

PROPIEDAD DE PROYECTO

PROPUESTA

BATALLA NAVAL

EQUIPO

LOS CODICIOSOS

INTEGRANTES

CANUL DURÁN NICOLAS ISRAEL
ORTIZ ZALDÍVAR ISAAC ALEJANDRO
FLORES VERA ALEJANDRO MANUEL
LEÓN CHAY JESÚS ISRAEL





ANTECEDENTES

Batalla Naval es una reinterpretación moderna del clásico juego de mesa, que será desarrollada como un videojuego de consola utilizando el lenguaje C. A pesar de ser un juego ampliamente replicado, la mayoría de las versiones digitales actuales no ofrecen novedades en su jugabilidad. Esta propuesta busca romper con esa monotonía añadiendo mecánicas modernas como cartas de poder y movimiento de unidades, lo que permitirá una experiencia más estratégica, dinámica y única.

Además de su valor lúdico, el proyecto tiene como finalidad servir como plataforma de aplicación de los conocimientos adquiridos durante el curso, incluyendo programación estructurada, uso de estructuras, manejo de memoria, ciclos, condicionales y lógica de control.

DESCRIPCION DEL PRODUCTO DE SOFTWARE

EL SOFTWARE DESARROLLADO ES UN PROTOTIPO DE VIDEOJUEGO TIPO BATALLA NAVAL, EJECUTABLE POR CONSOLA, ENFOCADO EN PARTIDAS MULTIJUGADOR LOCALES POR TURNOS.

EL SISTEMA PERMITE A CADA JUGADOR REGISTRAR SU NOMBRE, VISUALIZAR UN TABLERO VACÍO, Y COLOCAR ESTRATÉGICAMENTE SU FLOTA INICIAL DE BARCOS (UN TOTAL DE CINCO UNIDADES CON DISTINTAS LONGITUDES).

ACTUALMENTE, SE HA IMPLEMENTADO LA VALIDACIÓN DE COORDENADAS, LA VISUALIZACIÓN DEL TABLERO Y LA ESTRUCTURA MODULAR DEL SISTEMA. EL SISTEMA ESTÁ PREPARADO PARA INCORPORAR MÁS ADELANTE MECÁNICAS COMO CARTAS DE PODER, MOVIMIENTO DE BARCOS Y DETECCIÓN DE CONDICIONES DE VICTORIA.

LA INTERFAZ TEXTUAL ES CLARA, ACCESIBLE Y FUNCIONAL, UTILIZANDO CARACTERES Y COLORES DE CONSOLA PARA FACILITAR LA INTERACCIÓN.

REQUERIMIENTOS DEL SISTEMA FUNCIONALES

Código	Descripción
BNS-F01	El sistema permitirá registrar a dos jugadores solicitando un nombre identificador para cada uno.
BNS-F02	El sistema mostrará un menú principal con las opciones: iniciar partida, leer instrucciones y salir del juego.
BNS-F03	El sistema asignará de forma aleatoria una carta de poder a cada jugador al inicio de su turno, determinando la acción disponible.
BNS-F04	El sistema permitirá ejecutar un ataque a coordenadas del tablero enemigo solo si la carta recibida lo permite.
BNS-F05	El sistema permitirá mover un barco de posición únicamente si la carta activa del turno habilita esta acción.
BNS-F06	El sistema activará automáticamente efectos especiales (como radar, escudo o ataque doble) cuando la carta correspondiente esté activa.
BNS-F07	El sistema dará un reporte de estado de cada barco, incluyendo impactos, hundimientos y eliminación del tablero al jugador.
BNS-F08	El sistema alternará los turnos entre jugadores, mostrando con claridad cuál es el jugador activo en cada ronda.
BNS-F09	El sistema evaluará condiciones de victoria, finalizando la partida cuando uno de los jugadores pierda toda su flota.
BNS-F10	El sistema mostrará mensajes de fin de juego e invitará al jugador a salir o reiniciar la partida.

Describen el comportamiento del sistema desde el punto de vista del usuario. Incluyen acciones como registrar jugadores, alternar turnos o usar cartas especiales.

REQUERIMIENTOS DEL SISTEMA NO FUNCIONALES

Código	Descripción
BNS-NF01	El sistema debe ejecutarse completamente en entorno de consola, sin requerir conexión a internet.
BNS-NF02	El sistema debe ofrecer una interfaz textual clara y legible, con menús visibles y representaciones comprensibles del tablero.
BNS-NF03	El tiempo de respuesta entre las acciones del usuario debe ser inferior a 2 segundos.
BNS-NF04	El sistema debe estar modularizado en archivos independientes, cada uno asociado a un componente del juego.
BNS-NF05	El sistema debe utilizar estructuras, matrices y punteros para representar barcos y tableros.
BNS-NF06	El sistema usará <u>Raylib</u> y fuentes personalizadas para mejorar la experiencia visual.
BNS-NF07	La lógica del sistema debe estar separada de la visualización textual, facilitando mantenimiento y escalabilidad.
BNS-NF08	El sistema debe ser portable y compatible con compiladores compatibles con <u>Raylib</u> .
BNS-NF09	El sistema debe notificar al usuario sobre entradas erróneas y ofrecer retroalimentación textual inmediata.

Establecen criterios de calidad y restricciones técnicas del sistema, como rendimiento, portabilidad, organización del código y claridad visual.

OBJETIVOS DEL PROGRAMA

OBJETIVO GENERAL

Desarrollar un prototipo modular de un juego de estrategia por turnos en consola basado en Batalla Naval, aplicando estructuras de datos, funciones y principios de codificación en lenguaje C.

OBJETIVOS ESPECÍFICOS

- Diseñar e implementar estructuras de datos que representen con claridad a los barcos, tableros y jugadores.
- Programar una lógica de turnos automatizada, que controle el flujo de juego y permita su extensión futura.
- Incorporar validación de coordenadas y restricciones de solapamiento.
- Aplicar buenas prácticas de codificación (uso de funciones, macros, modularidad, documentación).
- Fomentar el trabajo colaborativo, documentado en bitácoras y repositorios compartidos como GitHub.
- Lograr una interfaz textual clara y amigable mediante el uso de símbolos y colores.

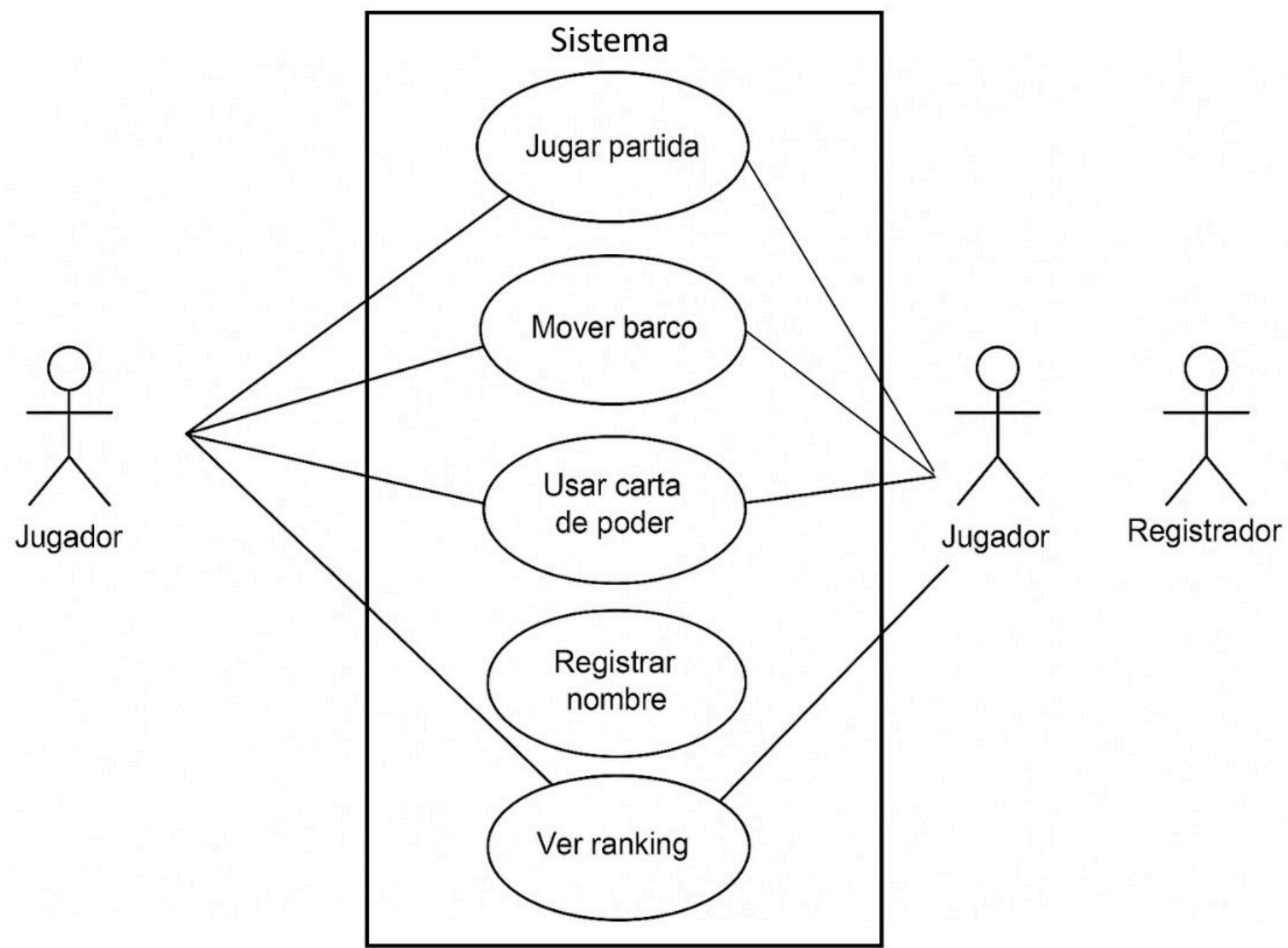


DIAGRAMA DE CASOS DE USOS

DIAGRAMA DE CASOS DE USOS

CASO DE USO 1: REGISTRAR JUGADOR

ACTOR PRINCIPAL: JUGADOR

DESCRIPCIÓN: EL JUGADOR INTRODUCE SU NOMBRE, EL CUAL SERÁ ALMACENADO Y UTILIZADO DURANTE LA PARTIDA.

PRECONDICIÓN: EL SISTEMA ESTÁ EN EJECUCIÓN.

FLUJO BÁSICO:

1. EL SISTEMA SOLICITA EL NOMBRE DEL JUGADOR.
2. EL JUGADOR INTRODUCE UN NOMBRE.
3. EL SISTEMA LO VALIDA (MÁXIMO 20 CARACTERES, SIN ESPACIOS INVÁLIDOS).
4. SE REGISTRA EL NOMBRE EN LA ESTRUCTURA CORRESPONDIENTE.

DIAGRAMA DE CASOS DE USOS

CASO DE USO 2: COLOCAR BARCOS

ACTOR PRINCIPAL: JUGADOR

DESCRIPCIÓN: EL JUGADOR POSICIONA BARCOS DE DIFERENTES TAMAÑOS EN SU TABLERO.

PRECONDICIÓN: EL JUGADOR YA FUE REGISTRADO.

FLUJO BÁSICO:

1. EL SISTEMA SOLICITA ORIENTACIÓN DEL BARCO.
2. EL JUGADOR INDICA COORDENADAS INICIAL Y FINAL.
3. EL SISTEMA VALIDA SI LAS COORDENADAS SON CORRECTAS Y NO HAY SOLAPAMIENTOS.
4. EL BARCO SE DIBUJA EN EL TABLERO CON LOS CARACTERES ☐ (PUNTA) Y ■ (CUERPO).
5. SE REPITE HASTA COMPLETAR LA FLOTA DEL JUGADOR.

ACTORES DEL SISTEMA

JUGADOR:

- Jugador:
Introduce su nombre.
- Coloca sus barcos en el tablero.
- Visualiza el tablero con símbolos.
- Interactúa mediante la consola.

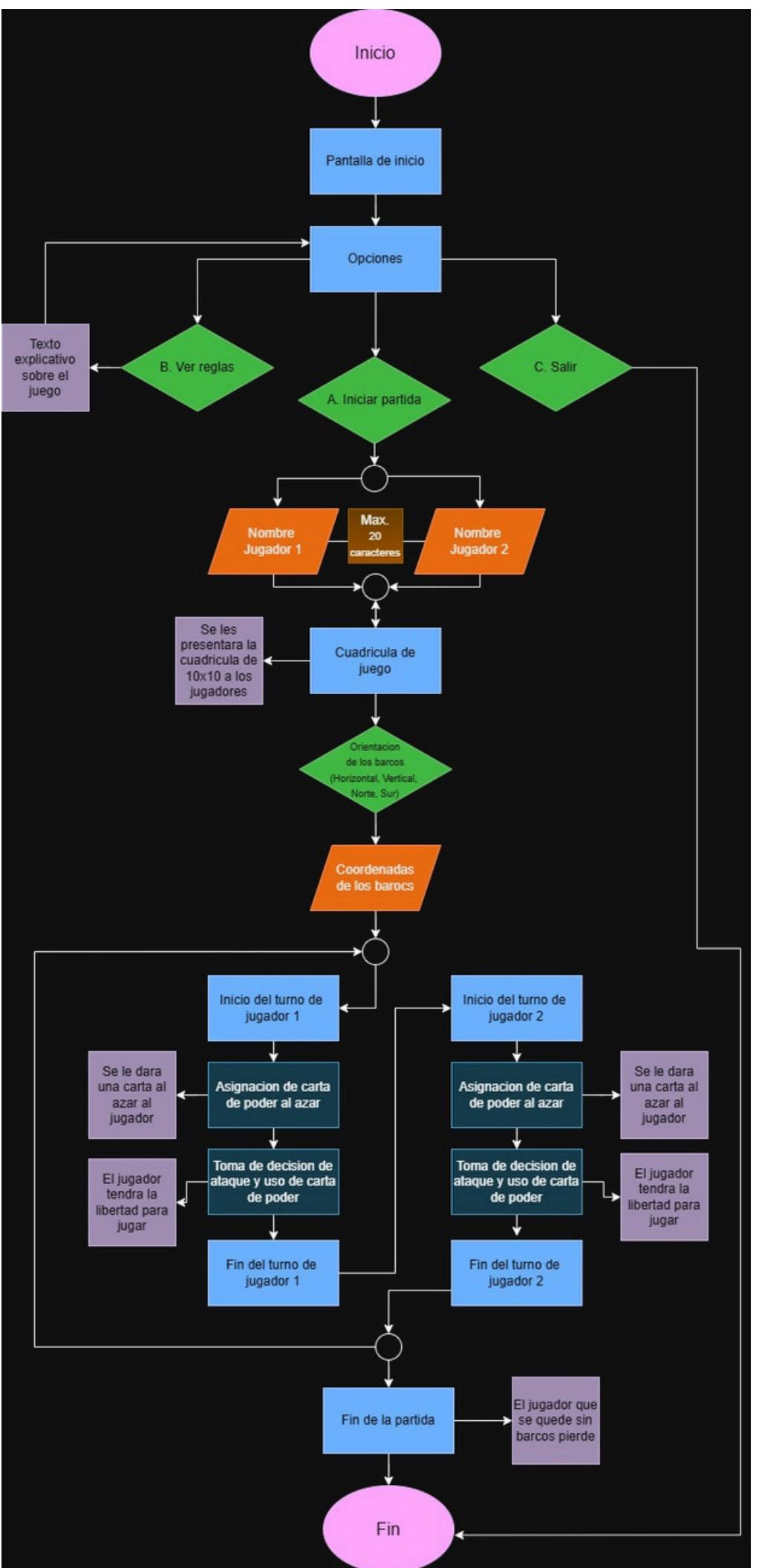
SISTEMA:

- Sistema:
 - Solicita y valida entradas.
 - Controla la estructura de cada barco y tablero.
 - Muestra retroalimentación visual.
 - Modulariza funcionalidades: colocación, validación, limpieza de pantalla, impresión del tablero

DIAGRAMA DE BLOQUES

El sistema fue diseñado de forma modular, con componentes que manejan el menú, el tablero, los turnos, la validación de jugadas y la detección de victoria.

Esta estructura permite mantener el código organizado, escalable y fácil de mantener.



Identificación	Descripción del requisito	Versión	Objetivo relacionado	Última fecha registrada
BNS-F01	El sistema permitirá registrar a dos jugadores solicitando un nombre identificador para cada uno.	1.0.0	OE8	12/05/2025
BNS-F02	El sistema mostrará un menú principal con las opciones: iniciar partida, leer instrucciones y salir del juego.	1.0.0	OE8	12/05/2025
BNS-F03	El sistema asignará de forma aleatoria una carta de poder a cada jugador al inicio de su turno.	1.0.0	OE4	12/05/2025
BNS-F04	El sistema permitirá ejecutar un ataque a coordenadas del tablero enemigo solo si la carta recibida lo permite.	1.0.0	OE5	12/05/2025
BNS-F05	El sistema permitirá mover un barco de posición solo si la carta activa del turno lo permite.	1.0.0	OE3	12/05/2025
BNS-F06	El sistema activará efectos especiales cuando la carta correspondiente esté activa.	1.0.0	OE4	12/05/2025
BNS-F07	El sistema gestionará el estado de cada barco, incluyendo impactos, hundimientos y eliminación del tablero.	1.0.0	OE1	12/05/2025
BNS-F08	El sistema alternará los turnos entre jugadores, mostrando cuál es el jugador activo en cada ronda.	1.0.0	OE2	12/05/2025
BNS-F09	El sistema evaluará condiciones de victoria, finalizando la partida cuando un jugador pierda toda su flota.	1.0.0	OE5	12/05/2025
BNS-F10	El sistema mostrará mensajes de fin de juego e invitará al	1.0.0	OE8	12/05/2025

MATRIZ DE REQUERIMIENTO

MATRIZ DE REQUERIMIENTO

	jugador a salir o reiniciar la partida.			
BNS-NF01	El sistema debe ejecutarse completamente en entorno de consola, sin requerir conexión a internet.	1.0.0	OE6	12/05/2025
BNS-NF02	La interfaz textual debe ser clara y legible, con menús visibles y representaciones comprensibles.	1.0.0	OE8	12/05/2025
BNS-NF03	El tiempo de respuesta entre acciones debe ser inferior a 2 segundos.	1.0.0	OE6	12/05/2025
BNS-NF04	El sistema debe estar modularizado en archivos independientes.	1.0.0	OE6	12/05/2025
BNS-NF05	Uso de estructuras, matrices y punteros para barcos y tableros.	1.0.0	OE1	12/05/2025
BNS-NF06	El sistema usará <u>Raylib</u> y fuentes personalizadas para mejorar la experiencia visual.	1.0.0	OE8	12/05/2025
BNS-NF07	Separación entre lógica y visualización para facilitar mantenimiento.	1.0.0	OE6	12/05/2025
BNS-NF08	El sistema debe ser portable y compatible con compiladores compatibles con <u>Raylib</u> .	1.0.0	OE6	12/05/2025
BNS-NF09	Retroalimentación inmediata ante entradas erróneas del usuario.	1.0.0	OE8	12/05/2025

ESTÁNDARES DE CODIFICACIÓN

- BASADO EN EL ESTÁNDAR ANSI C, GARANTIZANDO PORTABILIDAD Y CLARIDAD.
- USO DE MACROS (#DEFINE) PARA COLORES, CARACTERES VISUALES Y CONSTANTES CLAVE (BOARD_SIZE, SÍMBOLOS).
- FUNCIONES CON RESPONSABILIDAD ÚNICA, NOMBRES DESCRIPTIVOS Y REUTILIZABLES (IMPRIMIRTABLERO, PONERBARCOS).
- VARIABLES Y ESTRUCTURAS CON ESTILO COHERENTE: CAMELCASE PARA VARIABLES, MAYÚSCULAS PARA MACROS, Y SNAKE_CASE EN ESTRUCTURAS (SHIP_STATUS).
- COMENTARIOS ESTRUCTURADOS EN TRES PARTES:
 - // ENTRADA: DATOS RECIBIDOS
 - // PROCESO: LÓGICA INTERNA
 - // SALIDA: RESULTADO O IMPACTO
- MODULARIZACIÓN EN .C Y .H PARA SEPARAR LÓGICA Y DEFINICIONES.
- VALIDACIÓN DE ENTRADAS, CONTROL DE ERRORES Y CÓDIGO LIMPIO.
- PLANEACIÓN FUTURA: USO DE RAYLIB PARA INTERFAZ VISUAL SIN PERDER LA BASE ESTRUCTURADA.

PROCESO DE DESARROLLO

División de módulos y responsabilidades

Módulo	Funcionalidad Principal	Responsable
Menú e inicio de partida	Captura de nombres, inicio del juego	Nicolás Canul
Tablero y colocación	Impresión de matriz, validación de coordenadas, colocación de barcos	Alejandro Flores
Turnos y flujo del juego	Alternancia de jugadores, estructura inicial de partida	Jesús León
Documentación y estandarización	Comentarios, estilo de codificación, revisión de estructuras	Isaac Ortiz
Funciones generales y limpieza	Funciones de impresión, limpieza de consola, interacción básica	Todos

PROCESO DE DESARROLLO

Bitácora Semanal de Actividades

Semana	Nicolás Canul	Alejandro Flores	Jesús León	Isaac Ortiz
1	Diseño y estructura del menú	Diseño base del tablero	Estructura de turnos y juego	Organización de archivos y estructuras
2	Captura de nombres y entrada de datos	Colocación de barcos y validación de solapamiento	Inicio de turnos y verificación básica	Modularización del código y limpieza
3	Pruebas de flujo del menú	Revisión de impresión de tablero	Enlace entre menú y tablero	Comentarios y revisión del estándar
4	Revisión final y documentación	Ajustes de tablero	Apoyo en integración	Documentación del estándar de codificación

PROCESO DE DESARROLLO

Bitácora Semanal de Actividades

Semana	Nicolás Canul	Alejandro Flores	Jesús León	Isaac Ortiz
1	Diseño y estructura del menú	Diseño base del tablero	Estructura de turnos y juego	Organización de archivos y estructuras
2	Captura de nombres y entrada de datos	Colocación de barcos y validación de solapamiento	Inicio de turnos y verificación básica	Modularización del código y limpieza
3	Pruebas de flujo del menú	Revisión de impresión de tablero	Enlace entre menú y tablero	Comentarios y revisión del estándar
4	Revisión final y documentación	Ajustes de tablero	Apoyo en integración	Documentación del estándar de codificación

PROCESO DE DESARROLLO

Integrante	Rol Principal	% de Contribución Estimada
Nicolás Canul	Menú, entrada de datos	25%
Alejandro Flores	Tablero, colocación de barcos	30%
Jesús León	Turnos y flujo del juego	20%
Isaac Ortiz	Documentación, estructura modular	25%

projecto_prog_estructurada Public

Watch 1

main 1 Branch 0 Tags

Go to file Add file Code

Ozia112 Merge branch 'main' of https://github.com/Ozia112/proyecto_prog_estr... e9a4527 · 15 hours ago 16 Commits

Propuestas	Update propuesta_3.md	last week
codigo_fuente	Reestructuracion de variables a variables tipo struct para alm...	15 hours ago
informacion_administrativa	Creacion de apartado: Informacion administrativa	2 weeks ago
.gitignore	Modificaciones del codigo base y demás	18 hours ago
README.md	Update README.md	10 hours ago

README

Propuestas de proyecto

Las siguientes propuestas están pendientes a revisión y se requiere un mayor análisis de planteamiento para cada uno. Cuando sea decidido cuál se usará se emplearán todos los recursos en el proyecto elegido.

Propuestas:

- [Propuesta 1 \(Battleship\)](#)
- [Propuesta 2 \(Trivia\)](#)
- [Propuesta 3 \(Datos\)](#)

Proyecto elegido:

Battleship

REPOSITORIO DEL PROYECTO

GITHUB OFICIAL DEL EQUIPO “LOS CODICIOSOS”:

https://github.com/Ozia112/proyecto_prog_estructurada

En este repositorio encontrarás todo el código fuente del sistema, incluyendo los archivos .c, .h, documentación, diagramas y bitácoras de avance.

Fue gestionado de manera colaborativa por todos los integrantes a través de ramas y commits organizados, asegurando control de versiones y trazabilidad del desarrollo.

iGracias!