

## PROYECTO PY03

### Batalla Naval en Lenguaje Ensamblador ARM

---

#### 1. Objetivo General

Desarrollar un programa en consola que permita jugar **Battleship (Batalla Naval)** utilizando una **interfaz de línea de comandos (CLI)**, en lenguaje ensamblador ARM. Se incluirá soporte para partidas en modo solitario y una opción opcional para partidas multijugador a través de **sockets**.

---

#### 2. Requerimientos Funcionales

##### 2.1 Configuración del Juego

- Selección entre mapa estándar predefinido o creación de tablero personalizado.
- Almacenamiento de configuraciones personalizadas localmente para reutilización.

##### 2.2 Tamaño del Mapa

- Mapa estándar: 10x10 celdas.
- Mapa personalizado: entre 10x10 y 20x20 celdas.

##### 2.3 Colocación de Barcos

Cada jugador deberá colocar los siguientes barcos, con orientación horizontal o vertical:

##### Tipo de barco Celdas ocupadas

Portaaviones 5

Acorazado 4

Submarino 3

Crucero 3

Destructor 2

---

## 2.4 Modos de Juego

- **Solitario**  
El jugador compite contra un oponente automático que utiliza movimientos válidos y aleatorios.
- **Multijugador (opcional)**  
Permite juego entre dos jugadores en tiempo real mediante **tecnología de sockets**.

Si se implementa esta funcionalidad, debe ser completamente funcional para optar por puntos adicionales. La documentación debe detallar claramente el proceso de conexión entre clientes.

---

## 2.5 Jugabilidad

- Interacción mediante comandos en consola.
  - Los jugadores solo visualizan:
    - Sus propios barcos.
    - Resultados de sus ataques en el tablero enemigo.
  - Los ataques se realizan indicando coordenadas.
  - El juego notifica aciertos, fallos o destrucción de barcos.
  - La partida finaliza al hundir todos los barcos del oponente.
- 

## 3. Documentación Externa

El repositorio del proyecto deberá incluir:

- Descripción general y objetivos del juego.
  - Manual de usuario (instrucciones de ejecución y juego).
  - Detalles sobre el diseño y los algoritmos implementados.
  - Evaluación y reflexión sobre los objetivos alcanzados.
- 

## 4. Tecnología

- Lenguaje: **Ensamblador ARM**.

- Entorno: **Raspbian OS** en **Raspberry Pi** o **emulador QEMU**.
  - Herramientas de compilación: **AS** y **LD**.
  - No se permite utilizar funciones de C o C++ directamente en el código.
- 

## 5. Grupos de Trabajo

- Composición: Grupos de **2 a 3 estudiantes**.
  - Gestión de código a través del **repositorio oficial del curso**, con acceso para el profesor y asistente del curso (@hros).
- 

## 6. Evaluación

- Basada en la funcionalidad del programa, calidad del código y claridad de la documentación.
  - El proyecto debe poder ejecutarse y evaluarse sin la asistencia directa de los desarrolladores.
- 

## 7. Recomendaciones Finales

Se recomienda iniciar el desarrollo con suficiente antelación, realizando una planificación detallada y una implementación progresiva.

La claridad, estructura y coherencia del proyecto serán altamente valoradas.

---

**¡Mucho éxito con el trabajo!**