



Instituto Politécnico Nacional.

Escuela Superior de Cómputo.

**Materia:
Desarrollo de Sistemas Distribuidos.**

**Tema:
Cálculo de Pi.
(Tarea 01)**

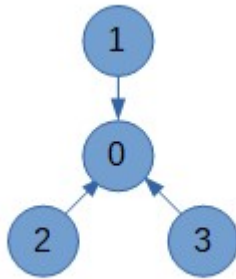
**Profesor:
Carlos Pineda Guerrero.**

**Alumno:
Mario Alberto Miranda Sandoval.**

**Grupo:
4CM5**

Objetivo.

Calcular el valor de Pi mediante la serie de Gregory-Leibniz a través de un programa multi-threading, armando la siguiente topología.



Donde el nodo 0 fungirá como servidor, mientras que los demás nodos se conectarán a este. Para esta tarea se usara la misma computadora por lo que se usaran diversas terminales para emular el funcionamiento.

Se debe mencionar que todos los nodos harán 10 millones de términos de la serie que se describe a continuación.

$$\pi = 4 \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} + \dots \right)$$

Pruebas.

Comenzaremos por compilar el programa.

```
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ javac Pi.java
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$
```

Una vez compilado el programa, proseguimos a correr el servidor (nodo 0).



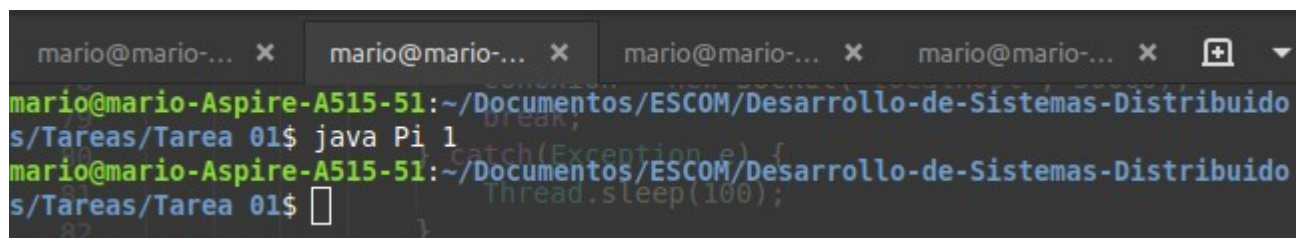
```
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ javac Pi.java
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 0
79
break;
```

Cuando el programa recibe el 0 por argumento, se crea el socket servidor y se queda a la espera de las conexiones de los clientes, para este programa usaremos 3 clientes.

Cuando un cliente se conecta se crea una instancia de la clase Worker, posteriormente se ejecuta el código del hilo de cada cliente (posteriormente se explica acerca de los clientes), después de que esto suceda el nodo 0 queda la escucha de la siguiente conexión de otro cliente.

Una vez conectados todos los clientes, el servidor procede a calcular los términos de la serie correspondiente, después de eso en el método synchronized actualizamos el valor de pi.

Ya por último llamamos al join de cada Worker, se cierra el servidor y mostramos el valor de pi.



```
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 1
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$
```

Aquí es cuando un cliente se conecta al servidor, sabemos que son 3 clientes pero para cada cliente el proceso es similar.

Cada cliente tiene predeterminada la conexión con re-intento, si la conexión falla el cliente intentará volverse a conectar cada 100 mili-segundos, una vez conectado el cliente, se crea el flujo de salida, se procede a calcular los términos de cada nodo, luego la siguiente línea es importante ya que

dependiendo del nodo se diferencia si al valor general se debe sumar o restar, posteriormente mandamos el valor calculado al servidor, cerramos el flujo de salida y la conexión del socket.

Por último, se puede ver como se calcula el valor de Pi de forma correcta sin importar el orden en el que son llamados, siempre y cuando no se repitan los nodos.

```
mario@mario-... x mario@mario-... x mario@mario-... x mario@mario-... x + v
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ javac Pi.java
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 0
Valor de pi: 3.141592628592157
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 0
Valor de pi: 3.141592628592157
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 0
Valor de pi: 3.141592628592157
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$
```

```
mario@mario-... x mario@mario-... x mario@mario-... x mario@mario-... x + v
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 1
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 2
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 3
```

```
mario@mario-... x mario@mario-... x mario@mario-... x mario@mario-... x + v
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 2
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 1
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 2
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
```

```
mario@mario-... x mario@mario-... x mario@mario-... x mario@mario-... x + v
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 3
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 3
mario@mario-Aspire-A515-51:~/Documentos/ESCOM/Desarrollo-de-Sistemas-Distribuido
s/Tareas/Tarea 01$ java Pi 1
```