

Identificación del problema y análisis de requerimientos Tarea Integradora I.

Nombres:

Cristian Camilo Herrera Ramírez – A00402478

Keren Valentina Rodríguez Rojas – A00405496

Caso de Estudio:

Cliente	Facultad de Negocios y Economía de la Universidad Icesi, Los estudiantes de Economía y Negocios internacionales
Usuario	Los estudiantes de Economía y Negocios internacionales
Requerimientos funcionales	RF1: Permitir al jugador humano ubicar sus barcos RF2: Permitir a la maquina ubicar sus barcos. RF3: Se debe realizar turno a turno RF4: Determinar Ganador del juego
Contexto del problema	Desde las directivas de la Facultad de Negocios y Economía de la Universidad Icesi, se busca desarrollar un juego de Batalla Naval en 1D con el objetivo de poner en práctica los conocimientos adquiridos sobre Teoría de Juegos. El propósito es que los estudiantes de Economía exploren diversos escenarios de competencia, tomando decisiones estratégicas y evaluando cómo la aleatoriedad influye en los resultados. Este software simulará una versión simplificada del juego, en la que participarán dos jugadores (uno humano y uno maquina). Los jugadores deberán ubicar sus barcos y jugarán de manera alternada. El sistema deberá determinar al ganador y finalizar el juego al cumplir con las condiciones de victoria. Además, contará con una interfaz intuitiva, ejecutándose en menos de un segundo por turno y permitiendo realizar varias iteraciones en una misma ejecución del programa. El desarrollo debe llevarse a cabo en Java.
Requerimientos no funcionales	1. La interfaz de usuario del juego debe ser clara y comprensible. 2. El programa debe ejecutarse en menos de un segundo por turno. 3. Permitir realizar varias iteraciones del juego en la misma ejecución del programa.
Requerimientos de proceso	1. El juego debe desarrollarse en Java

Identificador y nombre	RF1: Permitir al jugador humano ubicar sus barcos		
Resumen	El sistema debe permitir al jugador humano ubicar los barcos, los barcos no deben solaparse y deben ajustarse a las reglas de orientación y las coordenadas van a ser X, Y		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
	Posición de la lancha	int	<i>Se deben ingresar valores que se encuentre entre el 1-10 no puede salirse de ese rango.</i>
	coordenada 1 del barco medico	int	<i>Se deben ingresar valores que se encuentre entre el 1-10 no puede salirse de ese rango.</i>
	Coordenada 2 del barco médico	int	<i>Se deben ingresar valores que se encuentre entre el 1-10 no puede salirse de ese rango.</i>
	Coordenada 1 del barco de munición	int	<i>Se deben ingresar valores que se encuentre entre el 1-10 no puede salirse de ese rango.</i>
	Coordenada 2 del barco de munición	int	<i>Se deben ingresar valores que se encuentre entre el 1-10 no puede salirse de ese rango.</i>
Resultado o Postcondición	El sistema ubica la lancha, el barco médico y el barco de munición en el tablero		
Salidas	Nombre salida	Tipo de dato	Formato
	Posición de la lancha	string	La lancha ha sido ubicada (número)
	Posición del barco médico	string	El barco médico ha sido ubicado en la posición (número)
	Posición del barco de munición	string	El barco de munición ha sido ubicado en la posición (número)
	Disposición línea de mar	string	La disposición de tus barcos ha quedado así: los números donde están posicionados los barcos

Identificador y nombre	RF2: Permitir a la maquina ubicar sus barcos		
Resumen	El sistema debe permitir a la maquina ubicar aleatoriamente los barcos, los barcos no deben solaparse y deben ajustarse a las reglas de orientación.		
Resultado o Postcondición	Los barcos son posicionados correctamente teniendo en cuenta las reglas de orientación y el no solapamiento.		
Salidas	Nombre salida	Tipo de dato	Formato
	Mensaje de confirmación	String	"La disposición de los barcos ha sido completada, empezará el juego"

Identificador y nombre	RF3: Se debe realizar turno a turno.		
Resumen	El sistema debe pedir por consola al jugador humano el lugar donde desea atacar y modificar la línea de mar de la maquina en base a esta entrada, se debe verificar que la condenada es válida. Después, la maquina aleatoria realizará una jugada y modificará las líneas de mar del jugador humano según las coordenadas.		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
	Lugar daño Jugador	Int	El punto atacar debe estar entre 1-10
Resultado o Postcondición	Se le informará al jugador si le ha dado a un barco enemigo o lo contrario		
Salidas	Nombre salida	Tipo de dato	Formato
	Confirmación ataque Jugador	String	"Atacarás el punto (Lugar seleccionado)"
	Confirmación Daño Jugador	String	"¡Le has dado a un objetivo enemigo!"
	Línea mar rival	String	"Disposición de la Línea mar rival: 0 0 2 0 1....."
	Confirmación Daño Maquina	String	"¡Te han dado!"
	Desacierto de Daño	String1	"No ha pasado nada aquí"
	Línea de mar del Jugador	String	"Disposición de la línea mar jugador: 0 0 0 3..."
	Daño Barco Medico maquina	String	¡Han hundido tu barco medico!
	Daño Barco Munición maquina	String	¡Han hundido tu barco munición!
	Daño Lancha maquina	String	¡Han hundido tu Lancha!
	Daño Barco Medico Jugador	String	¡Has hundido tu barco medico!
	Daño Barco Munición Jugador	String	¡Has hundido tu barco munición!
	Daño Lancha Jugador	String	¡Has hundido tu Lancha!

Identificador y nombre	RF4: Determinar el ganador del juego		
Resumen	Se determinará que el juego se ganó cuando en cualquier línea solo hay 0 (agua), 2 (parte tocada del barco enemigo) y 3 (barco totalmente hundido), obteniendo dos salidas una es que el jugador gane o que el jugador maquina gane.		
Entradas	Nombre entrada	Tipo de dato	Condición valores válidos
Resultado o Postcondición	Muestra el ganador del juego		
Salidas	Nombre salida	Tipo de dato	Formato
	Gano el jugador humano	string	El ganador es el jugador humano
	Gano el jugador maquina	string	El ganador es el jugador maquina