

Primera entrega Proyecto 2 Pensamiento Computacional

Sebastian Villeda

Carné:

1032625

Universidad Rafael Landívar

Pensamiento Computacional

Estuardo Castillo

9 de mayo del 2025

Documento PDF que contiene el análisis y diseño (algoritmo) del programa **(40 Puntos)**. El cual debe de contener lo siguiente:

- Acciones del programa **(03 Puntos)**: Enumerar las acciones disponibles.
- Datos de entrada **(03 Puntos)**: Especificar la información que se solicitará al usuario y definir el tipo de dato para cada uno.
- Variables **(04 Puntos)**: Listar las variables a utilizar.
- Condiciones y cálculos **(05 Puntos)**: Describir las restricciones y fórmulas.
- Algoritmo **(25 Puntos)**: Presentar un diagrama de flujo (elaborado en Draw.io) que detalle la lógica completa del programa.

## Análisis y diseño del programa

### Acciones del programa

1. Mostrar una interfaz de bienvenida a los jugadores.
2. Solicitar el ingreso del nombre de los jugadores.
3. Asignar las posiciones de los barcos de manera aleatoria.
4. Verificar que los jugadores estén conformes con las posiciones en que los barcos fueron generados.
5. Mostrar el tablero actualizado de flota naval cada que vez que ocurre un turno.
6. Mostrar tablero de ataque.
7. Solicitar las coordenadas en las que desea realizar el ataque.
8. Mostrar menú de seguir jugando o rendirse.
9. Mostrar resultado de la partida.

### Datos de entrada

Datos solicitados al usuario	Tipo de dato	Descripción
Nombre del jugador	String	El jugador 1 y el jugador 2 ingresarán un nombre único para cada uno
Opciones de menú de iniciar, seguir jugando y rendirse	Int	Las opciones del menú estarán dadas por números para que el jugador identifique de mejor forma que número sirve para cada cosa, el número 1 servirá para continuar e iniciar la partida y el 2 para rendirse.
Coordenadas de ataque hacía el barco rival	string	Se le pedirá al jugador que coloque de primero el número de fila y luego el número de columna, para luego mostrar la coordenada que ingreso el jugador y luego mostrar en el tablero el ataque realizado.

Confirmación de posición de los barcos	Int	Se mostrará el tablero con la posición de los barcos establecidos por un random y si al jugador no le gusta el resultado lo puedo randomizar de nuevo
--	-----	---

### Variables

Main	Métodos que se llamaran en el main
	saludojugadores() verificarVictoria() dibujarTableros() lanzarMisil() recibirMisil() anotarResultado() sumarPuntos() posiciónBarcos()

Clases	Atributos	Métodos
Jugador	jugador1 jugador2 puntosxjugador	saludojugadores() verificarVictoria()
Tableros hereda la clase Jugador	submarino destructor fagata posiciónSubmarino [,] posiciónDestructor [,] posiciónFagata [,] flotaNaval [,] confirmaciónDeBarcos tableroAtaque [,] puntosSubmarino puntosDestructor puntosFagata conteoTurnos coordenadasAtaque	dibujarTableros() lanzarMisil() recibirMisil() anotarResultado() sumarPuntos() posiciónBarcos()

Nombre de variable	Tipo de variable	Descripción
jugador 1	String	Nombre del jugador número 1
jugador 2	String	Nombre del jugador número 2
puntosxjugador	Int	Lleva el conteo de los puntos acumulados por el jugador

submarino	Int	Cantidad de casillas que puede ocupar el submarino (su vida)
Destructor	Int	Cantidad de casillas que puede ocupar el destructor (su vida)
fagata	Int	Cantidad de casillas que puede ocupar la fagata (su vida)
posiciónSubmarino [,]	Matriz bidimensional tipo string	Es la posición en donde estará el submarino, esta se determinará usando un random
posiciónDestructor [,]	Matriz bidimensional tipo string	Es la posición en donde estará el destructor, esta se determinará usando un random
posiciónFagata [,]	Matriz bidimensional tipo string	Es la posición en donde estará la fagata, esta se determinará usando un random
confirmaciónDeBarcos	Bool	Se le preguntará al usuario si esta conforme con la posición en la que se generaron sus barcos y si no lo esta se volverá a randomizar las posiciones
flotaNaval [,]	Matriz bidimensional tipo string	Esta matriz dibujara el tablero en donde se encuentran los barcos del jugador
tableroAtaque [,]	Matriz bidimensional tipo string	Esta matriz dibujara el tablero en donde se encuentran los barcos del jugador
puntosSubmarino	int	Es la cantidad de puntos que da el submarino al ser destruido
puntosDestructor	int	Es la cantidad de puntos que da el destructor al ser destruido
puntosFagata	Int	Es la cantidad de puntos que da la fagata al ser destruido
conteoTurnos	Int	Cuenta los turnos que llevan los jugadores
coordenadasAtaque[,]	Matriz bidimensional tipo string	Es la coordenada de ataque escribirá el usuario

### Condiciones y cálculos

- El juego finalizará luego de 15 turnos, si ninguno de los 2 gana en los 15 turnos se hará un recuento de puntos para ver quién será el ganador, pero si ambos están empatados en puntos este terminará en empate.
- El juego finaliza si un jugador se rinde antes de terminar los 15 turnos o si a un jugador le destruyen o destruye todos los barcos.
- Si un jugador escribe una coordenada fuera de los límites del tablero (de A-F y de 1-6) o en una posición que ya ingreso antes, esta se marcará como incorrecta y lo hará escribirla de nuevo.
- El submarino ocupa 2 casillas en el tablero y al ser tumbado por completo otorga 2pts al jugador.
- La fragata ocupa 3 casillas en el tablero y al ser tumbada por completo otorga 3pts al jugador.
- El destructor ocupa 4 casillas en el tablero y al ser tumbada por completo otorga 4pts al jugador.
- La posición de los barcos debe determinarse de manera aleatoria y esta no podrá compartirse con otros barcos, por ejemplo: un submarino no puede estar en el mismo espacio que un destructor.
- Si un jugador intenta escribir una coordenada que en la que ya atacó con anterioridad, no lo contará y lo hará escribir la coordenada nuevamente.

## Diagrama de flujo

