BATALLA NAVAL

Proyecto Final: Programación Estructurada

Jesús Fernando Castro Hernandez

Mateo Emanuel Martín Tec

Luis Edward Chay Ascorra

Tecnológico de Software | 02/12/2025

DESCRIPCIÓN Y DINÁMICA

EL TABLERO

El juego se desarrolla en una matriz de dimensiones `8x8`.

Las filas están etiquetadas de la **A a la H** y las columnas del **1 al 8**.

El objetivo es localizar y destruir la flota enemiga (CPU) mediante coordenadas.

SIMBOLOGÍA

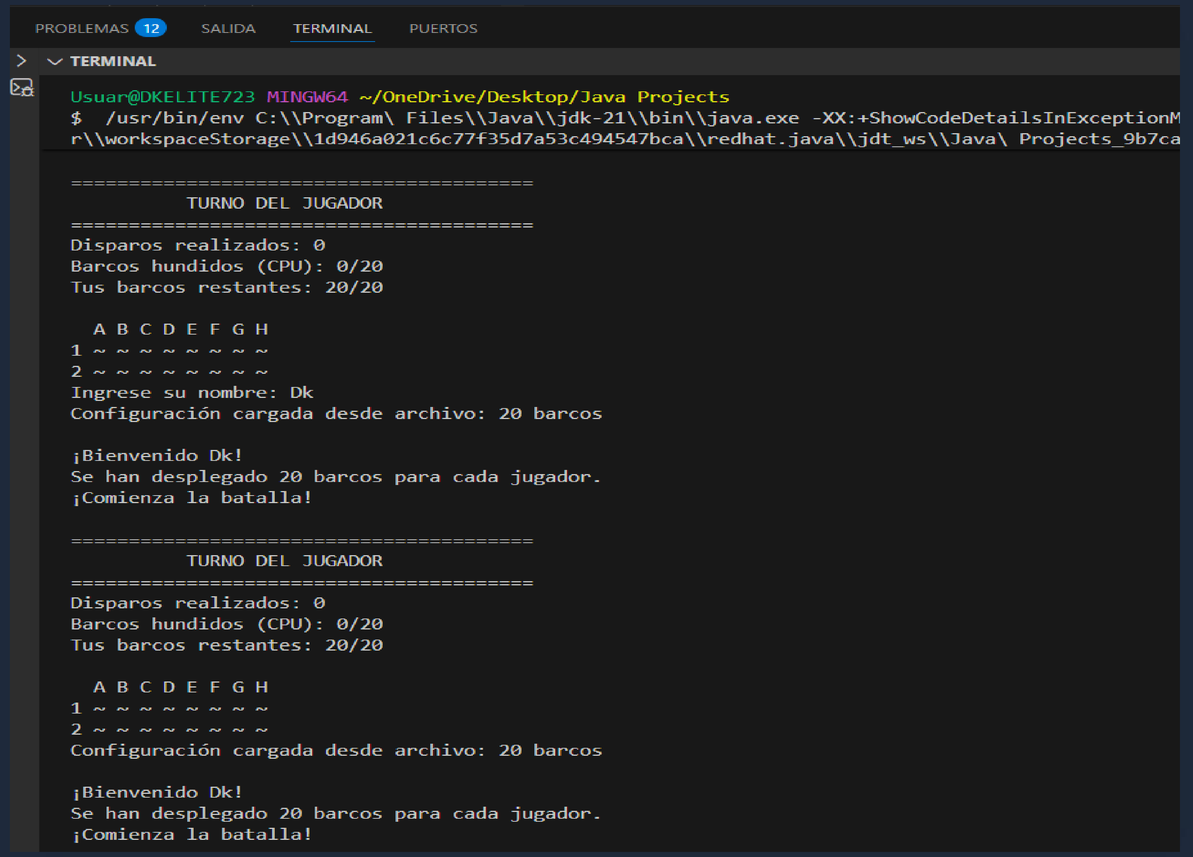
- `"~"` : Agua (Casilla desconocida)
- `"0"` : Agua Disparada (Fallo)
- `"X"` : Barco Impactado (Acierto)

"Estrategia por turnos en consola."

EJECUCIÓN DEL PROGRAMA

FLUJO DE JUEGO

- **Inicio:** Se generan los tableros y se colocan barcos aleatoriamente.
- **Turno Jugador:** Entrada por teclado (Ej: `"A5"`).
- **Validación:** El sistema verifica si es agua o barco.
- **Turno CPU:** La computadora dispara a una coordenada aleatoria válida.
- **Fin:** Gana quien hunda todos los barcos enemigos.



ESTRUCTURA DEL CÓDIGO

MATRICES (TABLEROS)

Dos matrices bidimensionales almacenan el estado del juego.

```
int tableroJugador[8][8]
```

```
int tableroCPU[8][8]
```

ARREGLOS AUXILIARES

Se utilizan arreglos unidimensionales temporales para convertir coordenadas.

Ej: `indices[0]` para fila, `indices[1]` para columna.

FUNCIONES JAVA

Uso de métodos nativos para la lógica:

`Math.random()` (IA)

`String.charAt()` (Entrada)

MÉTODOS PROPIOS

- **`procesarDisparo()`**
  Evalúa si la coordenada es agua o barco, actualiza la matriz e imprime el resultado ("Impacto" o "Agua").
- **`turnoCPU()`**
  Genera coordenadas aleatorias, valida que no se repitan disparos y ejecuta el ataque contra el jugador.
- **`convertirCoordenada()`**
  Traduce `"C4"` a índices de matriz `[2, 3]`.

```
while (!disparoValido) {
    System.out.print(s: "\nIngrese coordenada para disparar (ejemplo: A5, C7): ");
    String coordenada = scanner.nextLine().trim().toUpperCase();

    if (validarCoordenada(coordenada)) {
        int[] indices = convertirCoordenada(coordenada);
        int fila = indices[0];
        int columna = indices[1];

        if (tableroCPU[fila][columna] == aguaDisparada ||
                tableroCPU[fila][columna] == barcoImpactado) {
            System.out.println(x: "¡Ya disparaste a esta posición! Intenta otra coordenada.");
        } else {
            disparoValido = true;
            procesarDisparo(tableroCPU, fila, columna, esJugador: true);
            disparosRealizados++;
            // Verificar si el Jugador ganó
            if (barcosHundidosCPU == totalBarcos) {
                System.out.println("\n¡FELICIDADES " + nombreJugador + "! ¡HAS GANADO LA BATALLA!");
                juegoTerminado = true;
                guardarResultados(ganoJugador: true);
            }
        }
    } else {
        System.out.println(x: "Coordenada inválida. Use formato letra+número (A-H, 1-8). Ejemplo: A5");
    }
```

PERSISTENCIA DE DATOS

LECTURA DE ARCHIVOS

El programa inicia leyendo `barcos_config.txt`. Este archivo define la cantidad de barcos a colocar, permitiendo configuración dinámica sin recompilar.

ESCRITURA DE RESULTADOS

Al finalizar, se usa `FileWriter` en modo *append* para escribir en `battleship_scores.txt`.

Se guardan: Nombre, Barcos Hundidos, Disparos y Resultado (Victoria/Derrota).

RETO TÉCNICO: COORDENADAS

CONVERSIÓN ASCII

El mayor desafío fue traducir la entrada humana (ej. "C5") a índices numéricos para las matrices.

Solución: Manipulación de caracteres ASCII.

```
char letra = 'C';
int fila = letra - 'A'; // 2
int col = '5' - '1'; // 4
```

CONCLUSIONES

"La experiencia de construir un juego completo desde cero nos dio confianza en nuestras habilidades y consolidó el uso de matrices, modularidad y manejo de archivos en Java."

</> LÓGICA

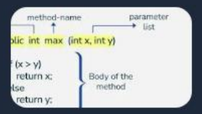
MODULARIDAD

PERSISTENCIA

IMAGE SOURCES

https://i.sstatic.net/sPXDQ.png

Source: unix.stackexchange.com



https://media.geeksforgeeks.org/wp-content/uploads/20240717150503/1.webp

Source: www.geeksforgeeks.org

https://media.geeksforgeeks.org/wp-content/uploads/20240514115348/Logical-Data-Flow-Diagram-(DFD).webp

Source: www.geeksforgeeks.org

https://i.ytimg.com/vi/ZGbVLKGUAnM/hq720.jpg?sqp=-oaymwEhCK4FEIIDSFryq4qpAxMIARUAAAAAGAElAADIQj0AgKJD&rs=AOn4CLCO2hKsVWRbogG8tpxY8b6vusSvLg

Source: www.youtube.com

https://media.istockphoto.com/id/959732362/vector/military-blue-radar-screen-with-target-futuristic-hud-interface-stock-vector-illustration.jpg?s=612x612&w=0&k=20&c=iUu5Riq4Y4iW5Og2BDPPzXZ_0NhM46DIUK3rZMjyVIA=

Source: www.istockphoto.com