



Instituto Politécnico Nacional

ESCOM

DESARROLLO DE SISTEMAS DISTRIBUIDOS

4CV2

Tarea 1

Gabriel Gómez Téllez

5 Marzo 2021

Compilación de archivo

```
+ javac PI.java 14:13
ml in ~/personal_dev/sistemas_distribuidos/tarea_1 on master !
+ 14:19
```

Utilizando javac creo los archivos PI.class y PI\$Worker.class, no hay ninguna warning ni error.

Ejecución de servidor (Nodo 0)

```
+ java PI 0 14:19
NODO: 0
```

Ejecuto la clase PI con el argumento 0, imprimo el número del nodo para confirmar que se está corriendo correctamente. El servidor se queda esperando las conexiones de los clientes.

Ejecución de cliente 1

```
0 master % java PI 1
NODO: 1

ml@C02XT2D2JK77 ~/personal_dev/sistemas_distribuidos/tarea_1
```

Ejecuto desde otra pestaña el cliente uno con la clase PI a través del argumento 1. Se conecta calcula su suma, se la manda al servidor y se desconecta.

```
+ java PI 0 14:19
NODO: 0
12.253294138380149
```

Del lado del servidor imprimo en la pantalla el número recibido así que podemos ver que imprime un 12.25.. de parte del cliente 1. Se queda esperando una siguiente conexión.

Ejecución de cliente 2

```
└─ java PI 2
NODO: 2
ml@C02XT2D2JK77 ~/personal_dev/sistemas_distribuidos/tarea_1 <master*>
└─
```

Nuevamente ejecuto el cliente desde otra pestaña ahora con el argumento 2. Hace su trabajo y se desconecta.

```
+ java PI 0
NODO: 0
12.253294138380149
-9.436047343801526
14:19
```

Podemos volver a ver en el servidor que imprime un segundo número, como lo indica la serie este número es negativo. Se queda esperando el servidor una siguiente conexión.

Ejecución de cliente 3

```
tarea_1> java PI 3
NODO: 3
```

Nuevamente desde una nueva pestaña ejecuto el cliente con el argumento 3, hace sus cálculos y los manda para terminar desconectandose del servidor.

```
+ java PI 0
NODO: 0
12.253294138380149
-9.436047343801526
8.785402214017193
14:19
```

Puedo ver por tercera vez como el servidor devuelve un número, este siendo el que calculo y envió el cliente número 3. Se queda esperando una 4 conexión.

Ejecución de cliente 4

```
> java PI 4      master [3fc6b3e] untracked
NODO: 4

ml@C02XT2D2JK77:~/personal_dev/sistemas_distribuidos/tarea_1 (*)
>      master [3fc6b3e] untracked
```

Desde una nueva pestaña volvemos a ejecutar un 4o cliente y su comportamiento es igual al resto.

```
+ java PI 0      14:19
NODO: 0
12.253294138380149
-9.436047343801526
8.785402214017193
-8.461056380003658
PI: 3.141592628592157

ml in ~/personal_dev/sistemas_distribuidos
/tarea_1 on master !
+      14:23
```

Por último vemos el servidor desplegar un 4o resultado del cliente y al final la suma final siendo una aproximación de PI.

Conclusión

Vemos que distribuyendo los cálculos de las diferentes partes de la serie podemos desacoplar el procesamiento y terminamos con una arquitectura escalable. Entre más computadoras tengamos más nos podríamos acercar a PI sin que el tiempo de ejecución sea afectado.