

## UD 7 PROGRAMACIÓN

### Trabajo 2: Hundir la flota

Programación  
CFGS DAW

Autores:

Lionel Tarazón [lionel.tarazon@ceedcv.es](mailto:lionel.tarazon@ceedcv.es)

Javier Valero [franciscojavier.valero@ceedcv.es](mailto:franciscojavier.valero@ceedcv.es)

Esta es una adaptación del trabajo original  
creado por Carlos Cacho y Raquel Torres

2019/2020

# HUNDIR LA FLOTA

## *El Videojuego*



## ÍNDICE

<b>1. OBJETIVO</b>	<b>4</b>
<b>2. DESCRIPCIÓN DEL TRABAJO</b>	<b>4</b>
<b>4. NÚMEROS ALEATORIOS</b>	<b>7</b>
<b>5. USO DE FUNCIONES</b>	<b>7</b>
<b>6. CONSEJOS Y CONSIDERACIONES</b>	<b>7</b>
<b>7. DESARROLLO POR ETAPAS Y CRITERIOS DE CALIFICACIÓN</b>	<b>8</b>
<b>8. ENTREGA</b>	<b>8</b>

## 1. OBJETIVO

Realizar un programa Java que permita jugar a una versión simplificada de **Hundir la flota**.



Imagen extraída de todocoleccion.net

## 2. DESCRIPCIÓN DEL TRABAJO

*En el juego original, cada jugador maneja dos tableros divididos en casillas. Cada tablero representa una zona diferente del mar abierto: la propia y la contraria. En uno de los tableros el jugador coloca sus barcos y registra los «tiros» del oponente; en el otro, se registran los tiros propios, al tiempo que se deduce la posición de los barcos del contrincante.*

*Al comenzar, cada jugador posiciona sus barcos en el primer tablero, de forma secreta, invisible al oponente.*

*Una vez todos los barcos han sido posicionados, se inicia una serie de rondas. En cada ronda, cada jugador, en su turno, «dispara» hacia la flota de su oponente, indicando una posición (las coordenadas de una casilla), la que registra en el segundo tablero. Si esa posición es ocupada por parte de un barco contrario, el oponente cantará ¡Tocado! si todavía quedan partes del barco (casillas) sin dañar, o ¡Hundido! si con ese disparo el barco ha quedado totalmente destruido. Si la posición indicada no corresponde a ninguna parte de un barco, cantará ¡Agua!.*

*Ganará el jugador que destruya primero todos los barcos de su oponente.*

Texto extraído de Wikipedia: [https://es.wikipedia.org/wiki/Batalla\\_naual\\_\(juego\)](https://es.wikipedia.org/wiki/Batalla_naual_(juego))

En la **versión simplificada** que debéis programar solo habrá **un jugador humano (usuario)** cuyo objetivo será **hundir todos los barcos del ordenador** en un número determinado de intentos. El ordenador tendrá un tablero con barcos y el usuario disparará para intentar hundirlos.

**Importante:** A continuación se describe la funcionalidad completa del videojuego pero se recomienda desarrollarlo por etapas como se describe en el apartado 7, empezando con el funcionamiento básico y luego añadiendo los demás aspectos poco a poco.

Se jugará sobre un **tablero de 10x10 posiciones** que se enumerarán del **0 al 9** en las **columnas** y de la **A a la J** en las **filas**.

Los **tipos de barcos** son:

- **Lancha (L):** ocupa una casilla del tablero.
- **Buque (B):** ocupa 3 casillas horizontales consecutivas del tablero.
- **Acorazado (Z):** ocupa 4 casillas horizontales consecutivas del tablero.
- **Portaaviones (P):** ocupa 5 casillas verticales consecutivas del tablero.

El videojuego tendrá cuatro **niveles de dificultad**:

- **Fácil:** El ordenador colocará 10 barcos (5 lanchas, 3 buques, 1 acorazado y 1 portaaviones) en el tablero y el jugador tendrá 50 intentos para hundirlos todos.
- **Medio:** El ordenador colocará 5 barcos (2 lanchas, 1 buque, 1 acorazado y 1 portaaviones) en el tablero y el jugador tendrá 30 intentos para hundirlos todos.
- **Difícil:** El ordenador colocará 2 barcos (1 lancha y 1 buque) en el tablero y el jugador tendrá 10 intentos para hundirlos todos.
- **Personalizado:** Se le preguntará al usuario el tamaño del tablero, el número de barcos de cada tipo y el número de intentos.

Al inicio de cada juego se le preguntará al jugador en qué nivel de dificultad quiere jugar y una vez seleccionado el ordenador colocará aleatoriamente los barcos en el tablero (al principio este estará oculto al usuario).

Los barcos del tablero podrán tocarse pero **hay que asegurarse de que no se solapen**.

La nomenclatura del **tablero** será la siguiente:

- Lancha: L
- Buque: B
- Acorazado: Z
- Portaaviones: P
- Agua: A
- Posición no disparada: -
- Posición tocada o hundida: X

Un ejemplo de **tablero del ordenador** (oculto) en el nivel fácil podría ser este:

	0	1	2	3	4	5	6	7	8	9
A	-	-	-	-	-	-	-	B	B	B
B	-	-	B	B	B	-	-	-	-	-
C	-	-	-	-	-	-	P	-	-	-
D	-	-	L	-	-	-	P	-	-	-
E	-	-	-	-	-	-	P	-	-	L
F	-	B	B	B	-	-	P	-	-	-
G	Z	Z	Z	Z	L	-	P	-	-	-
H	-	-	-	L	-	-	-	-	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	L	-	-	-

Al principio el **tablero visible al jugador** mostrará todas su posiciones a - (no disparado), así:

	0	1	2	3	4	5	6	7	8	9
A	-	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	-	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

Una vez el ordenador ha creado su tablero, se mostrará el tablero visible al jugador y se le preguntará a qué posición (fila y columna) quiere disparar. Si en esa posición hay una parte del barco, se mostrará el mensaje “Tocado”. En el caso de que no haya ningún barco en esa posición, se mostrará el mensaje “Agua”.

En ambos casos se mostrará el tablero visible al jugador actualizado con el disparo realizado. Hay que tener en cuenta que en la posición disparada solo se mostrará una A (si era agua) o X (si un barco ha sido tocado). No se deberá indicar el tipo de barco. Por lo tanto, el tablero visible al jugador solo contendrá casillas con -, A o X.

El usuario seguirá realizando disparos hasta que hunda todos los barcos, en cuyo caso le aparecerá el mensaje de “¡Has ganado!”, o bien, si el jugador no ha conseguido hundir todos los barcos en los intentos que tenía, con lo que le parecerá el mensaje de “¡Has perdido!”. En ambos casos se mostrará el tablero oculto del ordenador y el juego finalizará.

Durante el juego habrá que manejar la entradas erróneas (por ejemplo si el usuario introduce una coordenada de disparo incorrecta).

## 4. NÚMEROS ALEATORIOS

Tal y como se ha explicado, se utilizarán números aleatorios para la colocación de los barcos.

Recuerda que lo más sencillo es utilizar el método **Math.random()** que proporciona un número pseudoaleatorio tipo 'double' entre 0.0 y 1.0. Añadiendo algunas operaciones matemáticas podemos obtener un número entero aleatorio en el rango que nos interese:

Código Java	Número aleatorio generado
<code>Math.random()</code>	Tipo double entre 0.0 y 1.0
<code>(int) (Math.random() * (N+1))</code>	Tipo int entre 0 y N
<code>(int) (MIN + Math.random() * (MAX - MIN + 1))</code>	Tipo int entre MIN y MAX

## 5. USO DE FUNCIONES

Un aspecto importante de este trabajo son las funciones, que os permitirán crear un código más estructurado, menos repetitivo y sencillo de entender. Deberéis diseñar y crear las funciones que necesitéis para cada una de las características o funcionalidades del juego. Abajo tenéis la descripción de **algunas funciones recomendadas a implementar**:

- *crear\_tablero*. Crea el tablero del ordenador con todos los barcos necesarios.
- *insertar\_barco*. Inserta un barco en el tablero.
- *disparo*. Controla el disparo del usuario y actualiza el tablero.
- *mostrar\_tablero*. Muestra por pantalla el tablero al usuario.

Seguramente os sea de ayuda crear algunas funciones más. Por ejemplo, podrían ser útil crear *elegir\_dificultad*, *numero\_aleatorio*, *insertar\_lancha*, *comprobar\_???*, etc.

**Es obligatorio escribir arriba de cada función un comentario explicando qué hace.**

**Una buena práctica es que las funciones no tengan más de 30-40 líneas de código**, incluida la función main. No es obligatorio, es un consejo. Si una función tiene más líneas tal vez sea conveniente poner parte de ese código en una función.

## 6. CONSEJOS Y CONSIDERACIONES

**Antes de empezar** a programar lee bien el enunciado, subraya lo importante, hazte notas, escribe un borrador de las posibles funciones, etc. Intenta entender el problema y cómo realizarlo antes de empezar a escribir código.

**No intentes hacerlo todo de golpe**. Una estrategia clave en programación es dividir un problema grande en varios problemas pequeños y luego resolverlos uno a uno (es decir, programa y prueba las funciones una a una).

Empieza haciendo el funcionamiento básico descrito en el **apartado 7** y poco a poco ve añadiendo los demás. Es muy aconsejable hacer **copias de seguridad** del código tras cada etapa.

## 7. DESARROLLO POR ETAPAS Y CRITERIOS DE CALIFICACIÓN

1. **Videojuego Básico (5 puntos):** Se puede jugar pero solo hay 10 lanchas (no hay otros tipos de barcos) y sin niveles de dificultad (el juego termina tras 50 disparos).
2. **Añadir todos los barcos (2 puntos):** Añadir todos los tipos de barcos (Lancha, Buque, Acorazado y Portaaviones). Siguen sin haber niveles de dificultad, se juega en nivel fácil.
3. **Añadir 3 niveles de dificultad (2 puntos):** Añadir la opción de elegir el nivel de dificultad (solo fácil, medio y difícil).
4. **Añadir juego personalizado (1 punto):** Se añadirá la opción de juego personalizado.

En cada etapa desarrollada se valorará:

- Correcto funcionamiento del videojuego.
- Correcto uso de funciones y evitar repetir código (en la medida de lo posible).
- Código ordenado y bien estructurado.
- Nombres de variables apropiados y autoexplicativos.
- Comentarios útiles y breves que ayudan a entender el código.

## 8. ENTREGA

Deberás entregar toda la **carpeta del proyecto NetBeans** comprimida en un **archivo zip** a través de la tarea del aula virtual.

La fecha límite es el **domingo 12 de enero a las 23:55h**.

## Licencia



**Reconocimiento – NoComercial – CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.