

Universidad Rafael Landívar

Facultad de Ingeniería

Licenciatura en Ingeniería en Informática y Sistemas

Laboratorio de Pensamiento Computacional

Docente: Felipe Mendez

Análisis y Diseño: Proyecto 2 FASE 01

Estudiante: Morales Mazariegos, Martín Andrés

Carnet: 1217025

Guatemala, 07 de mayo de 2025

1. ALGORITMO

1.1. Acciones disponibles del usuario:

- Ingresar el nombre de usuario que desee usar cada jugador.
- Leer las reglas del juego y en que consiste.
- Cada usuario puede ver su tablero con tus barcos y los lugares donde el enemigo ha atacado. Y además el usuario podrá ver el tablero de su enemigo, pero solo con los lugares que el usuario ha atacado mostrando una "X" si falló o un "O" acertó.
- Cada usuario tendrá 15 intentos de darle a los barcos.
- Cada usuario puede ver su puntaje conforme avance el juego al igual que los tableros antes mencionados.
- Finalmente, los usuarios pueden reiniciar el juego si desean al finalizar el juego.

1.2. Información solicitada al usuario:

- 1.2.1. Primero se solicitará el nombre de usuario que desee cada usuario y cada uno sería un tipo de variable **string**.
- 1.2.2. Luego por cada intento de los 15 en el juego que tengan se les solicitará que ingrese una coordenada entre (A1-A6, B1-B6, C1-C6, D1-D6, E1-E6 y F1-F6) a cada usuario, y esta variable solicitada será de tipo **string**.
- 1.2.3. Y finalmente se le solicitará al usuario que ingrese si/no si desea reiniciar el juego al finalizar o no y será una variable tipo **string**.

1.3. Variables:

- **Tipo string:** usuario1, usuario2, ataqueUsuario1, ataqueUsuario2, tableroUsuario1, tableroUsuario2, coordenada (para un foreach), coordenadaBarco1, coordenadaBarco2, coordenadaBarco3 y reiniciar.
- **Tipo int:** puntajeUsuario1, puntajeUsuario2, i (variable de iteración), random y posicionBarcos (para determinar las posiciones de los barcos con él random).

1.4. Condiciones y cálculos:

Cálculos

Puntaje:

- `puntajeUsuario1` += 100 (por cada tiro acertado).
- `puntajeUsuario2` += 100 (por cada tiro acertado).
- `puntajeUsuario1` += 200 (por cada barco destruido).
- `puntajeUsuario2` += 100 (por cada barco destruido).

Condiciones

- Si `puntajeUsuario1` mayor que `puntajeUsuario2`, usuario1 gana.
- Si `puntajeUsuario2` mayor que `puntajeUsuario1`, usuario2 gana.
- Si `puntajeUsuario1` igual que `puntajeUsuario2`, empate.
- Si `ataqueUsuario` (1 o 2) es igual a `coordenadaBarco1`, `coordenadaBarco2` o `coordenadaBarco3` puntaje (1 o 2) incrementa a cien y “O” aparece en el tablero. De lo contrario el puntaje no aumenta y el aparecería “X” en el tablero.
- Si reiniciar igual a si el juego comienza desde cero, de lo contrario el juego finaliza.
- Si el random sale 1 (posición de barcos1), si el random sale 2 (posición de barcos2), si el random sale 3 (posición de barcos3), si el random sale 4 (posición de barcos4), si el random sale 5 (posición de barcos5), si el random sale 6 (posición de barcos6), si el random sale 7 (posición de barcos7), si el random sale 8 (posición de barcos8), si el random sale 9 (posición de barcos9), si el random sale 10 (posición de barcos10).
- Si el dato ingresado es diferente de (A1-A6, B1-B6, C1-C6, D1-D6, E1-E6 y F1-F6) el dato no será válido y se solicitará de nuevo hasta que lo sea.
- Si intentos de los usuarios menor o igual a 0 finaliza el juego.
- Si `coordenadaBarco1`, `coordenadaBarco2` y `coordenadaBarco3` es igual a “O” también finalizará el juego dando como ganar el que acierte los 3 barcos antes de los 15 intentos.

2. DIAGRAMA DE FLUJO

El archivo también fue enviado en la tarea del portal.

<https://github.com/mamoralesmaza/Laboratorios/tree/main/Proyecto%2002%20Fase%2001%20Martin%20Morales>