

Práctica 3: RMI ~ José Santos Salvador

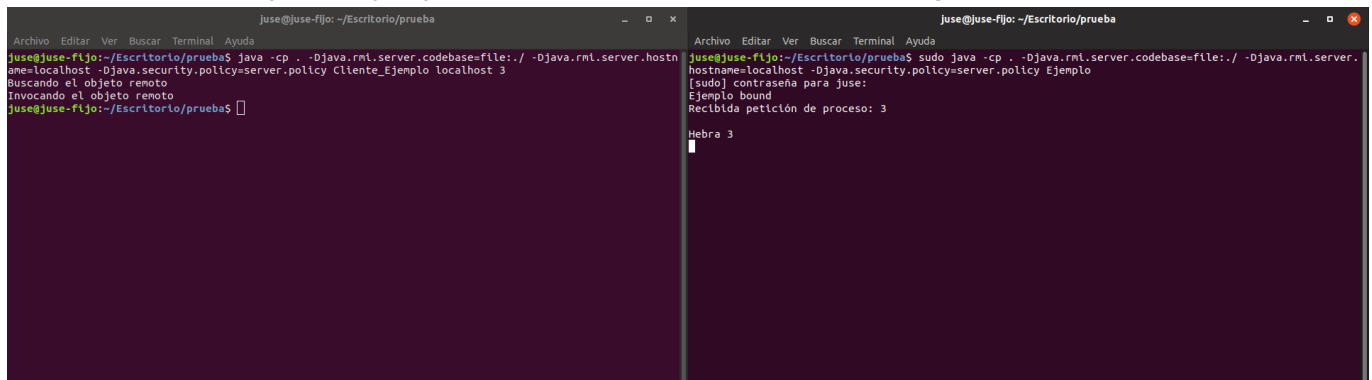
Parte 1

En esta práctica usamos la tecnología RMI que se basa en la implementación de una serie de formularios de petición de objetos. Estos objetos se pueden transportar entre cliente y servidor. Esto permite utilizar objetos de cualquier tipo, ofreciendo más versatilidad que las tecnologías usadas en la práctica 1 y la 2.

Ejemplo 1

El primer ejemplo consiste en un programa cliente-servidor básico. El servidor exporta los métodos de la interfaz y cuando un cliente le envía una petición, imprime el argumento enviado. El cliente por su parte, activa el gestor de seguridad, busca el objeto remoto e invoca el método `escribir_mensaje` con el argumento pasado.

Primero realicé este primer ejemplo usando el terminal como te indica el guión.



```
Juse@Juse-fijo: ~/Escritorio/prueba
juse@Juse-fijo:~/Escritorio/prueba$ java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente_Ejemplo localhost 3
Buscando el objeto remoto
Invocando el objeto remoto
juse@Juse-fijo:~/Escritorio/prueba$

Juse@Juse-fijo: ~/Escritorio/prueba
juse@Juse-fijo:~/Escritorio/prueba$ sudo java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Ejemplo
[sudo] contraseña para juse:
Ejemplo bound
Recibida petición de proceso: 3
Hebra 3
```

Después de compilarlo y realizarlo por terminal, decidí hacerlo con netbeans. Para ello copie los archivos `Cliente_Ejemplo.java`, `Ejemplo.java`, `Ejemplo_I.java` y `client.policy` en Netbeans. Cree una configuración para el servidor

```
-Djava.security.manager -Djava.security.policy=./src/server.policy
```

y otra para el cliente

```
-Djava.security.manager -Djava.security.policy=./src/client.policy
```

en VM option y para el cliente los argumentos que se le pasaba antes en la casilla Arguments

```
localhost 3
```

```

1
2 import java.rmi.registry.LocateRegistry;
3 import java.rmi.registry.Registry;
4
5 /**
6  *
7  * @author juse
8  */
9 public class Cliente_Ejemplo {
10     public static void main(String[] args) {
11         if (System.getSecurityManager() == null){
12             System.setSecurityManager(new SecurityManager());
13         }
14         try{
15             String nombre_objeto_remoto = "un_nombre_para_obj_remoto";
16             Registry registry = LocateRegistry.getRegistry(args[0]);
17             System.out.println("Buscando el objeto remoto");
18             Ejemplo_I instancia_local = (Ejemplo_I) registry.lookup(nombre_objeto_remoto);
19             System.out.println("Invocando el objeto remoto");
20             instancia_local.escribir_mensaje(Integer.parseInt(args[1]));
21         } catch (Exception e){
22             System.err.println("Ejemplo_I exception:");
23             e.printStackTrace();
24         }
25     }
26 }
27

```

```

run:
Buscando el objeto remoto
Invocando el objeto remoto
BUILD SUCCESSFUL (total time: 0 seconds)

```

```
1  /*
2   * To change this license header, choose License Headers in Project Properties
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  import java.rmi.RemoteException;
8  import java.rmi.registry.LocateRegistry;
9  import java.rmi.registry.Registry;
10 import java.rmi.server.UnicastRemoteObject;
11
12 /**
13  *
14  * @author juse
15  */
16 public class Ejemplo implements Ejemplo_I{
17
18     /**
19      * @param args the command line arguments
20      */
21
22     public Ejemplo(){
23         super();
24     }
25
26     public static void main(String[] args) {
27         if (System.getSecurityManager() == null){
28             System.setSecurityManager(new SecurityManager());
29         }
30         try{
31             Ejemplo_I prueba = new Ejemplo();
```

Output x

Ejemplo (run) x Ejemplo (run) #2 x

run:
Buscando el objeto remoto
Invocando el objeto remoto
BUILD SUCCESSFUL (total time: 0 seconds)

Uno de los problemas que tuve realizandolo en netbeans fué el archivo policy, que tuve que modificar pero al confundirme con el guión de la práctica, no lo modifica de forma correcta. Y también tuve algún contratiempo buscando donde lanzar rmiregistry (que se solventa en el ejemplo3, donde no es necesario lanzar en build/classes).

Ejemplo 2

En este ejemplo se lanzan varios clientes mediante heras, se puede observar que RMI es multihebrado, habilitando la gestión concurrente de peticiones de los clientes.

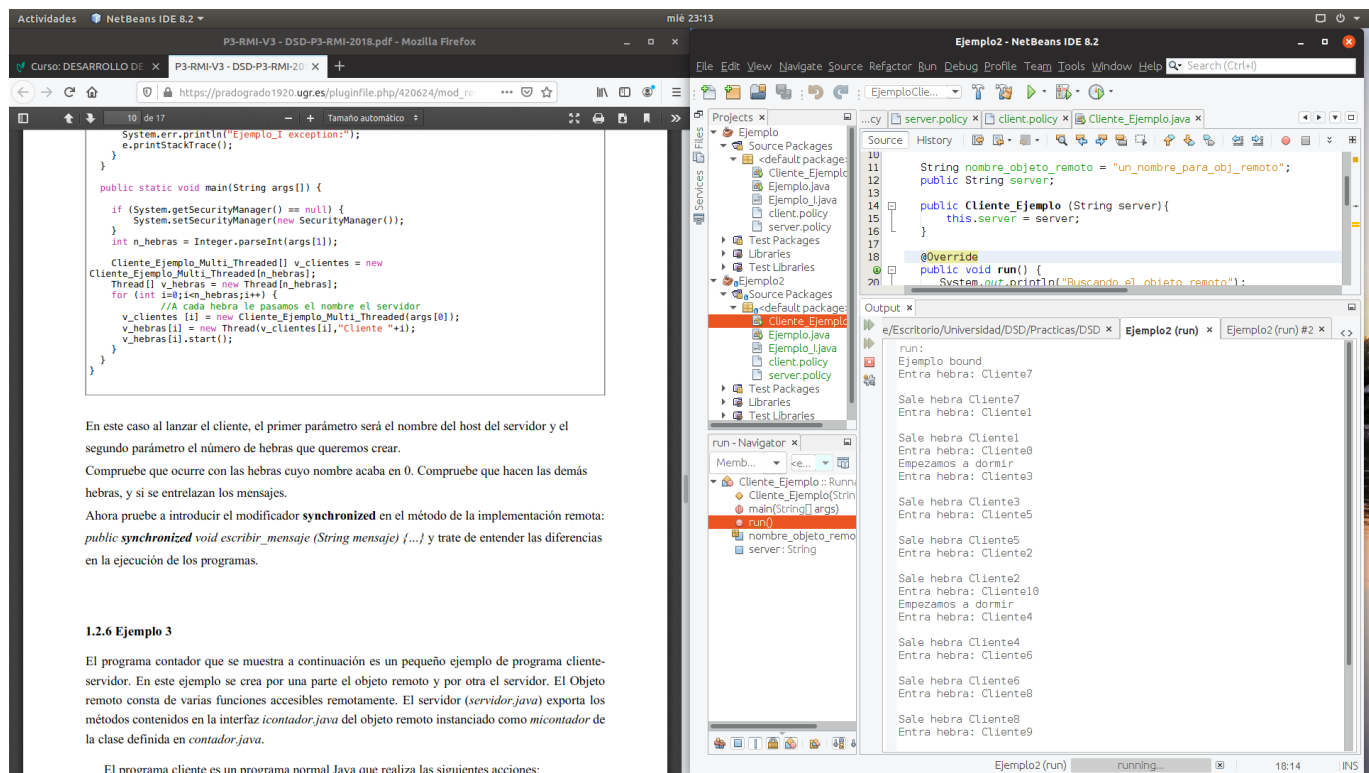
```
public class Cliente_Ejemplo implements Runnable{

String nombre_objeto_remoto = "un_nombre_para_obj_remoto";
public String server;

public Cliente_Ejemplo (String server){
    this.server = server;
}

@Override
public void run() {
    System.out.println("Buscando el objeto remoto");
    try{
        Registry registry = LocateRegistry.getRegistry(server);
        Ejemplo_I instancia_local = (Ejemplo_I)
registry.lookup(nombre_objeto_remoto);
        System.out.println("Invocando el objeto remoto");
        instancia_local.escribir_mensaje(Thread.currentThread().getName());
    } catch (Exception e){
        System.err.println("Ejemplo_I exception:");
        e.printStackTrace();
    }
}
```

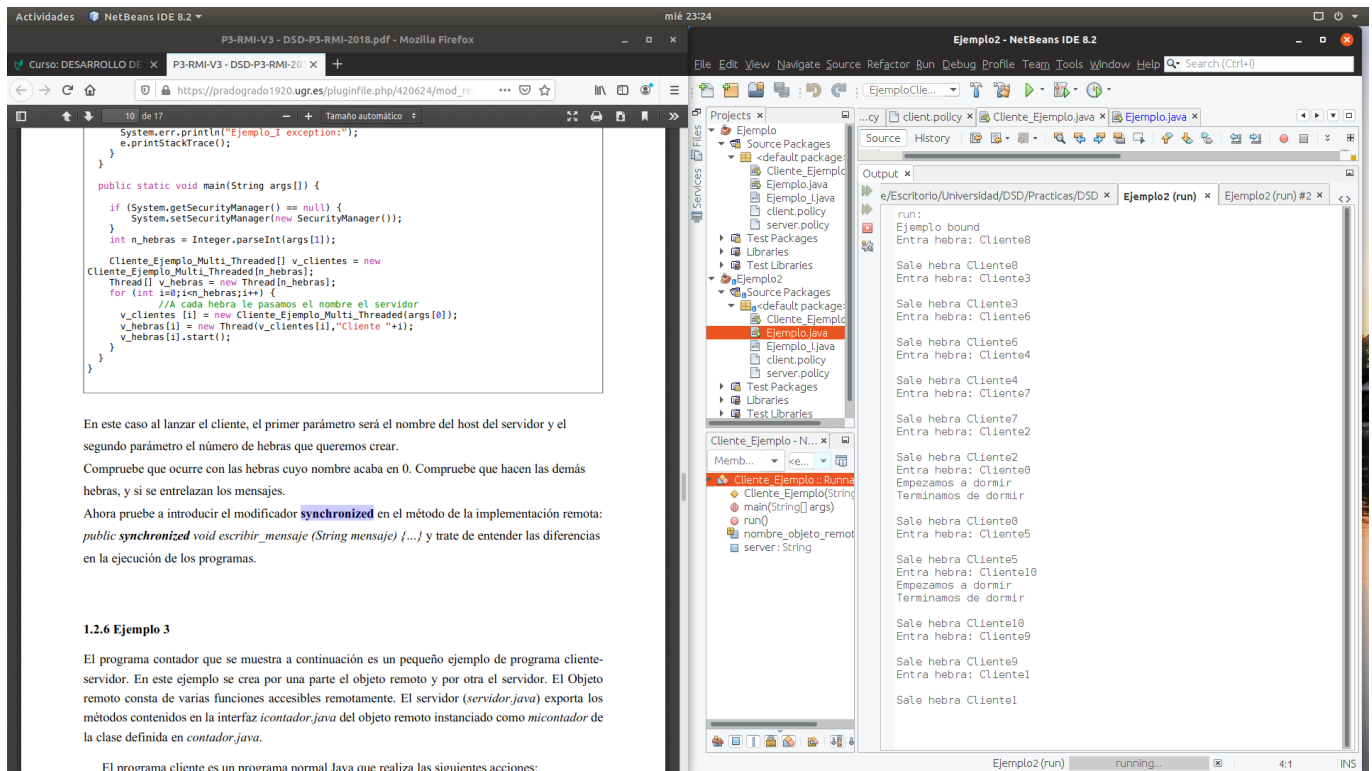
Este ejemplo lo realicé desde el inicio en netbeans usando las configuraciones para compilar tanto el servidor, como el cliente.



Como se puede observar cuando entra alguna hebra(Cliente) que termina con 0, se duerme y esto provoca que otras hebras entren sin esperar a que la hebra acaba en cero salga, entrelazándose los mensajes y las entradas.

```
public void escribir_mensaje(String mensaje) throws RemoteException {
    System.out.println("Entra hebra: "+mensaje);
    if (mensaje.endsWith("0")){
        try{
            System.out.println("Empezamos a dormir");
            Thread.sleep(5000);
            System.out.println("Terminamos de dormir");
        }
    }
}
```

Para solucionarlo, hace falta declarar el método como **synchronized**, provocando así que cuando una hebra entre, el resto tengan que esperar a que esa hebra salga, eliminando el entrelazamiento de los mensajes. [más información synchronized](#)



Ejemplo 3

En este ejemplo se crea un objeto remoto (contador.java que implementa icontador.java) en el servidor.java.

```
public class servidor {
    public static void main(String [] args){
        if (System.getSecurityManager() == null){
            System.setSecurityManager(new SecurityManager());
        }
        try {
            Registry reg = LocateRegistry.createRegistry(1099);
            contador micontador = new contador();
            Naming.rebind("micontador",micontador);

            ...

        }
    }

    //objeto contador
    public class contador extends UnicastRemoteObject implements icontador{
        private int suma;

        @Override
        public int sumar() throws RemoteException {
            return suma;
        }

        ...

    }
}
```

El cliente por su parte pone un valor inicial al contador del servidor e invoca al método incrementar del contador 1000 veces y por último muestra el valor de contador y el tiempo de respuesta que le ha llevado calcularlo.

```
public class cliente {
public static void main(String[] args){
    if (System.getSecurityManager() == null){
        System.setSecurityManager(new SecurityManager());
    }
    try {
        Registry mireg = LocateRegistry.getRegistry("127.0.0.1",1099);
        icontador micontador = (icontador)mireg.lookup("micontador");
        System.out.println("Poniendo contador a 0");
        micontador.sumar(0);

        long horacomienzo = System.currentTimeMillis();

        System.out.println("Incrementando...");
        for(int i = 0 ; i < 1000 ; i++){
            micontador.incrementar();
        }

        long horafin = System.currentTimeMillis();
        System.out.println("Media de las RMI realizadas = " + ((horafin -
horacomienzo)/1000f) + "mseg" );
        System.out.println("RMI realizadas = " + micontador.sumar());
    }catch (NotBoundException | RemoteException e){
        System.err.println("Exception del sistema: " + e);
    }
}
```

En este ejemplo no es necesario llamar a rmiregistry con el terminal, ya que se encarga el servidor de hacerlo (imágenes abajo). Tuve que crear también una compilación para el servidor y el cliente y en este caso, el cliente no tiene ningún parametro.

Source Refactor Run Debug Profile Team Tools Window Help

EjemploClie...

contador.java x servidor.java x cliente.java x

Source History

```
12 11 System.getSecurityManager();
13     System.setSecurityManager(new Sec
14 }
15 try {
16     Registry mireg = LocateRegistry.g
17     icontador micontador = (icontador
18     System.out.println("Poniendo conti
19     micontador.sumar(0);
20
21     long horacomienzo = System.curren
22
23     System.out.println("Incrementando
24     for(int i = 0 ; i < 1000 ; i++){
25         micontador.incrementar();
26     }
27
28     long horafin = System.currentTime
29     System.out.println("Media de las l
30     System.out.println("RMI realizada:
31 }catch (NotBoundException | RemoteExc
32     System.err.println("Exception del
33 }
34 System.exit(0);
35 }
36 }
37
```

Output x

Ejemplo3 (run) x Ejemplo3 (run) #2 x

run:
Poniendo contador a 0
Incrementando...
Media de las RMI realizadas = 0.136mseg
RMI realizadas = 1000
BUILD SUCCESSFUL (total time: 0 seconds)

EjemploClie...

contador.java x servidor.java x cliente.java x

Source History

```
1 import java.net.MalformedURLException;
2 import java.rmi.*;
3 import java.rmi.registry.LocateRegistry;
4 import java.rmi.registry.Registry;
5 import java.rmi.server.*;
6
7 /**
8  *
9  * @author juse
10  */
11 //en este ejemplo ya se lanza rmiregistry y no es necesario lar
12 public class servidor {
13     public static void main(String [] args){
14         if (System.getSecurityManager() == null){
15             System.setSecurityManager(new SecurityManager());
16         }
17         try {
18             Registry reg = LocateRegistry.createRegistry(1099);
19             contador micontador = new contador();
20             Naming.rebind("micontador",micontador);
21
22             System.out.println("Sevidor RemoteException | Malfo
23         } catch ( RemoteException | MalformedURLException e){
24             System.out.println("Exception: " + e.getMessage());
25         }
26     }
27 }
28
29
```

Output x

Ejemplo3 (run) x Ejemplo3 (run) #2 x

run:
Sevidor RemoteException | MalformedURLExceptionor preparada