

Ejercicio 5 - Definición e implementación de un protocolo de aplicación.

Índice:

1. Descripción de la aplicación, funcionalidad y actores que intervienen
2. Diagrama de estados del servidor
3. Mensajes que intervienen
4. Evaluación de la aplicación

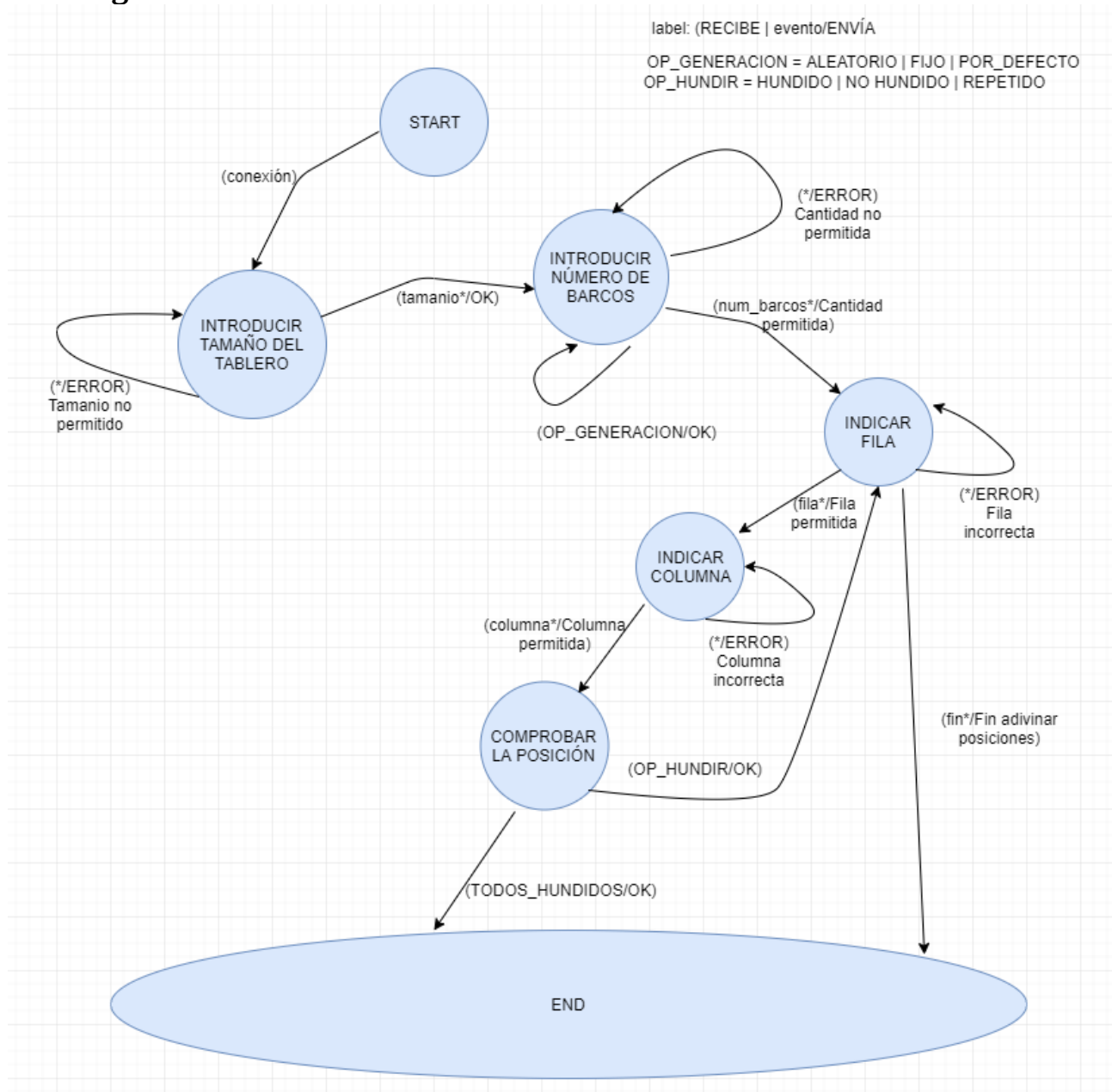
5.1 Descripción de la aplicación, funcionalidad y actores que intervienen

El objetivo de la aplicación es crear un servicio/juego de adivinar posiciones en un tablero. Las posiciones a adivinar son interpretadas como barcos y el cliente es el encargado de adivinar la posición de estos barcos en el tablero y “hundirlos”.

Respecto a su funcionalidad; la aplicación comienza solicitándole al cliente el tamaño (superior y diferente a 0) que desea que posea su tablero, este será una matriz cuadrada, por lo que solo necesita un parámetro para determinar sus dimensiones: “ $n \times n$ ”, siendo n el tamaño introducido por el cliente. Tras ello el servidor genera el tablero y solicita al cliente el número de barcos que desea introducir en él, cuyo valor no podrá ser 0 o inferior ni superior al número de huecos disponibles en el tablero. Se ofrecen tres formas de introducir el número de barcos: aleatoria, tomará un valor aleatorio adecuado al tablero; fijo, que le permitirá al cliente introducir el número de barcos, siempre que cumpla las restricciones; y por defecto, que establecerá como número de barcos la mitad del total que podrían insertarse en el tablero. Una vez tenemos el tablero y los barcos el cliente deberá introducir la fila y columna dónde cree que se encuentra un barco, cumpliendo que estos valores introducidos tengan sentido en nuestro tablero respecto a sus dimensiones. Podrá repetir este proceso hasta que hunda todos los barcos o decida finalizar el juego, para ello al indicar las filas deberá escribir “fin” en lugar de la fila donde desea “disparar”.

Por lo tanto en nuestra aplicación intervendrán un servidor que se encargará de procesar la información que le será introducida a través de un cliente que una vez a introduzca el tablero sobre el que jugará y el número de posiciones de barcos que tendrá que averiguar irá introduciendo filas y columna, posiciones dónde supondrá que se encuentran los “barcos” que deberá “hundir”.

5.2 Diagrama de estados del servidor



5.3 Mensajes que intervienen

Cliente:

Código	Cuerpo	Descripción
1000	Tamaño + tamaño	Mensaje que enviará el cliente para insertar el tamaño del tablero.

Código	Cuerpo	Descripción
2000	Tipo generación número de barcos + genera	Este mensaje será enviado por el cliente para indicar como desea que se genere el número de barcos que habrá en el tablero.

Código	Cuerpo	Descripción
2010	Generación aleatorio + genera	Indica que desea una generación aleatoria.

Código	Cuerpo	Descripción
2020	Generación fija + genera	El cliente indica que desea insertar el mismo el número de barcos.

Código	Cuerpo	Descripción
2030	Generación por defecto + genera	Indicar que desea una generación por defecto.

Código	Cuerpo	Descripción
2040	Numero de barcos + num_barcos	Mensaje enviado por el cliente para solicitar el número de barcos que al final se han insertado en el tablero.

Código	Cuerpo	Descripción
3000	Fila + fila	Mensaje que enviará el cliente para indicar una fila del tablero.

Código	Cuerpo	Descripción
3001	Finalizar juego + fin	Este mensaje será enviado por el cliente para finalizar de adivinar posiciones.

Código	Cuerpo	Descripción
3010	Columna + columna	Mensaje que enviará el cliente para indicar una fila del tablero.

Código	Cuerpo	Descripción
3020	HundirBarcos + hundir	Mensaje que enviará el cliente para solicitar que había en la posición que ha indicado.

Servidor:

Código	Cuerpo	Descripción
1001	"Error: Tamano no permitido"	Mensaje de error cuando el tamaño es menor o igual a 0.

Código	Cuerpo	Descripción
1002	"Bien, el tablero será de nxn"	Mensaje de confirmación para la petición de establecer el tamaño del tablero

Código	Cuerpo	Descripción
2021	"Error: Cantidad no permitida."	Mensaje de error cuando el número de barcos es menor o igual a 0 o mayor a los huecos disponibles en el tablero.

Código	Cuerpo	Descripción
2022	"Cantidad permitida."	Mensaje de confirmación para la petición de establecer el número de barcos.

Código	Cuerpo	Descripción
3002	"Fin de adivinar posiciones"	Mensaje tras la finalización de la aplicación por el cliente, sin haber hundido todos los "barcos".

Código	Cuerpo	Descripción
--------	--------	-------------

3003	"Continuemos"	El juego no ha finalizado y se desea seguir introduciendo posiciones.
------	---------------	---

Código	Cuerpo	Descripción
3004	"Error: Fila incorrecta"	Mensaje de error cuando la fila introducida no existe en el tablero.

Código	Cuerpo	Descripción
3005	"Fila valida"	Mensaje de confirmación para la petición de indicar una fila.

Código	Cuerpo	Descripción
3011	"Error: Columna incorrecta"	Mensaje de error cuando la columna introducida no existe en el tablero.

Código	Cuerpo	Descripción
3012	"Columna valida"	Mensaje de confirmación para la petición de indicar una columna.

Código	Cuerpo	Descripción
3021	"Has hundido Todos los barcos!!"	Mensaje que indica que todos los "barcos se han "hundido" fin de la aplicación.

Código	Cuerpo	Descripción
3022	"Aún quedan barcos!!"	Mensaje que indica la repetición del proceso de indicar una posición.

5.4 Evaluación de la aplicación

```

rafa@rafa-Z97P-D3:~/Escritorio/3año/Fundament de Redes/Practicas/Practica
2/ejercicios/Ejercicio5$ java BarcosClienteTCP
Bienvenido al simulador Hundir la flota
Prueba tu suerte intentando adivinar dónde estarán los barcos

Por favor ingrese el tamaño del tablero
0
Error: Tamaño no permitido

Por favor ingrese el tamaño del tablero
3
Bien, el tablero será de 3x3

Recibido:
---
---
---

```

La aplicación comienza pidiendo el tamaño al cliente, podemos ver que si insertamos un tamaño menor que o igual a 0 nos dará error y nos lo volverá a pedir. Y el servidor almacena ese tamaño:

```

rafa@rafa-Z97P-D3:~/Escritorio/3año/Fundament de Redes/Practicas/Practica
2/ejercicios/Ejercicio5$ java BarcosServidorIterativo
Servidor simulador de hundir la flota

Se ha establecido el tablero: 3x3
| - - - - |
| - - - - |
| - - - - |
| - - - - |

```

```

Cómo desea la generación del número de barcos:
Si la desea aleatoria escriba 'ale'
Si la desea fija escribe 'fija'
Si introduce otra palabra se hará por defecto
fija
Introduzca el número de barcos que desea:
10
Error: Cantidad no permitida.
Introduzca el número de barcos que desea:
3
Barcos creados: 3

```

A continuación se le solicita al cliente cómo quiere que se realice la generación del número de barcos. En este ejemplo el cliente solicita una creación fija, es decir el establece el número, pudiendo observar que si inserta un número mayor al número de huecos dará error. En el servidor se almacena el número de barcos y por medio de una función existente en su código llamada generarBarcosTablero, se colocan en posiciones aleatorias del tablero:

```

Numero de barcos generado: 3
De forma: fija
| - - - - |
| - - - - |
| * * * * |
| - - - - |

```

Finalmente nos encontramos en la parte en la que el cliente debe introducir las posiciones para adivinar dónde se encuentran los barcos:
Si da una fila no existente en el tablero dará fallo.

```
Introduzca la fila donde desea disparar:
4
Error: Fila incorrecta
Introduzca la fila donde desea disparar:
0
Fila valida

Introduzca la columna donde desea disparar:
0
Columna valida

Lo siento no has acertado el disparo

Aún quedan barcos!!

Recibido:
---
---
```

Esto se muestra en el cliente falla el “disparo”

```
Continuemos
Introduzca la fila donde desea disparar:
0
Fila valida

Introduzca la columna donde desea disparar:
2
Columna valida

Bingo!! Le has dado

Aún quedan barcos!!

Recibido:
--X
---
```

Cuando acertamos el “disparo” se muestra “X” en el tablero del cliente, indicando dónde había un barco.

```
Si desea finalizar el juego introduzca 'fin'
Si desea seguir introduzca cualquier otra cosa por teclado
fin

Fin de adivinar posiciones

Recibido:
---
---
```

Cuando el cliente decide dejar de adivinar posiciones deberá escribir “fin” y finalizará la aplicación.

```
Has hundido Todos los barcos!!

Recibido:
--X
X--
X--
```

Cuando el cliente acierta la posición de todos los barcos finaliza el programa.

```
| ?   -   X   |
| *   -   -   |
| *   -   -   |
|     -   -   |
|     |     |
```

El servidor mientras va mostrando el tablero indicando con ‘*’ los barcos aún no “hundidos” con un ‘?’ las posiciones dónde dispara el cliente y ni hay nada, y con una ‘X’ las posiciones que el cliente ha acertado, hundiendo el barco situado ahí.