

INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

TAREA 1 Calculo de PI

Desarrollo de Sistemas Distribuidos

Nombre: Morales Garcia Christian Arturo

Grupo: 4CV2

Fecha de entrega: mar 5 2020

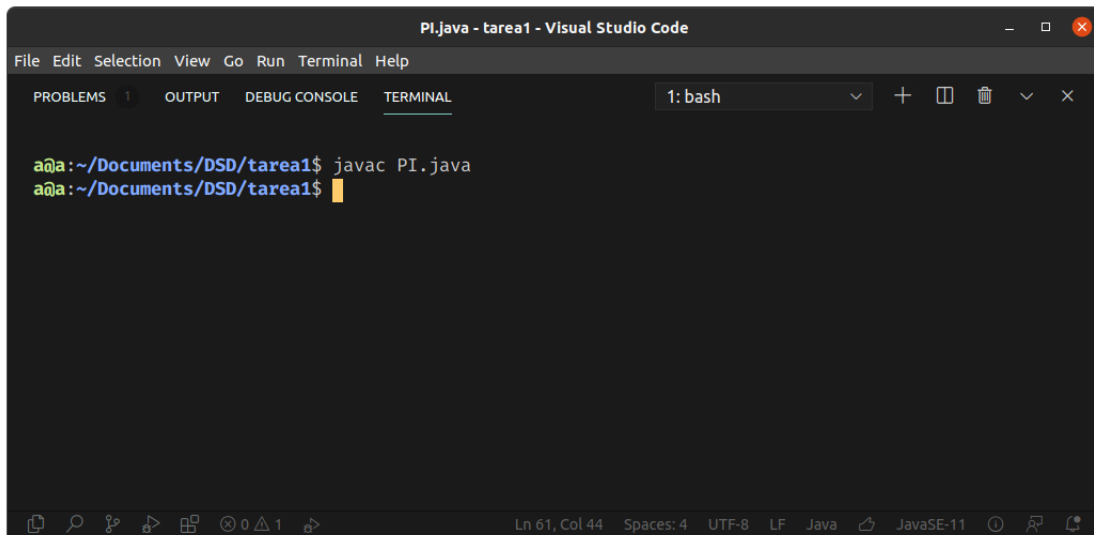
Fecha: Marzo 4 2020

1 Compilación

Para compilar usamos el siguiente comando:

```
javac <nombre_programa>.java
```

Podemos observar que se ha compilado sin ningún problema.



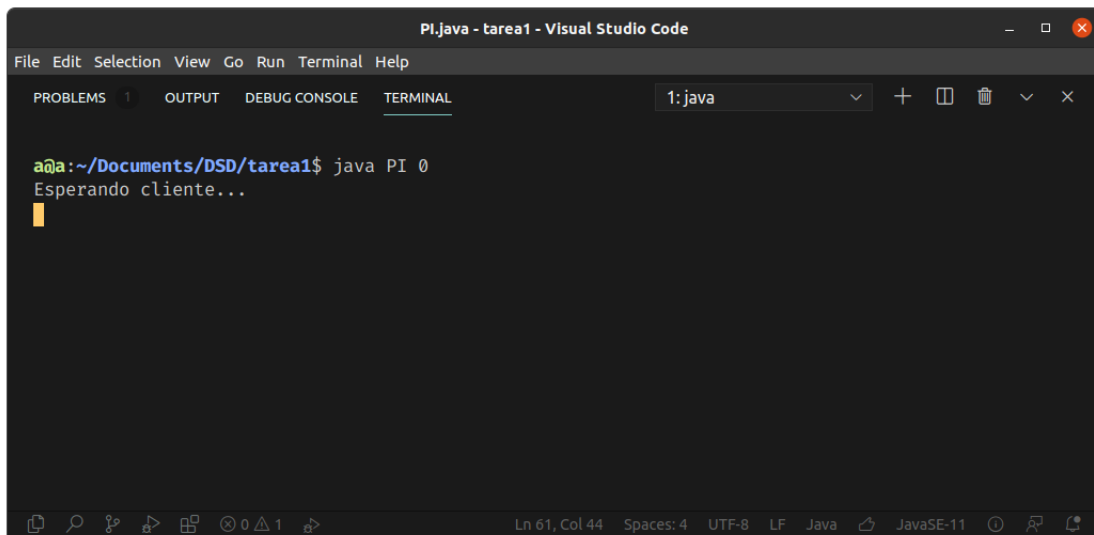
2 Ejecución de programa

El comando para ejecutar el programa es:

```
java <nombre_programa_compilado> <numero_de_nodo>
```

2.1 Inicio de ejecución de servidor

Recordemos que para iniciar el programa primero debemos iniciar el servidor y el servidor es el nodo cero por lo que es necesario colocar el "0" después de nombre del programa compilado. En la siguiente imagen se muestra el comando completo:



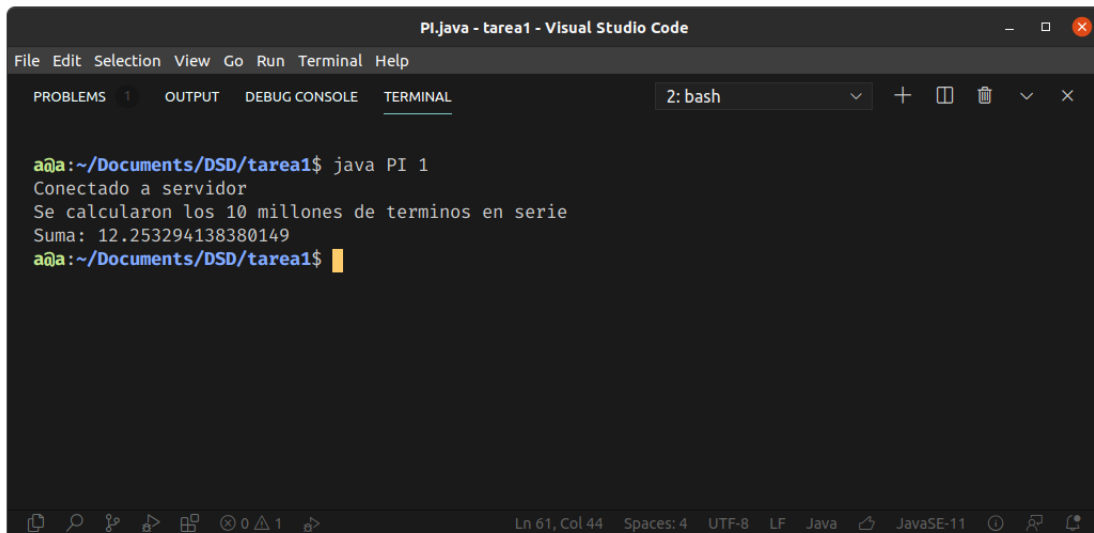
The screenshot shows a Visual Studio Code window titled "PI.java - tarea1 - Visual Studio Code". The terminal panel is active, displaying the command `java PI 0` and the output `Esperando cliente...`. The terminal prompt is `aa@:~/Documents/DSD/tarea1$`. The status bar at the bottom indicates the file is at line 61, column 44, with 4 spaces, UTF-8 encoding, LF line endings, and the Java language.

```
PI.java - tarea1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 1:java
aa@:~/Documents/DSD/tarea1$ java PI 0
Esperando cliente...
Ln 61, Col 44 Spaces: 4 UTF-8 LF Java JavaSE-11
```

Al iniciar el programa aun no se ha conectado ningún cliente al servidor, así que se colocó un texto que nos indica que se están esperando a los clientes y cuando se conecte algún cliente se mostrara un texto que indique que se conectó un cliente.

2.2 Ejecución cliente 1

En este punto el servidor ya está iniciado y esperando conexiones, como este es el primer cliente el valor del nodo es 1 y el comando queda como se observa en la imagen.

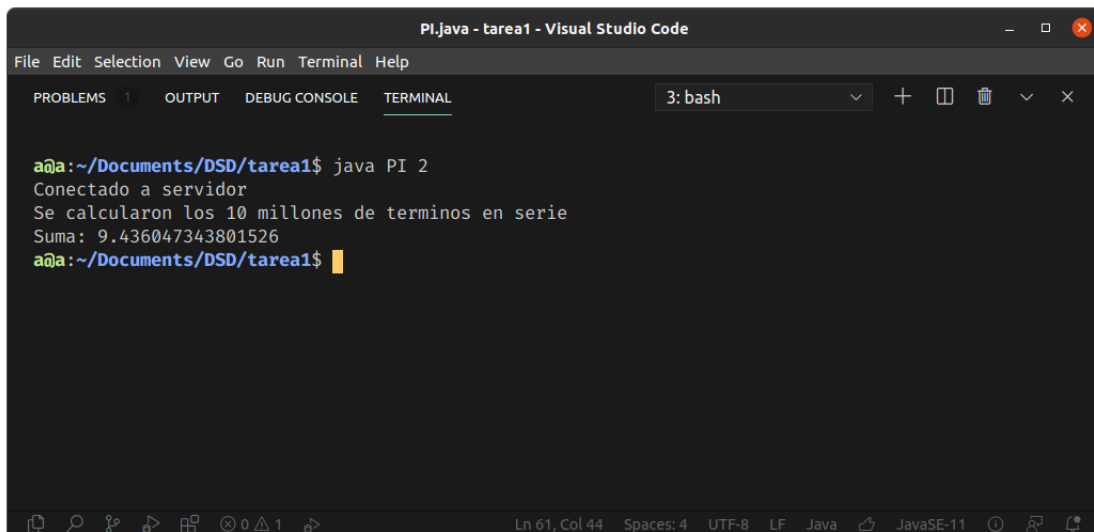


```
PI.java - tarea1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 2: bash
a@a:~/Documents/DSD/tarea1$ java PI 1
Conectado a servidor
Se calcularon los 10 millones de terminos en serie
Suma: 12.253294138380149
a@a:~/Documents/DSD/tarea1$
```

En el programa del cliente se colocó un texto que nos indica cuando el cliente ya está conectado al servidor, mas adelante comenzará a calcular la suma de los 10 millones de términos en serie, nos indicará cuando termine de calcularlos, envía la suma al servidor y al final nos dará el valor de la suma de todos los términos.

2.3 Ejecución cliente 2

Continuamos con el siguiente nodo de la misma manera en que lo hicimos con el nodo anterior pero como este es el segundo nodo el valor del nodo es 2 y el comando queda como se observa en la imagen.



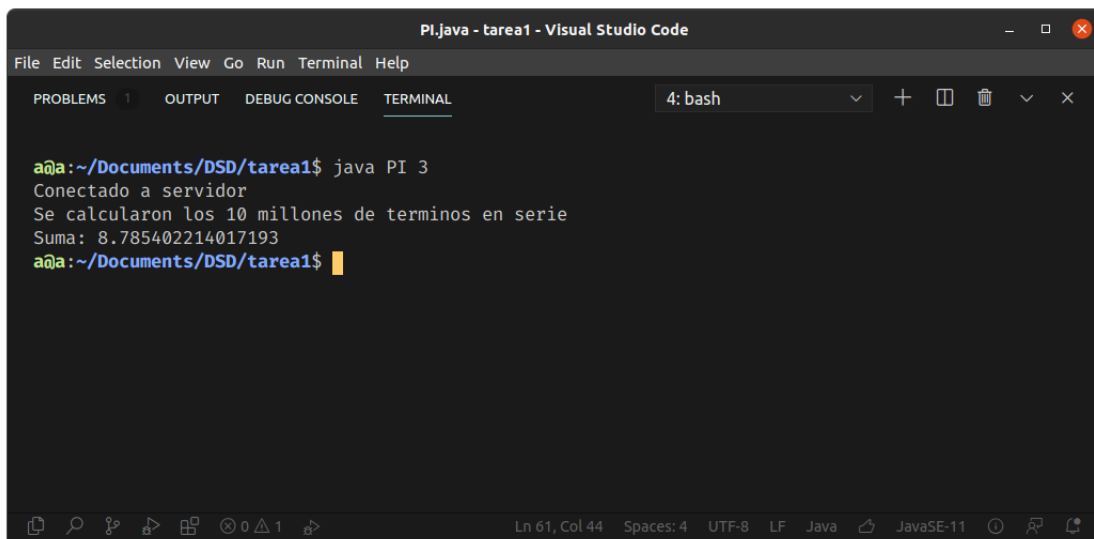
```
PI.java - tarea1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 3: bash
a@a:~/Documents/DSD/tarea1$ java PI 2
Conectado a servidor
Se calcularon los 10 millones de terminos en serie
Suma: 9.436047343801526
a@a:~/Documents/DSD/tarea1$
```

Como ya se indicó anteriormente en el programa del cliente se colocó un texto que nos indica cuando el cliente ya está conectado al servidor, mas adelante comenzará

a calcular la suma de los 10 millones de términos en serie, nos indicara cuando termine de calcularlos, envía la suma al servidor y al final nos dará el valor de la suma de todos los términos.

2.4 Ejecución cliente 3

Ahora continuamos con el nodo tres, de la misma manera en que lo hicimos con el nodo anterior pero como este es el tercero nodo el valor del nodo es 3 y el comando queda como se observa en la imagen.

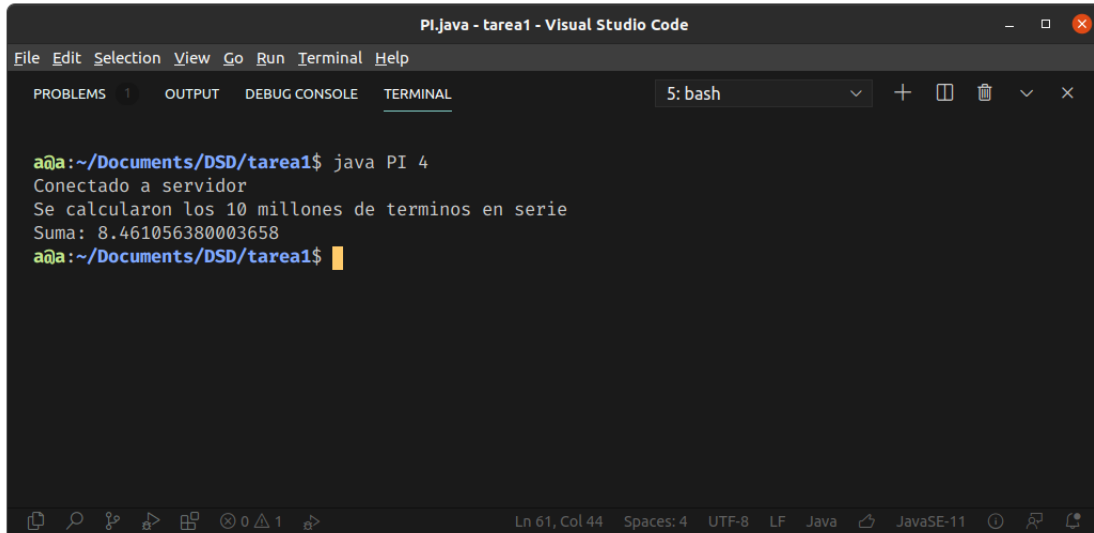


```
PI.java - tarea1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 4: bash
a@a:~/Documents/DSD/tarea1$ java PI 3
Conectado a servidor
Se calcularon los 10 millones de terminos en serie
Suma: 8.785402214017193
a@a:~/Documents/DSD/tarea1$
```

De la misma manera a como se hizo en el nodo pasado también se muestra un texto que nos indica cuando el cliente ya esta conectado al servidor, después comenzara a calcular la suma de los 10 millones de términos en serie, nos indicara cuando termine de calcularlos, envía la suma al servidor y al final nos dará el valor de la suma de todos los términos.

2.5 Ejecución cliente 4

Finalmente tenemos el cuarto nodo y de la misma manera en que lo hicimos con el nodo anterior pero como este es el cuarto nodo el valor del nodo es 4 y el comando queda como se observa en la imagen.

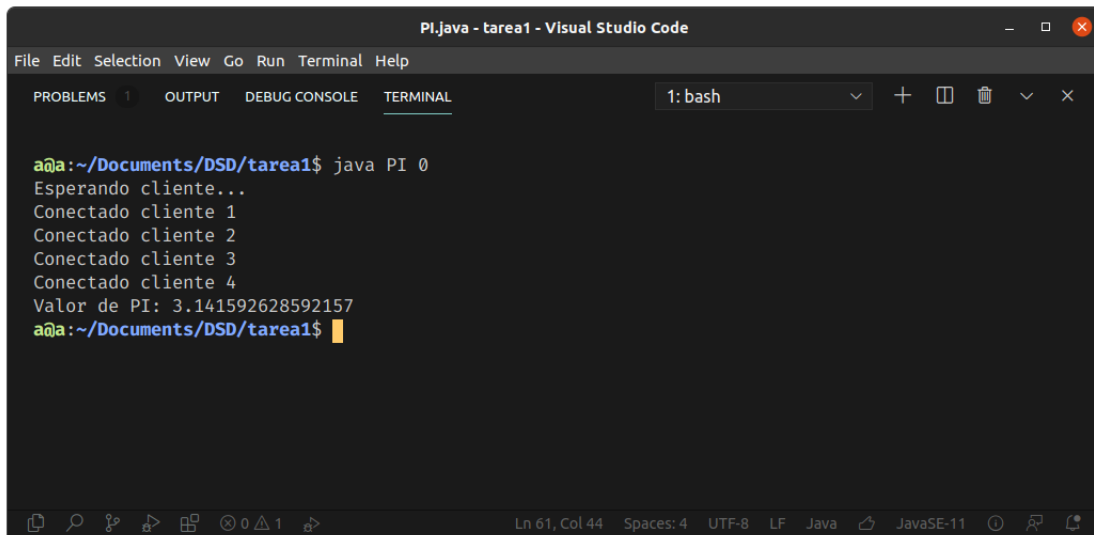


```
PI.java - tarea1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 5: bash
a@a:~/Documents/DSD/tarea1$ java PI 4
Conectado a servidor
Se calcularon los 10 millones de terminos en serie
Suma: 8.461056380003658
a@a:~/Documents/DSD/tarea1$
```

Terminamos haciendo por ultima vez el despliegue en la terminal del texto que nos indica cuando el cliente ya esta conectado al servidor, calcula la suma de los 10 millones de términos en serie, nos indicara cuando termine de calcularlos, envía la suma al servidor y al final nos dará el valor de la suma de todos los términos.

2.6 Final de ejecución de servidor y valor de PI

En la terminal del servidor podemos observar que se fue indicando el cliente que se iba conectado y al final cuando todos los cliente terminaron de hacer su calculo entregaron su resultado al servidor el cual lo sumo y entrego el valor final de PI, el cual se muestra al final de la ejecución del servidor.



```
PI.java - tarea1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 1: bash
ana:~/Documents/DSD/tarea1$ java PI 0
Esperando cliente...
Conectado cliente 1
Conectado cliente 2
Conectado cliente 3
Conectado cliente 4
Valor de PI: 3.141592628592157
ana:~/Documents/DSD/tarea1$
```

3 Conclusiones

Se puede concluir que el dividir la carga de trabajo para solucionar problemas grandes es muy útil pero si lo hacemos es necesario que exista una coordinación entre las elementos que realizan cada tarea, en este caso tenemos diferentes hilos que se encargan de resolver una parte específica del problema. Para este problema se necesita que cada hilo calculara 10 millones de términos pero como son varios hilos trabajando al mismo tiempo y no se sabe el orden en el que irán terminando se debe coordinar el momento en el que leen y escriben la variable que guarda el resultado obtenido del cálculo de cada hilo, esto con el objetivo de evitar la inconsistencia de datos y al hacer esta coordinación tengamos en cuenta que el programa tardara mas en ejecutarse.