

Memoria de la práctica 2

José Antonio Álvarez Ocete
Miguel Lentisco Ballesteros

7 de noviembre de 2017

1. Introducción y descripción del problema.

Para el ejercicio 5 hemos diseñado una arquitectura cliente/servidor P2P descentralizada para implementar un tres en raya online. Para ello hemos decidido que los clientes se conectarán al servidor mediante UDP para evitar sobrecargar al mismo.

Una vez hayan llegado al menos dos clientes al servidor, este los empareja y les envía la información necesaria para que se comuniquen mediante TCP (esto es, la dirección y el puerto). El servidor se olvida entonces de estos dos clientes, de ahí la descentralización de la arquitectura, y procede a esperar a que lleguen más.

Por otra parte los clientes se conectarán mediante TCP porque es orientado a conexión y evita errores, ambas características muy necesarias para un juego en línea. Tras terminar la partida tendrán que volver a ponerse en contacto con el servidor si quieren volver a jugar.

2. Diagramas

Hemos hecho tres diagramas de estados. El primero para el servidor, invariante durante toda su ejecución (mientras esté activo). El segundo para el cliente durante su conexión con el servidor. Una vez reciba la información del otro jugador pasará al tercer diagrama, donde vemos el desarrollo de la partida en si.

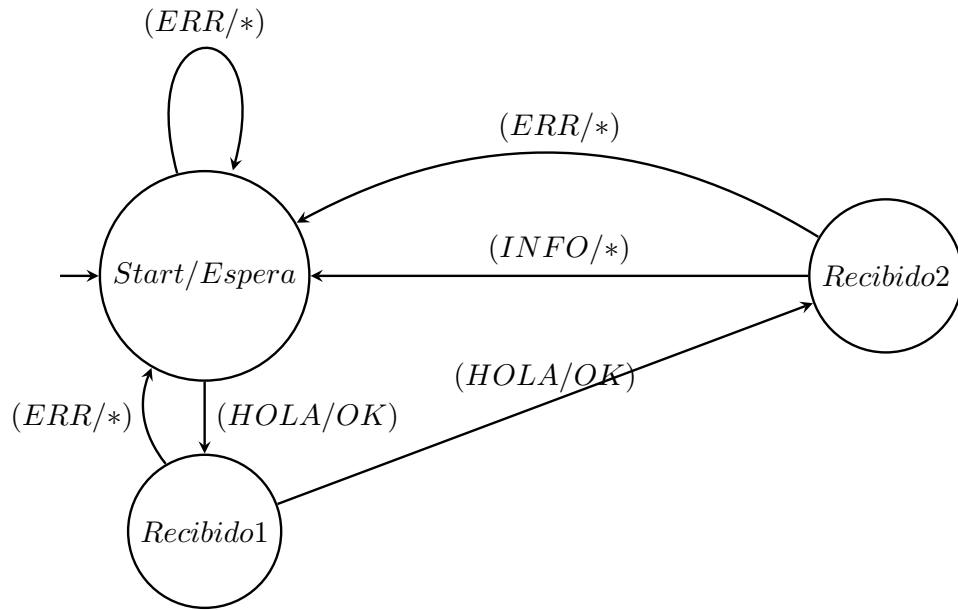


Figura 1: Diagrama de estados del servidor.

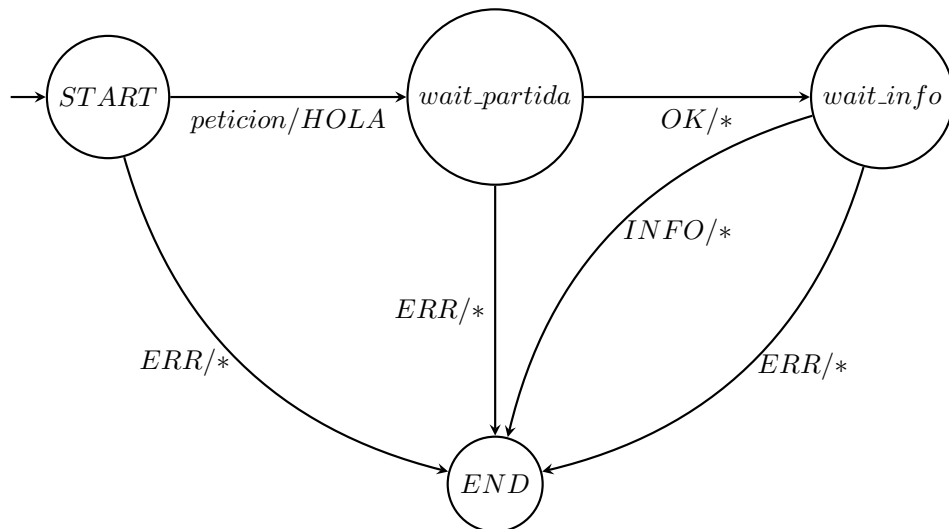


Figura 2: Diagrama de estados del cliente durante la conexión.

Código	Cuerpo	Descripción
000	ERR	Error - vuelve al inicio
001	OK	El servidor responde a la petición del cliente.
002	INFO	El servidor envía la información de un cliente a otro
003	HOLA	Recibe la petición por parte del cliente.
004	GO	Acaba el turno, le toca al cliente que recibe
005	FIN	Fin de partida

Tabla 1: Mensajes del protocolo.

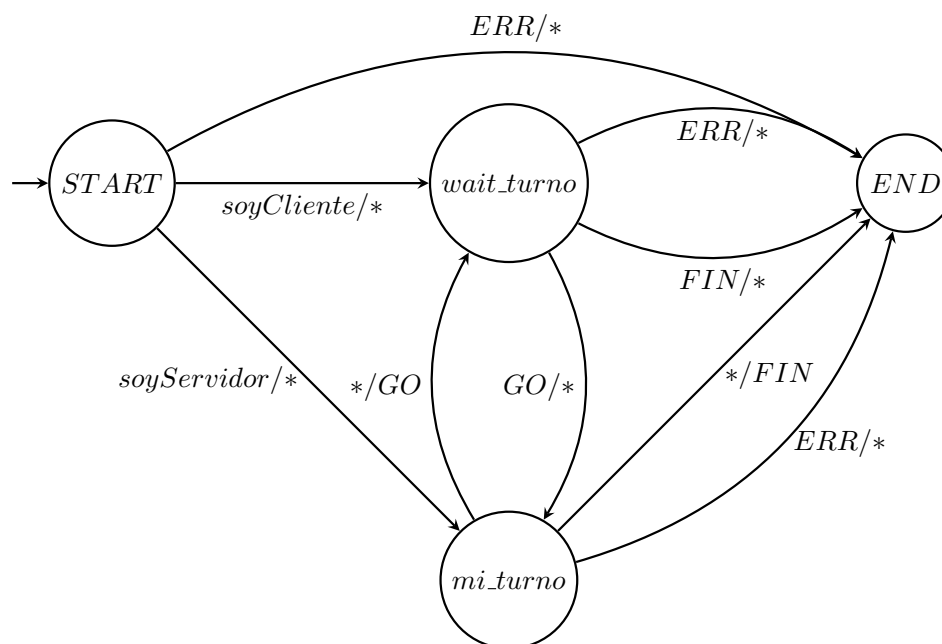


Figura 3: Diagrama de estados del cliente durante la partida.

3. Mensajes

Los mensajes utilizados son los del **Tabla 1**. Debido a un problema de formato está al principio de la página.

4. Evaluación

Por último comprobemos que el protocolo funciona de la forma deseada. Para ejecutarlo utilizaremos NetBeans. Primero pondremos en marcha el servidor: **ServidorConcurrenteUDP**. Tras ello enviaremos a dos clientes ejecutando **J1Conecta3** y **J2Conecta3**. Estas son las salidas obtenidas:

```
ant -f C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3 -Djavac.includes=conecta3/ServidorConcurrenteUDP
init:
deps-jar:
Created dir: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build
Updating property file: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\build-jar.properties
Created dir: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\classes
Created dir: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\empty
Created dir: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\classes
Note: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\src\conecta3\ServidorConcurrenteUDP.java u
Note: Recompile with -Xlint:unchecked for details.
compile-single:
run-single:
Se ha recibido un paquete
Se ha saludado al usuario de puerto: 8888
Se ha recibido un paquete
Se ha saludado al usuario de puerto: 8889
Hay dos en cola
Adios, jugadores en la cola
```

Figura 4: Salida de **ServidorConcurrenteUDP**.

```
ant -f C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3 -Djavac.includes=conecta3/J2Conecta3
init:
Deleting: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\build-jar.properties
deps-jar:
Updating property file: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\build-jar.properties
Compiling 1 source file to C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\classes
compile-single:
run-single:
Conectandose al servidor
Esperando a otro jugador
Dirección: 127.0.0.1
Puerto: 8888
Jugador encontrado, vas segundo.
Conexión establecida
Turno número: 1
Le toca a las: 0
-----
| | | |
-----
| | |X|
-----
| | | |
-----
Introduce fila y columna:
1 1
Turno número: 2
Le toca a las: X
-----
| | | |
-----
| |O|X|
-----
| | | |
-----
```

Figura 5: Salida de **J2Conecta3**.

```
ant -f C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3 -Djavac.includes=conecta3/J1Conecta3
init:
Deleting: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\build-jar.properties
deps-jar:
Updating property file: C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\build-jar.properties
Compiling 1 source file to C:\Users\Ocete\Desktop\Estudios\FR\Conecta3_v08\Conecta3\build\classes
compile-single:
run-single:
Conectandose al servidor
Esperando a otro jugador
Dirección: 127.0.0.1
Puerto: 8889
Jugador encontrado, empiezas tú.
Conexión establecida
Turno número: 0
Le toca a las: X
-----
| | |
| | |
| | |
-----
Introduce fila y columna:
1 2
Turno número: 1
Le toca a las: O
-----
| | | |
| | |X|
| | |
-----
Turno número: 2
Le toca a las: X
-----
| | | |
| |O|X|
| | |
-----
Introduce fila y columna:
```

Figura 6: Salida de **J1Conecta3**.