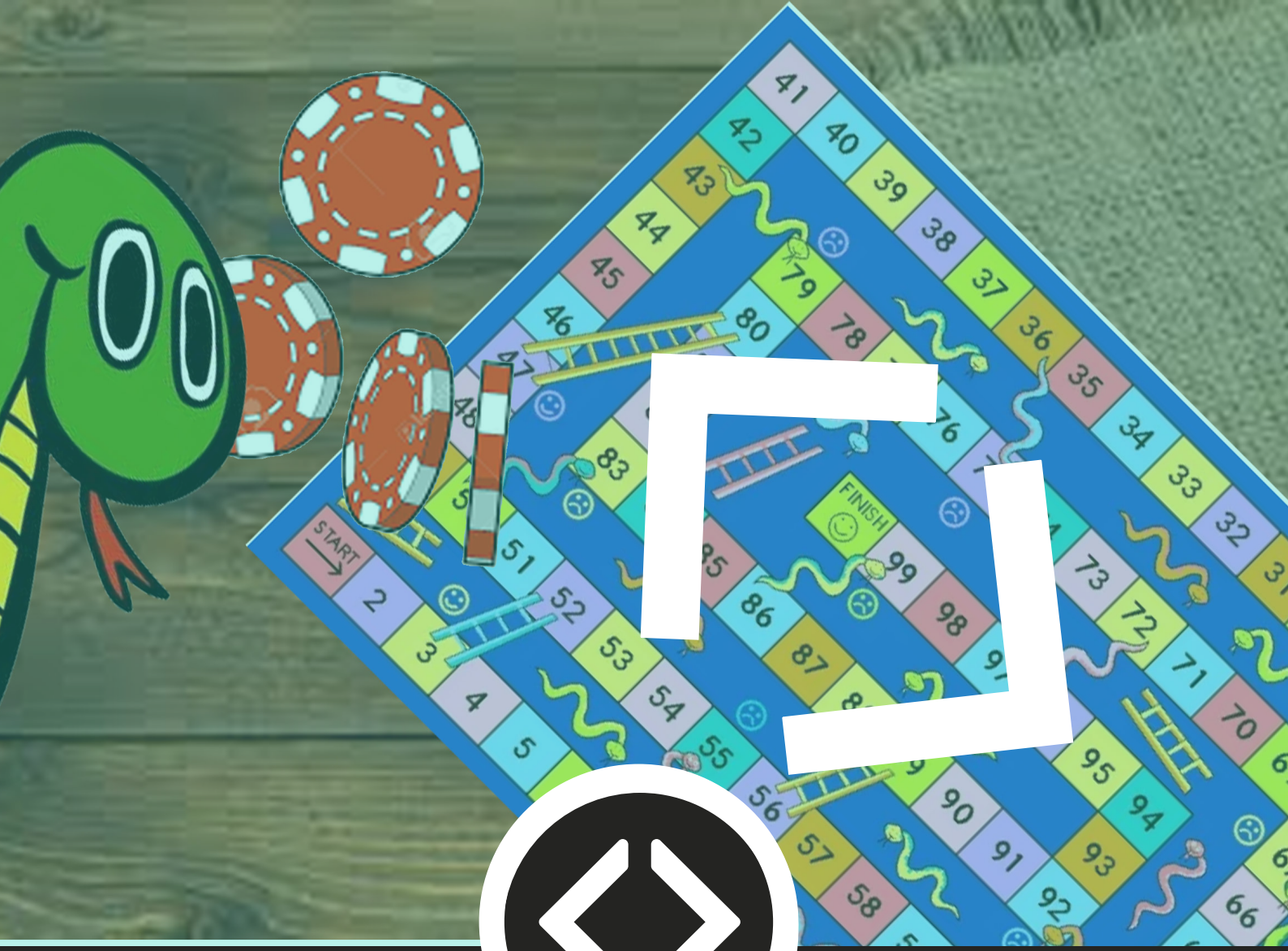
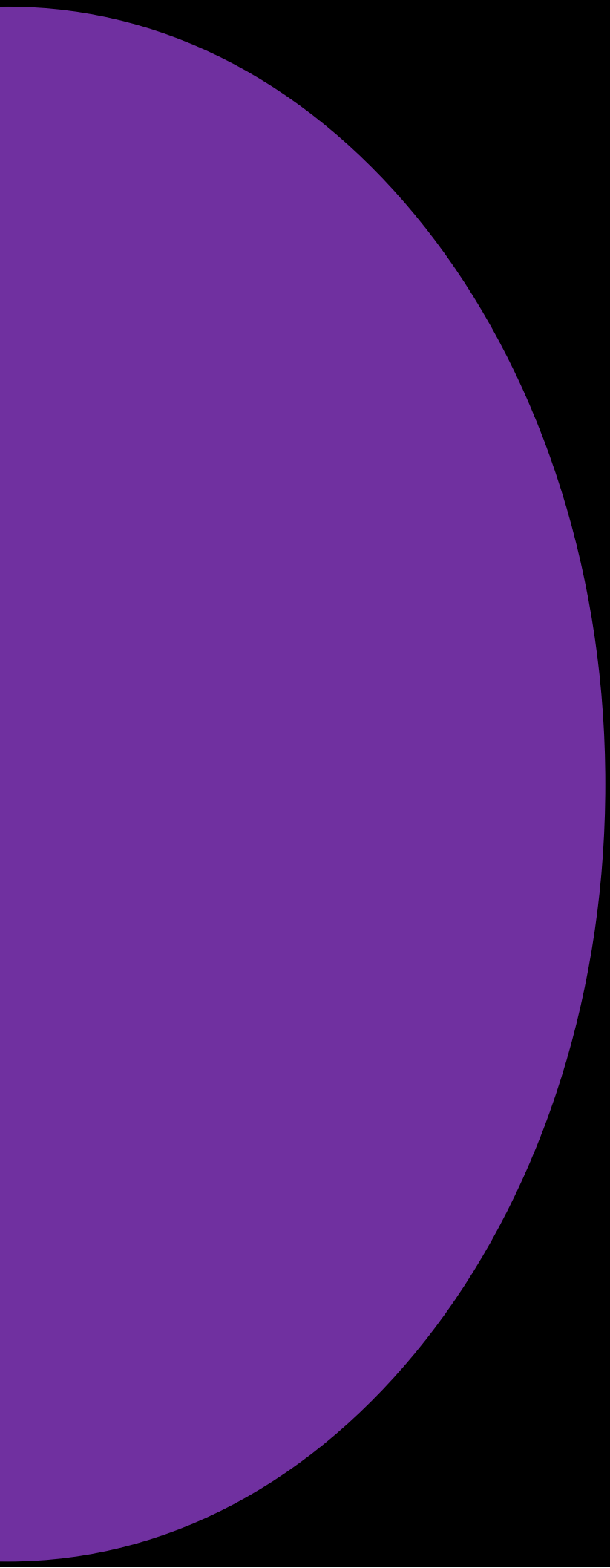


MANUAL TÉCNICO

Escaleras Matemáticas

CURSO	CARNET	NOMBRE
IPC1	*201900042*	*Rodrigo Hernández*

ÍNDICE



OBJETIVOS

- El programador pueda captar todas las clases y métodos hechos en la aplicación del lenguaje Java.
- Fundamentar el código con el cuál se desarrolló el programa.

INTRODUCCIÓN

Este manual tiene el propósito de que como desarrollador se pueda entender el funcionamiento del código con el cual fue desarrollada la aplicación, también que demuestre cada método utilizado dentro de este para fundamentar lo que llegó a formarse esta aplicación en sí.

REQUERIMIENTOS

- Tener instalado Java Runtime Enviroment 8.
- Tener instalado Java Development Kit 8.2
- Usarlo únicamente en PC.
- Lo recomendable es usar el sistema operativo de Windows 10.
- Tener un IDE compatible con Java 8.2 (ejemplo: Netbeans8.2).

CÓDIGO FUENTE

1. Clase Pra1

Esta clase es utilizada únicamente para mandar a llamar a la clase programap() y poder ejecutarlo

```
//CLASE PRINCIPAL LA CUAL DESDE AQUI INICIA TODO EL PROCESO DEL PROGRAMA
public class Pra1 {
    public static void main(String[] args) {
        //Se manda a llamar la clase programap para ejecutar las instrucciones de esa clase
        programap m = new programap();
        m.menu();
    }
}
```

2. Clase programap()

Esta clase es donde se encuentra toda la lógica del juego.

2.1 Método inicior1()

Este método es utilizado para la cabeza del reporte1 en html.

```
    //Metodo para la parte inicial del documento de reportaje
    public void inicior1(){
        lapiz.append("<!doctype html>\n" +
"<html lang=\"en\">\n" +
"\n" +
"<head>\n" +
"    <meta charset=\"utf-8\">\n" +
"    <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\n" +
"    <link rel=\"stylesheet\" href=\"https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css\">\
"    <script src=\"https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js\"></script>\n" +
"    <script src=\"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js\"></script>\n" +
"    <script src=\"https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js\"></script>\n" +
"    <title>Reporte1</title>\n" +
"</head>\n" +
```

2.2 Método menú()

Este método es el encargado de mostrar el menú principal y validar las opciones que el usuario esta ingresando en el Scanner.

```
//try{
    System.out.println("=============================================");
    System.out.println("===========     MENÚ PRINCIPAL     ==========");
    System.out.println("== 1. Iniciar Juego                        ==");
    System.out.println("== 2. Reanudar Juego                       ==");
    System.out.println("== 3. Generar Reportes                     ==");
    System.out.println("== 4. Salir                                ==");
    System.out.println("=============================================");
    opcion = leerm.nextInt();
    switch(opcion){
        //OPCION PARA INICIAR EL JUEGO
        case 1:
            tablero();
```

2.3 Método tablero().

Este método es utilizado para realizar tres cosas: la primera es imprimir el tablero de números de forma que inicie de derecha a izquierda y luego que cada fila cambie de dirección.

```
//TABLERO
boolean izquierda = false;
int contador = 1;
for (int i = 8-1; i >=0; i--) {
    if (izquierda) {
        for (int j = 0; j <tablero[i].length ; j++) {
            tablero[i][j]= contador;
            contador++;
        }
    }else{
        for (int j = tablero[i].length-1; j >=0 ; j--) {
            tablero[i][j]= contador;
            contador++;
        }

    }
    izquierda = !izquierda;
}
```

La segunda es la implementación del tablero de penalizaciones de forma aleatoria e imprima “#”.

```
for (int i = 0; i < 8; i++) {
    for (int j = 0; j < penalizaciones[i].length; j++) {
            if (r.nextBoolean() == true) {
                p = 1;
            } else {
                p = 0;
            }
            penalizaciones[i][j] = p;
    }
}
penalizaciones[7][7] = 0;
```

```
for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        // IMPRIMIR LAS PENALIZACIONES
        if (penalizaciones[i][j] == 1) {
            System.out.print("# " + tablero[i][j] + "\t|");
        } else {
            System.out.print("  " + tablero[i][j] + "\t|");
        }

    }
    System.out.println("");
```

Y tercero que es la parte donde la ficha se va a estar moviendo en todo el tablero y hasta se definió su posición inicial.

```
// IMPRIMIR LA MATRIZ DE MOVIMIENTOS
for (int j = 0; j < 8; j++) {

    System.out.print(movimiento[i][j] + "\t|");
}
System.out.println("");
System.out.println("-----------------------------------------------------------------");
```

```
//TABLERO DE MOVIMIENTOS
for (int i = 0; i <8; i++) {
    for (int j = 0; j < 8; j++) {
        movimiento[i][j] = " ";
    }
}
movimiento[7][7] = "@";
```

Se añadió la lógica de los movimientos del tablero en donde se iba utilizando un “do-while” para que el tablero se fuera actualizando comenzando con que imprima como vacio el tablero.

```
//----------MOVIMIENTOS DE LA FICHA----------
//IMPRIME EL TABLERO COMO VACIO
for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        movimiento[i][j] = " ";
    }
}
```

Luego el dado se ejecuta con que muestre un numero random entre 2 y 6, y llama a la función Dado que hace esa operación y se guarda la variable.

```
int d = Dado();

//VALIDAR LA OPCION 1
if (o == 1) {
    System.out.println("=============================================================================");
    //MUESTRA EL RESULTADO DEL DADO
    System.out.println("    DADO: " + d);
    System.out.println("");
```

Se llama a la función donde devolverá las posiciones del tablero cada vez que se tira el dado, se definió “x” como las filas y “y” como columnas:

```
//Se crea una variable String para leer la duncion posicion
String sp = posicion(base);
//Se guarda en un arreglo de string los numeros sacados de la funcion
String[] coordenadas = sp.split(",");
//Se guarda las coordenadas de la nueva posicion de la ficha
int x = Integer.parseInt(coordenadas[0]);
int y = Integer.parseInt(coordenadas[1]);
```

Se valido si las posiciones eran menor que 8 para que ejecute los movimientos por medio de un for e imprima la ficha y el tablero actualizado.

```
if (y < 8) {
    //IMPRIME LA FICHA
    movimiento[x][y] = "@";
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            // IMPRIMIR LAS PENALIZACIONES
            if (penalizaciones[i][j] == 1) {
                System.out.print("# " + tablero[i][j] + "\t|");
            } else {
                System.out.print("  " + tablero[i][j] + "\t|");
            }

        }
        System.out.println("");
        // IMPRIMIR LA MATRIZ DE MOVIMIENTOS
        for (int j = 0; j < 8; j++) {

            System.out.print(movimiento[i][j] + "\t|");
        }
        System.out.println("");
        System.out.println("-------------------------------------------------------------------");
    }
```

Se verifico si la ficha cae en la fila 7 o 6 que llame al método penalización fácil, si cae en la fila 5,4 o 3 que llame al método penalización intermedia y si cae en la fila 2,1 o 0 entonces que llame a la penalización difícil.

```
for (int i = 0; i < 8; i++) {
    for (int j = 0; j < penalizaciones[i].length; j++) {
        if (penalizaciones[i][j] == 1) {
            //SI LAS PENALIZACIONES SON 1 Y ESTAN EN LA MISMA POSICION DE LA FICHA ENTONCES QUE SE EJECUTEN LAS FUNCIONES
            /**
             * SI ESTA EN LA FILA 7 Y 6: SE LLAMA A LA PENALIZACIÓN FACIL
             * SI ESTA EN LA FILA 5, 4 y 3: SE LLAMA A LA PENALIZACIÓN INTERMEDIA
             * SI ESTA EN LA FILA 2, 1 Y 0: SE LLAMA A LA PENALIZACIÓN DIFICIL
            **/
            if (i == x && j == y) {
                if (i == 0 || x == 0) {
                    penalizaciondificil();
                }else if (i == 1 || x == 1) {
                    penalizaciondificil();
                }else if (i == 2 || x == 2) {
                    penalizaciondificil();
                }else if (i == 3 || x == 3) {
                    penalizacionIntermedia();
                }else if (i == 4 || x == 4) {
                    penalizacionIntermedia();
                }else if (i == 5 || x == 5) {
                    penalizacionIntermedia();
                }else if (i == 6 || x == 6) {
                    penalizacionfacil();
                }else if (i == 7 || x == 7) {
                    penalizacionfacil();
                }

            }
        }
    }
```

Se verifica si la ficha superó las 64 posiciones y si lo superó entonces finaliza el juego:

```
}else{
    //SI SUPERA AL TAMAÑO DEL TABLERO ENTONCES SE FINALIZA EL JUEGO
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            // IMPRIMIR LAS PENALIZACIONES
            if (penalizaciones[i][j] == 1) {
                System.out.print("# " + tablero[i][j] + "\t|");
            } else {
                System.out.print("  " + tablero[i][j] + "\t|");
            }

        }
        System.out.println("");
        // IMPRIMIR LA MATRIZ DE MOVIMIENTOS
        for (int j = 0; j < 8; j++) {

            System.out.print("" + "\t|");
        }
        System.out.println("");
        System.out.println("-----------------------------------------------------------------");
    }
    System.out.println("");
```

2.4 Función Dado()

En esta función se realiza la operación donde al llamar la librería random se llame un numero aleatorio entre 2 y 6 y retorne su valor.

```
//FUNCION PARA RETORNAR UN VALOR ALEATORIO ENTRE 2 Y 6
public int Dado(){
    Random r = new Random();
    int dado = 2+r.nextInt((6-2)+1);
    return dado;
}
```

2.5 Función posición()

En esta función lo que hace es que verifique la suma de la posición actual de la ficha con la suma de lo que retorno el valor de la función Dado y devuelva la posición de las coordenadas de la ficha.

```
public String posicion(int b){
    switch(b){
        case 2:
            return "7,6";
```

2.6 Método penalizacionfacil()

En este método solo muestra el mensaje de “Cayó en una penalización” y depende del numero random que aparezca entonces llama a cada opción que tiene esta penalización.

```
//METODO PARA EL MENU DE LA PENALIZACION FACIL
public void penalizacionfacil(){
    Random rf = new Random();
    int opn = 1+rf.nextInt((3-1)+1);
    System.out.println("=================================================");
    System.out.println("===     ¡HAS CAIDO EN UNA PENALIZACIÓN!       ===");
    System.out.println("===             MODALIDAD: FACIL              ===");
    System.out.println("=================================================");
    switch (opn) {
        case 1:
            opcionaf();
            break;
        case 2:
            opcionbf();
            break;
        case 3:
            opcioncf();
            break;
    }
}
```

2.7 Método opcionaf()

En este método se realiza la operación de calcular por medio de ley de cosenos el lado B, el ángulo gamma y Beta.

2.8 Método opcionbf()

En este método se realiza la operación del sistema donde aplica ley de cosenos para encontrar el lado A, el ángulo alfa y gamma.

2.9 Método opcioncf()

Este método es utilizado para que el sistema por medio de la ley de cosenos calcule el lado C, el ángulo alfa y beta.

2.10 Método repoopaaf()

Este método es utilizado para que realice el reporte del método opcionaf().

2.11 Método repoopbbf()

Este método es utilizado para que realice el reporte del método opcionbf().

2.12 Método repoopccf()

Este método es utilizado para que realice el reporte del método opcioncf().

2.13 Método penalizacionintermedia()

Este método es utilizado para que el sistema haga la operación de sumas de matrices A y B.

```
//METODO PARA EL MENU DE LA PENALIZACION INTERMEDIA
public void penalizacionIntermedia() {
    Scanner sc = new Scanner(System.in);
    Random rf = new Random();
    int opp = 1+rf.nextInt((3-1)+1);
    System.out.println("=============================================================");
    System.out.println("===      ¡HAS CAIDO EN UNA PENALIZACIÓN!                  ===");
    System.out.println("===           MODALIDAD: INTERMEDIA                       ===");
    System.out.println("=============================================================");
    switch (opp) {
```

Dentro de este método hay un switch case donde el caso 1 toma la opción 1 de la penalización, caso 2 para la opción 2 de la penalización y caso 3 para la opción 3 de su penalización.

2.14 Método piar1()

Este método es utilizado para reportar el caso 1 del método penalizacionintermedia.

2.15 Método pibr1()

Este método es utilizado para reportar el caso 2 del método penalizacionintermedia.

2.16 Método picr1()

Este método es utilizado para reportar el caso 1 del método penalizacionintermedia.

2.17 Método penalizaciondificil()

Este método es el encargado de que por medio de un numero aleatorio entre 1 y 3 llame a sus métodos.

```
//---------------------------------------PENALIZACIÓN DÍFICIL-----------------------------------------
//METODO PARA EL MENU DE LA PENALIZACION DÍFICIL
public void penalizaciondificil() {
    Random rf = new Random();
    int opc = 1+rf.nextInt((3-1)+1);
    System.out.println("=================================================");
    System.out.println("===     ¡HAS CAIDO EN UNA PENALIZACIÓN!       ===");
    System.out.println("===            MODALIDAD: DÍFCIL              ===");
    System.out.println("=================================================");
    switch (opc) {
        case 1:
            opcionad();
            break;
        case 2:
            opcionbd();
            break;
        case 3:
            opcioncd();
            break;
    }
}
```

2.18 Método opcionad()

Este método es el encargado de realizar la operación de la matriz inversa por medio de gauus jordan de la matriz B y multiplicarla con la matriz A para obtener la división de matrices entre A y B de la opción 1.

2.19 Método opcionbd()

Este método es el encargado de realizar la operación de la matriz inversa por medio de gauus jordan de la matriz B y multiplicarla con la matriz A para obtener la división de matrices entre A y B de la opción 2.

2.20 Método opcioncd()

Este método es el encargado de realizar la operación de la matriz inversa por medio de gauus jordan de la matriz B y multiplicarla con la matriz A para obtener la división de matrices entre A y B de la opción 3.

2.21 Método reop1d()

Este método es el encargado de reportar el método opcionad().

2.22 Método reop2d()

Este método es el encargado de reportar el método opcionbd().

2.23 Método reop3d()

Este método es el encargado de reportar el método opconcd().

2.24 Método finaldoc1()

Este método es el encargado de agregar la parte final del reporte1.

2.25 Método errorr1()

Este método es el encargado de mostrar un mensaje de error en el reporte.

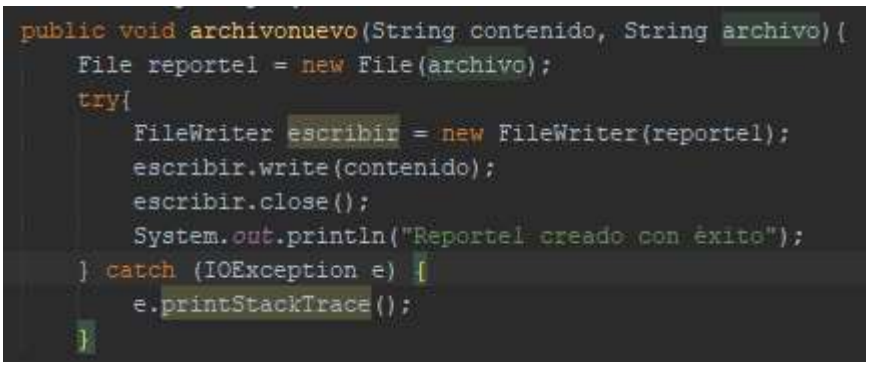
2.26 Método r1()

Este método es el encargado de unir todos los métodos que hacen el reportaje de la aplicación.

3. CLASE REPORTAJE

3.1 Método archivonuevo()

Este método es el encargado de crear el archivo de reportes donde ingresaran los parámetros que están en el método de r1() de la clase programap() y se puedan generar los reportes agregando también que se cierra el archivo.



```
public void archivonuevo(String contenido, String archivo){
    File reporte1 = new File(archivo);
    try{
        FileWriter escribir = new FileWriter(reporte1);
        escribir.write(contenido);
        escribir.close();
        System.out.println("Reporte1 creado con éxito");
    } catch (IOException e) {
        e.printStackTrace();
    }
```

LIBRERIAS UTILIZADAS

1. Java.util.Scanner.

Esta librería es utilizada para leer los datos del usuario.

2. Java.util.Random.

Esta librería es utilizada para números aleatorios.

3. Java.io.*.

Esta librería es utilizada para la creación de archivos en java.

CONCLUSIONES

Este manual es necesario para entender el código hecho por el programador encargado de este programa y que por medio de la descripción de cada método utilizado pueda ser útil para la buena comprensión de lo que se realizó, cada operación matemática, lógica y funcional dentro de la aplicación.