*****Instituto Politécnico Nacional***

***Escuela Superior de Cómputo***

*Desarrollo de Sistemas Distribuidos*

***Tarea 1. Cálculo Distribuido de PI***

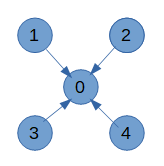
***Nombre:*** *Sampayo Hernández Mauro*

***Grupo:*** *4CV1*

***Profesor:*** *Pineda Guerrero Carlos*

**Introducción:**

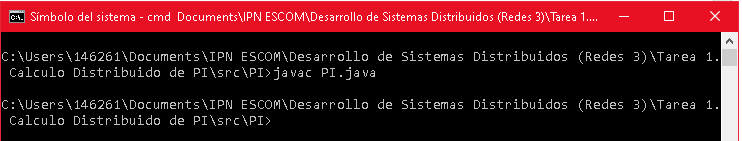
Se realizará a continuación la demostración de la compilación y ejecución de un programa Cliente-Servidor multithread que calcula una aproximación del número de pi mediante el uso de la serie de Gregory-Leibniz y mediante una tipología de estrella, consistente en 5 nodos como se muestra en la siguiente figura:



Para este caso, el nodo 0 actuará como un servidor en el programa, mientras que los nodos 1, 2, 3 y 4 actuaran como el cliente.

**Compilación del programa:**

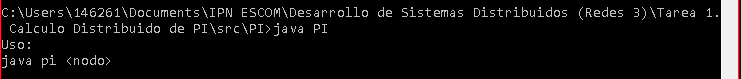
Se realiza la compilación del programa por medio del comando “javac PI.java” en la terminal de comandos de Windows.



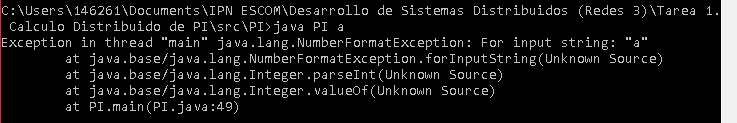
**Ejecución del programa:**

El programa por defecto requerirá de un argumento numérico que indique el nodo que se ejecutará al correr el programa.

En caso de que el programa no reciba ningún argumento, se mostrará el siguiente mensaje:



En caso de que el programa reciba un argumento no válido (que el argumento no sea un número) este arrojará el siguiente error:

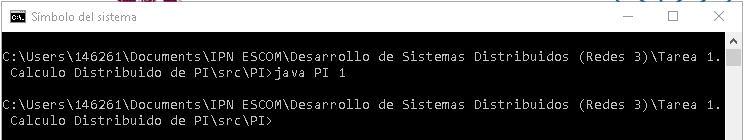


Finalmente, en el caso de que el programa reciba un argumento válido, esté se ejecutará exitosamente.

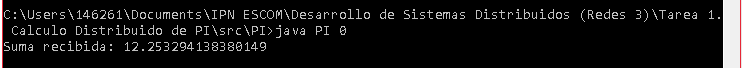
En este caso podemos ver que el programa recibió el número cero como argumento, por lo cual actuará como el servidor y quedará a la espera de que un cliente se conecte.



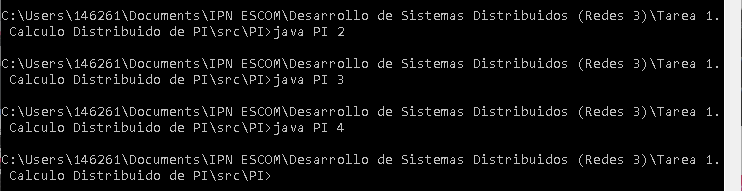
Ahora, para ejecutar al primer cliente, en otra terminal se realizará la ejecución del programa, pero recibiendo como argumento el número 1



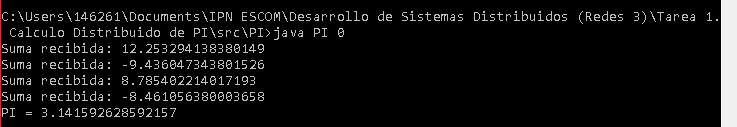
Una vez el nodo cliente haya establecido la conexión con el servidor, este mandará la suma que realizó al servidor, el cuál lo recibirá. Para mostrar el funcionamiento de esto, se incluyo una línea de código extra que imprimiese el valor recibido por el servidor desde cada uno de los 4 clientes.



Se realiza la ejecución del resto de los clientes (nodos 2, 3 y 4).



Las sumas de los clientes llegan al servidor y finalmente mediante el uso de estas, el servidor finaliza el proceso de aproximación del número pi y lo muestra en pantalla, para después finalizar su ejecución



Para finalizar, cabe mencionar que el programa al tratarse de un sistema distribuido, seguirá funcionando sin importar el orden en el que se ejecuten los nodos siempre y cuando ninguno de estos se repitan.

**Conclusión:**

El uso de nodos e hilos para la elaboración de programa de Cliente-Servidor resulta ser una opción bastante eficiente, pues de esta manera se puede realizar un proceso de envió y recibimiento de datos más rápida, y reutilizando el código ahorrándonos espacio en memoria.