Actividad 3 Metodo Monte Carlo

Uriel G. Ernesto A. Oscar H.

8 de septiembre de 2022

Resumen

En esta actividad se realizará una simulación donde obtendremos el valor de Pi, dado que no es un valor exacto y este mismo sigue en discusión, buscaremos un resultado más parecido al que se nos mencionó.

1. Introducción

Los números imaginarios han tenido una gran influencia para el desarrollo y cálculo de diversos problemas, este nos ayudado a resolver problemas numéricos más complejos en ciencia, ingeniería, finanzas y estadísticas. En este caso lo usaremos para el cálculo de Pi, un número muy debatido, en casos prácticos lo usamos como ¨3.1416¨ pero en realidad es un numero más largo y complejo, para ello el uso del método Monte Carlo, en el cual se simulan distinta cantidad de casos para llegar a la respuesta más exacta .

2. Simulación Monte Carlo

La simulación Monte Carlo es básicamente un muestreo experimental cuyo propósito es estimar las distribuciones de las variables de salida que depende de variables probabilísticas de entrada. Los investigadores acuñaron este término por su similaridad al muestreo aleatorio en los juegos de ruleta en los casinos de Monte Carlo. Así, por ejemplo, el modelo de Monte Carlo puede simular los resultados que puede asumir el VAN de un proyecto. Pero lo más relevante es que la simulación permite experimentar para observar los resultados que va mostrando dicho VAN[1].

3. Números aleatorios

En el corazón de los Métodos de Monte-Carlo encontramos un generador de números aleatorios, es decir un procedimiento que produce un flujo infinito de variables aleatorias, que generalmente se encuentran en el intervalo (0, 1); los cuales son independientes y están uniformemente distribuidos de acuerdo a una distribución de probabilidad. La mayoría de los lenguajes de programación hoy en día contienen un generador de números aleatorios por defecto al cual simplemente debemos ingresarle un valor inicial, generalmente llamado seed o semilla, y que luego en cada invocación nos va a devolver un secuencia uniforme de variables aleatorias independientes en el intervalo (0, 1)[2].

4. Números pseudoaleatorios

El concepto de una secuencia infinita de variables aleatorias es una abstracción matemática que puede ser imposible de implementar en una computadora. En la práctica, lo mejor que se puede hacer es producir una secuencia de números pseudoaleatorios con propiedades estadísticas que son indistinguibles de las de una verdadera secuencia de variables aleatorias. Aunque actualmente métodos de generación física basados en la radiación de fondo o la mecánica cuántica parecen ofrecer una fuente estable de números verdaderamente aleatorios , la gran mayoría de los generadores de números aleatorios que se utilizan hoy en día están basados en algoritmos simples que pueden ser fácilmente implementados en una computadora

Figura 1: Ejemplo del codigo de la simulacion Monte Carlo

5. Codigo a ejecutar

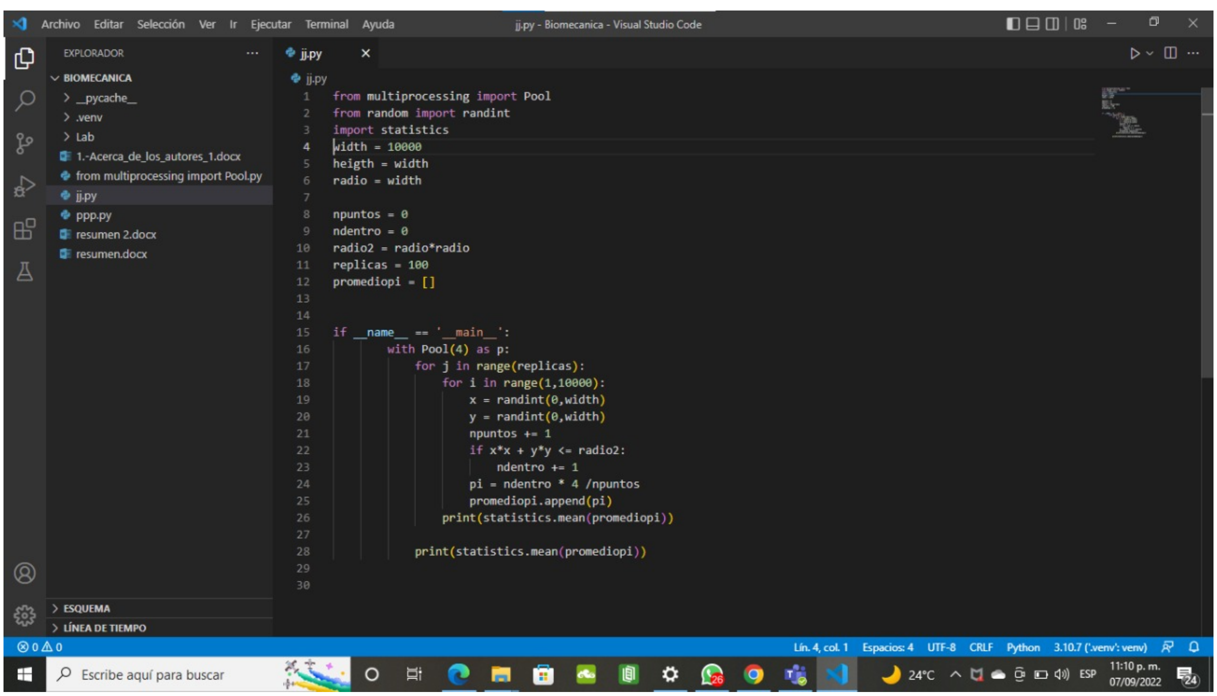
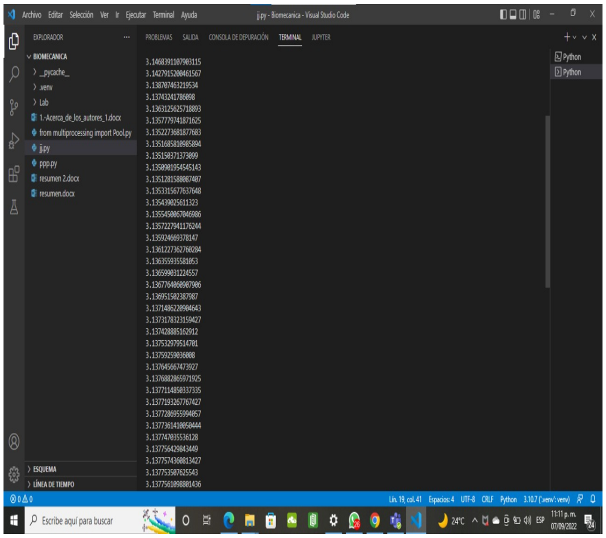
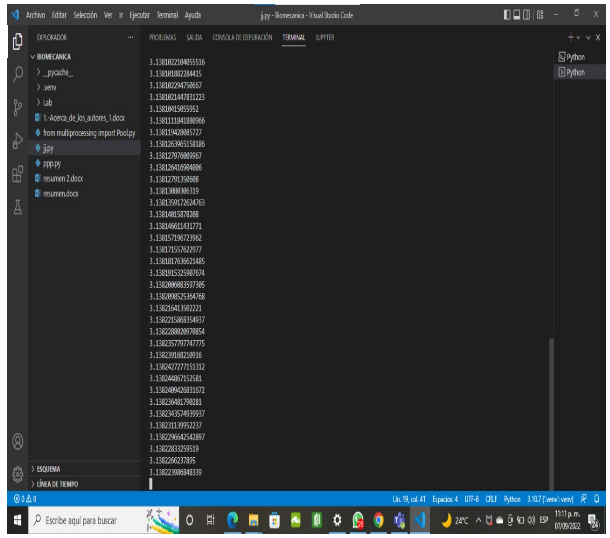


Figura 2: Codigo de la simulacion Monte Carlo

6. Resultados

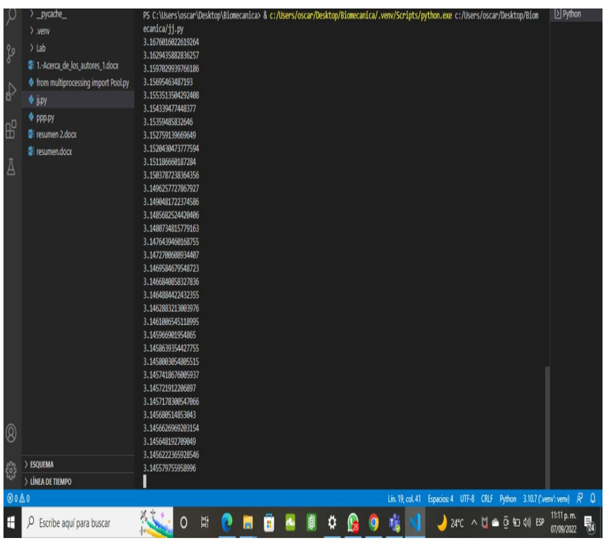


(a) Primeros resultados

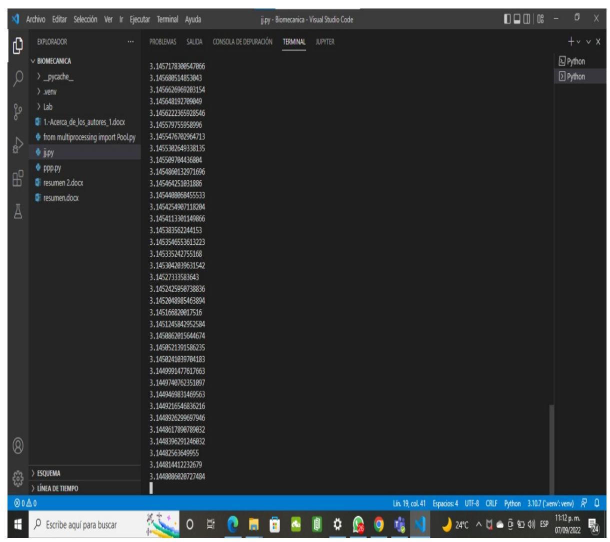


(b) Segunda lista resultados

Figura 3: Lista de resultados



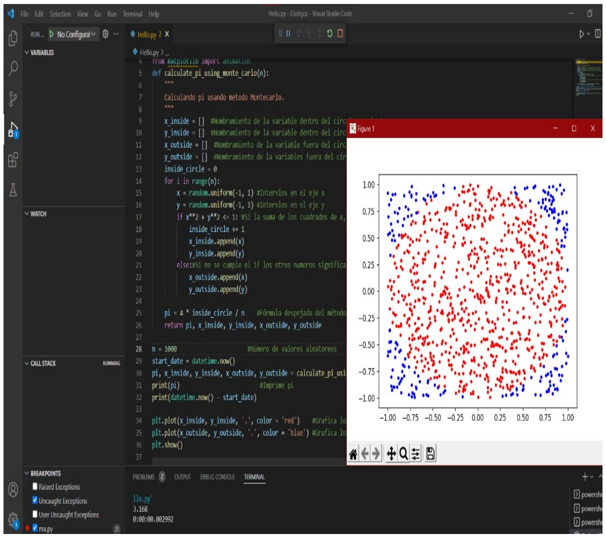
(a) Tercera lista resultados



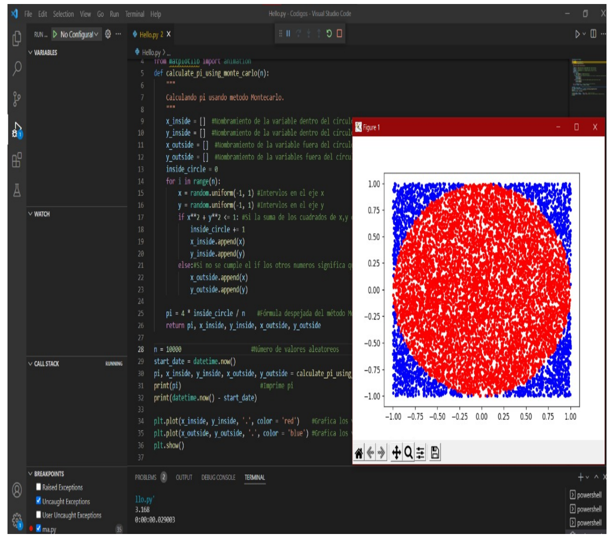
(b) Cuarta lista resultados

Figura 4: Lista de resultados

7. Graficas

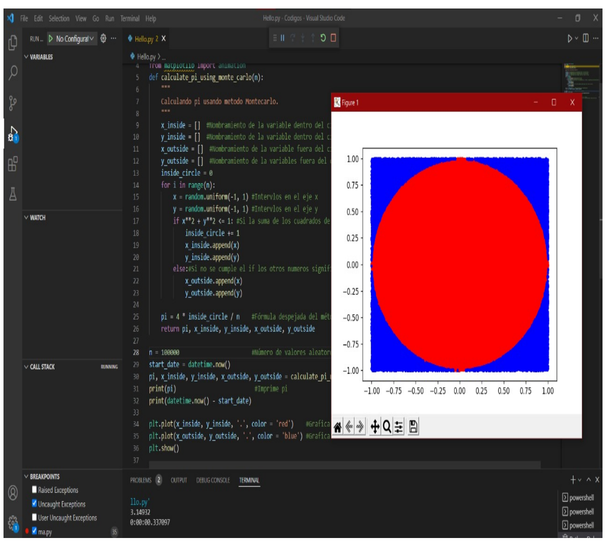


(a) Resultado con 1,000 números aleatorios

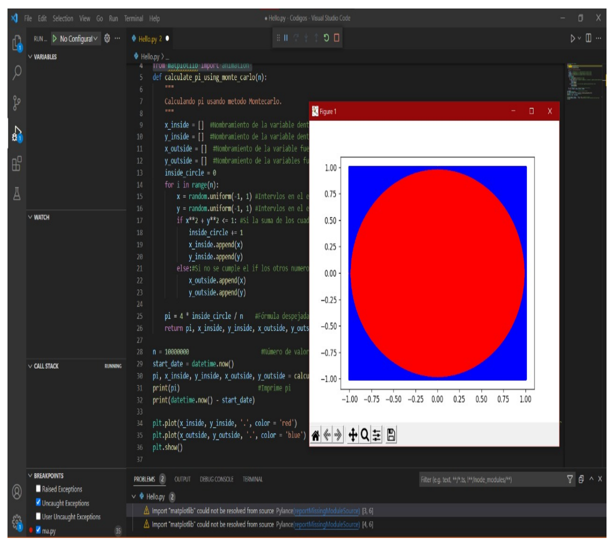


(b) Resultado con 10,000 números aleatorios

Figura 5: Graficas resultantes



(a) Resultado con 100,000 números aleatorios



(b) Resultado con 1,000,000 números aleatorios

Figura 6: Graficas resultantes

8. Conclusiones

Al realizar esta actividad nos dimos cuenta de lo importante que pueden llegar a ser la generación de números aleatorios, en este caso lo vimos con el método Monte Carlo, que entre más números generara más exacta fue nuestra respuesta, con esto lo podemos llegar aplicar en algún caso práctico de la vida diaria.

Referencias

[1] Danysoft. Cómo instalar python, bibliotecas pandas y matplotlib, power bi, spyder y jupyter notebooks. `https://www.youtube.com/watch?v=JbsiKRDnIns&t=744s`, 2021.

[2] R. E. Lopez. Introducción a los métodos de monte-carlo con python., enero 2017.