

Universidad Rafael Landívar  
Departamento de ciencias básicas  
Pensamiento computacional  
Catedrática: Vivian Damaris Campos Gonzales

## **PROYECTO 2 (PARTE A)**

Diego Ignacio Sánchez López – 1249123

Guatemala, 31 de octubre de 2023

## Análisis y diseño del programa

### **Nombre y descripción de las clases que utilizará:**

1. Nombre: "Barco" Descripción: La clase "Barco" representa un barco en el juego de batalla naval. Cada objeto de esta clase tiene un nombre que identifica al barco.
2. Nombre: "Tablero" Descripción: La clase "Tablero" representa el tablero de juego en el que se juega batalla naval. Cada objeto de esta clase contiene una representación del tablero como una matriz de 10x10 casillas y también esta clase mantiene una lista de los barcos e juego.

### **Nombre, descripción y tipo de dato de los Atributos que contendrá las clases que utilizará:**

Clase Tablero:

Nombre de la Clase: Tablero

Descripción: Representa el tablero de juego en el que se desarrolla la Batalla Naval.

Atributos:

- tablero (list): Una matriz que representa el tablero de juego. Cada elemento de esta matriz almacena un carácter (' ', 'O', 'X' o 'M') para representar el estado de cada casilla.
- barcos (list): Una lista que almacena objetos de la clase Barco, representando los barcos colocados en el tablero.

Tipos de Datos de los Atributos:

nombre (str)

tamaño (int)

ubicación (list) - Lista de tuplas (int, int)

tablero (list) - Matriz de caracteres (str)

barcos (list) - Lista de objetos de la clase Barco

### **Nombre, descripción y si retornará algún valor los métodos que contendrá las clases que utilizará:**

Clase Barco:

Nombre de la Clase: Barco

Descripción: Representa un barco en el juego de Batalla Naval.

Métodos:

1. `__init__(self, nombre, tamaño)`: Constructor de la clase que inicializa los atributos del barco, incluyendo su nombre, tamaño y una lista vacía para las ubicaciones.

2. No retorna valores. El constructor (`__init__`) no retorna ningún valor.

Clase Tablero:

Nombre de la Clase: Tablero

Descripción: Representa el tablero de juego en el que se desarrolla la Batalla Naval.

Métodos:

1. `__init__(self)`: Constructor de la clase que inicializa el tablero y la lista de barcos en el tablero.
2. `mostrar_tablero(self, ocultar_barcos)`: Muestra el estado actual del tablero, con la opción de ocultar los barcos del oponente si `ocultar_barcos` es `True`.
3. `agregar_barco(self, barco, fila, columna, orientacion)`: Permite a un jugador colocar un barco en el tablero en una posición y orientación específicas.
4. `disparar(self, fila, columna)`: Permite a un jugador disparar a una casilla del tablero y verifica si el disparo acertó o no. Luego, actualiza el tablero con 'X' para aciertos y 'M' para disparos fallidos.
5. `todos_barcos_hundidos(self)`: Verifica si todos los barcos en el tablero han sido hundidos.

Los métodos `mostrar_tablero`, `agregar_barco`, `disparar`, y `todos_barcos_hundidos` son métodos de instancia y operan sobre la instancia específica de la clase `Tablero`. Estos métodos no retornan ningún valor explícito, pero realizan acciones dentro de la instancia del objeto, como mostrar el tablero, agregar barcos, registrar disparos y verificar si todos los barcos han sido hundidos. El valor de retorno de los métodos es implícito, ya que su principal propósito es realizar modificaciones en el estado del tablero.

### Condiciones y restricciones que debe tener su programa:

#### 1. Coordenadas del Tablero:

- El tablero del juego es de 10x10 casillas numeradas del 1 al 10 en ambas filas y columnas. Las filas se representan con letras de la 'A' a la 'J'.
- Las coordenadas ingresadas por el jugador deben estar dentro de este rango.

#### 2. Colocación de Barcos:

- Cada jugador tiene 3 barcos pequeños y 2 barcos grandes para colocar en su tablero.
- El jugador debe especificar la coordenada superior izquierda de un barco y su orientación (horizontal 'h' o vertical 'v') al colocar un barco en el tablero.
- Los barcos no pueden superponerse ni estar en posiciones fuera del tablero.

#### 3. Disparos:

- Los jugadores toman turnos para disparar a las coordenadas del tablero del oponente.

- El juego verifica si un disparo acertó en un barco ('O') o fue un disparo fallido ('X' o 'M').

#### **4. Ganador:**

- El juego determina un ganador cuando todos los barcos de un jugador han sido hundidos.
- El juego muestra un mensaje de victoria al jugador ganador.

#### **5. Ocultar Barcos:**

- Los barcos del oponente se pueden ocultar en el tablero si se establece la opción `ocultar_barcos` como `True` al mostrar el tablero.
- Los barcos solo se revelan cuando son alcanzados por un disparo.

#### **6. Entradas de Usuario:**

- El código incluye manejo de errores para asegurarse de que las entradas del usuario sean válidas. Si un jugador ingresa coordenadas fuera del rango o una entrada no válida, se le solicita que lo intente nuevamente.

#### **7. Finalización del Juego:**

- El juego termina cuando uno de los jugadores ha hundido todos los barcos del oponente.

### **Algoritmo que implementará en el programa:**

1. Se definen dos clases: Barco y Tablero. La clase Barco se utiliza para representar los barcos en el juego, y la clase Tablero se utiliza para representar el tablero de juego y gestionar la colocación de barcos, disparos y determinar el ganador.
2. El juego comienza con un mensaje de bienvenida y la creación de dos tableros, uno para cada jugador, representados por `tablero_jugador1` y `tablero_jugador2`.
3. Cada jugador tiene una lista de barcos, donde se definen el nombre y el tamaño de cada barco. Los barcos se crean como objetos de la clase Barco y se almacenan en la lista `barcos_jugador1` y `barcos_jugador2`.
4. Los barcos se colocan en el tablero de cada jugador. Para ello, se muestra el tablero inicial con los barcos ocultos. El jugador debe especificar la coordenada superior izquierda de un barco (por ejemplo, "A1") y la orientación (horizontal "h" o vertical "v"). El método `agregar_barco` en la clase Tablero se encarga de colocar los barcos en el tablero y mantener un registro de su ubicación.

5. Una vez que todos los barcos han sido colocados en los tableros, el juego comienza. Los jugadores se turnan para disparar a las coordenadas del tablero del oponente.
6. Cuando un jugador realiza un disparo, se verifica si acertó en un barco (casilla marcada con "O") o si fue un disparo fallido (casilla marcada con "X" o "M"). El resultado del disparo se muestra en la consola.
7. El juego continúa hasta que uno de los jugadores ha hundido todos los barcos del oponente. Se verifica esta condición utilizando el método `todos_barcos_hundidos` en la clase `Tablero`.
8. Cuando se cumple la condición de victoria, se muestra un mensaje indicando el ganador.
9. El juego permite que los jugadores se alternen y disparen hasta que se determine un ganador, y luego se cierra.