



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Tarea 1. Cálculo de PI

Unidad de aprendizaje: Desarrollo de Sistemas Distribuidos

Grupo: 4CV11

Alumno:
Sanchez Mendez Edmundo Josue

Profesor:
Pineda Guerrero Carlos

2 de septiembre de 2021

Índice

1. Introducción	2
2. Desarrollo	2
2.1. Compilación del código	2
2.2. Ejecución del programa	3
3. Conclusiones	6
Referencias	6

1. Introducción

Mediante el desarrollo de un sistema distribuido la cual tendrá una topología lógica de estrella, se hará un cálculo aproximado del número PI haciendo uso de la serie de Gregory-Leibniz, la cual esta definida de la siguiente manera.

$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots\right) = 4 \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

En donde los denominadores son los números impares y el signo negativo se va alternando entre cada miembro de la suma. El sistema distribuidos contara con 5 nodos que se ejecutaran en una misma computadora, en donde uno de los 5 nodos sera el servidor y el encargado de mostrar el valor aproximado calculado por el sistema, mientras que los 4 nodos restantes actuaran como los clientes serán los encargados de hacer los cálculos correspondientes.

2. Desarrollo

Una vez programado el sistema distribuido con base en los algoritmos establecidos en la tarea se procedió a realizar la compilación de un archivo con nombre PI.java el cual se encuentra en el archivo comprimido que se entrego en la plataforma.

2.1. Compilación del código

En la figura 1 podemos ver como la compilación de nuestro programa PI.java se hace de manera exitosa y sin ningún error alguno, esto desde el Símbolo del sistema (CMD) de Windows 10.

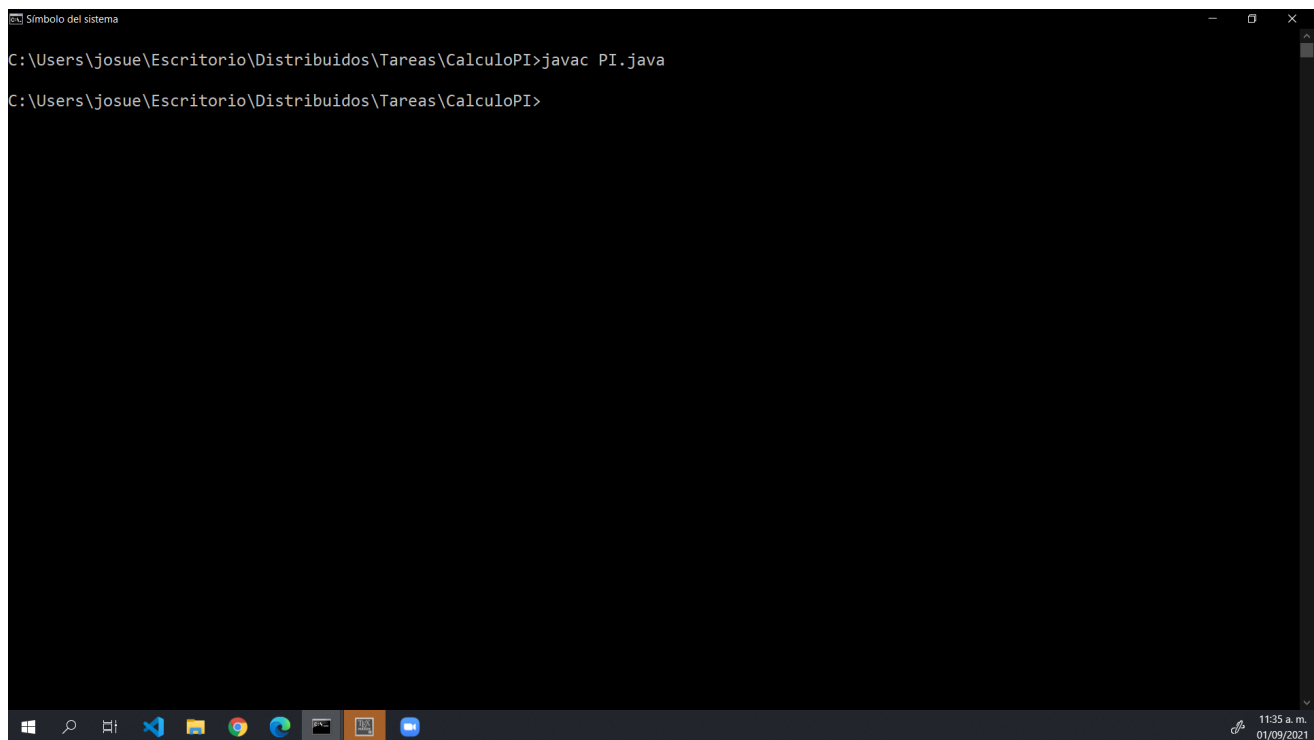
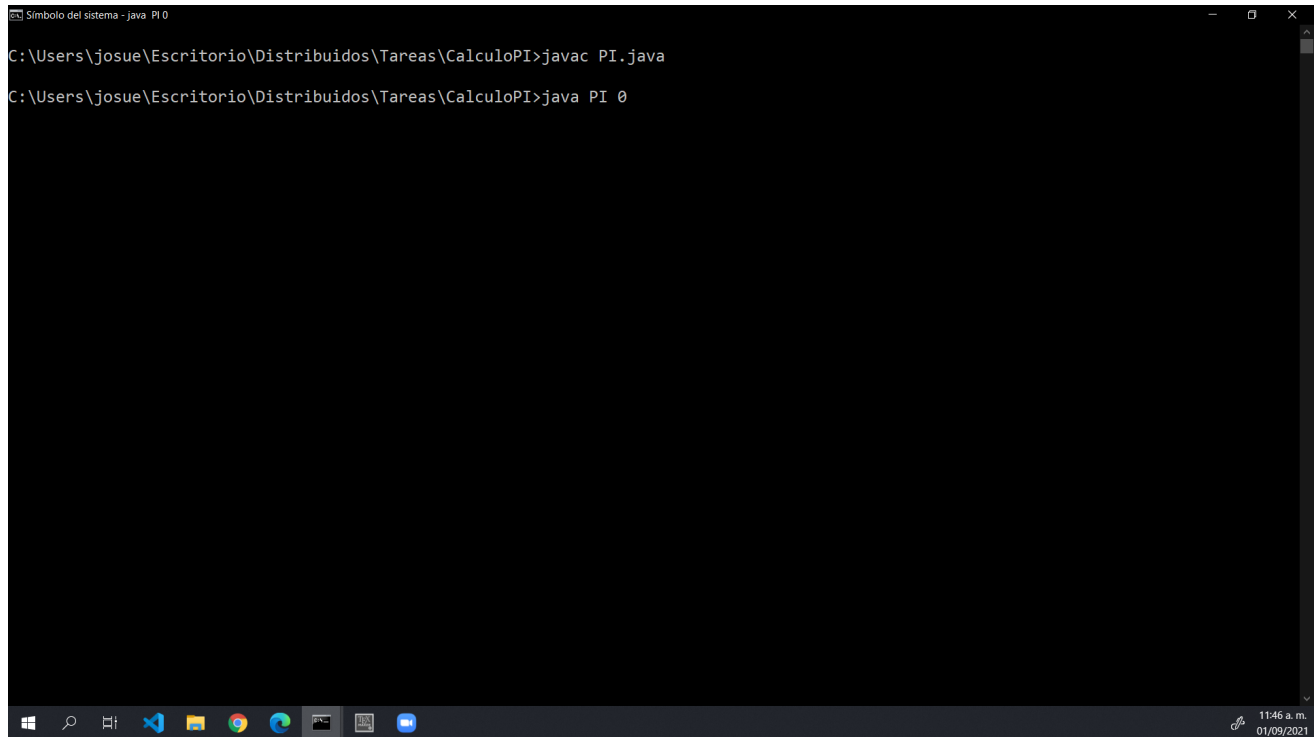


Figura 1: Compilación del código por medio de CMD.

2.2. Ejecución del programa

Primero ejecutamos el nodo del servidor, para ello se le pasa al programa el valor 0, desde la consola:



```
Símbolo del sistema - java PI 0
C:\Users\josue\Escritorio\Distribuidos\Tareas\CalculoPI>javac PI.java
C:\Users\josue\Escritorio\Distribuidos\Tareas\CalculoPI>java PI 0
```

Figura 2: Ejecución del Nodo 0 de nuestro sistema distribuido.

Después abrimos 4 nuevas terminales y ejecutamos a los 4 clientes, pasando el valor de 1, 2, 3, 4 a cada terminal para poder completar el numero total de nodos requeridos en nuestro sistema distribuido:

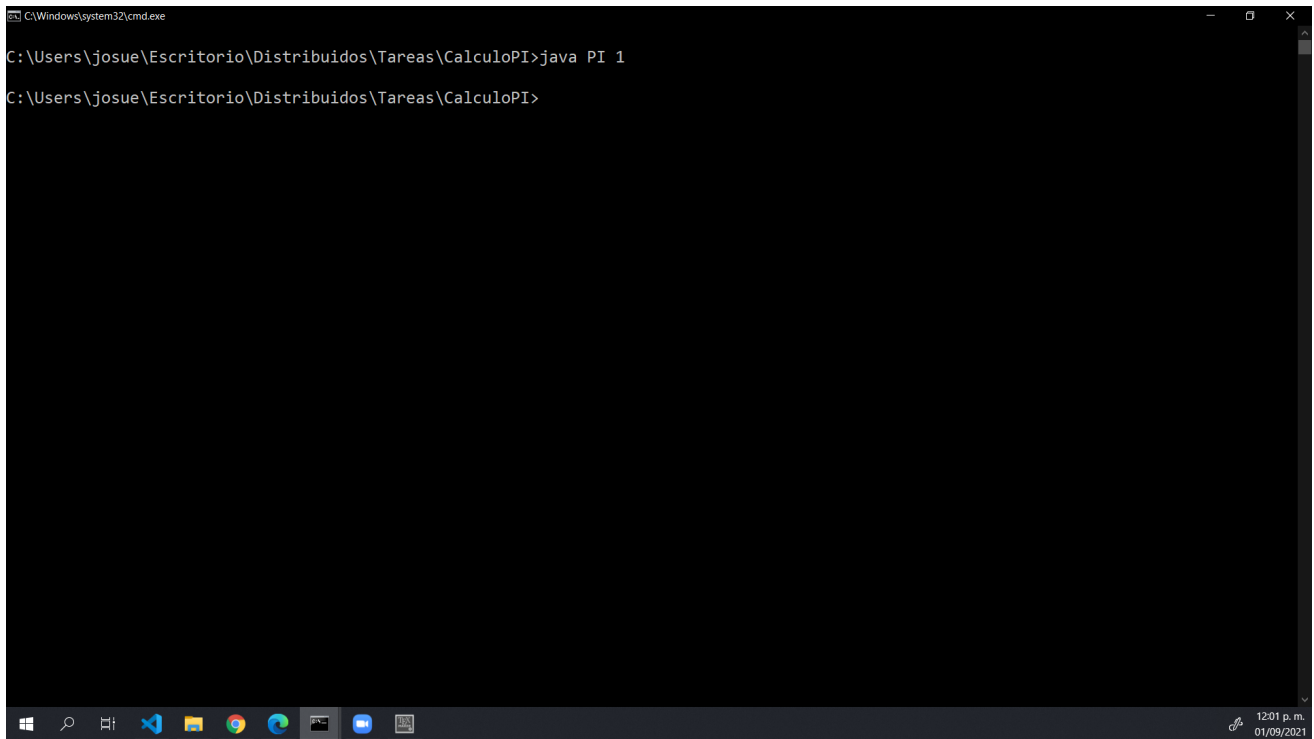


Figura 3: Ejecución del nodo 1 de nuestro sistema distribuido.

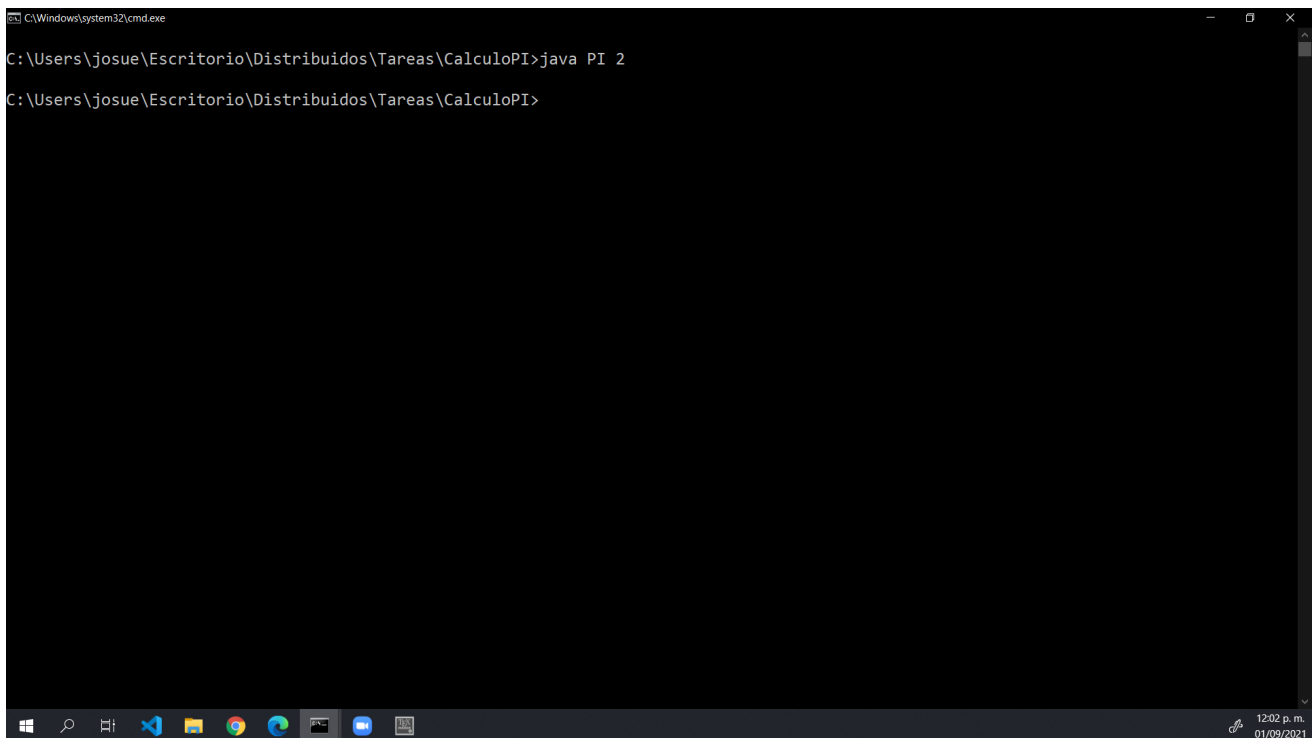


Figura 4: Ejecución del nodo 2 de nuestro sistema distribuido.

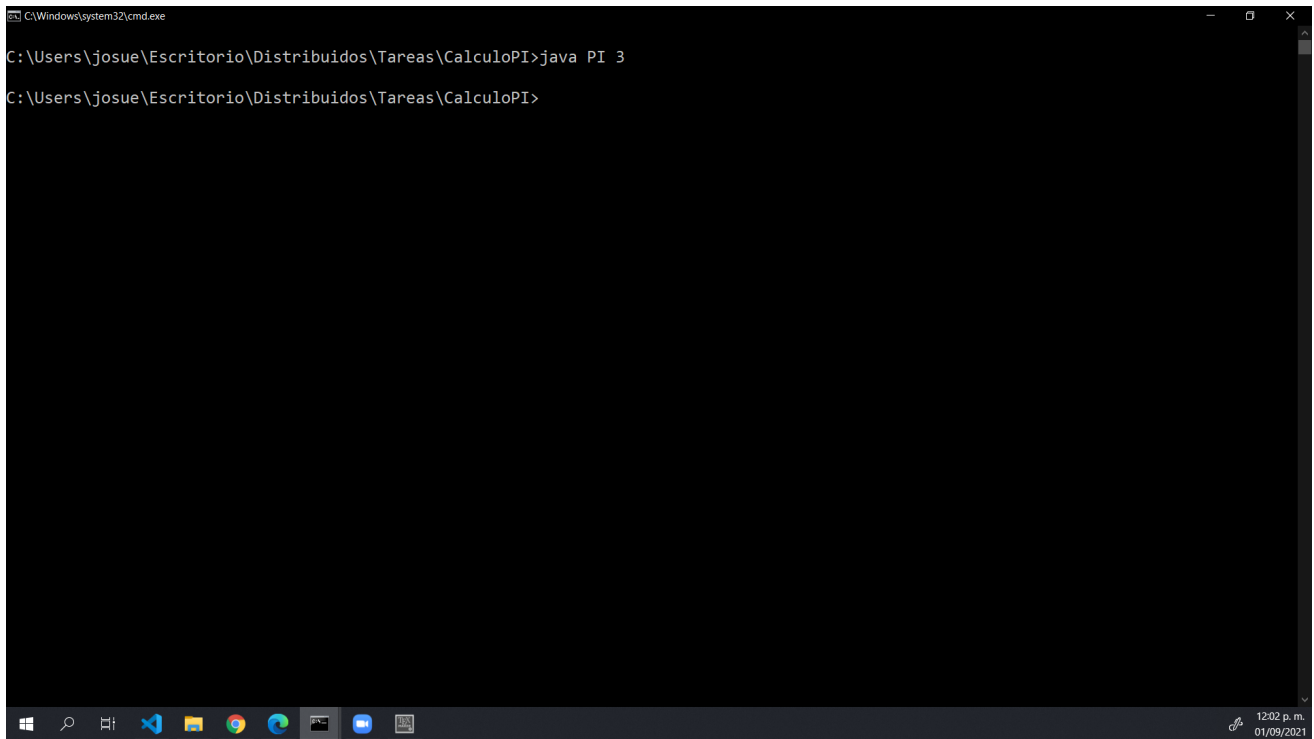


Figura 5: Ejecución del nodo 3 de nuestro sistema distribuido.

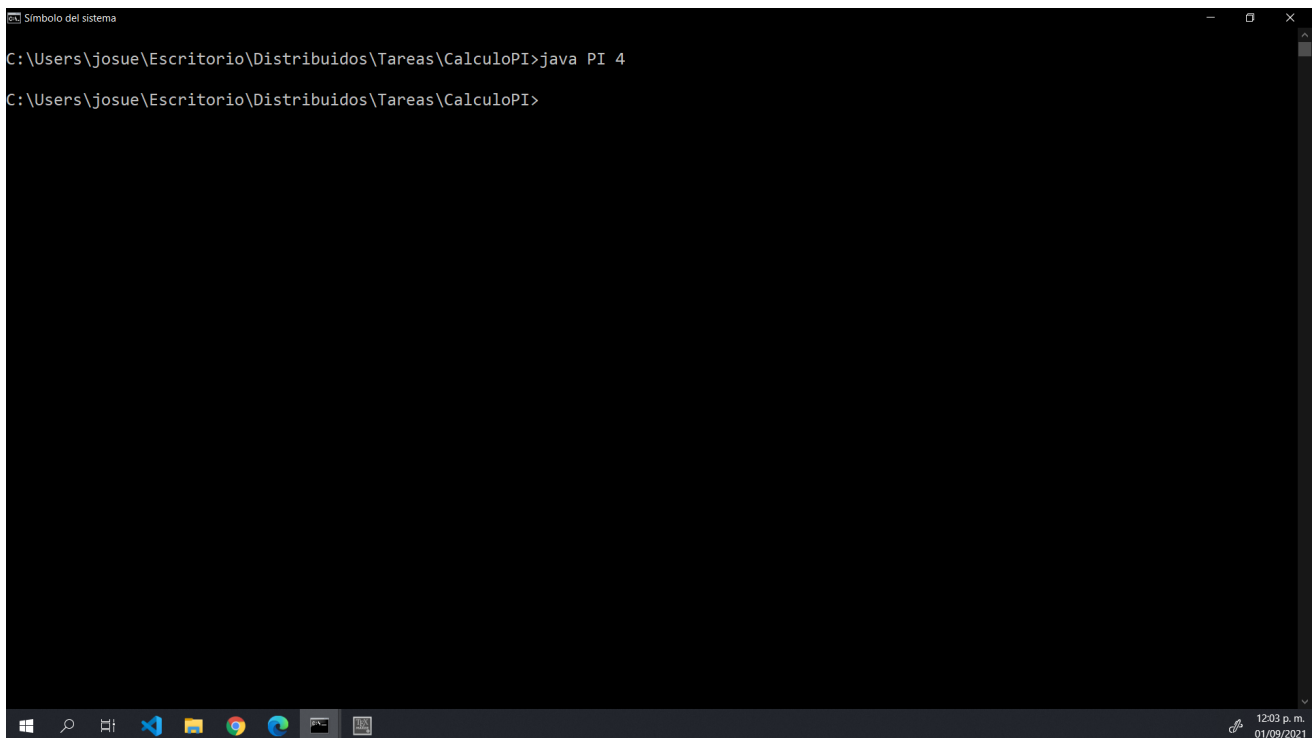
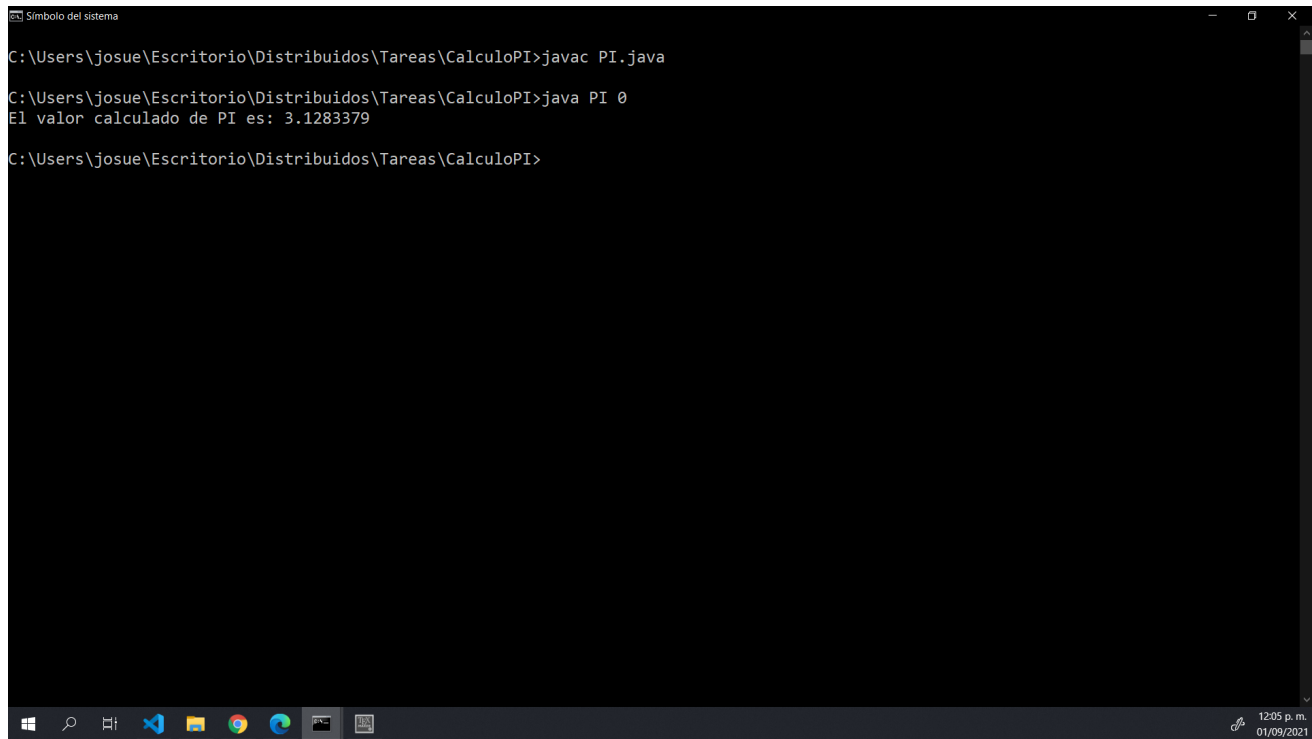


Figura 6: Ejecución del nodo 4 de nuestro sistema distribuido.

Cuando se ejecute al último cliente, la consola en donde esta el servidor mostrará el resultado obtenido de PI, el cual es el mostrado en la figura 7:



```
Símbolo del sistema
C:\Users\josue\Escritorio\Distribuidos\Tareas\CalculoPI>javac PI.java
C:\Users\josue\Escritorio\Distribuidos\Tareas\CalculoPI>java PI 0
El valor calculado de PI es: 3.1283379
C:\Users\josue\Escritorio\Distribuidos\Tareas\CalculoPI>
```

Figura 7: Valor arrojado por nuestro sistema distribuido.

3. Conclusiones

Esta tarea me ayudo a recordar los conocimientos de la materia de materias anteriores como redes 2 en el uso de java para aplicaciones de comunicaciones en red y conocimientos de redes 1 haciendo uso de una topología de tipo estrella, también un recordatorio en el uso de hilos y de herramientas para sincronizar los mismos, además de darme una pequeña introducción a los sistemas distribuidos, viendo cómo podemos dividir el trabajo haciendo uso distintos procesos o hilos para poder reducir el tiempo de ejecución. Finalmente comentar que el valor calculado de PI seria mas próximo al valor establecido de PI si usáramos el tipo de dato double en lugar de usar float esto debido a que double nos proporciona de 15 a 16 décimas en comparación con el tipo de dato float que nos proporciona de 6 a 7 décimas, es decir, double nos proporciona mayor cantidad de información la cual en este caso es útil para tener una mayor aproximación al valor de PI.

Referencias

WLynn, B. (s. f.). Pi - The Gregory-Leibniz Series. Applied Cryptography Group — Stanford University. <https://crypto.stanford.edu/pbc/notes/pi/glseries.html> [Accessed 01 September 2021].