

# BATTLESHIP

## GAME

START



Gisel Daniela Caicedo Soler  
Cristian Camilo Valbuena Bocanegra  
Universidad Antonio Nariño

# BATTLESHIP

## GAME

## Resumen

El presente proyecto tiene como objetivo el desarrollo de un juego de batalla naval en el lenguaje de programación C++. El programa permitirá al usuario interactuar a través de la consola, utilizando las mecánicas clásicas y dinámicas de combate naval. Los barcos enemigos se situarán aleatoriamente en un tablero y el objetivo del jugador será eliminarlos mediante ataques continuos hasta hundir todas las naves enemigas.

El desarrollo del juego ha sido completado, alcanzando así el 100% de funcionalidad del código correspondiente. La meta final lograda es un programa que proporciona al jugador un tablero de juego con un número determinado de casillas, realiza un seguimiento de los movimientos del usuario y brinda una retroalimentación adecuada.

**Código:** <https://onlinegdb.com/tWDotXuDD>

# Introducción

La temática de las batallas navales en distintas industrias ha sido un impacto significativo en los jugadores sin importar la edad, quienes fueron cautivados por la capacidad envolvente que logra el sumergirse en un enfrentamiento marítimo. Nuestro juego está diseñado para que el jugador despliegue sus flotas, seleccione cuidadosamente sus movimientos y realizar ataques precisos que terminen con sus enemigos.

A continuación se presentará una visión general detallada del proyecto, incluyendo una justificación de su desarrollo, el estado del arte en juegos de batalla naval y los objetivos que se pretende alcanzar. Además, se hablará del diseño y sus correspondientes funcionalidades acompañado del cronograma de desarrollo para su implementación.

Nuestra meta finalmente, es otorgarle a quien lo juegue una experiencia satisfactoria y divertida, que cumpla con sus expectativas con respecto a un combate ávido de estrategia dentro de las limitaciones que ofrece el lenguaje empleado. Estamos contentos con el resultado final y esperamos que nuestro trabajo cumpla con los requerimientos mínimos y mucho más que se espera de un juego tan clásico como este, puesto que la creación de este juego de batalla naval ha satisfecho las expectativas personales que se esperan de él.

# Justificación

El juego de batalla naval puede llegar a ser un juego donde la persona pone a prueba su pensamiento estratégico, ya que el jugador debe planificar correctamente sus movimientos para derrotar a su oponente. Por otra parte, podemos desarrollar la capacidad de analizar y deducir situaciones de manera más rápida, ayudando a los participantes a mejorar su concentración y con la posibilidad de llegar a ser muy útil en diferentes situaciones de la vida.

La demanda de juegos de estrategia y batalla naval ha sido un gran impacto en la industria de los videojuegos, nuestro desarrollo se basa en el atractivo de la modalidad empleada. La demanda que se nos exige nos ayudará a incrementar nuestro aprendizaje y crecimiento personal que obtuvimos mediante el desarrollo del programa. Se mantiene una postura segura acerca del potencial que tiene el proyecto para lograr una experiencia única y emocionante al jugarlo mientras exploramos y aplicamos habilidades y conocimientos en el campo de desarrollo de un programa en C++.

# Estado de Arte

Hay una variedad de tecnologías que se pueden usar para implementar un juego de batalla naval, incluidos lenguajes de programación como Python, Java o C++, y marcos de desarrollo de juegos como Unity o Unreal Engine. También hay bibliotecas y módulos disponibles para simplificar la implementación del juego en diferentes lenguajes de programación. A continuación, se presenta un resumen de los juegos más destacados y las tendencias actuales en el género de batalla naval.

Un título con la capacidad de resaltar por su originalidad y adaptabilidad es "Battleship" de Hasbro Interactive. Este juego está basado en el popular juego de mesa del mismo nombre que ha sido adaptado en diferentes plataformas digitales y en su momento ofreció novedosas mecánicas de adivinanzas y estrategias para lograr impactar en las ubicaciones de los barcos enemigos. Battleship logra hacer un gran trabajo manteniéndose fiel al juego original y al mismo tiempo estableciendo un sistema de combate interesante.

Uno de los juegos más reconocidos en el ámbito es "World of Warships", desarrollado por Wargaming. Un juego en línea que ofrece emocionantes batallas navales y reúne una armada de más de 600 barcos de la primera mitad del siglo XX, desde destructores hasta gigantes acorazados. Históricamente sigue una fidelidad grandiosa con famosos barcos y con todos los estilos de juego logra establecer una jugabilidad que entretiene al usuario.

Por último, pero no menos importante. Se busca destacar el juego online que influencio en demasía el desarrollo de este programa. Battleship Game, disponible en cualquier navegador y desarrollado con la versión de HTML5. Su diseño estético es sencillo de comprender y lograr en un lenguaje de programación de C++, el mapeo y la interfaz de usuario nos ofreció una facilidad creativa aliviadora para entender el funcionamiento de las mecánicas de este modo de juego.



# BATTLESHIP

## GAME

# Objetivos

Diseñar una interfaz gráfica atractiva y amigable: el objetivo principal era crear una interfaz visualmente atractiva que capturara la esencia de la guerra naval y brindara a los jugadores una experiencia inmersiva. Se prestará especial atención a la usabilidad y accesibilidad, asegurando que la interfaz sea intuitiva y fácil de navegar.

Implementar mecánicas de juego estratégicas y desafiantes: el objetivo es diseñar y desarrollar mecánicas de juego que fomenten la estrategia, la toma de decisiones tácticas y el pensamiento estratégico. Los jugadores deben tener en cuenta factores como la posición del barco.

# BATTLESHIP

## GAME

# Diseño de E/S

## Entrada:

Los datos de entrada que se necesitarán son:

1. Selección de usuario: El usuario tendrá que elegir las casillas dentro del tablero de juego, ingresando las coordenadas de fila y columna.
2. Selección automática de la posición de los barcos: Se generará de manera aleatoria la posición de los barcos en el tablero.
3. Selección de cancelación de partida: El usuario tiene que tener la posibilidad de cancelar la partida en cualquier momento.
4. Seleccionar para iniciar partida: Así como se podrá cancelar la partida, también se debe tener la posibilidad de iniciar una en cualquier momento.

## Salida:

Aunque especificar los datos de salida del programa a tan temprana etapa no sea posible.

1. Mensajes de retroalimentación: Se le informará al jugador, si ha acertado o errado en un movimiento.
2. Mensajes para indicar si ganó o perdió.
3. Estadísticas: Cuando se tengan la interacción del jugador, se podrá dar el análisis completo como; partidas ganadas o perdidas y el top cinco de puntajes.
4. La posición de los barcos.
5. Las posiciones atacadas por el jugador, acompañada de un informe si logran impactar o no.
6. Cuando se hunda un barco en su totalidad, indicar cual fue.
7. Puntuación final.

# BATTLESHIP GAME

## Diagrama de flujo





# BATTLESHIP

## GAME

# Cronograma

Actividades	Meses				(Semana s)	
Cortes: 1, 2, 3 y 4.	Febrero (4 semanas)	Marzo (4 semanas)	Abril (4 semanas)	Mayo (4 semanas)	Junio	
Definición de requisitos y diseño inicial.  2. Establecer los requisitos del juego (funcionalidades, reglas, etc.). 3. Diseñar la estructura general del juego y el tablero.	2 3 4					
Realización del mapeo y generación automática de barcos. Programa con funcionalidad mínima del 40%.		2 3 4				
Introducción, justificación, estado del arte, manual usuario. Programa con funcionalidad mínima del 70%. Optimizar el código y mejorar la eficiencia si es necesario. Refinar la interfaz de usuario y la presentación de los mensajes. Realizar ajustes adicionales en base a los comentarios y retroalimentación recibida.			1 2 3 4			
Documento final con normas APA y sus conclusiones. Programa 100%. Perfeccionar código y detalles, así como sus trucos. (ENTREGA FINAL)				1 2 3 4		

# Manual de Usuario

- Se requiere un compilador de C++ para compilar y ejecutar el código.
- Se necesita una terminal para interactuar con el juego.



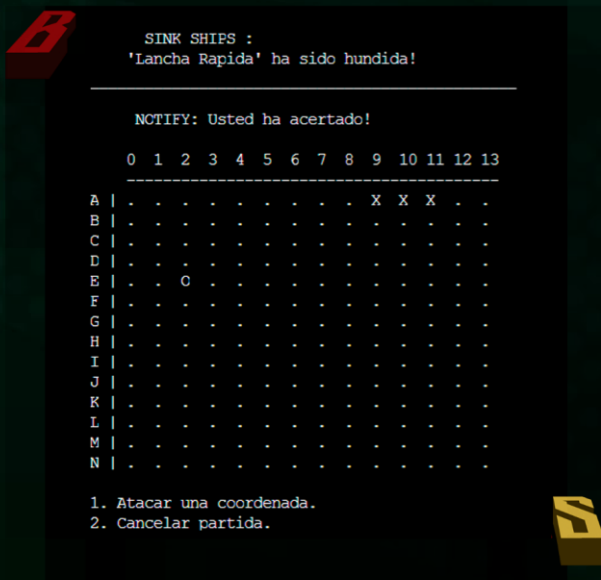
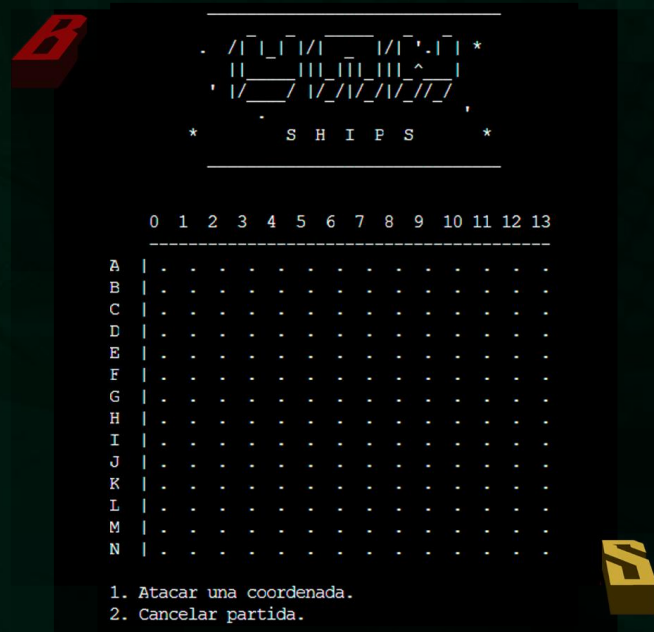
- Al iniciar el juego, se mostrará el menú principal con las siguientes opciones.
- Para seleccionar una opción, ingrese el número correspondiente y presione Enter.
- Si ingresa una opción inválida, se mostrará su correspondiente mensaje.

# BATTLESHIP

## GAME

### 2. Iniciar partida.

- Si selecciona la opción "1" en el menú principal, se iniciará una nueva partida del juego.
- Se mostrará un tablero vacío en la pantalla.
- Los barcos se colocarán aleatoriamente en el tablero, y el objetivo del juego es hundir todos los barcos en la menor cantidad de intentos posibles.
- Para atacar una coordenada, se le pedirá que ingrese las coordenadas X e Y.
- Ingrese las coordenadas y presione Enter.



- Si acierta y golpea un barco, se mostrará un mensaje de éxito y el tablero se actualizará con un marcador "X" en la ubicación del barco hundido.
- Si falla y golpea el agua, se mostrará un mensaje de error y el tablero se actualizará con un marcador "O" en la ubicación del intento fallido.
- Después de cada intento, se mostrará el tablero actualizado.
- El juego continúa

hasta que se hayan hundido todos los barcos y mostrará su correspondiente aviso si es que el

# BATTLESHIP

## GAME

jugador ha hundido un barco, acumulando el total de barcos hundidos.

- También puede cancelar la partida en cualquier momento ingresando el número negativo tanto para la coordenada X como la coordenada Y.



#### 4. Finalizar partida.

- Si selecciona la opción "3" en el menú principal, se finalizará el juego y se mostrará un mensaje de salida exitosa.

#### 3. Mostrar estadísticas.

- Si selecciona la opción "2" en el menú principal, se mostrarán las estadísticas del juego.
- Las estadísticas incluyen información como la cantidad de partidas canceladas y top cinco de puntajes.



The screenshot shows a Commodore 64 screen with a red 3D letter 'B' logo in the top-left corner. The title 'SHIPS' is displayed in a stylized font at the top center, flanked by asterisks. Below the title are two identical 14x14 grids representing a ship placement board. Each grid has columns numbered 0-13 and rows labeled A-N. The first grid is empty, while the second grid contains several ships represented by letters: 'A' at B4, B5, C5, D5; 'P' at F7, G7, H7, I7; 'L' at J6, K6, L6; and 'M' at B0, B1, B2, B3.



# Conclusiones

En definitiva, desarrollar un juego de batalla naval en C++ ha sido un proceso exitoso que ha traído innumerables beneficios. Al elegirlo como nuestro lenguaje de programación, aprovechamos su poder y eficiencia para crear un juego robusto y de alto rendimiento.

Una de las principales ventajas de usar C++ es su capacidad para administrar eficientemente los recursos del sistema. Esto es fundamental para garantizar un rendimiento de juego óptimo, especialmente cuando se trata de velocidad de ejecución y procesamiento de gráficos en tiempo real. La optimización del código C++ nos permite crear una experiencia de juego fluida y sin problemas que hace que los jugadores se sientan como si estuvieran allí.

Con todo, desarrollar un juego de batalla naval es una buena elección, nos brinda un alto rendimiento y un código bien estructurado.

La eficiencia y la flexibilidad de C++ ayudan a crear juegos de alta calidad que brindan a los jugadores una experiencia de juego emocionante y atractivo.



# Referencias

- Anton Oparin, Alexander Bogomolsky, Artur Tokhtash (2015). World of Warships. Wargaming Group Limited, Lesta Studio. Recuperado de [https://store.steampowered.com/app/552990/World\\_of\\_Warships/](https://store.steampowered.com/app/552990/World_of_Warships/)
- Marmalade Game Studio Ltd (2018). Hasbro's BATTLESHIP. Recuperado de [https://store.steampowered.com/app/893050/Hasbros\\_BATTLESHIP/](https://store.steampowered.com/app/893050/Hasbros_BATTLESHIP/)
- Battleship.org (2012-2023) Recuperado de <https://www.battleshiponline.org>