

Instituto Politécnico Nacional
Escuela Superior de Cómputo



Desarrollo de Sistemas Distribuidos

Tarea 1. Cálculo Distribuido de PI

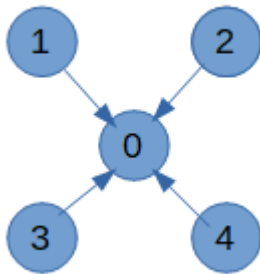
Nombre: Sampayo Hernández Mauro

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 actuarán 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.

```
Simbolo del sistema - cmd Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1...
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>javac PI.java
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>
```

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:

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>java PI
Uso:
java pi <nodo>
```

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

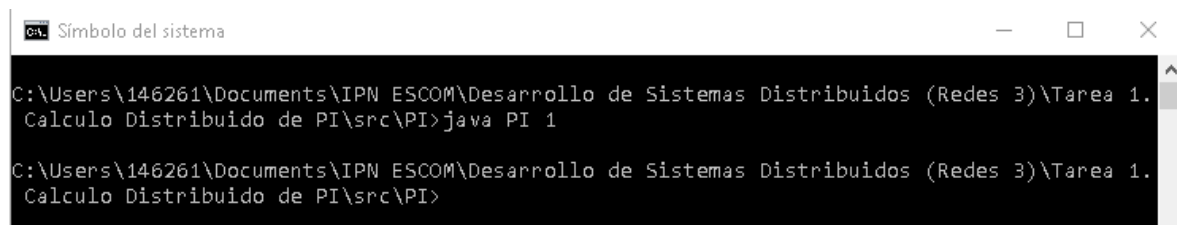
```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>java PI a
Exception in thread "main" java.lang.NumberFormatException: For input string: "a"
    at java.base/java.lang.NumberFormatException.forInputString(Unknown Source)
    at java.base/java.lang.Integer.parseInt(Unknown Source)
    at java.base/java.lang.Integer.valueOf(Unknown Source)
    at PI.main(PI.java:49)
```

Finalmente, en el caso de que el programa reciba un argumento válido, éste 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.

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>java PI 0
```

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



```
Simbolo del sistema
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>java PI 1
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>
```

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 incluyó una línea de código extra que imprimiese el valor recibido por el servidor desde cada uno de los 4 clientes.

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>java PI 0
Suma recibida: 12.253294138380149
```

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

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>java PI 2
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>java PI 3
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>java PI 4
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.
Calculo Distribuido de PI\src\PI>
```

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

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 1.  
Calculo Distribuido de PI\src\PI>java PI 0  
Suma recibida: 12.253294138380149  
Suma recibida: -9.436047343801526  
Suma recibida: 8.785402214017193  
Suma recibida: -8.461056380003658  
PI = 3.141592628592157
```

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 envío y recibimiento de datos más rápida, y reutilizando el código ahorrándonos espacio en memoria.