UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE PROGRAMACIÓN DE COMPUTADORAS
LICENCIATURA EN CIBERSEGURIDAD
PROYECTO FINAL

Asignatura

Programación I

Proyecto Semestral

Integrantes

Omar Polanco
8-1014-120

Andrea Avila
8-1010-2209

Henzel Menal
8-1011-2

Valentina Orozco
20-14-7870

Profesor

Paulo Picota

II Semestre

Programación I

26 de noviembre de 2023

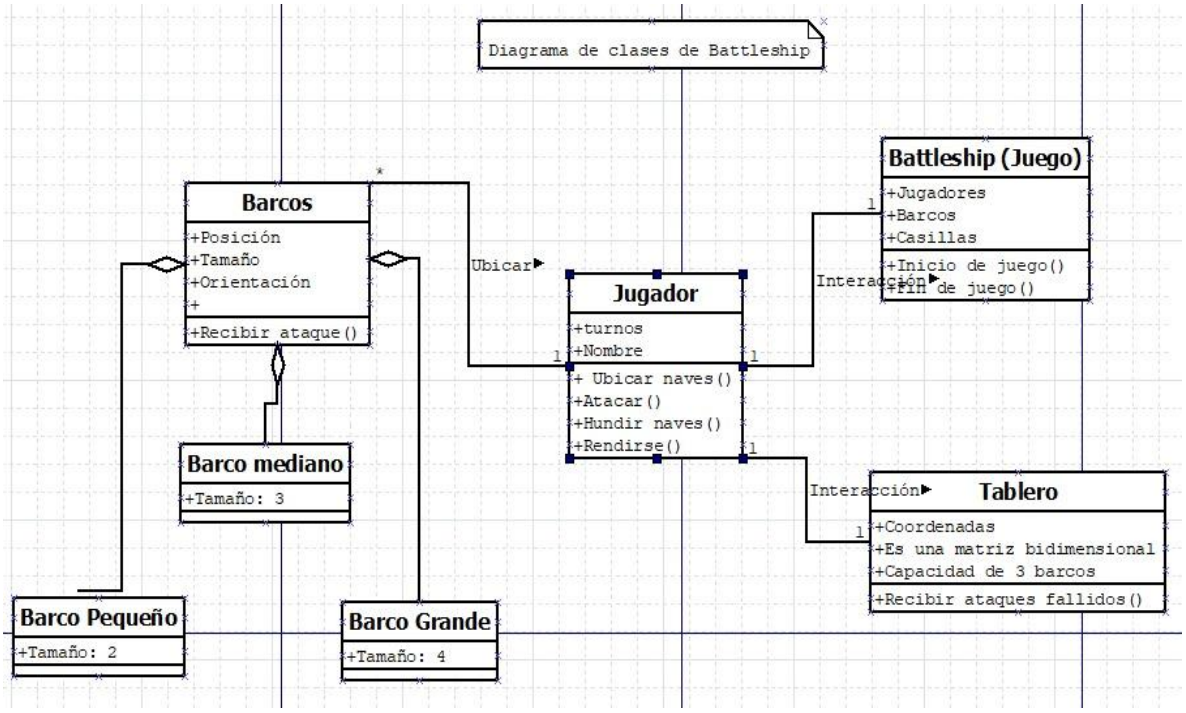
Índice

Introducción

En el siguiente trabajo explicaremos nuestro código, el cual intenta recrear el clásico juego de batalla naval "Battleship", un juego estratégico de dos jugadores. Esto, con las herramientas, elementos y materiales proporcionados a través del semestre académico.

Nuestro trabajo consiste en programar dicho juego en el lenguaje de programación: Java. Esto pone a prueba nuestras habilidades. No solo en aplicar los conocimientos adquiridos en programación, sino que también en lógica y abstracción, al tener que programar un juego debimos entenderlo por completo, sus reglas, parámetros, complejidad y estrategias, y así también amoldar toda esta lógica y abstracción a las instrucciones y peticiones dadas por nuestro profesor. Para así poder recrearlo de la mejor manera posible y brindarles la divertida experiencia de jugar este juego.

Diagrama de clases



Listado de clases

BattleshipMain
Esta es la clase main. Aquí el usuario ingresa todos los datos necesarios para poder jugar, llamando a las otras clases y métodos para así ejecutar todas las declaraciones y lógica dentro de estas de acuerdo con el juego.
Barcos
Esta clase en palabra simples se encarga de la ubicación de los barcos, mediante un método llamado barcos y utilizando condicionales para la orientación de los barcos, colocando la letra “O” en las coordenadas dadas por el jugador y así devolviendo una de las matrices con la ubicación de cada barco.
Boardload
Esta clase se encarga de cargar dos matrices 10x10 tipo String mediante un método llamado bload, con un contador para nombrar la primera fila con letras y columna con números, indicando los puntos de las coordenadas, así dejando una matriz 9x9 como el tablero sobre el que se jugara.
Attackboard
Esta clase se encarga de los ataques hacia el contrincante, mediante un método llamado Battack el cual utiliza parámetros como turno, fila y columna del ataque, al ingresar la coordenada que se desea atacar comprueba si hay un barco en dicha posición y lo marca con una “X” de lo contrario lo marcará con un “0” y devolverá una de las matrices con dicho ataque.

Lista de métodos

barcos
Este método se encarga se colocar los barcos en el tablero, siguiendo parámetros como tamaño del barco, orientación y coordenadas, estas ultimas las almacena en los arreglos fila y columna, crea una nueva matriz llamada ship y la devuelve con la ubicación de los barcos.
bload
Este método se encarga de iniciar y cargar una matriz que representa el tablero del juego, utiliza dos contadores para nombrar las filas y columnas de este, al imprimirla solo en la primer fila y columna se muestran el numero o letra que le corresponde y rellena lo que resta con el símbolo “~”.
battack
Este método se encarga de gestionar los ataques hechos por el jugador, tomando parámetros como las coordenadas, el turno y las matrices que representan el tablero de juego para cada jugador, también se encarga de gestionar los ataques y actualizar el tablero.

Listado de variables

String usu1 y String usu2
Estas variables se encargan de almacenar los nombres de los jugadores.
int tam
Esta variable se utiliza para almacenar el tamaño de los barcos que serán colocados en el tablero por los jugadores.
int selec
Esta variable se utiliza para gestionar el número de barcos seleccionado por cada jugador.
int ship2, int ship3, e int ship4
Esta variable le da seguimiento a la cantidad de barcos por tamaño que cada jugador coloque en el tablero.
int turno = 0;
Esta variable les da seguimiento a los turnos, así alternando estos entre cada jugador.
int atinar1 e int atinar2
Esta variable da seguimiento al numero de aciertos de cada jugador, esto ayuda a determinar el ganador bajo las condiciones dadas.
int lost
Este variable determina si algún jugador decide rendirse.
String ori
Esta variable determina la orientación de los barcos, ya sea vertical u horizontal.
String fila y String columna
Estas variables se utilizan para almacenar y gestionar las coordenadas dadas por los jugadores en diferentes partes del juego
String prueba

Esta variable se encarga de gestionar las coordenadas ya dadas al colocar los barcos, así si un jugador la repite se generará un error
String ataque
Esta variable almacena las coordenadas dadas por el jugador a la hora de atacar, su una de estas se repite se generará un error.
String [] resul1 y String [] resul2
Estas variables almacenan la información de los barcos, como tamaño, coordenadas y orientación.

Retos y dificultades

El reto inicial que más nos costó solucionar fue la lógica del programa, por ejemplo, cuantas clases se crearían o si necesitábamos especificarle al programa si los barcos fuesen horizontales o verticales (cosa que terminó siendo el caso). Además de esto, la lógica detrás de especificar las posiciones de las coordenadas mediante las matrices se nos llegó a complicar bastante. al inicio, la mayoría de las veces salía algo que no queríamos, pero eventualmente logramos que funcionara correctamente.

Tardamos bastante pensando en la abstracción de cómo hacer ciertos métodos dentro del programa, especialmente en la captura de errores. Probamos múltiples veces el código con todas las opciones incorrectas para detectar todas las excepciones dentro de la ejecución para una detectadas pudiéramos implementar un mensaje de error al código para que no se interrumpiera la ejecución.

Otro inconveniente particularmente molesto tuvo que ver con el hardware en vez del software. Algunos integrantes del grupo no pudieron correr el código en sus máquinas por razones que aún desconocemos. Este inconveniente nos afecta aún más ya que al estar trabajando desde casa no se tiene acceso a otras máquinas como pasaba en el centro universitario.

Para finalizar, el elemento en el que más dificultades tuvimos fue en la impresión de las matrices en la ejecución del juego. Las matrices no imprimían lo que queríamos que imprimieran específicamente, por ejemplo, a la hora de colocar los navíos, se podía repetir las coordenadas ya puestas, lo cual implicaba que los navíos se solapaban uno al otro, cosa que fue particularmente difícil de solucionar.

Conclusión

Logramos aplicar todos los conocimientos adquiridos durante el curso y referencia externa fueron los medios que utilizamos para lograr crear este código que adapta el juego de mesa “Battleship” a el lenguaje de programación Java. A pesar de no podernos reunir en persona, mediante reuniones en líneas se logró organizar cada faceta del trabajo.

La mayoría de nuestros esfuerzos fueron en la creación del código, más concretamente el main, el cual es la parte más compleja de nuestro trabajo. Aplicamos todo el contenido de los módulos como el de arreglos en java, que nos fue particularmente útil ya que nuestro programa involucra matrices. También, aplicamos contenido del módulo de diagramas de clases ya que se incluye un diagrama sobre el juego en este trabajo escrito.

Por último, La adaptación de un juego de mesa clásico a un entorno de programación no solo fue un desafío técnico, sino también una oportunidad para aplicar metodologías de diseño y trabajo en equipo. La experiencia de enfrentar problemas, discutir soluciones y tomar decisiones conjuntas durante las reuniones en línea contribuyó significativamente al desarrollo de habilidades colaborativas y a la resolución de problemas.

Bibliografía

1. AyeshaShaukat, «Project-Battle-Ships-Game/BattleShips.java at Master · AyeshaShaukat/Project-Battle-Ships-Game», *GitHub*. https://github.com/AyeshaShaukat/Project-Battle-Ships-Game/blob/master/BattleShips.java
2. abel balarezo, «JUEGO BATALLA NAVAL EN JAVA (IDE: NETBEANS)», *YouTube*. 12 de julio de 2017. [En línea]. Disponible en: https://www.youtube.com/watch?v=CywlIfNyLiw

Anexo

Código Fuente

```
import java.io.*;
import java.util.Set;
import java.util.HashSet;

public class BattleshipMain {
    public static void main(String[] args) throws IOException   {

        BufferedReader shinxlector = new BufferedReader(new InputStreamReader(System.in));
        Boardload dratini = new Boardload();
        Barcos vaporeon = new Barcos();
        Attackboard pancham = new Attackboard();

        String usu1 = "";
        String usu2 = "";
        int tam = 0;
        int selec = 0;
        int ship2 = 0;
        int ship3 = 0;
        int ship4 = 0;
        int turno = 0;
        int atinar1 = 0;
        int atinar2 = 0;
        int lost = 0;
        String ori = "";
        String fila = "";
        String columna = "";
        String prueba = "";
        String ataque = "";
        String [] vec1;
        String [] vec2;
        String [] resul1 = new String [4];
        String [] resul2 = new String [4];
        Set<String> santa1 = new HashSet<>();
        Set<String> santa2 = new HashSet<>();
        Set<String> gift1 = new HashSet<>();
        Set<String> gift2 = new HashSet<>();

        System.out.println("Bienvenido a nuestro increible juego de BattleShip!!!");
        System.out.println();
```

```
        System.out.print("Jugador 1. Ingrese su nombre de usuario: ");
        usu1 = shinxlector.readLine();
        System.out.println();
        System.out.print("Jugador 2. Ingrese su nombre de usuario: ");
        usu2 = shinxlector.readLine();
        System.out.println();
        System.out.println("Aqui están tus 2 tableros " + usu1 + " con los que jugarás. El
       tablero de la izquierda será para tus ataques y el de la derecha será para colocar
       los barcos.");
        System.out.println();
        String [][] mat1 = dratini.bload();
        String [][] mat2 = dratini.bload();
        String [][] mat4 = dratini.bload();

        for (int f = 0; f < 10; f++)    {
            for (int c = 0; c < 10; c++)    {
                System.out.print(mat1 [f][c]);
            }
            System.out.print("   ");
            for (int c = 0; c < 10; c++)    {
                System.out.print(mat2 [f][c]);
            }
            System.out.println();
        }
        System.out.println();
        while (selec < 4)    {
        System.out.println(usu1 +"! Empecemos a ubicar tus barcos. ¿Qué barco deseas poner
       primero? (4, 3, 2)");
        do{
            try{
            tam = Integer.parseInt(shinxlector.readLine());
            if (tam < 2 || tam > 4){
                System.out.println("Error. Ingrese solo los numeros 4, 3 y 2");

             }
            }catch (Exception e){
                System.out.println("Error. Ingrese solo los numeros 4, 3 y 2");

            }

       } while (tam < 2 || tam > 4);//fin do while

        /*Hacerlo a tu manera*/while (tam == 2 && ship2 > 1 || tam == 3 && ship3 > 0 ||
       tam == 4 && ship4 > 0)    {
```

```
        System.out.print("Este barco ya alcanzo la cantidad maxima de unidades a
    colocar. Por favor ingrese un nuevo tamaño de barco: ");
        tam = Integer.parseInt(shinxlector.readLine());
    }

    if (tam == 2)   {
        ship2 = ship2 + 1;
    }   else if (tam == 3)  {
        ship3 = ship3 + 1;
    }   else    {
        ship4 = ship4 + 1;
    }

    vec1 = new String[tam];
    vec2 = new String[tam];

    System.out.println("Ahora, escoge en qué dirección deseas colocar tu nave, ya sea
    horizontal (H) o vertical (V): ");
    ori = shinxlector.readLine();
    /*Hacerlo a tu maanera*/while (!ori.equalsIgnoreCase("H") &&
    !ori.equalsIgnoreCase("V"))  {
       System.out.println("Error. Solo puede ser 'V' o 'H'");
       ori=shinxlector.readLine();
    }

    if (ori.equalsIgnoreCase("H"))   {
    System.out.println("Muy bien, ahora escoge las coordenadas");
    System.out.println("Primero escoge la fila: ");
    fila = shinxlector.readLine();
    /*Hacerlo a tu manera*/while (fila.compareTo("I") > 0 || fila.compareTo("A") <
    0)  {
        System.out.println("Error. La letra debe estar dentro del rango de las filas,
    intenta otra vez: ");
        fila = shinxlector.readLine();
    }
    for (int a = 0; a < tam; a++)    {
        System.out.println("Ahora, escoge las coordenadas de las columnas: ");
        vec1 [a] = shinxlector.readLine();
        /*Hacerlo a tu manera*/while (Integer.parseInt(vec1[a]) > 9 ||
    Integer.parseInt(vec1[a]) < 1)  {
            System.out.println("Error. La cifra ingresada no concuerda con los valores
    de las columnas. Por favo ingresa de nuevo un valor: ");
            vec1 [a] = shinxlector.readLine();
        }
        prueba = fila + vec1 [a];
```

```
        while (santa1.contains(prueba) || Integer.parseInt(vec1[a]) > 9 ||
    Integer.parseInt(vec1[a]) < 1)    {
            System.out.println("Error. Coordenada repetida, ingresa una nueva");
            vec1 [a] = shinxlector.readLine();
            prueba = fila + vec1 [a];
        }
        santa1.add(prueba);
        resul1 [selec] = "Barco de tamaño " + tam + ", coordenada inicial es " + fila
    + vec1 [a] + " en " + ori;
     }

     }   else if (ori.equalsIgnoreCase("V"))  {
        System.out.println("Muy bien, ahora escoge las coordenadas");
     System.out.println("Primero escoge la columna: ");
     columna = shinxlector.readLine()
;
     /*Hacerlo a tu manera*/while (Integer.parseInt(columna) > 9 ||
    Integer.parseInt(columna) < 1)  {
        System.out.println("Error. El numero debe estar dentro del rango de las
    columnas, intenta otra vez: ");
        columna = shinxlector.readLine();
     }
     for (int a = 0; a < tam; a++)    {
        System.out.println("Ahora, escoge las coordenadas de las filas: ");
        vec2 [a] = shinxlector.readLine();
        /*Hacerlo a tu manera*/while (vec2 [a].compareTo("I") > 0 || vec2
    [a].compareTo("A") < 0)  {
            System.out.println("Error. La letra ingresada no concuerda con los valores
    de las filas. Por favo ingresa de nuevo un valor: ");
            vec2 [a] = shinxlector.readLine();
        }
        prueba = vec2 [a] + columna;


        while (santa1.contains(prueba) || vec2 [a].compareTo("I") > 0 || vec2
    [a].compareTo("A") < 0)    {
            System.out.println("Error. Coordenada repetida, ingresa una nueva");
            vec1 [a] = shinxlector.readLine();
            prueba = vec2 [a] + columna;
        }
        santa1.add(prueba);
        resul1 [selec] = "Barco de tamaño " + tam + ", coordenada inicial es " + vec2
    [a] + columna + " en " + ori;
```

```
        }

        }
    String [][] matbarcos1 =  vaporeon.barcos(tam, ori, fila, columna, vec1, vec2, mat2,
        mat4);

    for (int f = 0; f < 10; f++)  {
        for (int c = 0; c < 10; c++)  {
            if (matbarcos1[f][c].equals(" O "))     {
                mat2[f][c] = " O ";
            }
        }
    }
    for (int f = 0; f < 10; f++)  {
        for (int c = 0; c < 10; c++)    {
            System.out.print(mat1 [f][c]);
        }
        System.out.print("   ");
        for (int c = 0; c < 10; c++)    {
            System.out.print(mat2 [f][c]);
        }
        System.out.println();
    }
    selec = selec + 1;
        }//Fin while principal
    selec = selec * 0;
    tam = tam * 0;
    ship2 = ship2 * 0;
    ship3 = ship3 * 0;
    ship4 = ship4 * 0;

    System.out.print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
        \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");

        System.out.println();
        System.out.println("Aqui están tus 2 tableros " + usu2 + " con los que jugarás. El
        tablero de la izquierda será para tus ataques y el de la derecha será para colocar
        los barcos.");
        System.out.println();
        String [][] mat3 = dratini.bload();
        /*String [][]*/ mat4 = dratini.bload();

        for (int f = 0; f < 10; f++)    {
            for (int c = 0; c < 10; c++)    {
                System.out.print(mat3 [f][c]);
```

```
            }
            System.out.print("   ");
            for (int c = 0; c < 10; c++)    {
                System.out.print(mat4 [f][c]);
            }
            System.out.println();
        }
        System.out.println();
    while (selec < 4)    {
        System.out.println(usu2 +"! Empecemos a ubicar tus barcos. ¿Qué barco deseas poner
        primero? (4, 3, 2)");
        do{
            try{
            tam = Integer.parseInt(shinxlector.readLine());
            if (tam < 2 || tam > 4){
                System.out.println("Error. Ingrese solo los numeros 4, 3 y 2");

             }
            }catch (Exception e){
                System.out.println("Error. Ingrese solo los numeros 4, 3 y 2");

            }

        } while (tam < 2 || tam > 4);//fin do while

        /*Hacerlo a tu manera*/while (tam == 2 && ship2 > 1 || tam == 3 && ship3 > 0 ||
        tam == 4 && ship4 > 0)    {
            System.out.print("Este barco ya alcanzo la cantidad maxima de unidades a
        colocar. Por fvor ingrese un nuevo tamaño de barco: ");
            tam = Integer.parseInt(shinxlector.readLine());
        }

        if (tam == 2)   {
            ship2 = ship2 + 1;
        }   else if (tam == 3)  {
            ship3 = ship3 + 1;
        }   else    {
            ship4 = ship4 + 1;
        }

        vec1 = new String[tam];
        vec2 = new String[tam];

        System.out.println("Ahora, escoge en qué dirección deseas colocar tu nave, ya sea
        horizontal (H) o vertical (V): ");
```

```
        ori = shinxlector.readLine();
        /*Hacerlo a tu maanera*/while (!ori.equalsIgnoreCase("H") &&
       !ori.equalsIgnoreCase("V"))  {
           System.out.println("Error. Solo puede ser 'V' o 'H'");
           ori=shinxlector.readLine();
        }

        if (ori.equalsIgnoreCase("H"))   {
        System.out.println("Muy bien, ahora escoge las coordenadas");
        System.out.println("Primero escoge la fila: ");
        fila = shinxlector.readLine();
        /*Hacerlo a tu manera*/while (fila.compareTo("I") > 0 || fila.compareTo("A") <
       0)  {
            System.out.println("Error. La letra debe estar dentro del rango de las filas,
       intenta otra vez: ");
            fila = shinxlector.readLine();
        }
        for (int a = 0; a < tam; a++)    {
            System.out.println("Ahora, escoge las coordenadas de las columnas: ");
            vec1 [a] = shinxlector.readLine();
            /*Hacerlo a tu manera*/while (Integer.parseInt(vec1[a]) > 9 ||
       Integer.parseInt(vec1[a]) < 1)  {
                System.out.println("Error. La cifra ingresada no concuerda con los valores
       de las columnas. Por favo ingresa de nuevo un valor: ");
                vec1 [a] = shinxlector.readLine();
            }
            prueba = fila + vec1 [a];


            while (santa2.contains(prueba) || Integer.parseInt(vec1[a]) > 9 ||
       Integer.parseInt(vec1[a]) < 1)    {
                System.out.println("Error. Coordenada repetida, ingresa una nueva");
                vec1 [a] = shinxlector.readLine();
                prueba = fila + vec1 [a];
            }
            santa2.add(prueba);
            resul2 [selec] = "Barco de tamaño " + tam + ", coordenada inicial es " + fila
       + vec1 [a] + " en " + ori;
        }

        }   else if (ori.equalsIgnoreCase("V"))  {
            System.out.println("Muy bien, ahora escoge las coordenadas");
        System.out.println("Primero escoge la columna: ");
        columna = shinxlector.readLine();
```

```
        /*Hacerlo a tu manera*/while (Integer.parseInt(columna) > 9 ||
    Integer.parseInt(columna) < 1)  {
            System.out.println("Error. El numero debe estar dentro del rango de las
    columnas, intenta otra vez: ");
            columna = shinxlector.readLine();
        }
        for (int a = 0; a < tam; a++)    {
            System.out.println("Ahora, escoge las coordenadas de las filas: ");
            vec2 [a] = shinxlector.readLine();
            /*Hacerlo a tu manera*/while (vec2 [a].compareTo("I") > 0 || vec2
    [a].compareTo("A") < 0)  {
                System.out.println("Error. La letra ingresada no concuerda con los valores
    de las filas. Por favo ingresa de nuevo un valor: ");
                vec2 [a] = shinxlector.readLine();
            }
            prueba = vec2 [a] + columna;


            while (santa2.contains(prueba) || vec2 [a].compareTo("I") > 0 || vec2
    [a].compareTo("A") < 0)    {
                System.out.println("Error. Coordenada repetida, ingresa una nueva");
                vec2 [a] = shinxlector.readLine();
                prueba = vec2 [a] + columna;
            }
            santa2.add(prueba);
            resul2 [selec] = "Barco de tamaño " + tam + ", coordenada inicial es " + vec2
    [a] + columna + " en " + ori;
        }
        }
    String [][] matbarcos2 =  vaporeon.barcos(tam, ori, fila, columna, vec1, vec2, mat2,
        mat4);

    for (int f = 0; f < 10; f++)  {
        for (int c = 0; c < 10; c++)  {
            if (matbarcos2[f][c].equals(" O "))     {
                mat4[f][c] = " O ";
            }
        }
    }
    for (int f = 0; f < 10; f++)  {
        for (int c = 0; c < 10; c++)    {
            System.out.print(mat3 [f][c]);
        }
        System.out.print("   ");
        for (int c = 0; c < 10; c++)    {
```

```
            System.out.print(mat4 [f][c]);
        }
        System.out.println();
    }
    selec = selec + 1;
        }//Fin while principal

        selec = selec * 0;
    tam = tam * 0;
    ship2 = ship2 * 0;
    ship3 = ship3 * 0;
    ship4 = ship4 * 0;

    System.out.print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
       \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");

    System.out.println("Bien! Que comience la batalla!!!");

    while ((atinar1 != 11) && (atinar2 != 11))      {
        if (turno % 2 == 0)     {
    System.out.println("Es el turno de " + usu1 + ". Elije dónde vas a atacar: ");
    System.out.print("Escoge primero el valor de la fila: ");
        fila = shinxlector.readLine();

        if (turno % 2 == 0 && fila.equalsIgnoreCase("Exit"))   {
                    System.out.println("Parece que " + usu1 + " a decidido rendirce.");
                    System.out.println("El ganador ese caso es " + usu2 + "!!!");
                    lost = lost + 1;
                    break;
                }

        System.out.println();
        /*Hacerlo a tu manera*/while (fila.compareTo("I") > 0 || fila.compareTo("A") <
       0)  {
            System.out.print("Error. La letra debe estar dentro del rango de las filas,
       intenta otra vez: ");
            fila = shinxlector.readLine();
            if (turno % 2 == 0 && fila.equalsIgnoreCase("Exit"))   {
                    System.out.println("Parece que " + usu1 + " a decidido rendirse.");
                    System.out.println("El ganador en ese caso es " + usu2 + "!!!");
                    lost = lost + 1;
                    break;
                }
        }
            System.out.println("Ahora, escoge el valor de la columna: ");
```

```
        columna = shinxlector.readLine();
        if (turno % 2 == 0 && columna.equalsIgnoreCase("Exit"))   {
                System.out.println("Parece que " + usu1 + " a decidido rendirse.");
                System.out.println("El ganador ese caso es " + usu2 + "!!!");
                lost = lost + 1;
                break;
            }

        /*Hacerlo a tu manera*/while (Integer.parseInt(columna) > 9 ||
    Integer.parseInt(columna) < 1)  {
            System.out.println("Error. La cifra ingresada no concuerda con los valores
    de las columnas. Por favo ingresa de nuevo un valor: ");
            columna = shinxlector.readLine();
            if (turno % 2 == 0 && columna.equalsIgnoreCase("Exit"))   {
                System.out.println("Parece que " + usu1 + " a decidido rendirse.");
                System.out.println("El ganador ese caso es " + usu2 + "!!!");
                lost = lost + 1;
                break;
            }
        }
        ataque = fila + columna;

        while (gift1.contains(ataque) || Integer.parseInt(columna) > 9 ||
    Integer.parseInt(columna) < 1 || fila.compareTo("I") > 0 || fila.compareTo("A") <
    0)    {
            System.out.println("Coordenada repetida. Ingresa una nueva");
            System.out.println("Ingrese un nuevo valor para la fila: ");
            fila = shinxlector.readLine();
            System.out.println("Ingrese un nuevo valor para la columna: ");
            columna = shinxlector.readLine();
            ataque = fila + columna;
        }
        gift1.add(ataque);

     }   else if (turno % 2 == 1)  {
        System.out.println("Es el turno de " + usu2 + ". Elije dónde vas a atacar: ");
        System.out.print("Escoge primero el valor de la fila: ");
            fila = shinxlector.readLine();

            if (turno % 2 == 1 && fila.equalsIgnoreCase("Exit"))   {
                        System.out.println("Parece que " + usu2 + " a decidido
    rendirse.");
                        System.out.println("El ganador ese caso es " + usu1 + "!!!");
                        lost = lost + 2;
                        break;
```

```
                }

        System.out.println();
        /*Hacerlo a tu manera*/while (fila.compareTo("I") > 0 ||
fila.compareTo("A") < 0)  {
            System.out.print("Error. La letra debe estar dentro del rango de las
filas, intenta otra vez: ");
            fila = shinxlector.readLine();
            if (turno % 2 == 1 && fila.equalsIgnoreCase("Exit"))   {
                    System.out.println("Parece que " + usu2 + " a decidido
rendirse.");
                    System.out.println("El ganador en ese caso es " + usu1 +
"!!!");
                    lost = lost + 2;
                    break;
                }
        }
            System.out.println("Ahora, escoge el valor de la columna: ");
            columna = shinxlector.readLine();
            if (turno % 2 == 1 && columna.equalsIgnoreCase("Exit"))   {
                    System.out.println("Parece que " + usu2 + " a decidido
rendirse.");
                    System.out.println("El ganador ese caso es " + usu1 + "!!!");
                    lost = lost + 2;
                    break;
                }

            /*Hacerlo a tu manera*/while (Integer.parseInt(columna) > 9 ||
Integer.parseInt(columna) < 1)  {
                System.out.println("Error. La cifra ingresada no concuerda con los
valores de las columnas. Por favo ingresa de nuevo un valor: ");
                columna = shinxlector.readLine();
                if (turno % 2 == 1 && columna.equalsIgnoreCase("Exit"))   {
                    System.out.println("Parece que " + usu2 + " a decidido
rendirse.");
                    System.out.println("El ganador ese caso es " + usu1 + "!!!");
                    lost = lost + 2;
                    break;
                }
            }
            ataque = fila + columna;

            while (gift2.contains(ataque) || Integer.parseInt(columna) > 9 ||
Integer.parseInt(columna) < 1 || fila.compareTo("I") > 0 || fila.compareTo("A") <
0)    {
```

```
                    System.out.println("Error. Coordenada repetida, ingresa una
    nueva.");
                    System.out.println("Ingrese un nuevo valor para la fila: ");
                    fila = shinxlector.readLine();
                    System.out.println("Ingrese un nuevo valor para la columna: ");
                    columna = shinxlector.readLine();
                    ataque = fila + columna;
                }
                gift2.add(ataque);
            }
            String [][] attackmat = pancham.battack (turno, fila, columna, mat2, mat4);

            if (turno % 2 == 0)     {
                for (int f = 0; f < 10; f++)    {
                    for (int c = 0; c < 10; c++)    {
                        if (attackmat [f][c].equals(" X "))     {
                            mat1 [f][c] = " X ";
                            atinar1++;
                        }   else if (attackmat [f][c].equals(" 0 "))    {
                            mat1 [f][c] = " 0 ";
                        }
                    }
                }
                for (int f = 0; f < 10; f++)    {
                    for (int c = 0; c < 10; c++)    {
                        System.out.print(mat1[f][c]);
                    }
                    System.out.print("   ");
                    for (int c = 0; c < 10; c++)    {
                        System.out.print(mat2[f][c]);
                    }
                    System.out.println();
                }
            }   else if (turno % 2 == 1) {
                for (int f = 0; f < 10; f++)    {
                    for (int c = 0; c < 10; c++)    {
                        if (attackmat [f][c].equals(" X "))     {
                            mat3 [f][c] = " X ";
                            atinar2++;
                        }   else if (attackmat [f][c].equals(" 0 "))    {
                            mat3 [f][c] = " 0 ";
                        }
                    }
                }
```

```java
                for (int f = 0; f < 10; f++)    {
                    for (int c = 0; c < 10; c++)    {
                        System.out.print(mat3[f][c]);
                }
                System.out.print("    ");
                for (int c = 0; c < 10; c++)    {
                    System.out.print(mat4[f][c]);
                }
                System.out.println();
            }
        }
            System.out.print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
        \n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
            turno = turno + 1;
            }

        System.out.println("Partida terminada. Es momento de ver un pequeño resumen de la
        partida: ");

        if (atinar1 == 11 || lost == 2)   {
            System.out.println(usu1 + " a sido el ganador de esta partida. GG EZ!!!");
            System.out.println("La partida duró un total de " + turno + " turnos.");
            System.out.println("Este es el resumen de " + usu1 + ": ");
            System.out.println(resul1[0] + "\n" + resul1[1] + "\n" + resul1[2] + "\n" +
        resul1[3]);
            System.out.println("Este es el resumen de " + usu2 + ": ");
            System.out.println(resul2[0] + "\n" + resul2[1] + "\n" + resul2[2] + "\n" +
        resul2[3]);
        }   else if (atinar2 == 11 || lost == 1)     {
            System.out.println(usu2 + " a sido el ganador de esta partida. GG EZ!!!");
            System.out.println("La partida duró un total de " + turno + " turnos.");
            System.out.println("Este es el resumen de " + usu2 + ": ");
            System.out.println(resul2[0] + "\n" + resul2[1] + "\n" + resul2[2] + "\n" +
        resul2[3]);
            System.out.println("Este es el resumen de " + usu1 + ": ");
            System.out.println(resul1[0] + "\n" + resul1[1] + "\n" + resul1[2] + "\n" +
        resul1[3]);
        }

    }//Fin Public
}//Fin Main
```