

Informe práctica 3

Nombre: Francisco de Asís Carrasco Conde

DNI: 26301846N

1.Ejemplo 1

El ejemplo 1 es un ejemplo muy básico de la arquitectura cliente servidor. Se inicia un servidor, y se ejecuta un solo cliente que envía un valor(id_proceso) simulando el numero de proceso escogido por el cliente y se imprime en la terminal del servidor el numero de proceso invocado por el cliente.

```
String nombre_objeto_remoto = "Ejemplo_I";
System.out.println("Buscando el objeto remoto");
Registry registry = LocateRegistry.getRegistry(args[0]);
Ejemplo_I instancia_local = (Ejemplo_I) registry.lookup(nombre_objeto_remoto);
System.out.println("Invocando el objeto remoto");
instancia_local.escribir_mensaje(Integer.parseInt(args[1]));
} catch (Exception e) {
System.err.println("Ejemplo_I exception:");
e.printStackTrace();
}
}
```

Ejecución en terminal macro.sh:

```
Lanzando el servidor
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple.Ejemplo (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo1/)
WARNING: Please consider reporting this to the maintainers of simple.Ejemplo
WARNING: System::setSecurityManager will be removed in a future release
Ejemplo bound

Lanzando el primer cliente

WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple.Cliente_Ejemplo (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo1/)
WARNING: Please consider reporting this to the maintainers of simple.Cliente_Ejemplo
WARNING: System::setSecurityManager will be removed in a future release
Buscando el objeto remoto
Invocando el objeto remoto
Recibida petición de proceso: 0
Empezamos a dormir
Terminamos de dormir

Hebra 0

Lanzando el segundo cliente

WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple.Cliente_Ejemplo (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo1/)
WARNING: Please consider reporting this to the maintainers of simple.Cliente_Ejemplo
WARNING: System::setSecurityManager will be removed in a future release
Buscando el objeto remoto
Invocando el objeto remoto
Recibida petición de proceso: 3

Hebra 3
pacc@pacc-Victus-by-HP-Laptop-16-e0xxx:~/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo1$
```

2.Ejemplo 2

El ejemplo 2 evoluciona un poco, implica 3 hebras. Este ejemplo nos muestra que es posible la concurrencia usando 1 sola instancia del servidor. Entran 3 procesos de manera ordenada, y acaban saliendo de manera desordenada.

```

int n_hebras = Integer.parseInt(args[1]);
Cliente_Ejemplo_Multi_Threaded[] v_clientes = new
Cliente_Ejemplo_Multi_Threaded[n_hebras];
Thread[] v_hebras = new Thread[n_hebras];
for (int i=0;i<n_hebras;i++) {
//A cada hebra le pasamos el nombre el servidor
v_clientes [i] = new Cliente_Ejemplo_Multi_Threaded(args[0]);
v_hebras[i] = new Thread(v_clientes[i],"Cliente "+i);
v_hebras[i].start();
}

```

Ejecución en terminal macro.sh:

```

Lanzando el servidor
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple2.Ejemplo (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo2/)
WARNING: Please consider reporting this to the maintainers of simple2.Ejemplo
WARNING: System::setSecurityManager will be removed in a future release
Ejemplo bound

Lanzando el primer cliente

WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple2.Cliente_Ejemplo_Multi_Threaded (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo2/)
WARNING: Please consider reporting this to the maintainers of simple2.Cliente_Ejemplo_Multi_Threaded
WARNING: System::setSecurityManager will be removed in a future release

Lanzando el segundo cliente

WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple2.Cliente_Ejemplo_Multi_Threaded (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo2/)
WARNING: Please consider reporting this to the maintainers of simple2.Cliente_Ejemplo_Multi_Threaded
WARNING: System::setSecurityManager will be removed in a future release
Buscando el objeto remoto
Buscando el objeto remoto
Buscando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto

Entra Hebra Cliente 0
Empezamos a dormir

Entra Hebra Cliente 1

Entra Hebra Cliente 2
Sale Hebra Cliente 1
Sale Hebra Cliente 2
Terminamos de dormir
Sale Hebra Cliente 0
pacc@pacc-Victus-by-HP-Laptop-16-e0xxx:~/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo2$ 

```

3.Ejemplo 3

En el ejemplo 3, lanzamos 2 procesos cliente que tratan de aumentar un contador de un servidor, uno detrás de otro.

```

System.out.println("Poniendo contador a 0");
micontador.sumar(valor:0);
// Obtiene hora de comienzo
long horacomienzo = System.currentTimeMillis();
// Incrementa 1000 veces
System.out.println("Incrementando...");
for (int i = 0 ; i < 1000 ; i++ ) {
micontador.incrementar();
}
// Obtiene hora final, realiza e imprime calculos
long horafin = System.currentTimeMillis();
System.out.println("Media de las RMI realizadas = "
+ ((horafin - horacomienzo)/1000f)
+ " msecs");
System.out.println("RMI realizadas = " + micontador.sumar());
} catch (NotBoundException | RemoteException e) {
System.err.println("Exception del sistema: " + e);
}
System.exit(0);
}

```

Ejecución en terminal macro.sh:

```

Lanzando el servidor
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple.servidor (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo3/)
WARNING: Please consider reporting this to the maintainers of simple.servidor
WARNING: System::setSecurityManager will be removed in a future release
Servidor RemoteException | MalformedURLExceptionondor preparado

Lanzando el primer cliente

WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple.cliente (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo3/)
WARNING: Please consider reporting this to the maintainers of simple.cliente
WARNING: System::setSecurityManager will be removed in a future release
Poniendo contador a 0
Incrementando...
Media de las RMI realizadas = 0.174 msecs
RMI realizadas = 1000

Lanzando el segundo cliente

WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by simple.cliente (file:/home/pacc/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo3/)
WARNING: Please consider reporting this to the maintainers of simple.cliente
WARNING: System::setSecurityManager will be removed in a future release
Poniendo contador a 0
Incrementando...
Media de las RMI realizadas = 0.092 msecs
RMI realizadas = 1000
pacc@pacc-Victus-by-HP-Laptop-16-e0xxx:~/Escritorio/DSD/Practicas/Apuntes/P3/ejemplo3$

```

4.Ejercicio

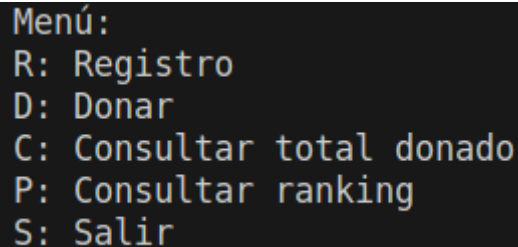
Este ejercicio esta compuesto por:

- 2 interfaces: MilInter (funciones del servidor) y conexión (conexión entre ambos servidores)
- Clase auxiliar, donación
- Clase cliente
- Clase servidor

El modo de ejecución del ejercicio es el siguiente:

1. Ejecutamos el script macro.sh desde ejercicio, este lanzará los 2 servidores. Luego en la misma terminal nos empezará a preguntar nuestro usuario y servidor al que conectarnos.

2. Se nos desplegará un menú , vamos seleccionando las opciones.



```
Menú:  
R: Registro  
D: Donar  
C: Consultar total donado  
P: Consultar ranking  
S: Salir
```

3. Para terminar la ejecución escribimos EXIT, una vez salgamos del servidor usando al cliente.

Cooperación entre servidores

Para comparar la información entre ambos servidores, guardo una copia del otro en cada replica, es decir, en el servidor1 tengo una copia del 2 y viceversa. El atributo en cuestión es br. Para las operaciones de que necesitan usar al otro servidor he usado una interfaz, conexión. La cual tiene 3 funciones:

- server registradoEn(String usuario): devuelve el servidor en el que esta registrado un usuario, si no está en ninguno devuelve null
- public server menosEntidades(): devuelve el servidor con menos entidades registradas

Métodos Cliente

En el cliente solo tengo el main que contiene el despliegue de un menú para manejar el servidor

Métodos Servidor

- updateR(): actualiza el ranking del servidor que llama a esta función
- setBr(server replic): añade una de otro server replica
- registrarse(String usuario): registra al usuario
- donar(String donante, int cantidad): realiza una donación
- consultar(String usuario): devuelve la cantidad total donada
- consulta_Ranking(): consulta el ranking

Funciones añadidas

He decidido añadir la funcionalidad de ranking, se enseña de mayor a menor donante en ambos servidores.

```
P
dfsddfsd --> 6546546 €
pac --> 4665465 €
opopo --> 64646 €

Menú:
R: Registro
D: Donar
C: Consultar total donado
P: Consultar ranking
S: Salir
```