mayor a 10.
6. Imprima los valores de posición en  $x$ , en  $y$  y número de paso en el archivo *caminata.txt* para cada paso de la caminata aleatoria. Cierre este archivo luego de concluir la caminata.
7. Lea el archivo de *caminata.txt* usando la función *open()*, en modo lectura y guarde las posiciones en  $x$  y  $y$  en arreglos.
8. Grafique la varianza del arreglo en función del número de pasos.
9. Grafique la caminata aleatoria.