

Manual de Programador

El presente documento fue creado con el propósito de poder apoyar al desarrollador respecto a cómo se programó el Software “Reloj análogo”.

1. Ventana principal, esta ventana nos ayuda a cargar el Reloj de una forma correcta y que no exista algún error al momento de cargar el sistema.



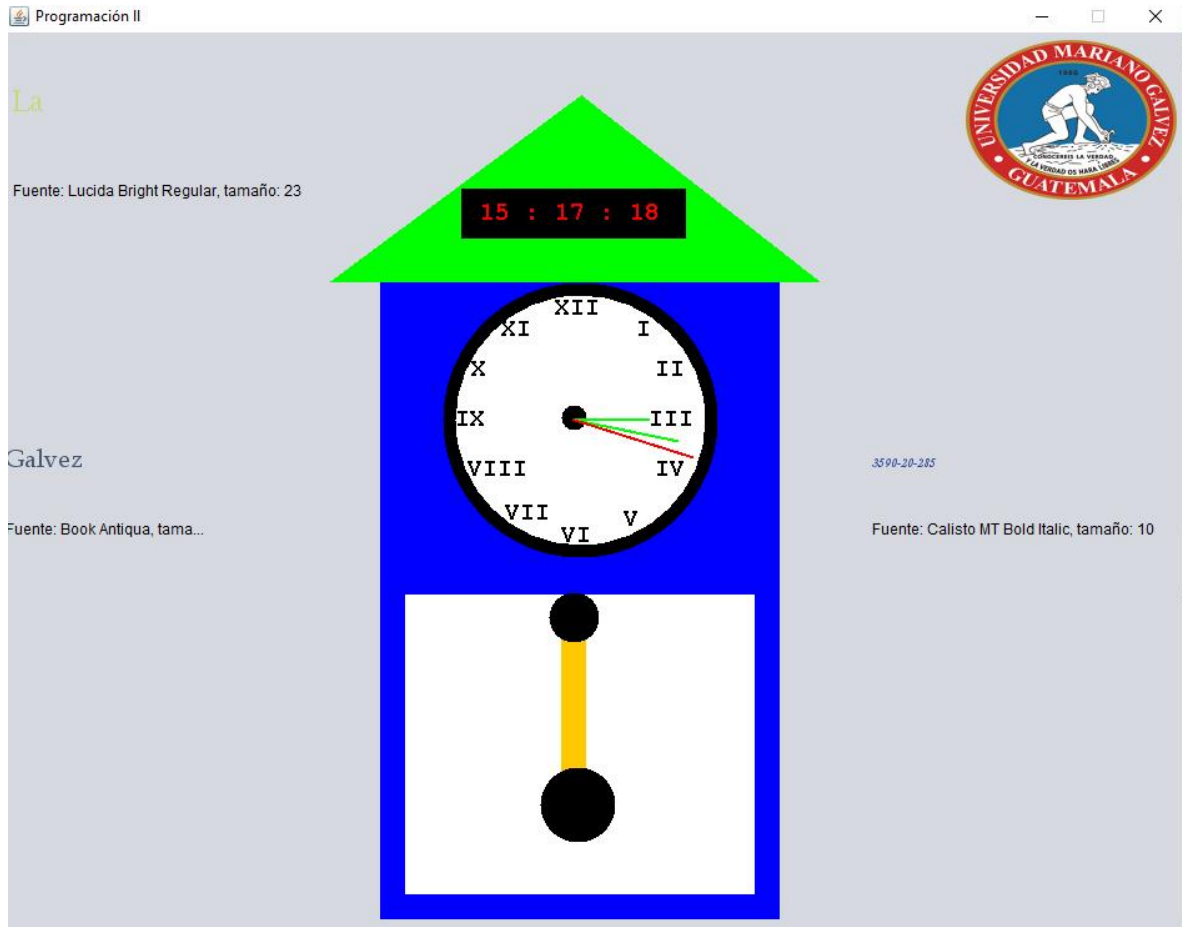
- 1.1 El código que ejecuta el botón Iniciar es el Siguiente, con esto nos aseguramos de que cuando la barra llega al 100% se cierre esta ventana principal y mandamos a llamar a la ventana del reloj.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    accion= new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent ae) {  
            progreso+=1;  
            carga.setValue(progreso);  
            verProgreso.setText(String.valueOf(progreso)+" %");  
            if(carga.getValue()==100){  
                tiempo.stop();  
                ver.parar();  
                avanzar();  
            }  
        }  
    };  
  
    tiempo = new Timer(100, accion);  
    tiempo.start();  
}
```

Una vez cargue a 100% detenemos el contador y llamamos a la función avanzar que es la que invoca la segunda venta y cierra la primera ventana con el objetivo que la primera ventana no se quede ahí sin hacer nada y ocupe mas espacio en la pantalla.

```
public void avanzar(){  
    Reloj ventana = new Reloj();  
    ventana.setVisible(true);  
    dispose();  
}
```

2. Una vez cargado nos mostrara el reloj donde podemos ver la hora en el reloj clásico con agujas y también podemos verlo de forma digital.



Para el proceso de mostrado de esta ventana se dibuja de arriba hacia abajo y se crea un objeto de la clase Hora que nos ayuda a obtener la hora actual del sistema y nos dibuja las líneas para que se pueda ver que las manecillas están caminando se usa el método repaint(); que refresca la ventana y da esa sensación de movimiento.

La forma de Dibujado es la siguiente:

```

public class Lienzo extends JPanel{

    @Override
    public void paint (Graphics g){
        super.paint(g);
        Graphics2D dibuja = (Graphics2D)g;

        dibuja.setColor(Color.GREEN);
        dibuja.fillPolygon(posX, posY, 3); //dibujamos el triangulo

        dibuja.setColor(Color.BLACK);
        dibuja.fillRect(365,125, 180, 40); // dibujamos el cuadrado

        dibuja.setColor(Color.BLUE);
        dibuja.fillRect(300,200, 320, 220); // dibujamos el cuadrado
        dibuja.setColor(Color.BLACK);
        dibuja.fillOval(350,200,220,220); // dibujamos el circulo GRANDE

        dibuja.setColor(Color.white);
        dibuja.fillOval(360,210,200,200); // dibujamos el circulo

        dibuja.setColor(Color.BLACK);
        dibuja.fillOval(445,298,20,20); // dibujamos el punto en el centro del circulo
        dibuja.setColor(Color.BLUE);
        dibuja.fillRect(300,420, 320, 290); // dibujamos el cuadrado
        dibuja.setColor(Color.white);
        dibuja.fillRect(320,450, 280, 240); // dibujamos el cuadrado
        dibuja.setColor(Color.orange);
        dibuja.fillRect(445,453 ,20, 140);
        dibuja.setColor(Color.BLACK);
        dibuja.fillOval(435,448,40,40); // dibujamos el punto en el centro del circulo
        dibuja.setColor(Color.BLACK);
        dibuja.fillOval(428,588,60,60); // dibujamos el punto en el centro del circulo

        Font myFont = new Font ("Courier New", 1, 20);
        dibuja.setFont(myFont);
        dibuja.setColor(Color.BLACK);
        dibuja.drawString("XII",440,226); //numero 12
        dibuja.drawString("I",505,243); // numero 1
        dibuja.drawString("II",520,275); // numero 2
        dibuja.drawString("III",515,315); // numero 3
        dibuja.drawString("IV",520,355); // numero 4
        dibuja.drawString("V",495,395); // numero 5
        dibuja.drawString("VI",445,408); // numero 6
        dibuja.drawString("VII",400,390); // numero 7
        dibuja.drawString("VIII",370,355); // numero 8
        dibuja.drawString("IX",360,315); // numero 9
        dibuja.drawString("X",373,275); // numero 10
        dibuja.drawString("XI",397,243); // numero 11

        Hora hilo = new Hora(dibuja);
        hilo.start();
        Pendulo hilo2 = new Pendulo(dibuja);
        hilo2.start();
        repaint();
    }
}

```

La Clase Hora funciona de la siguiente manera en la cual recibe por parámetros el objeto de tipo Graphics y nos ayuda a pintar las líneas en el círculo del centro.

```
public class Hora extends Thread {

    Calendar calendario = Calendar.getInstance();
    Calendar fecha = new GregorianCalendar();
    boolean seguir=false;
    String Seg="",Min="",Hor="";

    int hora = fecha.get(Calendar.HOUR_OF_DAY);
    int minuto = fecha.get(Calendar.MINUTE);
    int segundo = fecha.get(Calendar.SECOND);

    double grados = 0, radianes = 0, dato1 = 0, dato2 = 0, dato3 = 0;
    Graphics2D dibuja;
    AudioClip tic = java.applet.Applet.newAudioClip(getClass().getResource("/Ventana/campana.wav"));

    public Hora(Graphics2D dibuja) {
        this.dibuja = dibuja;
    }

    @Override
    public void run() {
        try {

            dibuja.setStroke(new BasicStroke(2));
            grados = hora * 30;
            radianes = Math.toRadians(grados);
            dato1 = Math.sin(radianes) * 60;
            dato2 = Math.cos(radianes) * 60;
            dibuja.setColor(Color.GREEN);
            dibuja.drawLine(455, 310, 455 + (int) dato1, 310 - (int) dato2);

            grados = minuto * 6;
            radianes = Math.toRadians(grados);
            dato1 = Math.sin(radianes) * 85;
            dato2 = Math.cos(radianes) * 85;
            dibuja.setColor(Color.GREEN);
            dibuja.drawLine(455, 310, 455 + (int) dato1, 310 - (int) dato2);

            grados = segundo * 6;
            radianes = Math.toRadians(grados);
            dato1 = Math.sin(radianes) * 100;
            dato2 = Math.cos(radianes) * 100;
            dibuja.setColor(Color.red);
            dibuja.drawLine(455, 310, 455 + (int) dato1, 310 - (int) dato2);
            if(minuto<10){
                Min = "0"+String.valueOf(minuto);
            }
            else{
                Min = String.valueOf(minuto);
            }
            if(segundo<10){
                Seg = "0"+String.valueOf(segundo);
            }
            else{
                Seg = String.valueOf(segundo);
            }
            String dato = (String.valueOf(hora)+" : "+Min+" : "+Seg);
            dibuja.drawString(dato,380,150); //mostramos hora
            Thread.sleep(1000);
        }
    }
}
```

```
        if(hora==12 && minuto==0 && segundo==0){
            tic.play();
            Thread.sleep(1000);
            tic.play();
            Thread.sleep(1000);
            tic.play();
        }

        if(minuto==0 && segundo==0){
            tic.play();
        }

    } catch (Exception e) {
        System.out.println("." + e);
    }
}
```