

MANUAL TECNICO

Este documento tiene el enfoque de ayudar al programador para realizar mantenimiento o actualizaciones al software.

Declaración de variables globales, las cuales tiene el objetivo de usarlas a lo largo del programa y poder modificarlas o imprimirlas en consola.

```
static Scanner ingreso;  
//tablero y penalizaciones  
static int[][] tablero = new int[8][8];  
static int[][] tablero2 = new int[8][8];  
//lecciones difisiles
```

En la matriz “tablero” se almacenarán las posiciones del tablero.

En la matriz “tablero2” se almacenarán las penalizaciones que habrá en todo el juego jejeje.

```
//primera operacion  
static double B1;  
static double Betha1;  
static double Gamma1;  
//segunda operacion  
static double A2;  
static double Alpha2;  
static double Gamma2;  
//tercera operacion  
static double C3;  
static double Alpha3;  
static double Betha3;
```

SUMA DE MATRICES

Acá se almacenarán los resultados de los primero 3 problemas de la ley de senos.

```
//ejercicios medios
static int[][] M2_1 = new int[5][5];
static int[][] M2_2 = new int[5][5];
static int[][] M2_3 = new int[5][5];
static int[][] M2_4 = new int[5][5];
static int[][] M2_5 = new int[5][5];
static int[][] M2_6 = new int[5][5];
```

Este contendrá las matrices de los ejercicios medios a realizar, la cuales se irán sumando y guardando en las siguientes matrices.

```
//suma de matices
static int[][] Suma1 = new int[5][5];
static int[][] Suma2 = new int[5][5];
static int[][] Suma3 = new int[5][5];
```

Se almacenará los resultados de las sumas realizadas por medio del método que se usó el cual es llamado

```
public static void suma() {
    for (int i = 0; i < M2_1.length; i++) {
        for (int j = 0; j < M2_1[i].length; j++) {
            Suma1[i][j] = M2_1[i][j] + M2_2[i][j];
        }
    }

    for (int i = 0; i < Suma1.length; i++) {
        for (int j = 0; j < Suma1[i].length; j++) {
            Suma2[i][j] = M2_3[i][j] + M2_4[i][j];
        }
    }

    for (int i = 0; i < M2_1.length; i++) {
        for (int j = 0; j < M2_1[i].length; j++) {
            Suma3[i][j] = M2_5[i][j] + M2_6[i][j];
        }
    }
}
```

MULTIPLICACIÓN DE MATRICES

```
//ejecicios dificiles
static double[][] M3_1 = new double[4][4];
static double[][] M3_2 = new double[4][4];
static double[][] M3_3 = new double[4][4];
static double[][] M3_4 = new double[4][4];
static double[][] M3_5 = new double[4][4];
static double[][] M3_6 = new double[4][4];
```

Acá se declaran las matrices las cuales contendrán los valores de las matrices a la hora de operar en el siguiente paso.

```
static double[][] M1 = new double[4][4];
static double[][] M2 = new double[4][4];
static double[][] M3 = new double[4][4];
```

Acá se almacenarán el resultado de la multiplicación.

```
public static void multilplicarIn() {
    for (int i = 0; i < M3_1.length; i++) {
        for (int j = 0; j < M3_1[i].length; j++) {
            for (int k = 0; k < M3_1[i].length; k++) {
                M1[i][j] += M3_1[i][k] * In1[k][j];
            }
        }
    }

    for (int i = 0; i < M3_1.length; i++) {
        for (int j = 0; j < M3_1[i].length; j++) {
            for (int k = 0; k < M3_1[i].length; k++) {
                M2[i][j] += M3_3[i][k] * In2[k][j];
            }
        }
    }

    for (int i = 0; i < M3_1.length; i++) {
        for (int j = 0; j < M3_1[i].length; j++) {
            for (int k = 0; k < M3_1[i].length; k++) {
                M3[i][j] += M3_5[i][k] * In3[k][j];
            }
        }
    }
}
```

Acá se hará la multiplicación de la operación de la matriz para poder almacenar y colocar el resultado en las matrices ya antes mencionadas.

DIVISION DE MATRICES

En este proceso se encuentra la inversa de una matriz

```
public static double[][] inverse(double a[][]) {
    int n = a.length;
    double w[] = new double[n/2];
    double b[] = new double[n/2];
    int index[] = new int[n];
    for (int i = 0; i < n; ++i) {
        b[i/2] = 1;
    }

    gaussiane(a, index);

    for (int i = 0; i < n; ++i) {
        for (int j = i + 1; j < n; ++j) {
            for (int k = 0; k < n; ++k) {
                b[index[j][k]] -= a[index[i][k]] * b[index[i][k]];
            }
        }
    }

    for (int i = 0; i < n; ++i) {
        w[n - 1 - i] = b[index[n - 1][i]] / a[index[n - 1][n - 1 - i];
        for (int j = n - 2; j >= 0; --j) {
            w[j/2] = b[index[j][i]];
            for (int k = j + 1; k < n; ++k) {
                w[j/2] -= a[index[j][k]] * w[k/2];
            }
        }

        w[j/2] /= a[index[j][j/2]];
    }

    return w;
}
```

```
public static void gaussiane(double a[][], int index[]) {
    int n = index.length;
    double w[] = new double[n];

    for (int i = 0; i < n; ++i) {
        index[i] = i;
    }

    for (int i = 0; i < n; ++i) {
        double o1 = 0;
        for (int j = 0; j < n; ++j) {
            double o0 = Math.abs(a[i][j]);
            if (o0 > o1) {
                o1 = o0;
            }
        }
        w[i] = o1;
    }

    int k = 0;
    for (int j = 0; j < n - 1; ++j) {
        double p11 = 0;
        for (int i = j; i < n; ++i) {
            double p10 = Math.abs(a[index[i][j]);
            p10 /= w[index[i][j]);
            if (p10 > p11) {
                p11 = p10;
                k = i;
            }
        }

        int itmp = index[j];
        index[j] = index[k];
        index[k] = itmp;
        for (int i = j + 1; i < n; ++i) {
            double p1 = a[index[i][j]] / a[index[j][j]);
            a[index[i][j]] = p1;

            for (int i = j + 1; i < n; ++i) {
                a[index[i][j]] -= p1 * a[index[j][i]];
            }
        }
    }
}
```

En estos dos métodos nos ayuda para saber la matriz inversa. La cual nos va a servir para que calcular con la división de las matrices.

```
public static void guardarInversa() {
    //funcion de Inversa
    //matriz inversa 1
    In1 = Inversa(M1_2);
    for (int i = 0; i < R1.length; i++) {
        for (int j = 0; j < R1[i].length; j++) {
            R1[i][j] = Math.round(In1[i][j] * 100.0) / 100.0;
        }
    }

    //matriz inversa 2
    In2 = Inversa(M1_4);
    for (int i = 0; i < R1.length; i++) {
        for (int j = 0; j < R1[i].length; j++) {
            R2[i][j] = Math.round(In2[i][j] * 100.0) / 100.0;
        }
    }

    //matriz inversa 3
    In3 = Inversa(M1_6);
    for (int i = 0; i < R3.length; i++) {
        for (int j = 0; j < R3[i].length; j++) {
            R3[i][j] = Math.round(In3[i][j] * 100.0) / 100.0;
        }
    }
}
```

En este proceso se realizará lo que es guardar la matriz y aproximarla a dos decimales.

```
public static void multiplicarIn() {
    for (int i = 0; i < M3_1.length; i++) {
        for (int j = 0; j < M3_1[i].length; j++) {
            for (int k = 0; k < M3_1[i].length; k++) {
                M1[i][j] += M3_1[i][k] * In1[k][j];
            }
        }
    }

    for (int i = 0; i < M3_1.length; i++) {
        for (int j = 0; j < M3_1[i].length; j++) {
            for (int k = 0; k < M3_1[i].length; k++) {
                M2[i][j] += M3_3[i][k] * In2[k][j];
            }
        }
    }

    for (int i = 0; i < M3_1.length; i++) {
        for (int j = 0; j < M3_1[i].length; j++) {
            for (int k = 0; k < M3_1[i].length; k++) {
                M2[i][j] += M3_5[i][k] * In3[k][j];
            }
        }
    }
}
```

Por lo último se realiza la multiplicación de la matriz “A” por la matriz inversa de “B”.

MOVIMIENTO DEL TABLERO

```
boolen penalization = false;

if (iposition <= 40) {
    boolen deracha = false;
    string s = "";
    for (int i = 0; i < tablero[i].length(); i++) {
        System.out.println(" ");
        if (deracha) {
            for (int j = 0; j < tablero[0].length(); j++) {
                System.out.print(tablero[i][j] + " " + "\t");
            }
            System.out.println("");
        } else {
            for (int j = 0; j < tablero[0].length(); j++) {
                if (tablero[i][j] == position || tablero[i][j] == 0) {
                    System.out.print("\t\t" + " " + "\t");
                } else if (iposition <= 39) {
                    continue;
                } else if (iposition > 39) {
                    continue;
                } else if (iposition > 40) {
                    diffusi++;
                }
            }
            penalization = true;
        }
        if (tablero[i][j] == position) {
            System.out.print("\t\t" + " " + "\t");
        } else if (tablero[i][j] == 0) {
            System.out.print("\t\t" + " " + "\t");
        } else {
            System.out.print("\t\t" + " " + "\t");
        }
    }
} else {
```

[illegible]

```

    } else {
        System.out.println("");
        System.out.println("¡Usted a ganado el juego!");
        menu();
        System.out.println("");
    }
    System.out.println("");
}

```

En este proceso se realiza el movimiento de la ficha en el tablero. En la cual se declaran 3 variables para realizar el conteo de penalizaciones según las filas correspondientes y según la dificultad del ejercicio. En la cual se llama a una función dado la cual contiene el movimiento que se realizar en el tablero y saber si el jugador callo en una penalización o no. Al mismo tiempo se imprimirá un mensaje sabiendo si el usuario

esta exonerado de las operaciones ya que solo puede caer dos veces en una penalización según en nivel de dificultad del programa.

PENALIZACIONES

En esta parte del código según en donde este la dicha del jugador se imprimirá el numero de problema que tiene que realizar.

```
static int problema;
public static void penalizaciones() {
    problema = generador.nextInt(3) + 1;
    if (posicion > 0 && posicion <= 16) {
        if (facil > 0 && facil <= 2) {
            System.out.println("Tiene que realizar el problema " + problema + " de la seccion facil");
            System.out.println("");
            if (problema == 1) {
                System.out.println("Segun la ley de cosenos este es el resultado: ");
                System.out.println("El lado B tiene el valor de: "+B1);
                System.out.println("El angulo Beta tiene el valor de: "+Beta1);
                System.out.println("El angulo Gamma tiene el valor de: "+Gamma1);
            } else if (problema == 2) {
                System.out.println("El lado A tiene el valor de: "+A2);
                System.out.println("El angulo Alpha tiene el valor de: "+Alpha2);
                System.out.println("El angulo Gamma tiene el valor de: "+Gamma2);
            } else if (problema == 3) {
                System.out.println("El lado C tiene el valor de: "+C3);
                System.out.println("El angulo Alpha tiene el valor de: "+Alpha3);
                System.out.println("El angulo Beta tiene el valor de: "+Beta3);
            }

            } else if (facil > 2) {
                System.out.println("ya esta exonerado de las penalizaciones faciles ");
            }
        }
    }
}
```

Este mismo proceso se realiza con las 3 dificultades del tablero.

REANUDAR EL JUEGO

En este paso se realiza casi lo mismo del tablero, pero sin tomar en cuenta los contadores de penalizaciones que tiene para no alterar el resultado de la misma.

```

public static void reanudar() {
    Boolean penalizacion = false;
    if (posicion == 0) {
        Boolean derecha = false;
        String ss = "";
        for (int i = 0; i < tablero.length; i++) {
            System.out.println(" ");
            if (derecha) {
                for (int j = 0; j < tablero[i].length; j++) {
                    System.out.print(tablero[i][j] + " " + " ");
                }
                System.out.println("");
            }
            for (int j = 0; j < tablero[i].length; j++) {
                if (tablero1[i][j] == posicion || tablero2[i][j] == 0) {
                    System.out.print("00" + " ");
                }
                penalizacion = true;
            }
            if (tablero1[i][j] == posicion) {
                System.out.print("0" + " ");
            }
            if (tablero2[i][j] == 0) {
                System.out.print("0" + " ");
            }
            else {
                System.out.print(" " + " ");
            }
        }
        for (int j = tablero[i].length - 1; j >= 0; j--) {
            System.out.print(tablero1[i][j] + " " + " ");
        }
    }
    else {
        for (int j = tablero[i].length - 1; j >= 0; j--) {
            System.out.print(tablero1[i][j] + " " + " ");
        }
        System.out.println("");
        for (int j = tablero[i].length - 1; j >= 0; j--) {
            if (tablero1[i][j] == posicion || tablero2[i][j] == 0) {
                System.out.print("00" + " ");
            }
            penalizacion = true;
        }
        if (tablero1[i][j] == posicion) {
            System.out.print("0" + " ");
        }
        if (tablero2[i][j] == 0) {
            System.out.print("0" + " ");
        }
        else {
            System.out.print(" " + " ");
        }
    }
    System.out.println("");
    derecha = !derecha;
    System.out.println(" ");
    if (penalizacion == true) {
        System.out.println("Duras Duras es una casilla penalizada");
        penalizaciones++;
    }
    else {
        System.out.println("");
        System.out.println("Duras a pasado al juego.");
        penurias;
        System.out.println("");
    }
}

```


GENERAR REPORTES

En es paso, por medio de FileWriter se realizará un documento de “html” para la impresión de los cálculos que se realizaron en la práctica.

```

public static void reporte() {
    FileWriter finhern = null;
    PrintWriter pw = null;

    try {
        finhern = new FileWriter("Reporte.html");
        pw = new PrintWriter(finhern);

        pw.println("<DOCTYPE html>");
        pw.println("<html>");
        pw.println("<head>");
        pw.println("<meta charset = 'utf-8'>");
        pw.println("<title>REPORTES DEL SUJETO/</title>");
        pw.println("<style type = 'text/css'>");
        pw.println("body{");
        pw.println("    * margin: 0;\n");
        pw.println("    * font-family: 'serif'; font-size: 20px; color: black;\n");
        pw.println("    * background-color: #f0f0f0;\n");
        pw.println("    * background: linear-gradient(to top, #f0f0f0, #f0f0f0, #f0f0f0, #f0f0f0, #f0f0f0, #f0f0f0, #f0f0f0, #f0f0f0, #f0f0f0);\n");
        pw.println("    * text-align: center;\n");
        pw.println("    * width: 100%;\n");
        pw.println("    * border-collapse: collapse;\n");
        pw.println("    * border: 1px solid black;\n");
        pw.println("    * padding: 10px;\n");
        pw.println("</style>");
        pw.println("</head>");
        pw.println("<body>");
        pw.println("<table>");
        pw.println("<tr>");
        pw.println("<th>");
        pw.println("<th>");
        pw.println("</tr>");
        pw.println("</table>");
        pw.println("</body>");
        pw.println("</html>");
    } catch (IOException e) {
        e.printStackTrace();
    }
}

boolean derecha = false;
for (int i = 0; i < tablero.length; i++) {
    pw.println("<tr class = 'table-primary'>");
    if (derecha) {
        for (int j = 0; j < tablero[i].length; j++) {
            pw.println("<td>"+ tablero[i][j] + "</td>");
        }
    } else {
        for (int j = tablero[i].length - 1; j >= 0; j--) {
            pw.println("<td>"+ tablero[i][j] + "</td>");
        }
    }
    pw.println("</tr>");
    derecha = !derecha;
}

pw.println("</table>");
pw.println("</div>");

pw.println("</div>");

//tablero de generalizaciones
pw.println("<div>");
pw.println("<table align='center'>");
pw.println("<tr>");
pw.println("<td colspan='8' style='background-color: #000000; color: white;'>Tablero de generalizaciones del juego</td>");
    for (int i = 0; i < tablero2.length; i++) {
        pw.println("<tr class = 'table-primary'>");
        for (int j = 0; j < tablero2[i].length; j++) {
            pw.println("<td>"+ tablero2[i][j] + "</td>");
        }
    }
    pw.println("</tr>");
}

```

Se imprimen los tablero y el resultado de cada una de las matrices que se usaron.