



Instituto Politécnico Nacional
Escuela Superior de Cómputo

Desarrollo de Sistemas Distribuidos

Tarea 1. Cálculo de PI

Diana Paola De la Cruz Sierra
Alan Ignacio Delgado Alarcon

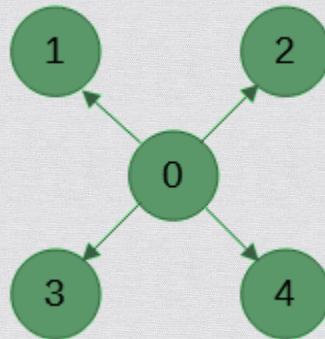
4CV11
2020/2

Unidad 1
21/02/2022

Carlos
Pineda Guerrero

Planteamiento

Desarrollar un programa distribuido que calcule una aproximación de PI haciendo uso de la serie de Gregory-Leibniz. El programa debe ejecutarse en forma distribuida sobre cinco nodos en una topología de estrella:



El nodo 0 debe actuar de cliente y los nodos 1-4 como servidores que calcularan el valor de Pi. El desarrollo deberá ser en un solo programa que decidirá si es un cliente o un servidor de acuerdo con el nodo que debe ser introducido como parámetro.

El objetivo es distribuir el cálculo de Pi en los nodos servidores. Y cada servidor deberá calcular distintas sumatorias:

- Nodo 'impar'
 - $\sum(4.0/(8 * i + 2 * (nodo - 2) + 3))$
- Nodo 'par'
 - $-\sum(4.0/(8 * i + 2 * (nodo - 2) + 3))$

Notar que cada nodo servidor calculará la sumatoria de un millón de términos, toda vez que a cada nodo le corresponde diferentes términos de la serie de Gregory-Leibniz.

El nodo 0 (nodo cliente) deberá hacer lo siguiente:

1. Conectarse a cada nodo servidor.
2. Esperar el resultado de la sumatoria calculada por cada servidor.
3. El valor de PI será la suma de las cuatro sumatorias calculadas por los servidores.
4. Desplegar el valor de PI calculado.
5. Terminar el programa.

Cada nodo servidor deberá hacer lo siguiente:

1. Esperar la conexión del nodo cliente (nodo 0).
2. Calcular la sumatoria del millón de términos que le corresponden.
3. Enviar al cliente el resultado de la sumatoria.
4. Terminar el programa.

Desarrollo

Cómo lo indica el planteamiento del problema, se codifico únicamente un solo programa, el cual fue llamado **PI**. Este a su vez se dividió en dos secciones: *cliente* y *servidor*. En cuanto al desarrollo del primero se utilizó de la clase llamada **Worker**, la cual hereda de la clase Thread; y en la parte del hilo se agregó la instrucción **synchronized** para que cada que se cree la conexión con un servidor este tenga que esperar a que el otro servidor termine de realizar las respectivas operaciones sobre la variable PI.

En total se tienen 5 nodos. En el hilo principal (*main*) se lee directo de consola el número de nodo que se quiere simular. Debido a que se tienen 4 servidores, una vez que se activa el nodo cliente (nodo 0) este creará 4 instancias de la clase *Worker* por medio de un *for*, pasando como parámetro el número actual del nodo servidor. Una vez se crean estos hilos son inicializados y posteriormente esperan a que todos terminen para poder mostrar el resultado final y así finalizar el programa.

A continuación, se muestra el procedimiento de compilación y ejecución del programa para los 5 nodos diferentes (el código fuente esta adjunto a este reporte en formato de texto):

En la siguiente imagen se observa el proceso de compilación de nuestro programa:

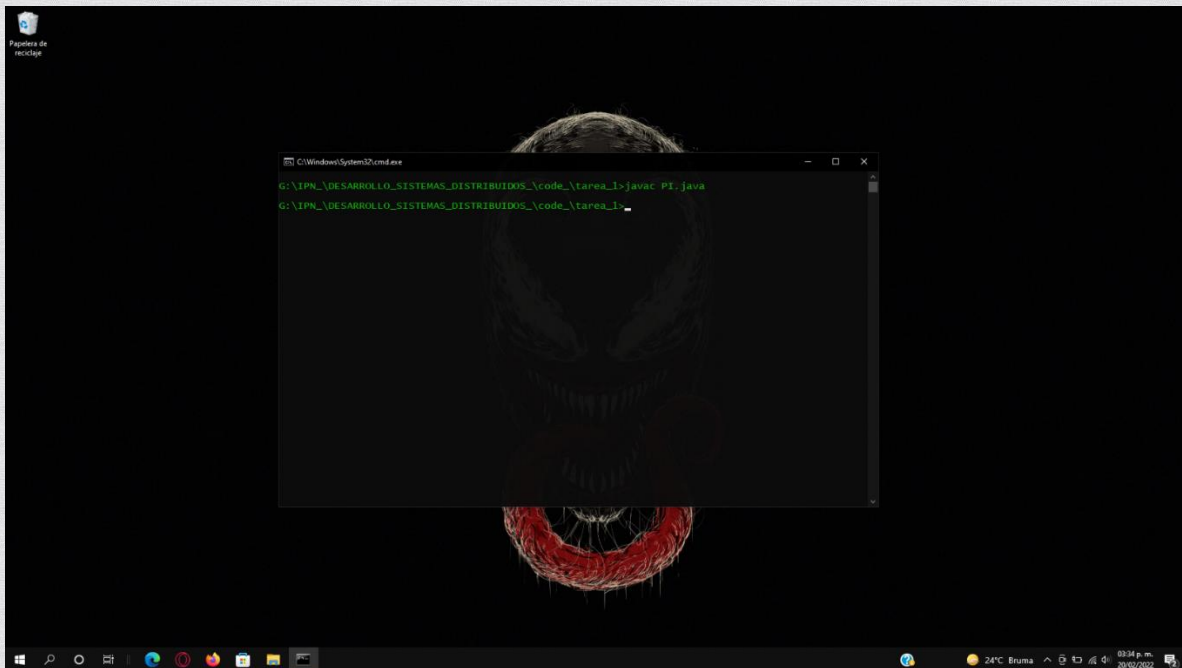


Imagen 1: Compilación del programa principal.

Posteriormente en las siguientes imágenes es mostrado el proceso de ejecución de nuestro programa PI. Primeramente, es ejecutado cada uno de los servidores y después el cliente, incluso se observa que primero termina con los primeros 3 y en la cuarto queda en espera hasta que terminen los demás.

Desarrollo de Sistemas Distribuidos

Tarea 1. Cálculo de PI

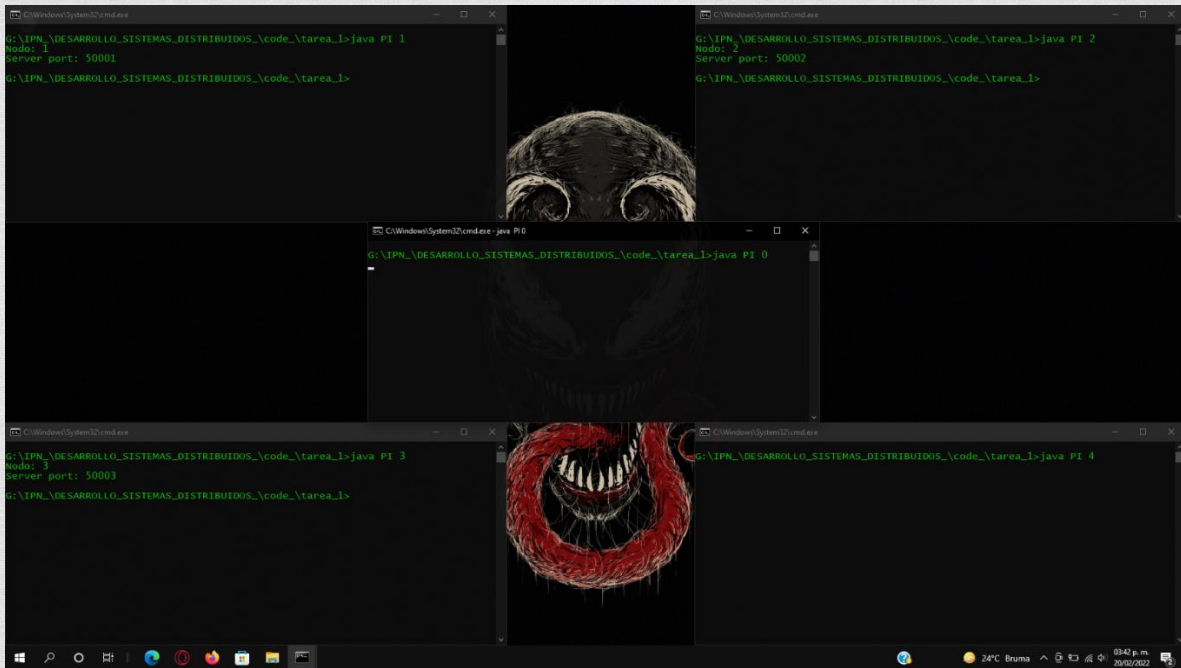


Imagen 2: Ejecución de los 4 nodos del programa

Finalmente, en la siguiente imagen se muestra cuando el nodo cliente termina de ejecutarse y por ende todos los servidores también. Una vez sucedido esto, se muestra el valor de la aproximación del valor PI.

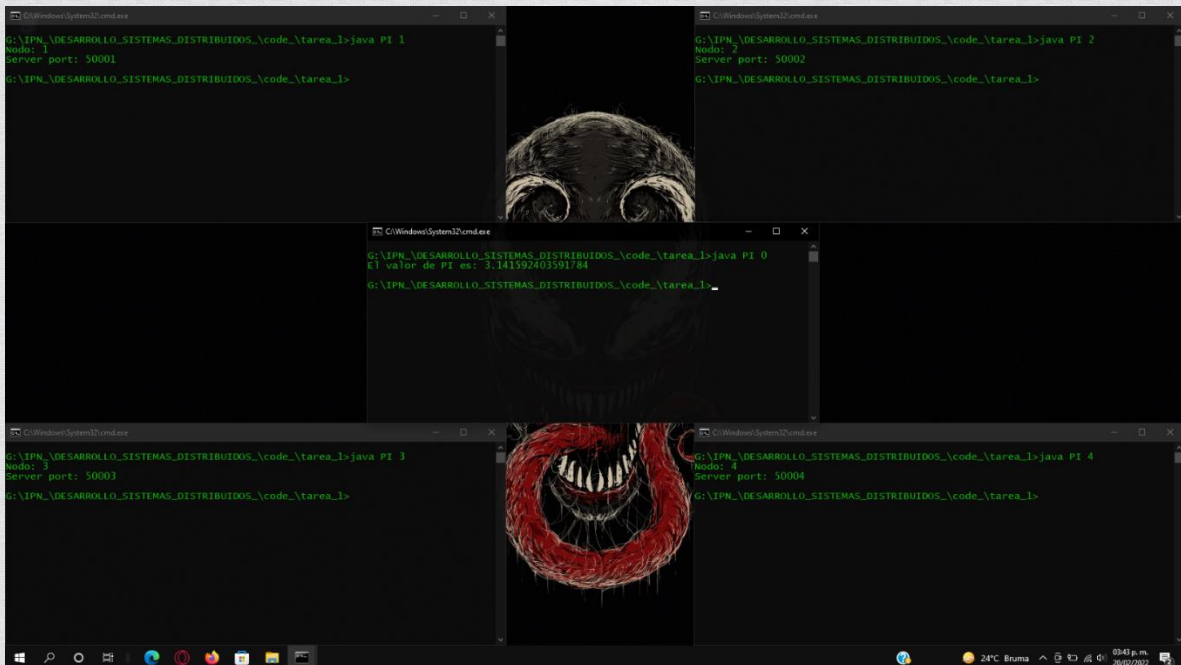


Imagen 3: Finalización de la ejecución del programa

Finalmente, como se observa en la captura de pantalla anterior la aproximación de pi es correcta.

Conclusiones

El desarrollo de esta práctica fue muy interesante ya que aplicamos gran parte de todo lo que vimos en lo que lleva del curso, lo que hace que pongamos en acción los conocimientos y comprensión que adquirimos de los diversos temas, de mi parte me ayudó a comprender la importancia de la sincronización de hilos, ya que con ello permitimos que solamente un servidor modificara a la vez a la variable PI, incluso aprendí como leer valores directos desde consola y no pedirlos al usuario por medio de Scanner como siempre he estado acostumbrada. Finalmente considero que esta práctica es la base de lo que se deberá de hacer en nuestros siguientes trabajos o prácticas.

Diana Paola De la Cruz Sierra

Durante esta tarea se aplicó un pequeño ejemplo del cómputo distribuido en aplicaciones de red, aplicando conocimientos anteriores de materias previas, haciendo uso de los hilos y sincronización de procesos que ayudan a modelar esta solución. Y que finalmente nos introducen en el desarrollo de los sistemas distribuidos y la importancia de dividir el cálculo computacional y sus ventajas.

Alan Ignacio Delgado Alarcón