

## Ejercicio 5: Definición e implantación de un protocolo de aplicación.

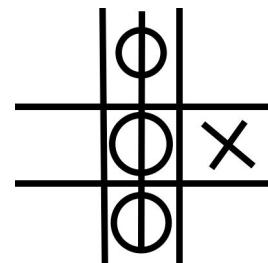
Sergio Fernández Vela  
Pablo Jiménez Jiménez

### Índice de contenidos:

<b>Descripción de la aplicación</b>	<b>1</b>
<b>Diagrama de estados</b>	<b>2</b>
<b>Mensajes que intervienen</b>	<b>2</b>
<b>El juego en acción</b>	<b>3</b>

### Descripción de la aplicación

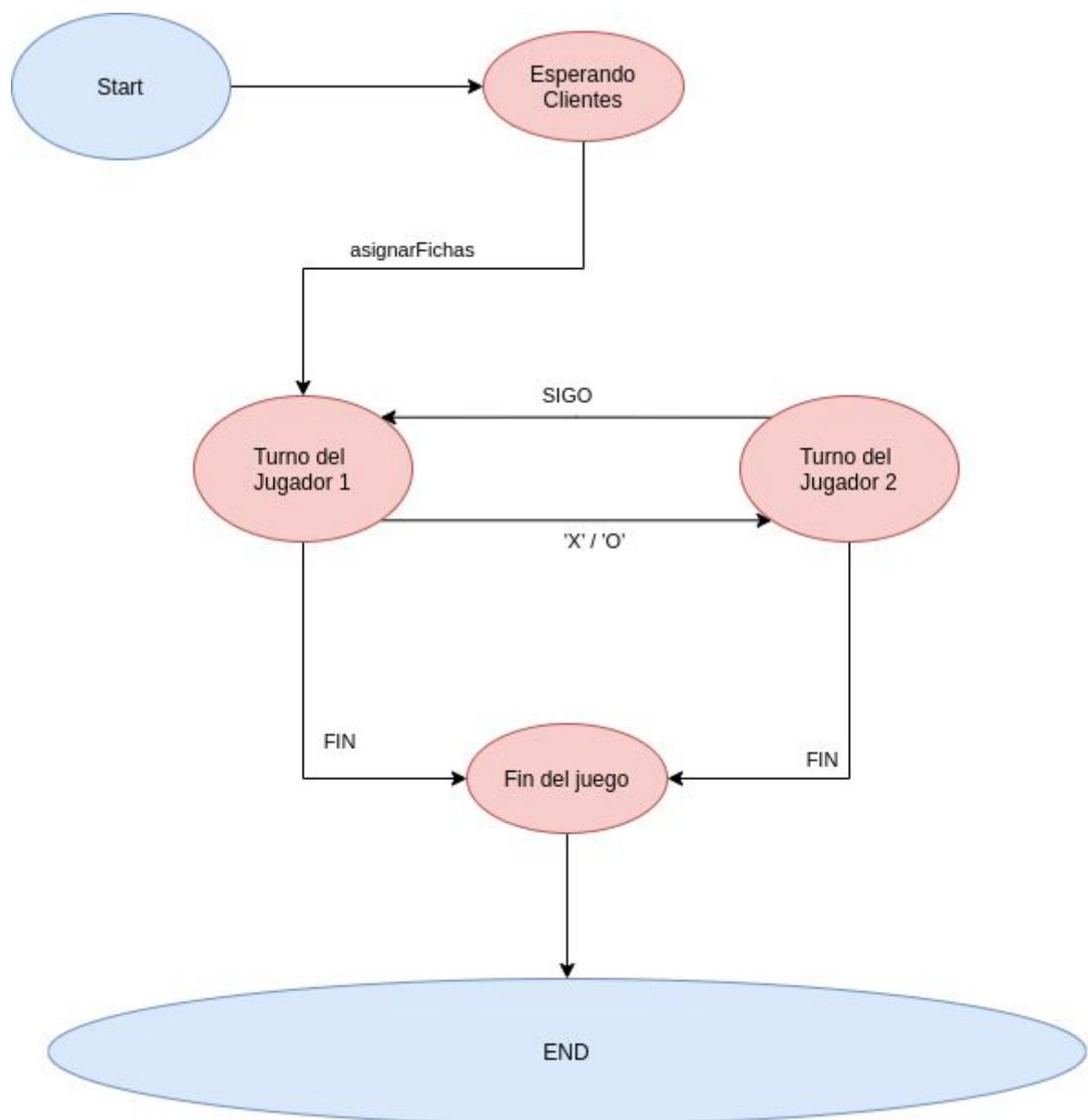
Hemos desarrollado el tradicional juego tres en raya (tic toe) para el que son necesarios dos jugadores. Como bien es sabido por todos el objetivo es conseguir tener 3 fichas alineadas vertical, horizontal o diagonalmente.



El proyecto implementa la arquitectura Cliente / Servidor. El servidor lleva toda la lógica del juego, encargándose de repartir los turnos, determinar las fichas, determinar el ganador... Mientras que los clientes (en este caso por la idiosincrasia del juego es obligatorio 2 instancias) se encargan de comunicarle al jugador lo que le llega del servidor y mandarle al servidor la jugada de cada jugador. El servidor se ejecuta concurrentemente por cada cliente.

El cliente y el servidor se conectan utilizando un protocolo de transmisión TCP, hemos elegido este en lugar de UDP debido a que es necesario que los datos lleguen íntegros, TCP dispone de control de flujo y de congestión que nos garantiza el funcionamiento íntegro.

## Diagrama de estados



## Mensajes que intervienen

### Servidor → Cliente

Código	Cuerpo	Descripción
ASIGNAR_FICHA	'X'   'O'	Manda la ficha correspondiente a cada jugador.
PRINT_TAB	tablero	Se corresponde con el estado del tablero en ese momento.
MSM	"introduce ficha: "	Mensaje informativo para el jugador que sea su turno.
FIN_JUEGO	FIN	Indica que el juego ha terminado y ha ganado el último jugador que tenía el turno.
NO_FIN_JUEGO	SIGUE	Indica que aún no hay un ganador y el juego continúa.

### Cliente → Servidor

Código	Cuerpo	Descripción
JUGADA	"13"	Devuelve al servidor la posición donde el jugador quiere introducir su ficha, en este caso fila ← 1 y columna ← 3

## El juego en acción

Vamos a reflejar la evolución de una partida.

```
pablo@pablo:~/projects/FR-ugr/Ejer5$ java Cliente
Tu ficha: X
|_|_|_
|_|_|_
|_|_|_
Introduce tu jugada (1-3,1-3):
|
pablo@pablo:~/projects/FR-ugr/Ejer5$ java Cliente
Tu ficha: 0
|_|_|_
|_|_|_
|_|_|_
Introduce tu jugada (1-3,1-3):
|
pablo@pablo:~/projects/FR-ugr/Ejer5$ java ServidorConcurrente
cliente conectado
cliente conectado
```

```
pablo@pablo:~/projects/FR-ugr/Ejer5$ java Cliente
Tu ficha: X
|_|_|_
|_|_|_
|_|_|_
Introduce tu jugada (1-3,1-3):
22
|0|_|_|
|_|X|_|
|_|_|_|
Introduce tu jugada (1-3,1-3):
|
pablo@pablo:~/projects/FR-ugr/Ejer5$ java Cliente
Tu ficha: 0
|_|_|_
|_|_|_
|_|_|_
Introduce tu jugada (1-3,1-3):
11
|
pablo@pablo:~/projects/FR-ugr/Ejer5$ java ServidorConcurrente
cliente conectado
cliente conectado
```

The screenshot shows a terminal window with two tabs, each representing a client session. Both clients are playing a game of tic-tac-toe.

**Client 1 (Left Tab):**

- Current board state:  
|\_|X|\_|  
|X|\_|X|  
|\_|O|\_|
- Last move: 21
- Next move prompt: Introduce tu jugada (1-3,1-3):

**Client 2 (Right Tab):**

- Current board state:  
|\_|X|\_|  
|X|\_|X|  
|\_|\_|O|
- Last move: 32
- Next move prompt: Introduce tu jugada (1-3,1-3):

**Terminal Bottom:**

- Java command: java ServidorConcurrente
- Output:
  - Cliente conectado
  - Cliente conectado
  - El juego ha finalizado