P3

# PRÁCTICA EVALUABLE

Desarrollando el juego Hundir la Flota


## OBJETIVOS DIDÁCTICOS

- Escribir un videojuego simulando el archiconocido Hundir la Flota.
- Consolidar los conceptos estudiados en la unidad de estructuras de datos estáticas.
- Escribir programas más complejos utilizando funciones.

## 1. OBJETIVO DE LA PRÁCTICA

A partir de las reglas del juego *Hundir la Flota* desarrollar un programa en Java con una **versión simplificada del juego**.

Ver reglas del juego: Wikipedia

## 2. ENUNCIADO

En esta versión **simplificada** solo habrá un jugador humando (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.

Se jugará con un tablero de 10x10 posiciones que se enumerarán del 0 al 9 en las columnas 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.


## OBJETIVOS DIDÁCTICOS

- Escribir un videojuego simulando el archiconocido Hundir la Flota.
- Consolidar los conceptos estudiados en la unidad de estructuras de datos estáticas.
- Escribir programas más complejos utilizando funciones.

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 sus 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 - - - - - - - - - -
```


## OBJETIVOS DIDÁCTICOS

- Escribir un videojuego simulando el archiconocido Hundir la Flota.
- Consolidar los conceptos estudiados en la unidad de estructuras de datos estáticas.
- Escribir programas más complejos utilizando funciones.

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 las entradas erróneas (por ejemplo, si el usuario introduce una coordenada de disparo incorrecta).

# 3. CONSEJOS

### 3.1 DISEÑO DEL JUEGO

En este documento se dan unas pautas de la dinámica del juego. Sin embargo, debes tomar tú las decisiones de qué mensajes vas a dar al usuario y cómo quieres que interactúe con el juego. Por ejemplo, es recomendable el uso de menú/s para facilitar al usuario la información de las acciones que pueda tomar.

### 3.2. USO DE NÚMEROS ALEATORIOS

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

Recuerda que puedes utilizar la clase **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:

```
Random rand = new Random();

int alealtorio =
(int)(rand.nextDouble()*cantidad_números_rango
+ término_inicial_rango)
```


# OBJETIVOS DIDÁCTICOS

- Escribir un videojuego simulando el archiconocido Hundir la Flota.
- Consolidar los conceptos estudiados en la unidad de estructuras de datos estáticas.
- Escribir programas más complejos utilizando funciones.

### 3.3. 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.

### 3.4. USO DE ARRAYS Y MATRICES

En este programa deberás utilizar arrays (matrices principalmente) estáticos para organizar los datos en un tablero. Repasar estos conceptos de los apuntes de la unidad.

### 3.5. PASO DE ARRAYS A FUNCIONES

Recordad que los tipos de datos primitivos y las estructuras estáticas de datos se pueden pasar como parámetros a funciones. También que los *arrays* se pasan a las funciones como referencia. Es decir, que cuando una función recibe un *array* como parámetro, esta no hace una copia, sino que recibe el array original. De esta manera, la función está tratando el array original y por tanto no falta hacer un return de este array a la llamada de la función.

### 3.6. USO DE VARIABLES STATIC

El uso de variables *static* (“globales”) **no está permitido** en esta práctica.


## OBJETIVOS DIDÁCTICOS

- Escribir un videojuego simulando el archiconocido Hundir la Flota.
- Consolidar los conceptos estudiados en la unidad de estructuras de datos estáticas.
- Escribir programas más complejos utilizando funciones.

# 4. DESARROLLO POR ETAPAS Y CRITERIOS DE CALIFICACIÓN

A continuación, una referencia a la forma de valorar esta práctica:

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.

El diseño del juego (uso de menús para que el usuario elija acción, mensajes a usuario, dinámica del juego, etc) es de libre elección.

En cada etapa desarrollada se valorará:

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

# 5. 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.

## 6. RECONOCIMIENTO DE AUTORÍA

Autor de este documento: José Ramón Simó

Esta práctica es una adaptación de la práctica creada y modificada por los siguientes autores: Carlos Cacho, Raquel Torres, Lionel Tarazón y Javier Valero.

## 7. LICENCIA