

Universidad Autónoma de Baja California

Ingeniero en computación



Programación Orientada a Objetos

Reporte Final De Proyecto

Nombre Del Proyecto: Mathematical Insanity Beta v1.0













Nombre: López Mercado Brayan

Matrícula: 1280838

Grupo: 542

Fecha de Entrega: 13 de junio de 2022

Índice

-  Descripción del proyecto... 3
-  Funcionalidades del sistema... 3
 - Aprendizaje... 4
-  Perfil de usuario... 5
 - Usuario Jugador... 5
-  Lógica del sistema... 5
-  Herramientas utilizadas... 5
 - Lenguaje de Programación... 5
 - Entorno de desarrollo... 5
 - Herramientas de diseño... 5
-  Diseño de la interfaz de usuario... 6
-  Formular la solución... 10
 - Arquitectura de clases... 10
 - Arquitectura de objetos... 12
-  Implementar la solución... 13
 - Clase Main... 13
 - Clase Ventana... 14
 - Clase VectorWindow... 15
 - Clase VectorGen... 19
 - Clase Score... 21
 - Clase QuadraticWindow... 21
 - Clase QuadraticGen... 25
 - Clase Menu... 26
 - Clase GaussWindow... 29
 - Clase GaussGen... 30
 - Clase exitWindow... 32
 - Clase ErrorInputWindow... 33
 - Clase aritmeticWindow... 33
 - Clase aritmeticGen... 37
-  Pruebas de ejecución... 38
 - Pruebas por parte de los programadores... 38
-  Resumen de los conceptos empleados de programación orientada a objetos... 40
-  Carpeta de Drive... 40
-  Repositorio... 40

Descripción Del Proyecto

- **¿Qué hará el proyecto?**
El proyecto tiene la meta de ser una herramienta para estudiar y practicar los temas de ecuaciones cuadráticas, vectores, números complejos, integrales definidas y aritmética simple.
- **¿Cómo funciona?**
La forma en la que funciona es haciendo uso del generador de números aleatorios, más específicamente, el constructor *Random*.
- **¿Qué acciones se llevarán a cabo?**
Las acciones que se llevan a cabo de manera interna sin contar el código que genera la parte visual, contiene métodos para solucionar los problemas que el mismo programa genera, por ejemplo, existe un método para resolución de producto cruz entre vectores.
- **Detalles en general, información extra, etc.:**
El programa, aunque tenga una interfaz excesivamente simple, no se ve tan mal comparado a como se vería si se utilizara en la consola del IDE.
El programa originalmente era más ambicioso en cuanto a funcionalidades, pero se descartaron por falta de tiempo y falta de conocimiento sobre cómo hacer el código para resolver el problema de manera óptima.
El nombre de Mathematical Insanity se debe a que el programador estaba escuchando la canción "*It's Almost Easy*" de Avenged Sevenfold mientras estaba pensando en el nombre.
- **¿Cuál es la relación entre los usuarios y los datos?**
La única relación que existe entre los usuarios y los del programa son los datos que captura el usuario al momento de resolver los problemas, todos los demás datos son generados por el programa.

Funcionalidades Del Sistema

- 1- Función de sumatoria para integrales definidas.
- 2- Selector de dificultad (fácil y difícil).
- 3- Función mousePressed para cambiar el estado de la interfaz.
- 4- Inclusión de estados de la interfaz principal.
- 5- Clase quadraticWindow para mostrar los problemas de ecuaciones cuadráticas.
- 6- Utilización de la Clase JFrame para diseñar las sub-interfaces gráficas.
- 7- Utilización de la Clase Canvas para la interfaz principal.
- 8- Función render para diseñar los botones de la interfaz principal.
- 9- Matriz de puntos para el Método de gauss-Kronrod.
- 10- Función para producto cruz entre vectores.

- 11-La pantalla de inicio será color verde claro.
- 12-La cantidad de problemas generados serán 5 por ronda.
- 13-No se podrá dejar respuesta en blanco.
- 14-La puntuación de los problemas será 100 puntos por pregunta.
- 15-En caso de que capture una letra en lugar de un número, se mostrara un mensaje de error.
- 16- El programa se ejecutará mientras el usuario lo desee.
- 17- El Selector de dificultad tendrá dos opciones: Normal y Difícil.
- 18- En los problemas de Integrales se darán 500 puntos si la respuesta es correcta.
- 19- La casilla de puntuación no se podrá modificar.
- 20- Se mostrará un mensaje de confirmación al presionar el botón de salida.
- 21- La casilla de las respuestas será blanca.
- 22- Botón de redirección al menú principal.

2.1 Aprendizaje

- El aprendizaje obtenido en el programa es más que nada un apoyo para repasar los temas que se incluyen en el programa.

Perfil De Usuario

- Rango de edad entre 16 y 20 años
- Nivel educativo de preparatorio o tronco común de universidad
- El estilo de aprendizaje no influye en nada al momento de usar el programa, aunque los que sean del tipo auditivo tendrán dificultades durante largos periodos de tiempo.
- Cualquier persona puede usar el programa, a no ser que tenga discapacidad visual.
- La única experiencia con equipo de cómputo requerida es saber usar el teclado y el mouse para moverse en el programa.
- Si el usuario ya hecho uso de plataformas de quiz en internet, no debería existir dificultad para utilizar el programa, en caso de que no, solo debe capturar datos en una casilla de texto, es como llenar un formulario en línea.
- Las preferencias de los usuarios objetivos tienen como base el gusto por las matemáticas y hacer uso de programas para la realización de problemas, no tendría caso que el usuario haga uso del programa en caso de que no le agrade mucho la idea de usar un programa que genera los problemas de forma aleatoria o simplemente que no le agrade resolver problemas fuera del horario de clase.

3.1 Usuario Jugador

- Las funcionalidades a las que tiene acceso el usuario del programa son: acceso al selector de dificultad y los tipos de problemas que se generan. No existen funcionalidades que sean exclusivas del administrador.

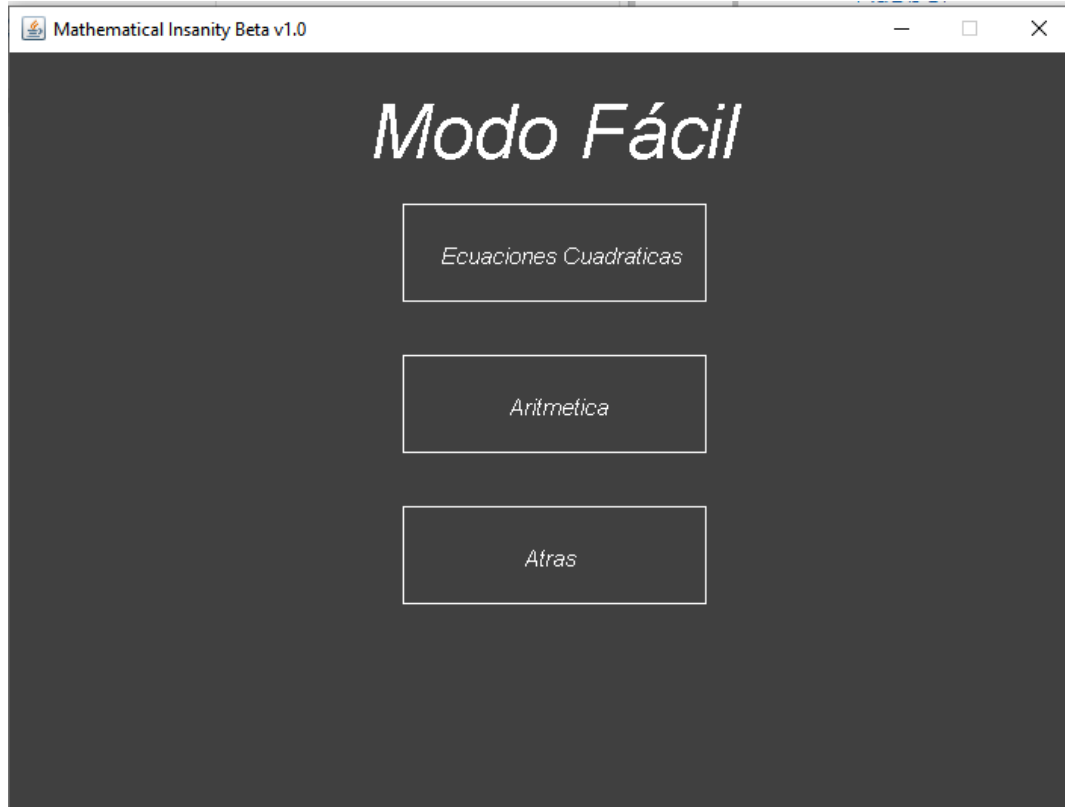
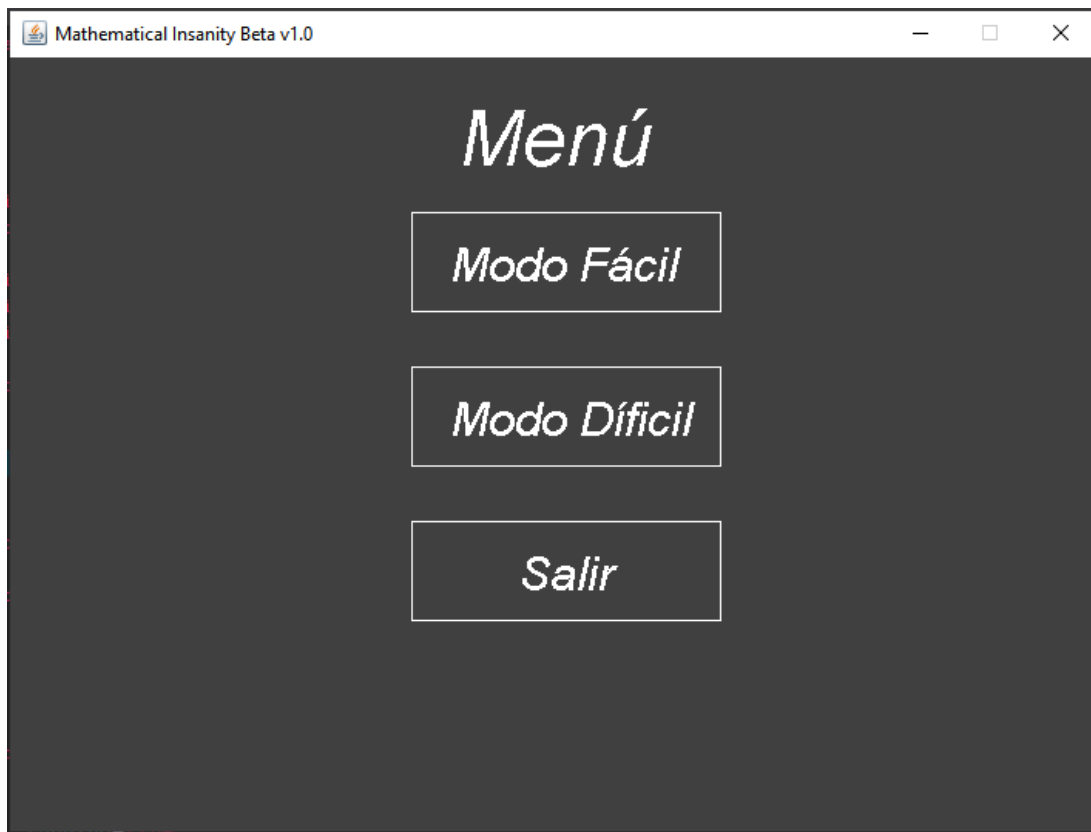
Lógica Del Sistema

La lógica del programa resumida de manera sencilla es: *“genero un problema y ahora resuélvelo no importa como lo hagas”*, o de una manera más detallada: se genera un vector con los números que forma parte del problema, se envían a un método que muestra el problema, se envía al método correspondiente para determinar la solución, y luego se le pide al usuario la respuesta del problema planteado y al final se muestra una ventana con la puntuación obtenida.

Herramientas Utilizadas

- **5.1 Lenguajes de Programación**
 - o Java
- **5.2 Entorno de desarrollo**
 - o Eclipse Mars 2
- **5.3 Herramientas de Diseño**
 - o Todo fue hecho solamente con código

Diseño De La Interfaz De Usuario



Modo Díficil


Integrales Definidas

*Numeros Complejos
y Vectores*

Atras

Cancelar


Salir


Ecuaciones Cuadraticas
—
□
×

Problema 1 :	$1x^2 - 6x + 7$	X1 :	<input type="text"/>	X2 :	<input type="text"/>
Problema 2 :	$-1x^2 + 15x - 4$	X1 :	<input type="text"/>	X2 :	<input type="text"/>
Problema 3 :	$-7x^2 + 14x - 8$	X1 :	<input type="text"/>	X2 :	<input type="text"/>
Problema 4 :	$-1x^2 - 1x + 10$	X1 :	<input type="text"/>	X2 :	<input type="text"/>
Problema 5 :	$10x^2 + 16x - 10$	X1 :	<input type="text"/>	X2 :	<input type="text"/>

Responder


Exit


Aritmetica
—
□
×

Problema 1 :	$-8 + (-8x - 18)x - 8$	=	<input type="text"/>
Problema 2 :	$-15 + (-15x - 7)x - 15$	=	<input type="text"/>
Problema 3 :	$-20 + (2x - 4)x - 2$	=	<input type="text"/>
Problema 4 :	$5 + (14x - 4)x - 14$	=	<input type="text"/>
Problema 5 :	$2 + (-9x - 19)x - 19$	=	<input type="text"/>

Responder


Exit

 Integrables Definidas

b =

dx =

a =

 Vectores y Numeros Complejos


Problema 1 : =()

Problema 2 : =()

Problema 3 : =

Problema 4 : =()

Problema 5 : =()

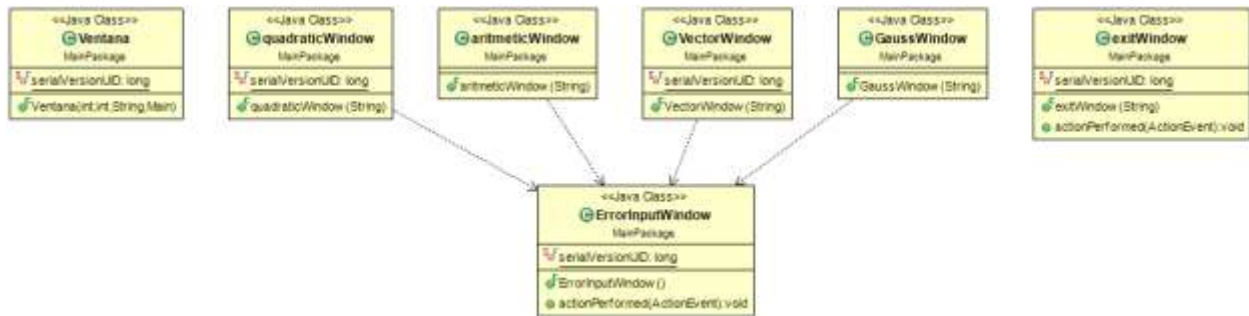
 Error

Debe Llenar Los Recuadros con números

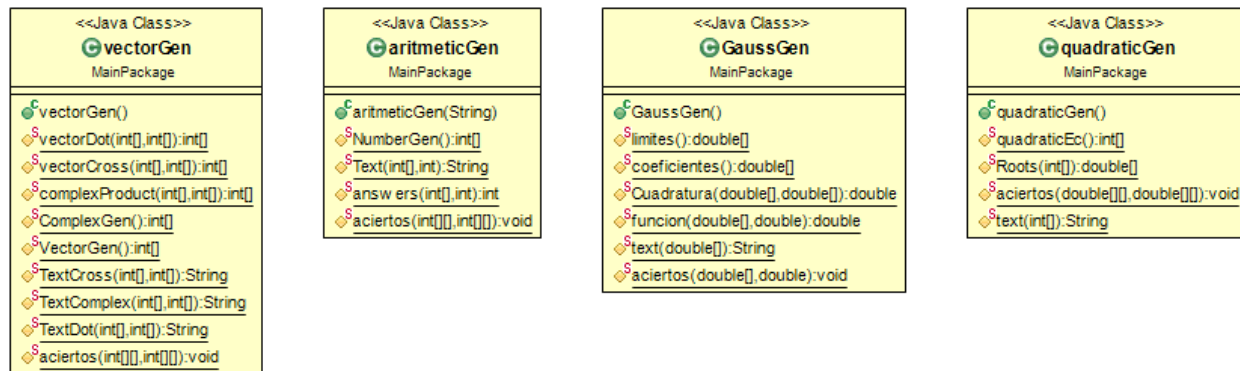
Formular La Solución

- 7.1 Arquitectura De Clases

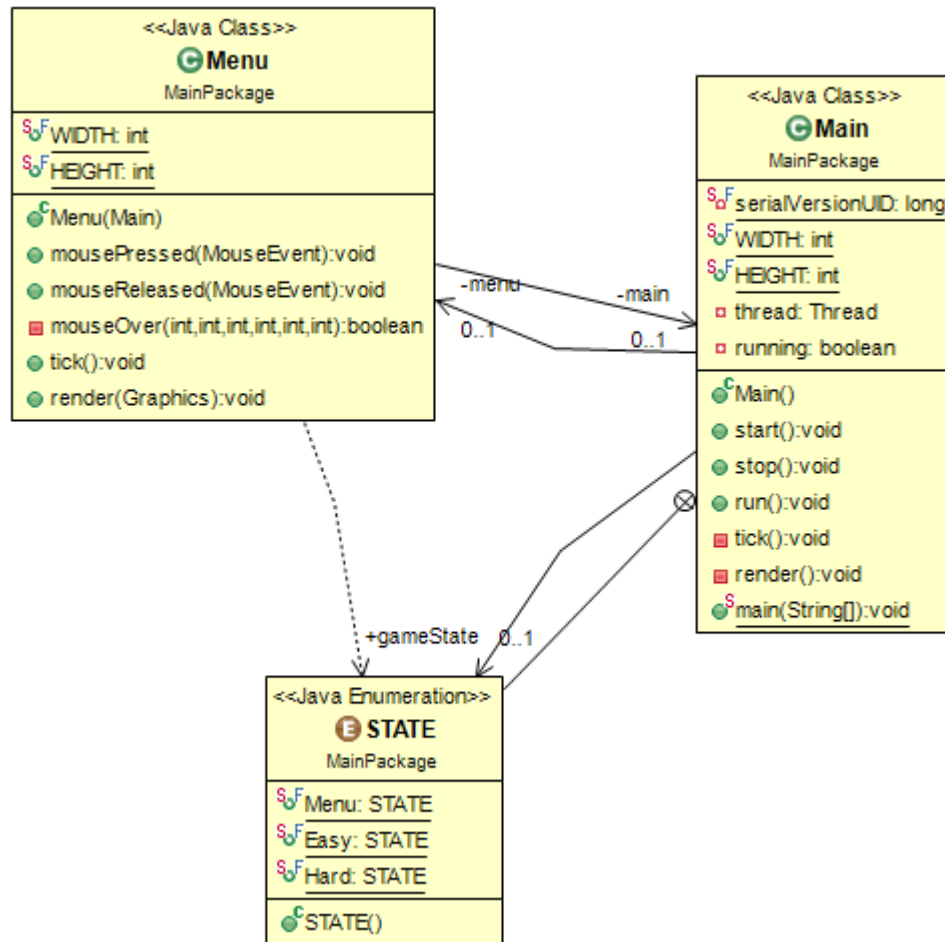
○ Interfaz De Usuario



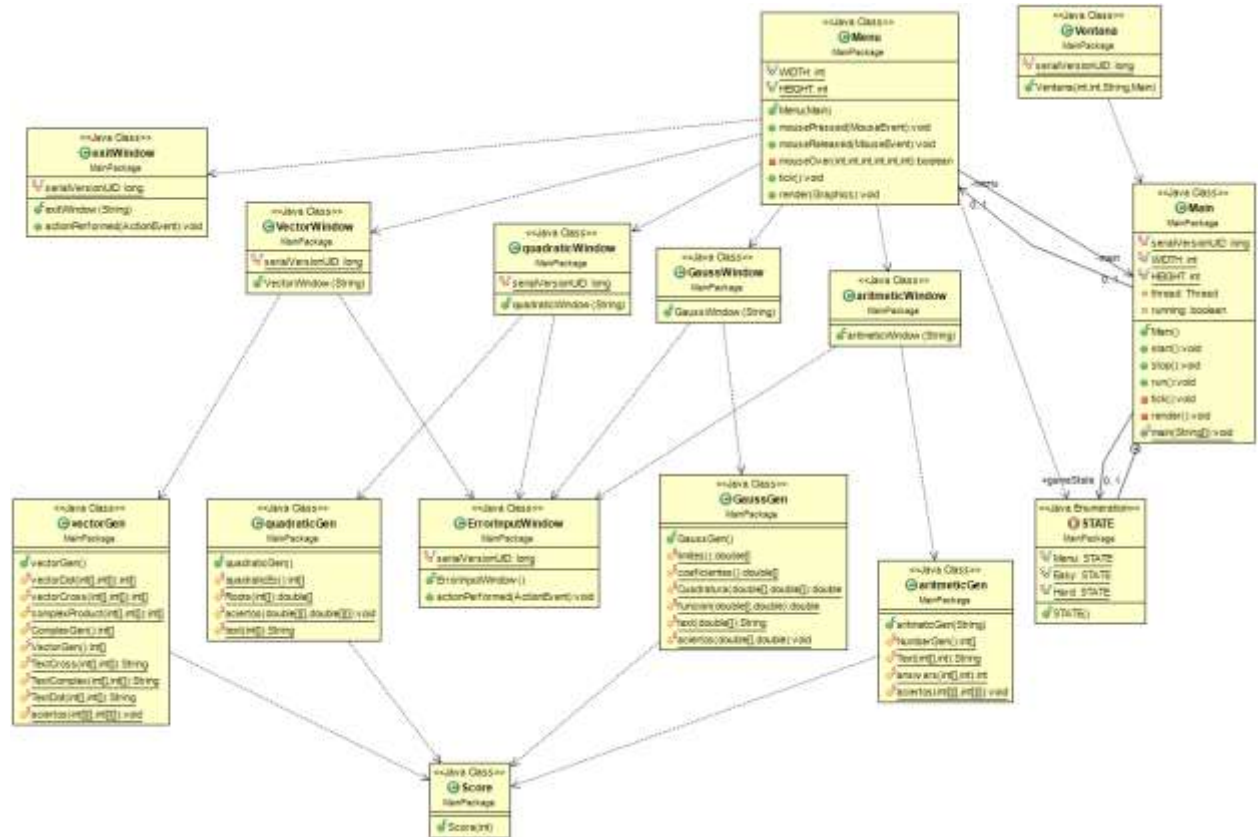
○ Datos



○ Lógica



- General



- 7.2 Arquitectura de Objetos

- Main
 - Menú
 - VectorWindow
 - ErrorInputWindow
 - Score
 - GaussWindow
 - ErrorInputWindow
 - Score
 - QuadraticWindow
 - ErrorInputWindow
 - Score
 - ArithmeticWindow
 - ErrorInputWindow
 - Score

Implementar La Solución

Clase Main

```
package MainPackage;

import java.awt.Canvas;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.image.BufferStrategy;

import MainPackage.Main.STATE;

public class Main extends Canvas implements Runnable{

    private static final long serialVersionUID = 5050370347925006000L;
    public static final int WIDTH=720,HEIGHT=WIDTH/12*9;

    private Thread thread;
    private boolean running=false;
    private Menu menu;

    public enum STATE{
        Menu,
        Easy,
        Hard,
    }

    public STATE gameState= STATE.Menu;

    public Main(){
        new Ventana(WIDTH,HEIGHT, "Mathematical Insanity Beta v1.0",this);
        menu = new Menu(this);
        this.addMouseListener(menu);
    }

    public synchronized void start(){
        thread= new Thread(this);
        thread.start();
        running=true;
    }

    public synchronized void stop(){
        try{
            thread.join();
            running=false;
        }catch(Exception e){
            e.printStackTrace();
        }
    }

    public void run(){
        long lastTime=System.nanoTime();
        double amountOfTicks=60.0;
        double ns=1000000000/amountOfTicks;
        double delta=0;
        long timer=System.currentTimeMillis();
        int frames=0;
        while(running){
            long now=System.nanoTime();
            delta+=(now-lastTime)/ns;
```

```

        lastTime=now;
        while(delta>=1){
            tick();
            delta--;
        }
        if(running)
            render();
        frames++;
        if(System.currentTimeMillis()-timer>1000){
            timer+=1000;
            frames=0;
        }
    }
    stop();
}

private void tick(){
    if(gameState==STATE.Menu){
        menu.tick();
    }
}

private void render(){
    BufferStrategy bs= this.getBufferStrategy();
    if(bs==null){
        this.createBufferStrategy(3);
        return;
    }

    Graphics g= bs.getDrawGraphics();

    g.setColor(Color.DARK_GRAY);
    g.fillRect(0,0,WIDTH,HEIGHT);
    if(gameState==STATE.Menu || gameState==STATE.Hard
        || gameState==STATE.Easy){
        menu.render(g);
    }
    g.dispose();
    bs.show();
}

public static void main(String[] args) {
    new Main();
}
}

```

Clase Ventana

```
package MainPackage;
```

```
import java.awt.Dimension;
```

```
import javax.swing.JFrame;
```

```
public class Ventana extends JFrame{
```

```
    private static final long serialVersionUID = -6768037413073049807L;
```

```
    public Ventana(int width,int height, String title, Main game){
```

```
        JFrame frame= new JFrame(title);
```

```

        frame.setPreferredSize(new Dimension(width,height));
        frame.setMaximumSize(new Dimension(width,height));
        frame.setMinimumSize(new Dimension(width,height));

        frame.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        frame.setResizable(false);
        frame.setLocationRelativeTo(null);
        frame.add(game);
        frame.setVisible(true);
        game.start();
    }
}

```

Clase VectorWindow

```

package MainPackage;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class VectorWindow extends JFrame{

    private static final long serialVersionUID = -3823889479953068613L;

    public VectorWindow (String title){
        JFrame frame = new JFrame(title);
        JButton b=new JButton("Responder");
        JButton ex=new JButton("Exit");

        JLabel label1= new JLabel("Problema 1 : ");
        label1.setBounds(10, 10, 90, 90);
        JLabel label2= new JLabel("Problema 2 : ");
        label2.setBounds(10, 40, 90, 90);
        JLabel label3= new JLabel("Problema 3 : ");
        label3.setBounds(10, 70, 90, 90);
        JLabel label4= new JLabel("Problema 4 : ");
        label4.setBounds(10, 100, 90, 90);
        JLabel label5= new JLabel("Problema 5 : ");
        label5.setBounds(10, 130, 90, 90);

        JLabel labelR1= new JLabel("=( ");
        labelR1.setBounds(225, 10, 90, 90);
        JLabel labelR2= new JLabel("=( ");
        labelR2.setBounds(225, 40, 90, 90);
        JLabel labelR4= new JLabel(")");
        labelR4.setBounds(360, 10, 90, 90);
        JLabel labelR5= new JLabel(")");
        labelR5.setBounds(360, 40, 90, 90);
        JLabel labelR6= new JLabel("= ");
        labelR6.setBounds(225,75,90,90);
        JLabel labelR7= new JLabel("=( ");
        labelR7.setBounds(225,105,90,90);
        JLabel labelR8= new JLabel(") ");
        labelR8.setBounds(315,105,90,90);
        JLabel labelR9= new JLabel("=( ");
    }
}

```

```

labelR9.setBounds(225,135,90,90);
JLabel labelR10 = new JLabel(" ");
labelR10.setBounds(315,135,90,90);

// Generador De Vectores
int [] vector1= new int[3];
int [] vector2 = new int [3];
int [] vector3= new int[3];
int [] vector4 = new int [3];
int [] vector5 = new int [3];

int [] NumCX1 = new int [2];
int [] NumCX2 = new int [2];
int [] NumCX3 = new int [2];
int [] NumCX4 = new int [2];

int cross[] = new int[3];
int cross2[] = new int[3];
int productNComplex[] = new int[2];
int productNComplex2[] = new int[2];

int dot[]= new int[1];

vector1=vectorGen.VectorGen();
vector2=vectorGen.VectorGen();
vector3=vectorGen.VectorGen();
vector4=vectorGen.VectorGen();
vector5=vectorGen.VectorGen();
NumCX1=vectorGen.ComplexGen();
NumCX2=vectorGen.ComplexGen();
NumCX3=vectorGen.ComplexGen();
NumCX4=vectorGen.ComplexGen();

// Resultados De Los Problemas
int resultados[][] = new int [5][];

dot = vectorGen.vectorDot(vector5,vector3);
cross = vectorGen.vectorCross(vector1,vector2);
cross2 = vectorGen.vectorCross(vector4,vector3);
productNComplex= vectorGen.complexProduct(NumCX1,NumCX2);
productNComplex2= vectorGen.complexProduct(NumCX4,NumCX3);

resultados[0]=cross;
resultados[1]=cross2;
resultados[2]=dot;
resultados[3]=productNComplex;
resultados[4]=productNComplex2;

// Cuadros De Problemas
JTextField f1= new JTextField(vectorGen.TextCross(vector1,vector2));
f1.setBounds(90,50,135,20);
f1.setEditable(false);

JTextField f2= new JTextField(vectorGen.TextCross(vector4,vector3));
f2.setBounds(90,80,135,20);
f2.setEditable(false);

JTextField f3= new JTextField(vectorGen.TextDot(vector5, vector3));
f3.setBounds(90,110,135,20);
f3.setEditable(false);

```

// 4,3


```
JTextField f4= new JTextField(vectorGen.TextComplex(NumCX1, NumCX2));
f4.setBounds(90,140,135,20);
f4.setEditable(false);
```

```
JTextField f5= new JTextField(vectorGen.TextComplex(NumCX4, NumCX3));
f5.setBounds(90,170,135,20);
f5.setEditable(false);
```

```
JTextField Af1= new JTextField(); // Cuadros de respuestas
Af1.setBounds(240,50,35,20);
JTextField Af2= new JTextField();
Af2.setBounds(240,80,35,20);
JTextField Af3= new JTextField();
Af3.setBounds(240,110,35,20);
JTextField Af4= new JTextField();
Af4.setBounds(240,140,35,20);
JTextField Af5= new JTextField();
Af5.setBounds(240,170,35,20);
JTextField Af6= new JTextField();
Af6.setBounds(280,50,35,20);
JTextField Af7= new JTextField();
Af7.setBounds(280,80,35,20);
JTextField Af9= new JTextField();
Af9.setBounds(280,140,35,20);
JTextField Af10= new JTextField();
Af10.setBounds(280,170,35,20);
JTextField Af11= new JTextField();
Af11.setBounds(320,50,35,20);
JTextField Af12= new JTextField();
Af12.setBounds(320,80,35,20);
```

```
frame.getContentPane().setBackground(Color.DARK_GRAY);
```

```
b.setBounds(90,210,100,30);
ex.setBounds(240,210,100,30);
```

```
label1.setForeground(Color.white);
label2.setForeground(Color.white);
label3.setForeground(Color.white);
label4.setForeground(Color.white);
label5.setForeground(Color.white);
labelR1.setForeground(Color.white);
labelR2.setForeground(Color.white);
labelR4.setForeground(Color.white);
labelR5.setForeground(Color.white);
labelR6.setForeground(Color.white);
labelR7.setForeground(Color.white);
labelR8.setForeground(Color.white);
labelR9.setForeground(Color.white);
labelR10.setForeground(Color.white);
```

```
b.setBackground(Color.white);
ex.setBackground(Color.white);
```

```
frame.add(f1);
frame.add(f2);
frame.add(f3);
frame.add(f4);
frame.add(f5);
frame.add(label1);
frame.add(label2);
```

```

frame.add(label3);
frame.add(label4);
frame.add(label5);
frame.add(Af1);
frame.add(Af2);
frame.add(Af3);
frame.add(Af4);
frame.add(Af5);
frame.add(Af6);
frame.add(Af7);
frame.add(Af9);
frame.add(Af10);
frame.add(Af11);
frame.add(Af12);
frame.add(labelR1);
frame.add(labelR2);
frame.add(labelR4);
frame.add(labelR5);
frame.add(labelR6);
frame.add(labelR7);
frame.add(labelR8);
frame.add(labelR9);
frame.add(labelR10);
frame.add(b);
frame.add(ex);

frame.setSize(450,300);
    frame.setResizable(false);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);

ActionListener a = new ActionListener(){
    @SuppressWarnings("deprecation")

    @Override
    public void actionPerformed(ActionEvent ae) {
        if(ae.getSource()==ex){
            frame.hide();
        }
        if(ae.getSource()==b){
            int a,b,c,d,e,f,g,h,i,j,k;

            int answerVec1[] = new int [3];
            int answerVec2[] = new int [3];

            int answerNC1[] = new int [2];
            int answerNC2[] = new int [2];
            int answerDot[] = new int [1];

            int answers[][] = new int [5][];

            try{
                a= Integer.parseInt(Af1.getText());
                b= Integer.parseInt(Af6.getText());
                c= Integer.parseInt(Af11.getText());

                d= Integer.parseInt(Af2.getText());
                e= Integer.parseInt(Af7.getText());
                f= Integer.parseInt(Af12.getText());

```

```

}
Clase VectorGen

```

```
crossNum[0] = (vector1[1]*vector2[2]) - (vector1[2]*vector2[1]);
crossNum[1] = (vector1[2]*vector2[0]) - (vector1[0]*vector2[2]);
crossNum[2] = (vector1[0]*vector2[1]) - (vector1[1]*vector2[0]);
```

```

        return crossNum;
    }

    protected static int [] complexProduct(int Nc1[], int Nc2[]){
        int complexNum[] = new int[2];

        complexNum[0] = (Nc1[0]*Nc2[0]) - (Nc1[1]*Nc2[1]);
        complexNum[1] = (Nc1[0]*Nc2[1]) + (Nc1[1]*Nc2[0]);

        return complexNum;
    }

    protected static int[] ComplexGen(){
        Random random = new Random();
        int NumCx [] = new int [2];
        for(int j=0;j<NumCx.length;j++){
            NumCx[j]=(random.nextInt(1+30)+1) - (random.nextInt(1+30)+1) ;
        }
        return NumCx;
    }

    protected static int[] VectorGen(){
        Random random = new Random();
        int Vector [] = new int [3];
        for(int j=0;j<Vector.length;j++){
            if(j==0){
                Vector[j]=(random.nextInt(1+30)+1);
            }
            else{
                Vector[j]=(random.nextInt(1+30)+1) - (random.nextInt(1+30)+1);
            }
        }
        return Vector;
    }

    protected static String TextCross(int[] vector,int[] vector2){
        String Text="("+ vector[0] + "," + vector[1] + "," + vector[2] + ")" + " x " +
            "(" + vector2[0] + "," + vector2[1] + "," + vector2[2] + ")";
        return Text;
    }

    protected static String TextComplex(int[] NumCX1,int[] NumCX2){

        String Text="("+ NumCX1[0] + " " + NumCX1[1] + "i )" + " * " + "(" + NumCX2[0] + " " + NumCX2[1] + "i )";

        if(NumCX1[1]>=0){
            Text="("+ NumCX1[0] + " " + NumCX1[1] + "i )" + " * " + "(" + NumCX2[0] + " " + NumCX2[1] + "i )";
        }
        else if(NumCX2[1]>=0){
            Text="("+ NumCX1[0] + " " + NumCX1[1] + "i )" + " * " + "(" + NumCX2[0] + " " + NumCX2[1] + "i )";
        }
        else if(NumCX1[1]>=0 && NumCX2[1]>=0){
            Text="("+ NumCX1[0] + " " + NumCX1[1] + "i )" + " * " + "(" + NumCX2[0] + " " + NumCX2[1] + "i )";
        }
        return Text;
    }

    protected static String TextDot(int[] vector,int[] vector2){
        String Text="("+ vector[0] + "," + vector[1] + "," + vector[2] + ")" + " * " +
            "(" + vector2[0] + "," + vector2[1] + "," + vector2[2] + ")";
        return Text;
    }

    protected static void aciertos (int answers[][], int resultados[][]){
        int points=0;

        for(int z=0;z<answers.length;z++){
            for(int y=0;y<answers[z].length;y++){

```

```

        if(answers[z][y]== resultados[z][y]){
            points+=100;
        }
    }
    }
    new Score(points);
}

```

Clase Score

```
package MainPackage;
```

```
import java.awt.Color;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JTextField;
```

```
public class Score {
```

```
    public Score(int points){
```

```
        JFrame frame = new JFrame ("Puntaje");
```

```
        frame.getContentPane().setBackground(Color.DARK_GRAY);
```

```
        JLabel label1= new JLabel ("Puntaje : ");
```

```
        label1.setForeground(Color.white);
```

```
        label1.setBounds(10,30, 90, 90);
```

```
        JTextField f1 = new JTextField();
```

```
        f1.setEditable(false);
```

```
        f1.setBounds(85,65,100,20);
```

```
        f1.setText(String.valueOf(points));
```

```
        frame.add(label1);
```

```
        frame.add(f1);
```

```
        frame.setSize(250,200);
```

```
        frame.setResizable(false);
```

```
        frame.setLayout(null);
```

```
        frame.setVisible(true);
```

```
    }
```

```
}
```

Clase QuadraticWindow

```
package MainPackage;
```

```
import java.awt.Color;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JTextField;
```

```
public class quadraticWindow extends JFrame{
```

```
    private static final long serialVersionUID = -6059720196692888831L;
```

```
    public quadraticWindow(String title){
```

```
JFrame frame = new JFrame(title);
JButton b1=new JButton("Responder");
JButton e=new JButton("Exit");
```

```
JLabel label1= new JLabel("Problema 1 : ");
label1.setBounds(10, 10, 90, 90);
JLabel label2= new JLabel("Problema 2 : ");
label2.setBounds(10, 40, 90, 90);
JLabel label3= new JLabel("Problema 3 : ");
label3.setBounds(10, 70, 90, 90);
JLabel label4= new JLabel("Problema 4 : ");
label4.setBounds(10, 100, 90, 90);
JLabel label5= new JLabel("Problema 5 : ");
label5.setBounds(10, 130, 90, 90);
```

```
JLabel labelR1= new JLabel("X1 : ");
labelR1.setBounds(210, 10, 90, 90);
JLabel labelR2= new JLabel("X1 : ");
labelR2.setBounds(210, 40, 90, 90);
JLabel labelR3= new JLabel("X1 : ");
labelR3.setBounds(210, 70, 90, 90);
JLabel labelR4= new JLabel("X1 : ");
labelR4.setBounds(210, 100, 90, 90);
JLabel labelR5= new JLabel("X1 : ");
labelR5.setBounds(210, 130, 90, 90);
```

```
JLabel labelW1= new JLabel("X2 : ");
labelW1.setBounds(350, 10, 90, 90);
JLabel labelW2= new JLabel("X2 : ");
labelW2.setBounds(350, 40, 90, 90);
JLabel labelW3= new JLabel("X2 : ");
labelW3.setBounds(350, 70, 90, 90);
JLabel labelW4= new JLabel("X2 : ");
labelW4.setBounds(350, 100, 90, 90);
JLabel labelW5= new JLabel("X2 : ");
labelW5.setBounds(350, 130, 90, 90);
```

```
int [] Ecu1= new int[3];
int [] Ecu2 = new int [3];
int [] Ecu3= new int[3];
int [] Ecu4 = new int [3];
int [] Ecu5 = new int [3];
```

```
double raices1 [] = new double[2];
double raices2 [] = new double[2];
double raices3 [] = new double[2];
double raices4 [] = new double[2];
double raices5 [] = new double[2];
```

```
double resultados[][]= new double [5][];
```

```
Ecu1=quadraticGen.quadraticEc();
Ecu2=quadraticGen.quadraticEc();
Ecu3=quadraticGen.quadraticEc();
Ecu4=quadraticGen.quadraticEc();
Ecu5=quadraticGen.quadraticEc();
```

```
raices1=quadraticGen.Roots(Ecu1);
raices2=quadraticGen.Roots(Ecu2);
raices3=quadraticGen.Roots(Ecu3);
```

```

raices4=quadraticGen.Roots(Ecu4);
raices5=quadraticGen.Roots(Ecu5);

resultados[0]=raices1;
resultados[1]=raices2;
resultados[2]=raices3;
resultados[3]=raices4;
resultados[4]=raices5;

JTextField f1= new JTextField(quadraticGen.text(Ecu1)); // Cuadros De Problemas
f1.setBounds(90,50,105,20);
f1.setEditable(false);

JTextField f2= new JTextField(quadraticGen.text(Ecu2));
f2.setBounds(90,80,105,20);
f2.setEditable(false);

JTextField f3= new JTextField(quadraticGen.text(Ecu3));
f3.setBounds(90,110,105,20);
f3.setEditable(false);

JTextField f4= new JTextField(quadraticGen.text(Ecu4));
f4.setBounds(90,140,105,20);
f4.setEditable(false);

JTextField f5= new JTextField(quadraticGen.text(Ecu5));
f5.setBounds(90,170,105,20);
f5.setEditable(false);

JTextField Af1= new JTextField(); // Cuadros de respuestas
Af1.setBounds(235,50,105,20);
JTextField Af2= new JTextField();
Af2.setBounds(235,80,105,20);
JTextField Af3= new JTextField();
Af3.setBounds(235,110,105,20);
JTextField Af4= new JTextField();
Af4.setBounds(235,140,105,20);
JTextField Af5= new JTextField();
Af5.setBounds(235,170,105,20);
JTextField Af6= new JTextField();
Af6.setBounds(375,50,105,20);
JTextField Af7= new JTextField();
Af7.setBounds(375,80,105,20);
JTextField Af8= new JTextField();
Af8.setBounds(375,110,105,20);
JTextField Af9= new JTextField();
Af9.setBounds(375,140,105,20);
JTextField Af10= new JTextField();
Af10.setBounds(375,170,105,20);

frame.getContentPane().setBackground(Color.DARK_GRAY);
b1.setBounds(100,210,100,30);
e.setBounds(310,210,100,30);

label1.setForeground(Color.white);
label2.setForeground(Color.white);
label3.setForeground(Color.white);
label4.setForeground(Color.white);
label5.setForeground(Color.white);
labelR1.setForeground(Color.white);
labelR2.setForeground(Color.white);

```

```

labelR3.setForeground(Color.white);
labelR4.setForeground(Color.white);
labelR5.setForeground(Color.white);
labelW1.setForeground(Color.white);
labelW2.setForeground(Color.white);
labelW3.setForeground(Color.white);
labelW4.setForeground(Color.white);
labelW5.setForeground(Color.white);

b1.setBackground(Color.white);
e.setBackground(Color.white);

frame.add(f1);
frame.add(f2);
frame.add(f3);
frame.add(f4);
frame.add(f5);
frame.add(Af1);
frame.add(Af2);
frame.add(Af3);
frame.add(Af4);
frame.add(Af5);
frame.add(Af6);
frame.add(Af7);
frame.add(Af8);
frame.add(Af9);
frame.add(Af10);

frame.add(label1);
frame.add(label2);
frame.add(label3);
frame.add(label4);
frame.add(label5);
frame.add(labelR1);
frame.add(labelR2);
frame.add(labelR3);
frame.add(labelR4);
frame.add(labelR5);
frame.add(labelW1);
frame.add(labelW2);
frame.add(labelW3);
frame.add(labelW4);
frame.add(labelW5);
frame.add(b1);
frame.add(e);

frame.setSize(550,300);
    frame.setResizable(false);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);

ActionListener a = new ActionListener(){
    @SuppressWarnings("deprecation")
    @Override
    public void actionPerformed(ActionEvent at) {
        if(at.getSource()==e){
            frame.hide();
        }
        if(at.getSource()==b1){
            double a,b,c,d,e,f,g,h,i,j;

```



```

double answerEc1[] = new double [2];
double answerEc2[] = new double [2];
double answerEc3[] = new double [2];
double answerEc4[] = new double [2];
double answerEc5[] = new double [2];

double answers[][] = new double [5][];

try{
    a= Double.parseDouble(Af1.getText());
    b= Double.parseDouble(Af6.getText());
    c= Double.parseDouble(Af2.getText());
    d= Double.parseDouble(Af7.getText());
    e= Double.parseDouble(Af3.getText());
    f= Double.parseDouble(Af8.getText());
    g= Double.parseDouble(Af4.getText());
    h= Double.parseDouble(Af9.getText());
    i= Double.parseDouble(Af5.getText());
    j= Double.parseDouble(Af10.getText());

    answerEc1 [0]=a; answerEc1 [1]=b;
    answerEc2 [0]=c; answerEc2 [1]=d;
    answerEc3 [0]=e; answerEc3 [1]=f;
    answerEc4 [0]=g; answerEc4 [1]=h;
    answerEc5 [0]=i; answerEc5 [1]=j;

    answers[0]=answerEc1;
    answers[1]=answerEc2;
    answers[2]=answerEc3;
    answers[3]=answerEc4;
    answers[4]=answerEc5;

    quadraticGen.aciertos(answers,resultados);
}
catch (Exception ext){
    ErrorInputWindow z= new ErrorInputWindow();
}
}

};
e.addActionListener(a);
b1.addActionListener(a);
}
}

```

Clase QuadraticGen

package MainPackage;

import java.util.Random;

public class quadraticGen {

 public quadraticGen(){

 }

 protected static int [] quadraticEc(){

 int terms[] = new int [3];

 int a=24,b=1,c=4;

 Random random = new Random();

 while(((b*b)<(4*a*c)) || a==0){

 a=(random.nextInt(1+30)+1) - (random.nextInt(1+30)+1);

 b=(random.nextInt(1+30)+1) - (random.nextInt(1+30)+1);

```

        c=(random.nextInt(1+30)+1) - (random.nextInt(1+30)+1);
    }
    terms[0]=a;
    terms[1]=b;
    terms[2]=c;
    return terms;
}

protected static double [] Roots (int [] ecu){
    double roots [] = new double [2];
    double disc,a,b;

    disc=Math.sqrt((ecu[1]*ecu[1])-(4*ecu[0]*ecu[2]));
    a=(-1*ecu[1] + disc)/(2*ecu[0]);
    b=(-1*ecu[1] - disc)/(2*ecu[0]);
    roots[0]=Math.round(a*100.0)/100.0;
    roots[1]=Math.round(b*100.0)/100.0;
    return roots;
}

protected static void aciertos (double answers[][], double resultados[][]){

    int points=0;
    for(int z=0;z<answers.length;z++){
        for(int y=0;y<answers[z].length;y++){
            if(answers[z][y]== resultados[z][y]){
                points+=50;
            }
        }
    }
    new Score(points);
}

protected static String text(int Ecu[]){
    String Text = Ecu[0] + "x^2 " + Ecu[1] + "x " + Ecu[2]; ;
    if(Ecu[1]>=0){
        Text=Ecu[0] + "x^2 + " + Ecu[1] + "x " + Ecu[2];
    }
    else if(Ecu[2]>=0){
        Text=Ecu[0] + "x^2 " + Ecu[1] + "x + " + Ecu[2];
    }
    return Text;
}
}

```

Clase Menú

```

package MainPackage;

import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;

import MainPackage.Main.STATE;

public class Menu extends MouseAdapter{
    public static final int WIDTH=540,HEIGHT=WIDTH/12*9;

    private Main main;

    public Menu(Main main){
        this.main=main;
    }

    public void mousePressed(MouseEvent e){
        int mx = e.getX();
    }
}

```

```

int my = e.getY();

if(main.gameState==STATE.Menu){
    if (mouseOver(mx,my,260, 300, 200, 64)){// Exit
        //System.exit(1);
        new exitWindow("Confirmar Salida");
    }
    if (mouseOver(mx,my,260, 100, 200, 64)){// Modo Facil
        main.gameState = STATE.Easy;
    }

    if (mouseOver(mx,my,260, 200, 200, 64)){// Modo Dificil
        main.gameState = STATE.Hard;
    }
}

else if (main.gameState==STATE.Hard){

    if (mouseOver(mx,my,260, 270, 200, 64)){// Retroceso Modo Dificil
        main.gameState = STATE.Menu;
        return;
    }

    if (mouseOver(mx,my,260, 185, 200, 64)){// Vectores
        new VectorWindow("Vectores y Numeros Complejos");
    }

    if (mouseOver(mx,my,260, 100, 200, 64)){// Integración
        new GaussWindow("Integrales Definidas");
    }
}

else if (main.gameState==STATE.Easy){
    if (mouseOver(mx,my,300, 300, 200, 64)){// Retroceso Modo Facil
        main.gameState = STATE.Menu;
        return;
    }

    if (mouseOver(mx,my,260, 100, 200, 64)){// Ecuaciones Cuadraticas
        new quadraticWindow("Ecuaciones Cuadraticas");
    }

    if (mouseOver(mx,my,260, 200, 200, 64)){// Aritmetica
        new arithmeticWindow("Aritmetica");
    }

}

}

public void mouseReleased(MouseEvent e){

}

private boolean mouseOver(int mx ,int my ,int x, int y, int width, int height){
    if (mx > x && mx<x+width){
        if (my>y && my<y+height){
            return true;
        }else return false;
    }else return false;
}

```

```
}
```

```
public void tick(){
```

```
}
```

```
public void render(Graphics g){
```

```
    Font fnt = new Font("arial",2,50);
```

```
    Font fnt2 = new Font("arial",2,30);
```

```
    Font fnt3 = new Font("arial",2,15);
```

```
    if (main.gameState==STATE.Menu){
```

```
        g.setFont(fnt);
```

```
        g.setColor(Color.white);
```

```
        g.drawString("Menú", 290, 70);
```

```
        g.setFont(fnt2);
```

```
        g.setColor(Color.white);//Easy Mode
```

```
        g.drawString("Modo Fácil",285,145);
```

```
        g.drawRect(260, 100, 200, 64);
```

```
        g.setFont(fnt2);
```

```
        g.setColor(Color.white);//Hard Mode
```

```
        g.drawString("Modo Díficil",285,245);
```

```
        g.drawRect(260, 200, 200, 64);
```

```
        g.setColor(Color.white);//Salir
```

```
        g.drawString("Salir",330,345);
```

```
        g.drawRect(260, 300, 200, 64);
```

```
    }
```

```
    else if(main.gameState==STATE.Easy){
```

```
        g.setFont(fnt);
```

```
        g.setColor(Color.white);
```

```
        g.drawString("Modo Fácil", 237, 70);
```

```
        g.setFont(fnt3);
```

```
        g.setColor(Color.white);//Ecuaciones De Segundo Grado
```

```
        g.drawString("Ecuaciones Cuadraticas",285,140);
```

```
        g.drawRect(260, 100, 200, 64);
```

```
        g.setColor(Color.white);
```

```
        g.drawString("Aritmetica", 330, 240); // Texto, Coordenada X, Coordenada Y
```

```
        g.drawRect(260, 200, 200, 64);
```

```
        g.setColor(Color.white);//Retroceder
```

```
        g.drawString("Atras",340,340);
```

```
        g.drawRect(260, 300, 200, 64);
```

```
    }
```

```
    else if(main.gameState==STATE.Hard){
```

```
        g.setFont(fnt);
```

```
        g.setColor(Color.white);
```

```
        g.drawString("Modo Díficil", 225, 70);
```

```
        g.setFont(fnt3);
```

```
        g.setColor(Color.white);
```

```

        g.drawString("Numeros Complejos", 300, 215);
        g.drawString("y Vectores", 330, 235);
        g.drawRect(260, 185, 200, 64);

        g.setColor(Color.white); //Integración
        g.drawString("Integrales Definidas", 295, 140);
        g.drawRect(260, 100, 200, 64);

        g.setColor(Color.white); //Retroceder
        g.drawString("Atras", 340, 310);
        g.drawRect(260, 270, 200, 64);
    }
}
}

```

Clase GaussWindow

```
package MainPackage;
```

```
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

```
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;
```

```
public class GaussWindow extends JFrame {
```

```
    public GaussWindow(String title) {
        JFrame frame = new JFrame(title);
        JButton b1 = new JButton("Responder");
        JButton e = new JButton("Exit");
        frame.getContentPane().setBackground(Color.DARK_GRAY);
        b1.setBounds(100, 210, 100, 30);
        e.setBounds(300, 210, 100, 30);

        JLabel label1 = new JLabel("b = ");
        label1.setBounds(10, 35, 90, 90);

        JLabel label2 = new JLabel("a = ");
        label2.setBounds(10, 115, 90, 90);

        JLabel label3 = new JLabel("dx = ");
        label3.setBounds(255, 75, 90, 90);

        double limites[] = new double[2];
        double coef[] = new double[4];
        double resultado;

        limites = GaussGen.limites();
        coef = GaussGen.coeficientes();
        resultado = GaussGen.Cuadratura(limites, coef);

        JTextField f1 = new JTextField(GaussGen.text(coef)); // Cuadros De Problemas
        f1.setBounds(60, 110, 190, 20);
        f1.setEditable(false);

        JTextField l1 = new JTextField("" + limites[1]); // Intervalos
    }
}

```

```

        l1.setBounds(40,70,45,20);
        l1.setEditable(false);

        JTextField l2= new JTextField(""+ limites[0]);
        l2.setBounds(40,150,45,20);
        l2.setEditable(false);

        JTextField Af1= new JTextField(); // Cuadros de respuestas
        Af1.setBounds(295,110,105,20);

        b1.setBackground(Color.white);
        e.setBackground(Color.white);
        label1.setForeground(Color.white);
        label2.setForeground(Color.white);
        label3.setForeground(Color.white);

        frame.add(b1);
        frame.add(l1);
        frame.add(l2);
        frame.add(e);
        frame.add(f1);
        frame.add(Af1);
        frame.add(label1);
        frame.add(label2);
        frame.add(label3);
        frame.setSize(450,300);
        frame.setResizable(false);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);

        ActionListener a = new ActionListener(){
            @SuppressWarnings("deprecation")
            @Override
            public void actionPerformed(ActionEvent at) {
                if(at.getSource()==e){
                    frame.hide();
                }
                if(at.getSource()==b1){
                    double answerP1[] = new double [1];

                    try{
                        answerP1[0]=Integer.parseInt(Af1.getText());
                        GaussGen.aciertos(answerP1, resultado);
                    }
                    catch (Exception ext){
                        ErrorInputWindow z= new ErrorInputWindow();
                    }
                }
            }
        };
        e.addActionListener(a);
        b1.addActionListener(a);
    }
}

```

Clase GaussGen

```
package MainPackage;
```

```
import java.util.Random;
```

```

public class GaussGen {
    public GaussGen(){

    }

    protected static double [] limites(){
        double limites[] = new double[2];
        Random random = new Random();
        double a=0, b=-1;
        while(b<a || b==a){
            a=(random.nextInt(1+7)+1)-(random.nextInt(1+7)+1);
            b=(random.nextInt(1+7)+1)-(random.nextInt(1+7)+1);
        }
        limites[0]=a;
        limites[1]=b;
        return limites;
    }

    protected static double [] coeficientes(){
        double coef[] = new double[4];
        Random random = new Random();

        for(int u=0;u<coef.length;u++){
            coef[u]=(random.nextInt(1+30)+1)-(random.nextInt(1+30)+1);
        }
        return coef;
    }

    protected static double Cuadratura(double[] inter, double[] coef){
        double result=0;
        double variable;
        double [] points = new double [] {-0.57735026, 0.57735026};
        double inter1=(inter[1]-inter[0])*0.5;
        double inter2=(inter[0]+inter[1])*0.5;
        for(int w=0;w<2;w++){
            variable = (inter1 * points[w]) + inter2;
            result+=funcion(coef,variable);
        }
        result*=(inter[1]-inter[0])*0.5;
        result=Math.round(result*100.0)/100.0;
        return result;
    }

    protected static double funcion(double[] coef, double point){
        return ((Math.pow(coef[0],3)*point) + (Math.pow(coef[1], 2)*point) + (Math.pow(coef[2], 1)*point) + coef[3]);
    }

    protected static String text(double Ecu[]){
        String Text = Ecu[0] + "x^3 " + Ecu[1] + "x^2 " + Ecu[2] + "x " +Ecu[3] ;
        if(Ecu[1]>=0){
            Text = Ecu[0] + "x^3 + " + Ecu[1] + "x^2 " + Ecu[2] + "x " +Ecu[3] ;
        }
        else if(Ecu[2]>=0){
            Text = Ecu[0] + "x^3 " + Ecu[1] + "x^2 + " + Ecu[2] + "x " +Ecu[3] ;
        }
        else if(Ecu[3]>=0){
            Text = Ecu[0] + "x^3 " + Ecu[1] + "x^2 " + Ecu[2] + " x + " +Ecu[3] ;
        }
        return Text;
    }

    protected static void aciertos (double answers[], double resultado){

        int points=0;
        if(answers[0]==resultado){
            points+=500;
        }
        new Score(points);
    }
}

```

```
}
```

Clase ExitWindow

```
package MainPackage;
```

```
import java.awt.Color;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
```

```
public class exitWindow extends JFrame implements ActionListener {
```

```
    private static final long serialVersionUID = 359460016246043594L;
```

```
    public exitWindow(String title){
```

```
        JFrame frame = new JFrame(title);
```

```
        JButton b=new JButton("Cancelar");
```

```
        JButton c=new JButton("Salir");
```

```
        b.setBounds(90,30,100,30);
```

```
        c.setBounds(90,80,100,30);
```

```
        b.setBackground(Color.white);
```

```
        c.setBackground(Color.white);
```

```
        frame.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
```

```
        frame.add(b);
```

```
        frame.add(c);
```

```
        frame.getContentPane().setBackground(Color.DARK_GRAY);
```

```
        frame.setSize(320,180);
```

```
        frame.setResizable(false);
```

```
        frame.setLayout(null);
```

```
        frame.setVisible(true);
```

```
        c.addActionListener(this);
```

```
        ActionListener a = new ActionListener(){
```

```
            @SuppressWarnings("deprecation")
```

```
            @Override
```

```
            public void actionPerformed(ActionEvent e) {
```

```
                if(e.getSource()==c){
```

```
                    System.exit(0);
```

```
                }
```

```
                if (e.getSource()==b){
```

```
                    frame.hide();
```

```
                }
```

```
            }
```

```
        };
```

```
        c.addActionListener(a);
```

```
        b.addActionListener(a);
```

```
    }
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        // TODO Auto-generated method stub
```

```
    }
```

```
}
```


Clase ErrorInputWindow

```
package MainPackage;

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class ErrorInputWindow extends JFrame implements ActionListener {

    private static final long serialVersionUID = -1249194514106409352L;

    public ErrorInputWindow(){
        JFrame frame = new JFrame("Error");
        JButton b=new JButton("Regresar");
        b.setBounds(90,90,100,30);
        b.setBackground(Color.white);
        frame.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        frame.add(b);
        frame.getContentPane().setBackground(Color.DARK_GRAY);
        frame.setSize(320,180);
        frame.setResizable(false);
        frame.setLayout(null);
        frame.setVisible(true);
        JLabel adver = new JLabel();
        adver.setBounds(30,15,250,50);
        adver.setText("Debe Llenar Los Recuadros, No sea Imbécil");
        //adver.setText("Debe Llenar Los Recuadros con números");
        adver.setForeground(Color.white);
        adver.setVisible(true);
        frame.add(adver);

        ActionListener a = new ActionListener(){
            @SuppressWarnings("deprecation")
            @Override
            public void actionPerformed(ActionEvent e) {
                if (e.getSource()==b){
                    frame.hide();
                }
            }
        };
        b.addActionListener(a);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
    }
}
```

Clase AritmeticWindow

```
package MainPackage;

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ImageIcon;
import javax.swing.JButton;
```

```

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class aritmeticWindow extends JFrame {
    public aritmeticWindow(String title){
        JFrame frame = new JFrame(title);
        JButton b1=new JButton("Responder");
        JButton e=new JButton("Exit");

        JLabel label1= new JLabel("Problema 1 : ");
        label1.setBounds(10, 10, 90, 90);
        JLabel label2= new JLabel("Problema 2 : ");
        label2.setBounds(10, 40, 90, 90);
        JLabel label3= new JLabel("Problema 3 : ");
        label3.setBounds(10, 70, 90, 90);
        JLabel label4= new JLabel("Problema 4 : ");
        label4.setBounds(10, 100, 90, 90);
        JLabel label5= new JLabel("Problema 5 : ");
        label5.setBounds(10, 130, 90, 90);

        JLabel labelR1= new JLabel(" = ");
        labelR1.setBounds(270, 10, 90, 90);
        JLabel labelR2= new JLabel(" = ");
        labelR2.setBounds(270, 40, 90, 90);
        JLabel labelR3= new JLabel(" = ");
        labelR3.setBounds(270, 70, 90, 90);
        JLabel labelR4= new JLabel(" = ");
        labelR4.setBounds(270, 100, 90, 90);
        JLabel labelR5= new JLabel(" = ");
        labelR5.setBounds(270, 130, 90, 90);

        int [] array1 = new int[2];
        int [] array2 = new int [2];
        int [] array3 = new int [2];
        int [] array4 = new int [2];
        int [] array5 = new int [2];

        int resultado1 [] = new int[1];
        int resultado2 [] = new int[1];
        int resultado3 [] = new int[1];
        int resultado4 [] = new int[1];
        int resultado5 [] = new int[1];

        int resultados[][]= new int [5][];

        array1=aritmeticGen.NumberGen();
        array2=aritmeticGen.NumberGen();
        array3=aritmeticGen.NumberGen();
        array4=aritmeticGen.NumberGen();
        array5=aritmeticGen.NumberGen();

        resultado1[0]=aritmeticGen.answers(array1,0);
        resultado2[0]=aritmeticGen.answers(array2,0);
        resultado3[0]=aritmeticGen.answers(array3,1);
        resultado4[0]=aritmeticGen.answers(array4,1);
        resultado5[0]=aritmeticGen.answers(array5,2);

        resultados[0]=resultado1;
        resultados[1]=resultado2;
        resultados[2]=resultado3;

```

```

resultados[3]=resultado4;
resultados[4]=resultado5;

JTextField f1= new JTextField(aritmeticGen.Text(array1,0)); // Cuadros De Problemas
f1.setBounds(90,50,180,20);
f1.setEditable(false);

JTextField f2= new JTextField(aritmeticGen.Text(array2,0));
f2.setBounds(90,80,180,20);
f2.setEditable(false);

JTextField f3= new JTextField(aritmeticGen.Text(array3,1));
f3.setBounds(90,110,180,20);
f3.setEditable(false);

JTextField f4= new JTextField(aritmeticGen.Text(array4,1));
f4.setBounds(90,140,180,20);
f4.setEditable(false);

JTextField f5= new JTextField(aritmeticGen.Text(array5,2));
f5.setBounds(90,170,180,20);
f5.setEditable(false);

JTextField Af1= new JTextField(); // Cuadros de respuestas
Af1.setBounds(285,50,105,20);
JTextField Af2= new JTextField();
Af2.setBounds(285,80,105,20);
JTextField Af3= new JTextField();
Af3.setBounds(285,110,105,20);
JTextField Af4= new JTextField();
Af4.setBounds(285,140,105,20);
JTextField Af5= new JTextField();
Af5.setBounds(285,170,105,20);

frame.getContentPane().setBackground(Color.DARK_GRAY);
b1.setBounds(100,210,100,30);
e.setBounds(285,210,100,30);

label1.setForeground(Color.white);
label2.setForeground(Color.white);
label3.setForeground(Color.white);
label4.setForeground(Color.white);
label5.setForeground(Color.white);
labelR1.setForeground(Color.white);
labelR2.setForeground(Color.white);
labelR3.setForeground(Color.white);
labelR4.setForeground(Color.white);
labelR5.setForeground(Color.white);

b1.setBackground(Color.white);
e.setBackground(Color.white);
Imagelcon ba = new Imagelcon("src/MainPackage/Sniki.png");

frame.add(f1);
frame.add(f2);
frame.add(f3);
frame.add(f4);
frame.add(f5);
frame.add(Af1);
frame.add(Af2);
frame.add(Af3);

```

```

frame.add(Af4);
frame.add(Af5);

frame.add(label1);
frame.add(label2);
frame.add(label3);
frame.add(label4);
frame.add(label5);
frame.add(labelR1);
frame.add(labelR2);
frame.add(labelR3);
frame.add(labelR4);
frame.add(labelR5);

frame.add(b1);
frame.add(e);

frame.setSize(450,300);
    frame.setResizable(false);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);

ActionListener a = new ActionListener(){
    @SuppressWarnings("deprecation")
    @Override
    public void actionPerformed(ActionEvent at) {
        if(at.getSource()==e){
            frame.hide();
        }
        if(at.getSource()==b1){
            int a,b,c,d,e;

            int answerP1[] = new int [1];
            int answerP2[] = new int [1];
            int answerP3[] = new int [1];
            int answerP4[] = new int [1];
            int answerP5[] = new int [1];

            int answers[][] = new int [5][];

            try{
                a= Integer.parseInt(Af1.getText());
                b= Integer.parseInt(Af2.getText());
                c= Integer.parseInt(Af3.getText());
                d= Integer.parseInt(Af4.getText());
                e= Integer.parseInt(Af5.getText());

                answerP1 [0]=a;
                answerP2 [0]=b;
                answerP3 [0]=c;
                answerP4 [0]=d;
                answerP5 [0]=e;

                answers[0]=answerP1;
                answers[1]=answerP2;
                answers[2]=answerP3;
                answers[3]=answerP4;
                answers[4]=answerP5;

                aritmeticGen.aciertos(answers,resultados);

```

```

    }
    catch (Exception ext){
        ErrorInputWindow z= new ErrorInputWindow();
    }
}

};
e.addActionListener(a);
b1.addActionListener(a);
}
}

```

ClaseAritmeticGen

package MainPackage;

import java.util.Random;

```

public class aritmeticGen {
    public aritmeticGen(String title){

    }

    protected static int[] NumberGen(){
        Random random = new Random();
        int a=1,b=0;
        int Numero [] = new int [3];

        for(int j=0;j<Numero.length;j++){
            Numero[j]=(random.nextInt(1+30)+1) - (random.nextInt(1+30)+1) ;
        }
        return Numero;
    }

    protected static String Text(int[] array, int id){
        String Text="";
        if(id==0){
            Text=array[0] + " + (" + " (" + array[0] + " x " + array[1]+ " ) " + " x " + array[0] + " )";
        }
        if (id==1){
            Text=array[0] + " + ( " + array[2] + " * " + array[1] + " ) x " + "(" + array[2] + " )";
        }
        else if(id==2){
            Text=array[2] + " + ( " + array[0] + " * " + array[1] + " ) x " + "(" + array[1] + " )";
        }
        return Text;
    }

    protected static int answers(int[] array, int id){
        int answer=0;
        if (id==0){
            answer=( (array[0]*array[1]) * array[0]) + array[0] ;
        }
        if (id==1){
            answer=array[0] + (array[2]*array[1])* array[2] ;
        }
        else if(id==2){
            answer=array[2] + ( array[0]* array[1])*array[1] ;
        }
        return answer;
    }

    protected static void aciertos (int answers[], int resultados[][]){

        int points=0;
        for(int z=0;z<answers.length;z++){
            for(int y=0;y<answers[z].length;y++){
                if(answers[z][y]== resultados[z][y]){

```

```

        }
        new Score(points);
    }
}

points+=100;
}

```

Pruebas De Ejecución

Ecuaciones Cuadraticas

Problema 1 :

$15x^2 + 1x - 20$

X1 :

2

X2 :

845

Problema 2 :

$4x^2 - 19x + 10$

X1 :

2

X2 :

1

Problema 3 :

$-24x^2 + 7x - 20$

X1 :

4

X2 :

23

Problema 4 :

$-4x^2 - 13x + 4$

X1 :

3

X2 :

421

Problema 5 :

$7x^2 + 15x - 19$

X1 :

2

X2 :

3

Responder

Exit

Puntaje

Puntaje : 0

Ecuaciones Cuadraticas

Problema 1 :

$-1x^2 - 4x + 4$

X1 :

-4.83

X2 :

0.83

Problema 2 :

$-1x^2 + 10x - 2$

X1 :

-0.2

X2 :

10.2

Problema 3 :

$-3x^2 + 4x - 18$

X1 :

-1.87

X2 :

3.21

Problema 4 :

$-19x^2 + 10x - 15$

X1 :

-0.66

X2 :

1.19

Problema 5 :

$-7x^2 - 18x + 2$

X1 :

-2.68

X2 :

0.11

Responder

Exit

Puntaje

Puntaje : 500

Vectores y Numeros Complejos

Problema 1 :

(1,3,23) x (24,-8,-15)

=

(

139

567

-80

)

Problema 2 :

(25,20,7) x (17,20,1)

=

(

-120

94

160

)

Problema 3 :

(16,-11,12)*(17,20,1)

=

64

Problema 4 :

(-3 -5i) * (-7 + 7i)

=

(

56

14

)

Problema 5 :

(3 + 0i) * (-2 -10i)

=

(

-6

-30

)

Responder

Exit

Puntaje

Puntaje :

1100

Aritmetica

Problema 1 :

5 + ((5 x 14) x 5)

=

123321

Problema 2 :

-1 + ((-1 x 6) x -1)

=

123

Problema 3 :

-3 + (17 * -2) x (17)

=

321

Problema 4 :

-13 + (2 * -17) x (2)

=

1

Problema 5 :

18 + (-6 * -21) x (-21)

=

-2628

Responder

Exit

Puntaje

Puntaje :

100

Integrales Definidas

b =

2.0

-18.0x^3 -2.0x^2 + 11.0x -17.0

dx

=

1

a =

0.0

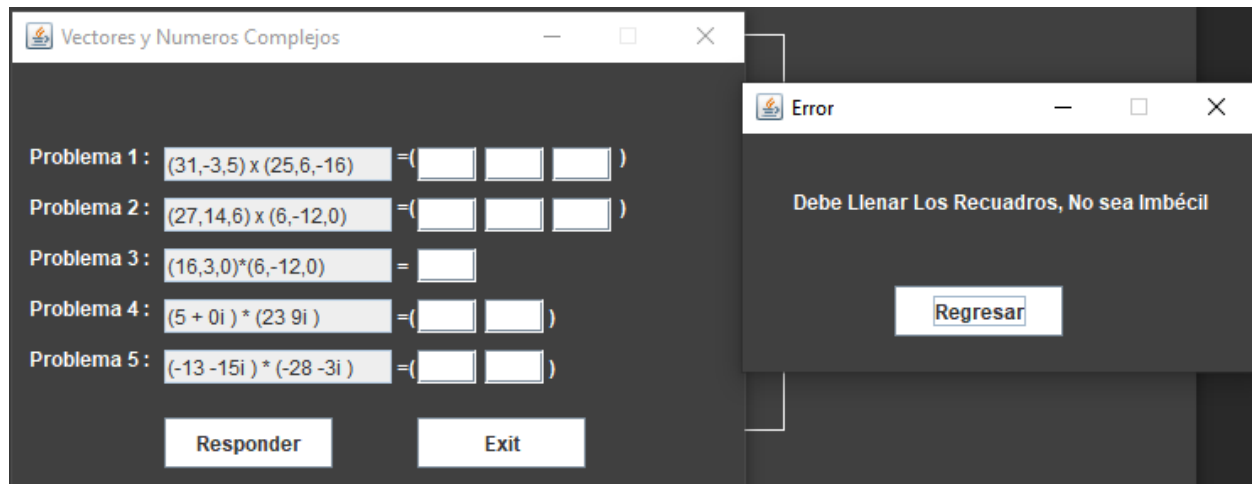
Responder

Exit

Puntaje

Puntaje :

0



Resumen De Conceptos Empleados De Programación Orientada A Objetos

- Clases Normales
- Interfaces
- Jerarquía
- Asociación
- Polimorfismo
- Manejo de eventos
- Concurrencia (un solo hilo)

Carpeta En Drive

<https://drive.google.com/drive/folders/14qSp9Dh9ZT6alh37KyefB0wrwwcGSven?usp=sharing>

Repositorio

https://github.com/BrayanLMercado/POO_Proyecto_BLM.git

Conclusiones

- Al realizar el proyecto experimente experiencias variadas, por ejemplo, desesperación por no hacer que los métodos funcionaran correctamente, que los resultados que implicaban divisiones resultaran en cero el 90% de las veces y tuviera que eliminarlas del programa, la parte positiva de esto fue que al final logre hacer que todo funcionara de manera medianamente correcta, todavía hay algunos fallos que no logré solucionar en su totalidad.
- En conceptos teóricos se podría decir que aprendí algunas cosas que no sabía antes, por ejemplo, hacer que un método regresara un array con valores (sí, no

sabía cómo se hacía, incluso de llevar 3 materias de programación), algunos conceptos los aprendí, pero no los llegue a implementar por falta de tiempo y para evitarme más problemas al momento de programar.

- En cuanto a las opiniones de los usuarios finales, es decir unos amigos míos en Discord, nos reímos bastante por la cantidad de cosas que dijimos durante las pruebas y los momentos de cómicos a causa del mensaje de error que agregue en la versión editada del programa.