

ESCALERAS MATEMÁTICAS

MANUAL TÉCNICO

**PROGRAMA REALIZADO POR EL ALUMNO ALLEN
GIANKARLO ROMÁN VÁSQUEZ CARNET 202004745 PARA
EL CURSO INTRODUCCIÓN A LA PROGRAMACIÓN DE
COMPUTADORAS I DE LA ESCUELA DE CIENCIAS Y
SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD SAN CARLOS DE GUATEMALA**

CONTENIDO

Identificación del documento

Lugar, fecha y responsables de la elaboración

Objetivos y alcances del sistema

Especificación Técnica

Interfaz Gráfica

Lógica del programa

Identificación del documento

El presente documento constituye el manual técnico del proyecto “Escaleras Matemáticas”, en el cual se proporciona al lector los aspectos que se consideraron para su elaboración.

Es importante destacar que este manual no es un curso de aprendizaje de las funciones de programación utilizadas para el desarrollo del programa, sino es una herramienta que provee los aspectos a conocer para la forma correcta de operación y aplicación del mismo.

Programa realizado por el alumno Allen Giancarlo Román Vásquez carnet 202004745 para el curso Introducción a la Programación de Computadoras 1 de la escuela de ciencias y sistemas de la Facultad de Ingeniería de la Universidad san Carlos de Guatemala

Lugar, fecha y responsables de la elaboración

El programa se elaboro en Huehuetenango para la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, durante las últimas dos semanas del mes de agosto del año 2021 durante el segundo semestre de este y el programa fue realizado por el alumno Allen Giancarlo Román Vásquez carnet 202004745 para el curso Introducción a la Programación de Computadoras 1 de la escuela de ciencias y sistemas de la Facultad de Ingeniería de la Universidad san Carlos de Guatemala

Objetivos y alcances del sistema

Instruir a docentes y estudiantes sobre el uso adecuado del programa diseñado para aprender de forma activa los temas de algebra matricial y trigonometría.

Proveer de una herramienta de aprendizaje a docentes y estudiantes de ingeniería y de otras carreras, que su pensum incluya matemáticas en las cuales se deban desarrollar los temas de algebra matricial y trigonometría.

Que los estudiantes interesados en aprender a realizar operaciones de ley de cosenos, sumas y divisiones de matrices lo puedan realizar de forma interactiva a través del juego.

Especificación Técnica (Hardware y Software)

Los requisitos para que el sistema pueda ser ejecutado adecuadamente son:

- Procesador: Intel Pentium III 800 MHz (800MHz Intel Pentium III u otro equivalente)
- RAM: 512 MB
- Espacio en disco: 750 MB
- Sistema Operativo: Windows 7, Windows XP, Windows Vista) Windows XP Profesional SP3/Vista SP1/Windows 7 Professional)
- Resolución gráfica: 1024 x 728
- Navegador de internet: Google Chrome, Microsoft Edge, Mozilla Firefox, Vivaldi u Opera
- Herramientas: Java y algún IDE que corra el mismo, en este caso se utilizo NetBeans 8.2.

Lógica del Programa

```

public class Proyecto1 {

    static int [][] tablero = new int [8][8]; //Control de las posiciones
    static int [][] tablero2 = new int [8][8]; // Penalizacion
    static boolean verificarpenal1 = false; //facil
    static boolean verificarpenal2 = false; //inter
    static boolean verificarpenal3 = false; //dificil
    static boolean nopenal1 = true;
    static boolean nopenal2 = true;
    static boolean nopenal3 = true;
    static boolean nopenal4 = true;
    static boolean nopenal5 = true;
    static boolean nopenal6 = true;
    static boolean nopenal7 = true;
    static boolean nopenal8 = true;
    static boolean nopenal9 = true;
    static int sumadado=0;
    static int contfacil=0;
    static int continter=0;
    static int contdif=0;
    static double alphas; static double ar2; static double gamma1;
    static double alphas; static double cr; static double bethar;
    static double bethar; static double br; static double gamma1;
    static int [][] suma1 = new int [5][5]; static int [][] suma2 = new int [5][5]; static int [][] respuesta1 = new int [5][5];
    static int [][] suma3 = new int [5][5]; static int [][] suma4 = new int [5][5]; static int [][] respuesta2 = new int [5][5];
    static int [][] suma5 = new int [5][5]; static int [][] suma6 = new int [5][5]; static int [][] respuesta3 = new int [5][5];
    static int[][] dividendo2 = new int [4][4]; static int[][] divisora2 = new int [4][4]; static double[][] resultado2= new double[4][4];
    static int[][] dividendo0 = new int [4][4]; static int[][] divisora0 = new int [4][4]; static double[][] respuesta0= new double[4][4];
    static int[][] dividendo1 = new int [4][4]; static int[][] divisora1 = new int [4][4]; static double[][] resultado1= new double[4][4];
}

```

De primero declaro las variables globales que me serán útiles, principalmente como para generar reporte, utilizar en varios procedimientos, etc.

```

public static void main(String[] args) {
    int menu=0;
    Scanner lector = new Scanner(System.in);

    do{
        System.out.println("===== MENU PRINCIPAL =====");
        System.out.println("1. Iniciar Juego");
        System.out.println("2. Retomar Juego");
        System.out.println("3. Generar Reportes");
        System.out.println("4. Salir");
        System.out.println("");
        try{
            menu = lector.nextInt();
        }catch(Exception error){
            lector.nextLine();
            System.out.println("Ingrese un valor numerico");
            System.out.println("");
        }
        switch(menu){
            case 1:
                iniciarJuego();
                break;
            case 2:
                retomarJuego();
            case 3:
                generarReportes();
            case 4:
                break;
            default:
                System.out.println("Seleccione un numero entre 1 y 4");
                System.out.println("");
                break;
        }
    }while(menu != 4);
}

```

En este parte del main, despliegue el menú a utilizar, que básicamente tenía cuatro opciones Iniciar juego, Retornar juego, generar reportes y salir.

```

public static void iniciarJuego(){
    sumadad=0;
    boolean verificarpenal1 = false; //facil
    boolean verificarpenal2 = false; //inter
    boolean verificarpenal3 = false;
    boolean nopenal1= true;
    boolean nopenal2= true;
    boolean nopenal3= true;
    boolean nopenal4= true;
    boolean nopenal5= true;
    boolean nopenal6= true;
    boolean nopenal7= true;
    boolean nopenal8= true;
    boolean nopenal9= true;
    int contfacil=0;
    int continer=0;
    int contdif=0;
    int contador =1;
    Random penalizacion = new Random();
    for(int i=0; i<tablero.length; i++){
        for (int j = 0; j < tablero[i].length; j++){
            tablero[i][j] = contador;
            contador++;
        }
    }
    int bodega=0;
    for(int i=0; i<tablero2.length; i++){
        int restriccion =0;
        for (int j = 0; j < tablero2[i].length; j++){
            tablero2[0][0]=0;
            if (restriccion<=3){
                tablero2[i][j] = penalizacion.nextInt(2);
            }
            if(tablero2[i][j]==1){
                restriccion = restriccion;
                restriccion++;
            }
        }
    }
}

```

En esta parte reinicio las variables, para iniciar de nuevo el juego, así como establezco el tablero de posición y de penalización.

```

boolean derecha = false;
for (int i = tablero.length-1; i >=0; i--){
    System.out.println("-----");
    if(derecha){
        for (int j=0; j<tablero[i].length; j++){
            if(i==0&&j==0){
                System.out.print("#"+"\t"+tablero[i][j]+" |");
            }
            else{
                System.out.print("\t"+tablero[i][j]+" |");
            }
        }
        System.out.println("");
        for (int j=0; j<tablero2[i].length; j++){
            bodega= tablero2[i][j];
            if(bodega==0){
                String reemplazo = " ";
                System.out.print("\t "+reemplazo+" |");
            }
            else{
                String reemplazo = "#";
                System.out.print("\t "+reemplazo+" |");
            }
        }
    }else{
        for (int j = tablero[i].length-1; j >= 0; j--){
            System.out.print("\t"+tablero[i][j]+" |");
        }
        System.out.println("");
        for (int j = tablero2[i].length-1; j >= 0; j--){
            bodega= tablero2[i][j];
            if(bodega==0){
                String reemplazo = " ";
                System.out.print("\t "+reemplazo+" |");
            }
            else{
                String reemplazo = "#";
                System.out.print("\t "+reemplazo+" |");
            }
        }
    }
}
System.out.println("");
System.out.println("-----");
derecha != derecha;

```

En esta parte de código se imprime el tablero inicial, la ficha en su posición inicial, así como se apoya de un término boolean para el recorrido.

```

System.out.println("");
System.out.println("");
movimiento();
    System.out.println("Usted ha ganado el juego, Felicidades");
}

```

Y por último en iniciar juego llamo al movimiento, que es donde realiza todo el movimiento las fichas y por último felicito al jugador.

```

boolean derecha = false;
int bodega=0;

while(sumadado<64){
verficarpenal1=false;
verficarpenal3=false;
verficarpenal2=false;

String nombre=null;
Scanner teclado = new Scanner(System.in);
System.out.println("");
System.out.println("Seleccione para la siguiente tirada");
System.out.println("p. MENU PRINCIPAL");
System.out.println("Presione enter para tirar el dado");
try{
    nombre = teclado.nextLine();
}catch(Exception error){
    teclado.nextLine();
    System.out.println("Ingrese un caracter valido");
    System.out.println("");
}

if(!nombre.equals("p")){
    int dado = 0;
Random dadol= new Random();
dado = (int) (dadol.nextInt(5)+2);
sumadado+= dado;

System.out.println("El valor que devuelve el dado es: "+dado);
System.out.println("");
System.out.println("-----");
for (int i = tablero.length-1; i >=0; i--){
    if(derecha){
        for (int j=0; j<tablero[i].length;j++){
            if(j==sumadado && i==0){
                System.out.print("@"+"\t"+tablero[i][j]+" |");
                if(tablero2[i][j]==1){
                    verficarpenal1=true;
                }
            }
            else if(j==sumadado-16 && i==2){
                System.out.print("@"+"\t"+tablero[i][j]+" |");
                if(tablero2[i][j]==1){
                    verficarpenal3=true;
                }
            }
        }
    }
}
}

```

Aquí uso un ciclo while hasta que la ficha termine el proceso del juego, así como esta la parte de acción si regresar al menú principal o tirar el dado utilizando un número random, luego sumo lo que me dio el dado para encontrar la posición de la ficha en mi tablero, realizo la impresión y comparo valores con el tablero de penalización para encontrar si hay penalización en esa casilla.

```

for (int i = tablero.length-1; i >=0; i--){
    if(derecha){
        for (int j=0; j<tablero[i].length;j++){
            if(j==sumadado && i==0){
                System.out.print("@+"\t"+tablero[i][j]+" |");
                if(tablero2[i][j]==1){
                    verificarpenal1=true;
                }
            }
            else if(j==sumadado-16 && i==2){
                System.out.print("@+"\t"+tablero[i][j]+" |");
                if(tablero2[i][j]==1){
                    verificarpenal2=true;
                }
            }
            else if(j==sumadado-32 && i==4){
                System.out.print("@+"\t"+tablero[i][j]+" |");
                if(tablero2[i][j]==1){
                    verificarpenal2=true;
                }
            }
            else if(j==sumadado-48 && i==6){
                System.out.print("@+"\t"+tablero[i][j]+" |");
                if(tablero2[i][j]==1){
                    verificarpenal3=true;
                }
            }
            else{
                System.out.print("\t"+tablero[i][j]+" |");
            }
        }
        System.out.println("");
        for (int j=0; j<tablero2[i].length;j++){
            bodega= tablero2[i][j];
            if(bodega==0){
                String reemplazo = " ";
                System.out.print("\t "+reemplazo+" |");
            }
            else{
                String reemplazo = "#";
                System.out.print("\t "+reemplazo+" |");
            }
        }
    }
}

```

Para la impresión de ambos tableros uso for anidados, ya sea que venga para la derecha o para la izquierda, y utilizo if para saber si la posición [i][j] es la que satisface mi ficha, así como activo un booleano si se cae en una casilla con penalización.


```

if (verficarpenal1==true) {
    if(contfacil<2){
        System.out.println("");Has caido en una penalizacion facil!");
        leykos();
        contfacil= contfacil;
        contfacil++;
    }
}

if (verficarpenal2==true) {
    if(continter<2){
        System.out.println("");Has caido en una penalizacion intermedia!");
        suma();
        continter= continter;
        continter++;
    }
}

if (verficarpenal3==true) {
    if(contdif<2){
        System.out.println("");Has caido en una penalizacion dificil!");
        division();
        contdif= contdif;
        contdif++;
    }
}
}

```

Con el booleano obtenido, comparo si necesito ejecutar algún tipo de penalización.

```

if(nombre.equals("p")) {
    int menu=0;
    Scanner lector = new Scanner(System.in);

    do{
        System.out.println("==== MENU PRINCIPAL =====");
        System.out.println("1. Iniciar Juego");
        System.out.println("2. Retomar Juego");
        System.out.println("3. Generar Reportes");
        System.out.println("4. Salir");
        System.out.println("");
        try{
            menu = lector.nextInt();
        }catch(Exception error){
            lector.nextLine();
            System.out.println("Ingrese un valor numerico");
            System.out.println("");
        }
        switch(menu) {
            case 1:
                iniciarJuego();
                break;
            case 2:
                retomarJuego();
            case 3:
                generarReportes();
            case 4:
                salida();
            default:
                System.out.println("Seleccione un numero entre 1 y 4");
                System.out.println("");
                break;
        }
    }while(menu != 4);
}

```

Luego ya si el usuario ingreso p, despliego nuevamente el menú.

```

public static void retomarJuego() {
    movimiento();
}

```

Para retornar juego, solamente llamó otra vez al movimiento ()

```

public static void salida() {
    System.exit(0);
}

```

De ser necesaria la salida, sólo se realiza.

```

public static void generarReportes(){
    FileWriter fichero=null;
    PrintWriter pw=null;

    String texto = "<!DOCTYPE html>\n"
        + "<html>\n"
        + "<head>\n"
        + "    <meta charset=\"utf-8\">\n"
        + "    <title>Reporte 1</title>\n"
        + "<style type=\"text/css\">\n"
        + "body{\n"
        + "    margin: 0;\n"
        + "    font-family: times, serif;\n"
        + "    background-color: #E1DBB4; \n"
        + "}\n"
        + "\n"
        + "</style>\n"
        + "\n"
        + "</head>\n"
        + "<body>\n"
        + " \n"
        + "<h1 align=\"center\">Reporte 1</h1>\n"
        + "\n"
        + "<h2>1. Operación</h2>\n"
        + "<h3>1.1. Operaciones faciles</h3>\n "
        + "<h4>1.1.1. Operacion facil no.1</h4>\n "
        + "<h5>Se tiene un triangulo escaleno con lado A=15, lado C=20 y angulo alpha=25</h5>\n "
        + "<h4>1.1.2. Operacion facil no.2</h4>\n "
        + "<h5>Se tiene un triangulo escaleno con lado B=10, lado C=20 y angulo betha=30</h5>\n "
        + "<h4>1.1.3. Operacion facil no.3</h4>\n "
        + "<h5>Se tiene un triangulo escaleno con lado B=10, lado C=20 y angulo betha=30</h5>\n "
        + "<h3>1.2. Operaciones intermedias</h3>\n "
        + "<h4>1.2.1. Operacion intermedia no.1</h4>\n ";

```

El código para generar reportes

```

        for (int i = 0; i < suma1.length; i++) {
            texto+= "<br></br>";

            for (int j = 0; j < suma1.length; j++) {
                texto+= "(" + suma1[j][i] + ")";
            }
        }

        texto+="

##### 


```

Utilizando las variables globales, que también serán utilizadas en los diferentes tipos de penalizaciones, así sucesivamente con el código de generar reportes.

```

        + "</html>";

    try
    {
        fichero= new FileWriter("D:\\UNIVERSIDAD\\2021 Segundo Semestre\\Introducción a la Programación y Computación I\\Proyecto 1\\Proyecto1/Reporte 1.html");
        pw = new PrintWriter(fichero);

        for (int i=0; i<=0; i++)
            pw.println(texto);

    }catch (Exception e){
        e.printStackTrace();
    }finally {
        try{
            if (null!=fichero)
                fichero.close();
        } catch (Exception e2){
            e2.printStackTrace();
        }
    }
}
}

```

Y el fin del código del reporte con su lugar para llegada del file.

```

public static int casteardado(int sumadado){
    if(sumadado==0){
        sumadado=7;
        return sumadado;
    }
    if(sumadado==1){
        sumadado=6;
        return sumadado;
    }
    if(sumadado==2){
        sumadado=5;
        return sumadado;
    }
    if(sumadado==3){
        sumadado=4;
        return sumadado;
    }
    if(sumadado==4){
        sumadado=3;
        return sumadado;
    }
    if(sumadado==5){
        sumadado=2;
        return sumadado;
    }
    if(sumadado==6){
        sumadado=1;
        return sumadado;
    }
    if(sumadado==7){
        sumadado=0;
        return sumadado;
    }

    return sumadado;
}

```

Este método es utilizado para emparejar la posición del recorrido, cuando es al revés de mi movimiento original

```
public static void leycos() {
    int opcion;
    int aleatorio;

    Random opfacil = new Random();
    opcion = opfacil.nextInt(3);

    if(opcion==0 && nopenal1==false){
        opcion = opfacil.nextInt(2)+1;
    }
    else if (opcion==1 && nopenal2==false){
        aleatorio = opfacil.nextInt(2);
        if(aleatorio==0){
            opcion=0;
        }
        else{
            opcion=2;
        }
    }
    if(opcion==2 && nopenal3==false){
        opcion = opfacil.nextInt(2);
    }
}
```

Para obtener al azar alguna operación de realizar de cualquier tipo de penalización utilizando un número random y si ya fue seleccionada, como requiere un booleano, se utilizan if para complacer las necesidades.

```
if (opcion==0 && nopenal1==true){
    double a= 15;
    double c= 20;
    double alpha= 25;
    double b; double betha; double gamma;
    System.out.println("Se tiene un triangulo escaleno con lado A=15, lado C=20 y angulo alpha=25");
    System.out.println("Calcule: el Lado B, angulo betha y angulo gamma");
    b= Math.sqrt((a*a)+(c*c)-2*a*c*Math.cos(alpha*Math.PI/180));
    betha= Math.acos(((a*a)-(c*c)-(b*b))/(-2*b*c))*180/Math.PI;
    gamma = 180 - alpha -betha;

    b= Math.round(b*1000.0)/1000.0;
    betha= Math.round(betha*1000.0)/1000.0;
    gamma= Math.round(gamma*1000.0)/1000.0;

    System.out.println("");
    System.out.println("Lado B: "+b);
    System.out.println("Angulo betha: "+betha);
    System.out.println("Angulo gamma: "+gamma);
    nopenal1=false;
}
```

Para las tres operaciones de ley de cosenos se utilizo el despeje para el lado, así

como un ángulo, para el segundo ángulo se utilizo la suma de ángulos internos de un triángulo, para redondear e imprimir resultados, se redondeo a tres decimales.

```
if (opcion==0 && nopenal4==true){
    sumal [0][0]=7; sumal [1][0]=48; sumal [2][0]=5; sumal [3][0]=0; sumal [4][0]=1;
    sumal [0][1]=57; sumal [1][1]=8; sumal [2][1]=4; sumal [3][1]=6; sumal [4][1]=14;
    sumal [0][2]=0; sumal [1][2]=5; sumal [2][2]=6; sumal [3][2]=78; sumal [4][2]=15;
    sumal [0][3]=21; sumal [1][3]=14; sumal [2][3]=8; sumal [3][3]=19; sumal [4][3]=54;
    sumal [0][4]=32; sumal [1][4]=20; sumal [2][4]=26; sumal [3][4]=47; sumal [4][4]=12;

    suma2 [0][0]=9; suma2 [1][0]=5; suma2 [2][0]=2; suma2 [3][0]=1; suma2 [4][0]=8;
    suma2 [0][1]=4; suma2 [1][1]=2; suma2 [2][1]=3; suma2 [3][1]=47; suma2 [4][1]=8;
    suma2 [0][2]=48; suma2 [1][2]=55; suma2 [2][2]=32; suma2 [3][2]=19; suma2 [4][2]=6;
    suma2 [0][3]=7; suma2 [1][3]=56; suma2 [2][3]=32; suma2 [3][3]=14; suma2 [4][3]=8;
    suma2 [0][4]=32; suma2 [1][4]=87; suma2 [2][4]=0; suma2 [3][4]=1; suma2 [4][4]=7;

    for (int i = 0; i<5; i++) {
        for (int j = 0; j < 5; j++){
            respuesta[i][j]=sumal[i][j]+suma2[i][j];
        }
    }
    System.out.println("");
    System.out.println("La respuesta correcta es: ");
    System.out.println("");
    for (int i = 0; i<5; i++) {
        for (int j = 0; j < 5; j++){
            System.out.print("(" +respuesta[j][i]+")");
            if (j==4){
                System.out.println("");
            }
        }
    }
    nopenal4=false;
}
```

Para la suma de matrices se ingresa manual, luego se utilizan for anidados para encontrar el resultado, así como para realizar la impresión.

```
dividendo0 [0][0]=5; dividendo0 [1][0]=10; dividendo0 [2][0]=1; dividendo0 [3][0]=3;
dividendo0 [0][1]=9; dividendo0 [1][1]=14; dividendo0 [2][1]=2; dividendo0 [3][1]=6;
dividendo0 [0][2]=7; dividendo0 [1][2]=8; dividendo0 [2][2]=15; dividendo0 [3][2]=3;
dividendo0 [0][3]=6; dividendo0 [1][3]=8; dividendo0 [2][3]=9; dividendo0 [3][3]=2;

divisora0 [0][0]=5; divisora0 [1][0]=13; divisora0 [2][0]=9; divisora0 [3][0]=4;
divisora0 [0][1]=1; divisora0 [1][1]=9; divisora0 [2][1]=6; divisora0 [3][1]=3;
divisora0 [0][2]=8; divisora0 [1][2]=11; divisora0 [2][2]=69; divisora0 [3][2]=33;
divisora0 [0][3]=25; divisora0 [1][3]=6; divisora0 [2][3]=7; divisora0 [3][3]=4;

btraspuesta [0][0]=divisora0 [0][0]; btraspuesta [1][0]=divisora0 [0][1]; btraspuesta [2][0]=divisora0 [0][2]; btraspuesta [3][0]=divisora0 [0][3];
btraspuesta [0][1]=divisora0 [1][0]; btraspuesta [1][1]=divisora0 [1][1]; btraspuesta [2][1]=divisora0 [1][2]; btraspuesta [3][1]=divisora0 [1][3];
btraspuesta [0][2]=divisora0 [2][0]; btraspuesta [1][2]=divisora0 [2][1]; btraspuesta [2][2]=divisora0 [2][2]; btraspuesta [3][2]=divisora0 [2][3];
btraspuesta [0][3]=divisora0 [3][0]; btraspuesta [1][3]=divisora0 [3][1]; btraspuesta [2][3]=divisora0 [3][2]; btraspuesta [3][3]=divisora0 [3][3];
```

Para la división se ingreso la matriz divisora y la dividendo, la traspuesta de la divisora se consiguió igualando en la posición requerida.

```

adjunta[0][0] = "btraspuesta [1][1]*(btraspuesta [2][2]*btraspuesta [3][3])-(btraspuesta [2][3]* btraspuesta [3][2])-(btraspuesta [2][1]*(btraspuesta [1][2]*btraspuesta [3][3])-(btraspuesta [1][3]* btraspuesta [3][2]))+
adjunta[0][1] = "-1*(btraspuesta [1][0]*(btraspuesta [2][2]*btraspuesta [3][3])-(btraspuesta [2][3]* btraspuesta [3][2])-(btraspuesta [1][2]*btraspuesta [3][3])-(btraspuesta [1][3]* btraspuesta [3][2]))
adjunta[0][2] = "(btraspuesta [1][0]*(btraspuesta [2][1]*btraspuesta [3][3])-(btraspuesta [2][0]* btraspuesta [3][1])-(btraspuesta [2][1]*(btraspuesta [1][1]*btraspuesta [3][2])-(btraspuesta [1][2]* btraspuesta [3][1]))
adjunta[0][3] = "-1*(btraspuesta [1][0]*(btraspuesta [2][1]*btraspuesta [3][2])-(btraspuesta [2][0]* btraspuesta [3][1])-(btraspuesta [2][1]*(btraspuesta [1][1]*btraspuesta [3][2])-(btraspuesta [1][2]* btraspuesta [3][1]))+btraspu
adjunta[1][0] = "-1*(btraspuesta [0][1]*(btraspuesta [2][2]*btraspuesta [3][3])-(btraspuesta [2][3]* btraspuesta [3][2])-(btraspuesta [2][1]*(btraspuesta [0][2]*btraspuesta [3][3])-(btraspuesta [0][3]* btraspuesta [3][2]))
adjunta[1][1] = "-1*(btraspuesta [0][0]*(btraspuesta [2][2]*btraspuesta [3][3])-(btraspuesta [2][3]* btraspuesta [3][2])-(btraspuesta [2][1]*(btraspuesta [0][2]*btraspuesta [3][3])-(btraspuesta [0][3]* btraspuesta [3][2]))+btraspu
adjunta[1][2] = "-1*(btraspuesta [0][0]*(btraspuesta [2][1]*btraspuesta [3][3])-(btraspuesta [2][0]* btraspuesta [3][1])-(btraspuesta [2][1]*(btraspuesta [0][1]*btraspuesta [3][2])-(btraspuesta [0][1]* btraspuesta [3][1]))
adjunta[1][3] = "-1*(btraspuesta [0][0]*(btraspuesta [2][1]*btraspuesta [3][2])-(btraspuesta [2][0]* btraspuesta [3][1])-(btraspuesta [2][1]*(btraspuesta [0][1]*btraspuesta [3][2])-(btraspuesta [0][1]* btraspuesta [3][1]))+btraspu
adjunta[2][0] = "(btraspuesta [0][1]*(btraspuesta [1][2]*btraspuesta [3][3])-(btraspuesta [1][3]* btraspuesta [3][2])-(btraspuesta [1][1]*(btraspuesta [0][2]*btraspuesta [3][3])-(btraspuesta [0][3]* btraspuesta [3][2]))
adjunta[2][1] = "-1*(btraspuesta [0][0]*(btraspuesta [1][2]*btraspuesta [3][3])-(btraspuesta [1][3]* btraspuesta [3][2])-(btraspuesta [1][1]*(btraspuesta [0][2]*btraspuesta [3][3])-(btraspuesta [0][3]* btraspuesta [3][2]))+btraspu
adjunta[2][2] = "(btraspuesta [0][1]*(btraspuesta [1][1]*btraspuesta [3][3])-(btraspuesta [1][0]* btraspuesta [3][1])-(btraspuesta [1][1]*(btraspuesta [0][1]*btraspuesta [3][2])-(btraspuesta [0][1]* btraspuesta [3][1]))
adjunta[2][3] = "-1*(btraspuesta [0][0]*(btraspuesta [1][1]*btraspuesta [3][2])-(btraspuesta [1][0]* btraspuesta [3][1])-(btraspuesta [1][1]*(btraspuesta [0][1]*btraspuesta [3][2])-(btraspuesta [0][1]* btraspuesta [3][1]))+btraspu
adjunta[3][0] = "-1*(btraspuesta [0][1]*(btraspuesta [1][2]*btraspuesta [3][3])-(btraspuesta [1][3]* btraspuesta [3][2])-(btraspuesta [1][1]*(btraspuesta [0][2]*btraspuesta [3][3])-(btraspuesta [0][3]* btraspuesta [3][2]))
adjunta[3][1] = "-1*(btraspuesta [0][0]*(btraspuesta [1][2]*btraspuesta [3][3])-(btraspuesta [1][3]* btraspuesta [3][2])-(btraspuesta [1][1]*(btraspuesta [0][2]*btraspuesta [3][3])-(btraspuesta [0][3]* btraspuesta [3][2]))+btraspu
adjunta[3][2] = "-1*(btraspuesta [0][0]*(btraspuesta [1][1]*btraspuesta [3][3])-(btraspuesta [1][0]* btraspuesta [3][1])-(btraspuesta [1][1]*(btraspuesta [0][1]*btraspuesta [3][2])-(btraspuesta [0][1]* btraspu
adjunta[3][3] = "-1*(btraspuesta [0][0]*(btraspuesta [1][1]*btraspuesta [3][2])-(btraspuesta [1][0]* btraspuesta [3][1])-(btraspuesta [1][1]*(btraspuesta [0][1]*btraspuesta [3][2])-(btraspuesta [0][1]* btraspu

```

La adjunta de la traspuesta se encontró por el método de cofactores.

```

d= ( btraspuesta [0][0]*adjunta[0][0])+( btraspuesta [1][0]*adjunta[1][0])+(btraspuesta [2][0]*adjunta[2][0])+( btraspuesta[3][0]*adjunta[3][0]);

```

El determinante multiplicando la primera fila de la traspuesta por la adjunta.

```

if(d!=0){
    for (int i = 0; i<4; i++) {
        for (int j = 0; j < 4; j++) {
            inversa[i][j] = adjunta[i][j] / d;
        }
    }
}

```

Se utilizo un if porque el determinante no puede ser igual a cero, con for anidados se encontró la inversa.

```

respuesta [0][0]=(inversa[0][0]*btraspuesta [0][0])+(inversad[1][0]*inversa[0][1])+(inversad[2][0]*inversa[0][2])+(inversad[3][0]*inversa[0][3]);
respuesta [1][0]=(inversad[0][0]*inversa[1][0])+(inversad[1][0]*inversa[1][1])+(inversad[2][0]*inversa[1][2])+(inversad[3][0]*inversa[1][3]);
respuesta [2][0]=(inversad[0][0]*inversa[2][0])+(inversad[1][0]*inversa[2][1])+(inversad[2][0]*inversa[2][2])+(inversad[3][0]*inversa[2][3]);
respuesta [3][0]=(inversad[0][0]*inversa[3][0])+(inversad[1][0]*inversa[3][1])+(inversad[2][0]*inversa[3][2])+(inversad[3][0]*inversa[3][3]);
respuesta [0][1]=(inversad[0][1]*inversa[0][0])+(inversad[1][1]*inversa[0][1])+(inversad[2][1]*inversa[0][2])+(inversad[3][1]*inversa[0][3]);
respuesta [1][1]=(inversad[0][1]*inversa[1][0])+(inversad[1][1]*inversa[1][1])+(inversad[2][1]*inversa[1][2])+(inversad[3][1]*inversa[1][3]);
respuesta [2][1]=(inversad[0][1]*inversa[2][0])+(inversad[1][1]*inversa[2][1])+(inversad[2][1]*inversa[2][2])+(inversad[3][1]*inversa[2][3]);
respuesta [3][1]=(inversad[0][1]*inversa[3][0])+(inversad[1][1]*inversa[3][1])+(inversad[2][1]*inversa[3][2])+(inversad[3][1]*inversa[3][3]);
respuesta [0][2]=(inversad[0][2]*inversa[0][0])+(inversad[1][2]*inversa[0][1])+(inversad[2][2]*inversa[0][2])+(inversad[3][2]*inversa[0][3]);
respuesta [1][2]=(inversad[0][2]*inversa[1][0])+(inversad[1][2]*inversa[1][1])+(inversad[2][2]*inversa[1][2])+(inversad[3][2]*inversa[1][3]);
respuesta [2][2]=(inversad[0][2]*inversa[2][0])+(inversad[1][2]*inversa[2][1])+(inversad[2][2]*inversa[2][2])+(inversad[3][2]*inversa[2][3]);
respuesta [3][2]=(inversad[0][2]*inversa[3][0])+(inversad[1][2]*inversa[3][1])+(inversad[2][2]*inversa[3][2])+(inversad[3][2]*inversa[3][3]);
respuesta [0][3]=(inversad[0][3]*inversa[0][0])+(inversad[1][3]*inversa[0][1])+(inversad[2][3]*inversa[0][2])+(inversad[3][3]*inversa[0][3]);
respuesta [1][3]=(inversad[0][3]*inversa[1][0])+(inversad[1][3]*inversa[1][1])+(inversad[2][3]*inversa[1][2])+(inversad[3][3]*inversa[1][3]);
respuesta [2][3]=(inversad[0][3]*inversa[2][0])+(inversad[1][3]*inversa[2][1])+(inversad[2][3]*inversa[2][2])+(inversad[3][3]*inversa[2][3]);
respuesta [3][3]=(inversad[0][3]*inversa[3][0])+(inversad[1][3]*inversa[3][1])+(inversad[2][3]*inversa[3][2])+(inversad[3][3]*inversa[3][3]);

System.out.println("");
System.out.println("La respuesta correcta es: ");
System.out.println("");
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) {
        System.out.print (" (" + Math.round (respuesta [j][i]*1000.0)/1000.0 + ")");
        if (j == 3) {
            System.out.println("");
        }
    }
}

```

Para la respuesta, se realizo la multiplicación de matrices en la posición requerida.