

Batalla Naval Algebraica

Documento Final – Casos de Uso

Proyecto Universitario

Contents

1	Introducción	3
2	Descripción General del Sistema	4
3	Reglas del Juego	5
4	Actores del Sistema	6
4.1	Jugador	6
4.2	Sistema	6
4.3	IA	6
5	Diagramas del Sistema	7
5.1	Diagrama UML de Casos de Uso (Representación Textual)	7
5.2	Diagrama de Clases (Representación Textual)	7
6	Casos de Uso Completos	8
6.1	UC-01: Iniciar Partida	8
6.2	UC-02: Colocar Barcos	8
6.3	UC-03: Realizar Disparo	9
6.4	UC-04: Determinar Fin de Partida	9
6.5	UC-05: Mostrar Tablero	10
6.6	UC-06: Colocación Automática	10
6.7	UC-07: Guardar Partida	10
6.8	UC-08: Cargar Partida	10
6.9	UC-09: Reiniciar Partida	10
6.10	UC-10: Estadísticas del Juego	10
6.11	UC-11: Juego Contra IA	10
6.12	UC-12: Validación Avanzada de Reglas	10
6.13	UC-13: Validar Coordenada Algebraica	10

7	Escenarios Extendidos	11
7.1	UC-02 Colocar Barcos	11
7.2	UC-03 Realizar Disparo	11
8	Reglas Algebraicas del Sistema	12
9	Glosario	13
10	Conclusiones	14

Introducción

Este documento presenta la versión final y extendida de los **Casos de Uso del sistema Batalla Naval Algebraica**. El proyecto combina mecánicas clásicas del juego Batalla Naval con elementos matemáticos avanzados como funciones cuadráticas, transformaciones lineales, desigualdades y restricciones geométricas.

Ejemplos de funciones empleadas:

$$f(x) = 2x^2, \quad g(x) = x + 3$$

Estas funciones permiten validar coordenadas, generar posiciones automáticas, construir patrones para la IA y aplicar reglas algebraicas al tablero.

Descripción General del Sistema

El sistema Batalla Naval Algebraica integra mecánicas tradicionales del juego con un motor algebraico capaz de validar coordenadas, evaluar funciones, aplicar desigualdades y generar estrategias basadas en transformaciones matemáticas. Los módulos principales incluyen:

- **Interfaz del Jugador:** Entrada de disparos y colocación de barcos.
- **Motor Algebraico:** Evalúa expresiones como $y = 2x^2$, $y = x + 3$ y desigualdades.
- **Motor de Juego:** Reglas, impactos, hundimientos y condiciones de victoria.
- **IA Matemática:** Estrategias algorítmicas basadas en funciones y transformaciones.

El objetivo del sistema es enriquecer el juego clásico mediante la incorporación de razonamiento matemático.

Reglas del Juego

- Cada jugador utiliza un tablero de 10×10 .
- Los barcos pueden colocarse manualmente o mediante funciones matemáticas.
- Las funciones empleadas deben generar coordenadas válidas.
- Los disparos pueden ingresarse como pares ordenados o como valores evaluados.
- El juego finaliza cuando todos los barcos de un jugador se destruyen.
- La IA puede generar disparos basados en funciones como $f(x) = 2x^2$.

Actores del Sistema

Jugador

Usuario que interactúa directamente con el sistema para jugar.

Sistema

Módulo backend encargado de reglas, validación, cálculos algebraicos y administración del juego.

IA

Oponente automático basado en estrategias matemáticas.

Diagramas del Sistema

Diagrama UML de Casos de Uso (Representación Textual)

- Jugador → Iniciar Partida
- Jugador → Colocar Barcos
- Jugador → Realizar Disparo
- Sistema **include**: Validar Coordenada
- Realizar Disparo **extends**: Determinar Fin de Partida
- Colocar Barcos **include**: Validar Coordenada
- IA → Realizar Disparo Automático
- Sistema **include**: Generación Algebraica para colocación automática

Diagrama de Clases (Representación Textual)

Clases principales:

- **Tablero**: matriz, validar_posicion(), colocar_barco()
- **Barco**: tamaño, coordenadas, esta_hundido()
- **Disparo**: x, y, resultado
- **FuncionAlgebraica**: expresión, evaluar(x)
- **IA**: generar_disparo(), estrategia_matematica()

Casos de Uso Completos

UC-01: Iniciar Partida

Actor: Jugador

Precondición: No existe una partida activa.

Postcondición: Se generan tableros vacíos.

Flujo Normal

Paso 1: El jugador selecciona “Iniciar Partida”.

Paso 2: El sistema genera tableros.

Paso 3: El sistema confirma la creación de la partida.

UC-02: Colocar Barcos

Actor: Jugador

Descripción: El jugador coloca barcos manualmente o usando funciones como $y = 2x^2$.

Precondición: Tablero en fase de colocación.

Postcondición: El barco queda colocado o se rechaza.

Flujo Normal

Paso 1: El jugador selecciona un barco.

Paso 2: Ingresa coordenadas o una función algebraica.

Paso 3: El sistema valida límites, colisiones y reglas algebraicas.

Paso 4: El sistema aprueba o rechaza la colocación.

Flujos Alternos

- Coordenada fuera del tablero.
- La función genera puntos inválidos.
- Colisión con otro barco.

UC-03: Realizar Disparo

Actor: Jugador

Descripción: Evalúa impacto, agua o hundimiento.

Incluye: Validar Coordenada.

Flujos Alternos

- Disparo repetido.

UC-04: Determinar Fin de Partida

Actor: Sistema

Extiende: Realizar Disparo

UC-05: Mostrar Tablero

UC-06: Colocación Automática

UC-07: Guardar Partida

UC-08: Cargar Partida

UC-09: Reiniciar Partida

UC-10: Estadísticas del Juego

UC-11: Juego Contra IA

UC-12: Validación Avanzada de Reglas

UC-13: Validar Coordenada Algebraica

Actor: Sistema

Descripción: Valida coordenadas generadas manualmente o mediante funciones.

Flujo Normal

Paso 1: Se recibe una coordenada o función.

Paso 2: Se evalúa la expresión (ejemplo: $y = 2x^2$).

Paso 3: Se verifican las reglas del tablero.

Paso 4: Se acepta o rechaza la coordenada.

Flujos Alternos

- Coordenada fuera de rango.
- Colisión.
- Violación de desigualdades algebraicas.

Escenarios Extendidos

UC-02 Colocar Barcos

- Escenario 1: Colocación manual.
- Escenario 2: Colocar barco siguiendo $y = 2x^2$.
- Escenario 3: Rechazo por colisión.

UC-03 Realizar Disparo

- Disparo válido.
- IA usando trayectoria $f(x) = 2x^2$.
- Disparo que termina la partida.

Reglas Algebraicas del Sistema

- Curvas cuadráticas: $y = 2x^2$
- Líneas rectas: $y = x + 3$
- Zonas prohibidas: $2x^2 + y > 15$
- Transformaciones lineales para IA

Glosario

- **Coordenada algebraica:** Punto generado por una función.
- **Curva cuadrática:** Ecuación de forma $ax^2 + bx + c$.
- **Transformación lineal:** Reubicación matemática de puntos.
- **Heurística algebraica:** Estrategia basada en funciones matemáticas.

Conclusiones

El sistema Batalla Naval Algebraica demuestra que es posible integrar conceptos matemáticos avanzados dentro de una mecánica lúdica conocida. Los casos de uso presentados describen de manera completa el comportamiento del sistema, permitiendo una comprensión clara de su funcionamiento y una base sólida para futuras mejoras, expansiones o implementación real.