

BATALLA NAVAL

ETAPA 1: Jugador vs. Máquina (máquina pone barcos, jugador dispara)

Objetivo:

- La máquina pone barcos en una **matriz oculta**.
- El jugador tiene X intentos para adivinar y hundirlos.
- Se muestra un mapa actualizado tras cada intento.

Pasos:

1. **Definir el tamaño del tablero** (matriz 5x5, 6x6, etc.).
 2. **Inicializar la matriz de juego** (por ejemplo, con símbolos como - para agua).
 3. **Generar posiciones aleatorias para los barcos:**
 - Evitar solapamientos (que tengan la misma posición).
 - Decidir si los barcos son de una sola casilla o más grandes.
 4. **Pedir coordenadas al usuario** (fila y columna).
 5. **Evaluar el disparo:**
 - Si acierta, marcar con X (hundido).
 - Si erra, marcar con O (agua).
 6. **Mostrar el mapa actualizado** (sin mostrar los barcos no hundidos).
 7. **Contar intentos y verificar si ganó o perdió.** Por ejemplo damos como máximo 10 intentos.
 8. **Mostrar mensaje final con resultado.**
-

ETAPA 2: Máquina vs. Jugador (ambos colocan y disparan)

Objetivo:

- Añadir una matriz para los **barcos del jugador**.
- La máquina también **dispara a posiciones aleatorias**.

Pasos adicionales:

1. **Permitir al jugador colocar sus barcos manualmente.**
2. **Agregar lógica para el turno de la máquina:**
 - Selección aleatoria de coordenadas (evitar repetir).
 - Guardar historial de tiros.
3. **Mostrar dos tableros:**
 - Uno del jugador disparando.
 - Uno del enemigo disparando.
4. **Alternar turnos entre jugador y máquina.**
5. **Determinar quién gana primero.**

Después de cada disparo se puede agregar `printf("Presione ENTER para continuar...");`
`getchar();` // Espera a que el jugador presione ENTER

Y para limpiar la pantalla:
`system("cls");` // En Windows
// o
`system("clear");` // En Linux/macOS

Para simular que la máquina está "pensando" antes de disparar se puede usar:

`sleep` o `Sleep` dependiendo del sistema operativo:

LINUX
`#include <unistd.h>`

`sleep(2);` // espera 2 segundos

WINDOWS
`#include <windows.h>`

`Sleep(2000);` // espera 2000 milisegundos = 2 segundos