



UNIVERSIDAD DEL BÍO-BÍO

# INFORME PROYECTO SISTEMAS OPERATIVOS

Sockets

Integrantes: Carlos Fernández

Felipe Hernández

Profesor: Carlos Faúndez

Fecha: 25 de marzo 2020

## 0. Introducción

La batalla naval, del nombre en inglés battleship, es un juego tradicional de estrategia y algo de suerte, que involucra a dos participantes. Los jugadores manejan un tablero de océano y un tablero de tiro; cada uno divididos en casillas. Cada tablero representa una zona diferente del mar abierto: la propia y la contraria. En el primer tablero, el jugador coloca sus barcos y registra los «tiros» del oponente; en el otro, se registran los tiros propios contra el otro jugador, diferenciando los impactos y los que dan al agua. Al tiempo, se deduce la posición de los barcos del contrincante.

Para este proyecto recrearemos este clásico juego usando el lenguaje de programación C y sockets para lograr la comunicación entre los jugadores (clientes) con el tablero (servidor)

## 1. Principales funciones utilizadas.

- ☐ Sockets: Para la comunicación entre los 2 clientes con el servidor.
- ☐ Función randómica: Para asignar aleatoriamente los barcos en el tablero.
- ☐ Matrices: Para la creación del tablero.
- ☐ Atoi (Librería stdlib.h): Función para convertir un string a un entero.

## 2. Preguntas.

- ☐ ¿Por qué es necesario emplear el uso de semáforos en su sistema?

R: Los semáforos en teoría se usan para controlar la entrada de un sólo proceso a la zona crítica, en este caso los semáforos controlan los turnos de cada jugador. En nuestro caso no nos fue posible implementar semáforos ya que no siempre funcionaban correctamente, y ante la duda decidimos no dejarlo. Debido a esto tampoco pudimos implementar los fork por cada cliente ya que no habría un control de acceso a la zona crítica del código.

- ☐ ¿Cuál es la diferencia entre usar tuberías con nombre y socket? ¿Cuál recomendaría ustedes?

R: En las tuberías con nombre para que haya comunicación entre procesos se necesita un archivo, en este caso llamado "myfifo", mientras que en el socket, él se conecta a una dirección IP que el servidor levanta cuando este se ejecuta. En el caso de los sockets basta con hacer sólo un cliente, mientras que en tuberías con nombre se necesita un cliente por cada jugador.

En nuestro caso recomendamos usar sockets porque tanto su implementación y su uso nos pareció más sencillo y se obtienen los mismos resultados.

- ☐ ¿Fue necesario el uso de señales? Si es que si, ¿En qué caso fue utilizado? ¿Qué señal o señales fueron utilizados?

R: En nuestro caso no fue necesario el uso de señales.

### 3. Conclusión

Si bien logramos crear un juego perfectamente jugable y a la vez entretenido, aún así no pudimos implementar todo lo requerido. Faltando principalmente la implementación de semáforos, que en nuestro caso no echamos de menos, ya que el resultado fue el mismo, un juego donde se respetan los turnos de cada jugador y la experiencia de juego es gratificante.