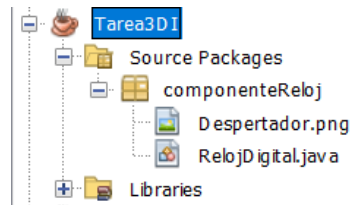


En la tarea se nos pide crear un componente gráfico o JavaBean de un reloj digital con función de alarma y que muestre un mensaje cuando se active la alarma.

Comenzamos creando un nuevo proyecto en NetBeans, un nuevo paquete llamado “componenteReloj” que contendrá una clase del tipo JavaBean llamada “RelojDigital” esto hará que la clase ya implemente la interfaz Serializable:



Hemos añadido dentro del proyecto una imagen (Despertador.png) porque luego la vamos a usar para mostrarla como icono dentro de la pantalla del mensaje de alarma.

Ya dentro de la nueva clase lo primero que vamos a hacer será crear los atributos del formato de hora, alarma, hora de la alarma, minutos de la alarma y mensaje de la alarma, tal y como nos indica la tarea cada uno con el formato correspondiente:

```
public class RelojDigital extends JLabel implements Serializable, ActionListener {

    //Declaración de atributos de clase
    private boolean formatoHora;
    private boolean alarma;
    private int hora_Alarma;
    private int minuto_Alarma;
    private String mensaje_Alarma;
```

Insertamos los métodos getter & setter de cada uno de ellos y ya tendremos lo que más adelante serán las propiedades de nuestro componente.

Una vez hecho esto ya estamos en disposición de empezar a programar lo que será el comportamiento de nuestro componente, en este caso el reloj. Empezamos insertando el método constructor donde le vamos a decir que se visualicen los dígitos del reloj y la alarma, en caso de que esta esté activada dentro del JLabel del que hereda nuestra clase principal:

```
public RelojDigital() {

    date = new Date();
    //Inicializamos las variables con valores por defecto
    setFormatoHora(true);
    setHora_Alarma(00);
    setMinuto_Alarma(00);
    setMensaje_Alarma("LEVANTATEEEEEEE !!!!!");

    if (isFormatoHora()) {
        formatoDeHora = new SimpleDateFormat("HH:mm:ss");//Formato de 24 horas
    } else {
        formatoDeHora = new SimpleDateFormat("hh:mm:ss");//Formato de 12 horas
    }

    if (isAlarma()) {
        visualizaAlarma(formatoDeHora);
    } else {
        visualizaDigitosReloj(formatoDeHora);
    }

    segundero = new Timer(1000, this);//Cada segundo lanza un evento a esta misma clase
    segundero.start();//Actúa de segundero
}
```

Desde el constructor vamos a lanzar una instancia de la clase Timer que nos va a lanzar un evento de esta clase cada segundo, haciendo la simulación de segundero porque en cada llamada al evento se nos va a crear una instancia de la clase Date. Esta clase lo que hace es recoger la hora del sistema:

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

    date = new Date();

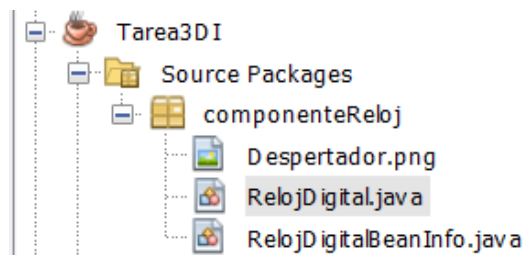
    if (isFormatoHora()) {
        formatoDeHora = new SimpleDateFormat("HH:mm:ss");//Formato de 24 horas
    } else {
        formatoDeHora = new SimpleDateFormat("hh:mm:ss");//Formato de 12 horas
    }
    if (isAlarma()) {
        if (date.getHours() == getHora_Alarma() && date.getMinutes() == getMinuto_Alarma() && date.getSeconds() == 0) {
            visualizaMensajeAlarma();
        }
        visualizaAlarma(formatoDeHora);
    } else {
        visualizaDigitosReloj(formatoDeHora);
    }
}
```

Para hacer el código más claro, legible, modular y eficiente he creado 3 métodos a mayores al final de la clase para las distintas visualizaciones de la etiqueta y el panel.

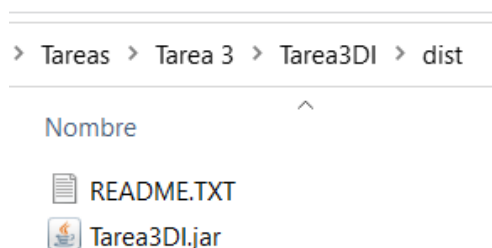
CREACIÓN DEL COMPONENTE

Ahora que ya tenemos creada la clase con sus atributos, funcionamiento y libre de errores, nos disponemos a crear el componente para poder utilizarlo posteriormente en cualquier aplicación gráfica.

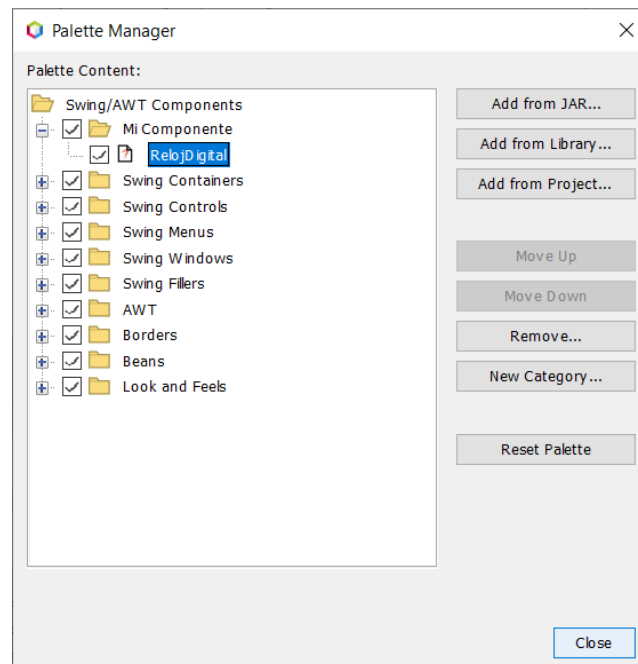
Lo primero que hacemos es pinchar con el botón derecho encima de la clase, seleccionamos “JavaBean info” esto hará que se nos cree otra clase que nos servirá como editor de propiedades de nuestro componente:



A continuación pinchamos encima del nombre de nuestro proyecto, con el botón derecho también, y seleccionamos la opción “clean and build” lo que nos va a generar un archivo .jar dentro de la carpeta “dist” del proyecto:

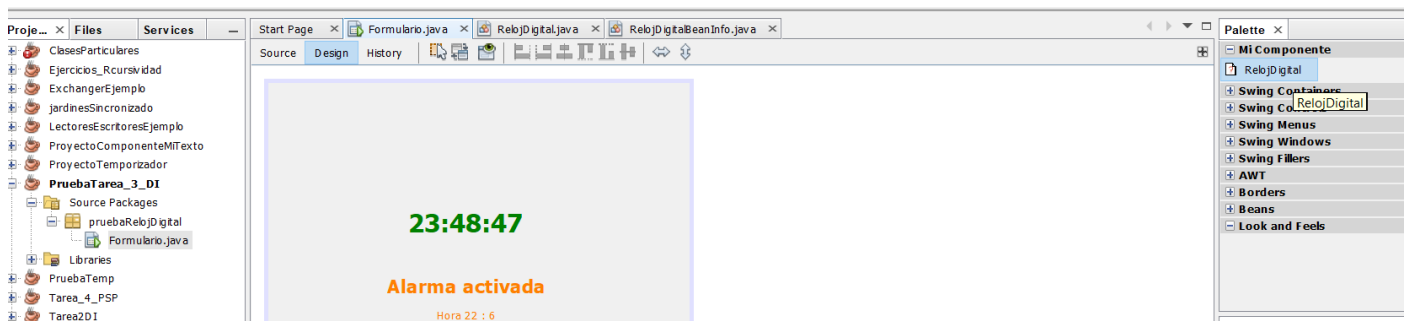


Ya solo nos falta agregar nuestro componente a la tableta de componentes de swing, lo haremos añadiendo a la misma el componente .jar que acabamos de crear:

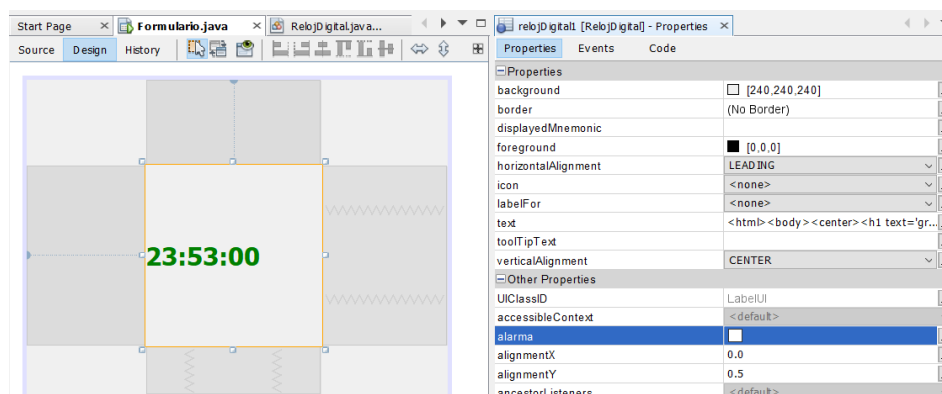


PRUEBAS DEL PROYECTO

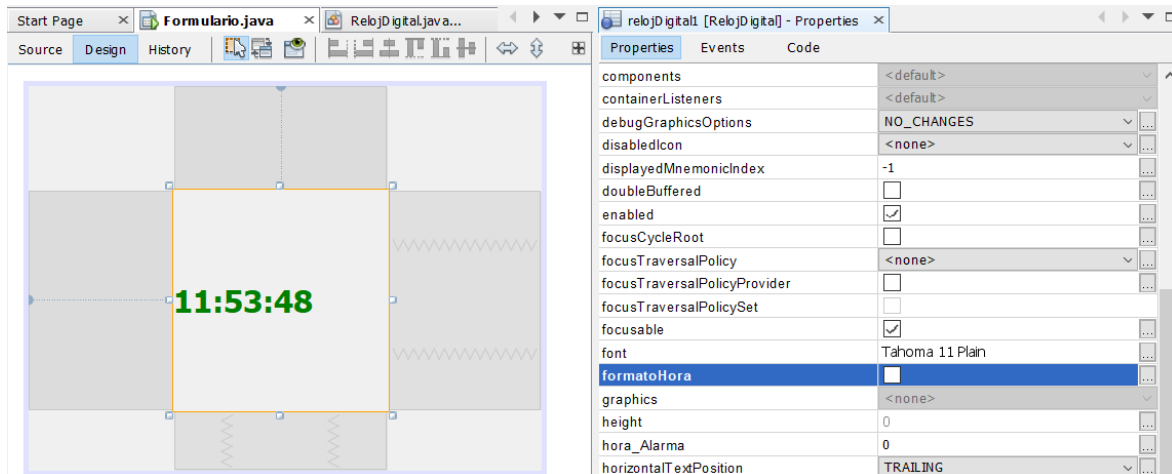
Para poder probar el nuevo componente tendremos que crear un nuevo proyecto con una clase que herede de la clase JFrame y así nos permita añadir desde la paleta nuestro componente de reloj:



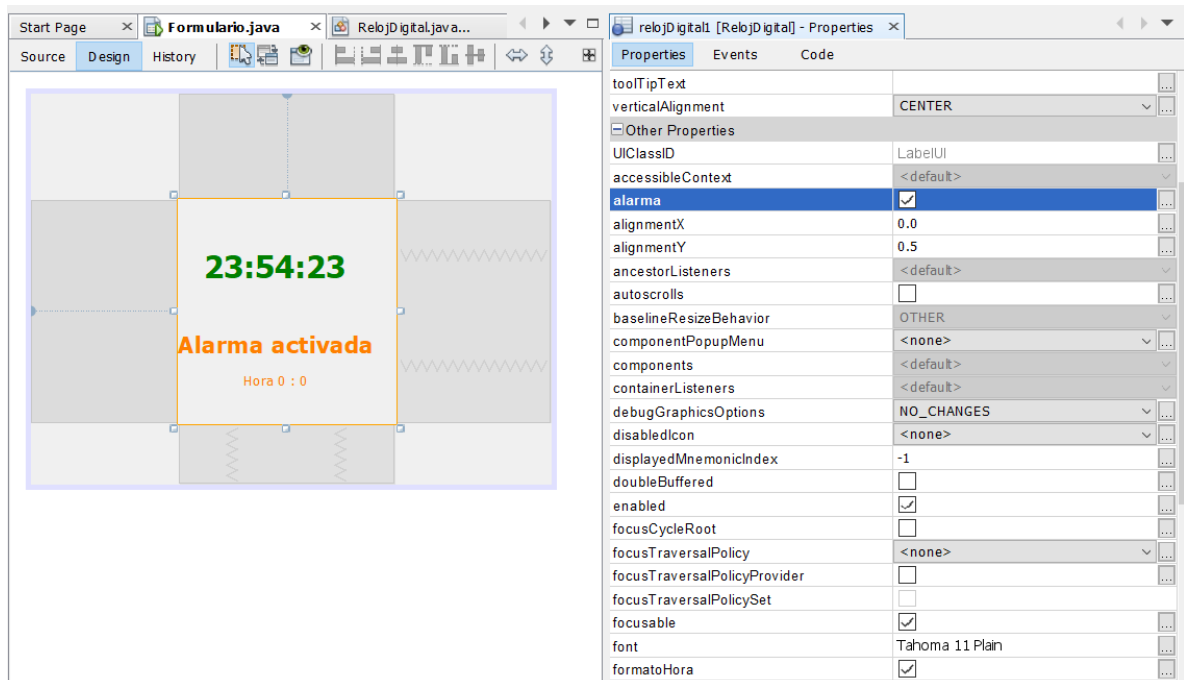
Podemos modificar sus propiedades a través de la paleta y vemos por ejemplo como se activa y desactiva la alarma dejando de visualizarse o como cambia el formato de hora a través de la propiedad correspondiente:



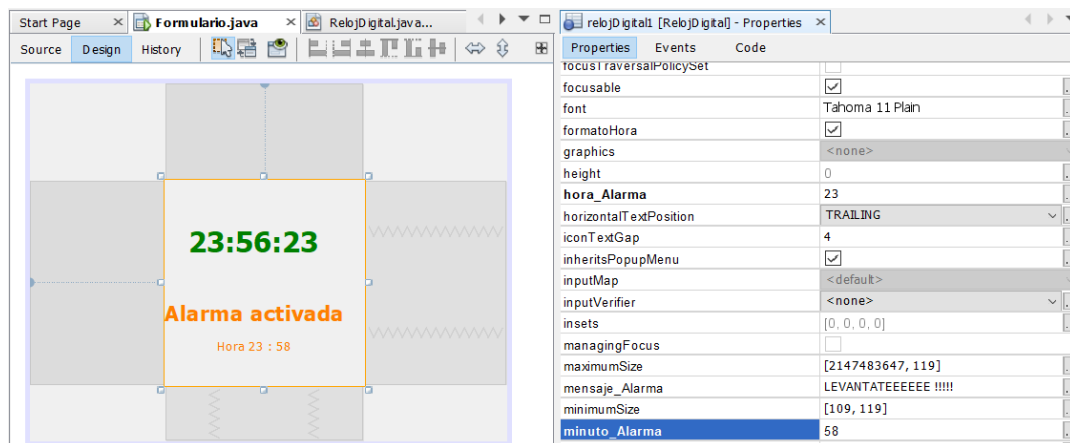
Alarma desactivada



Formato de 12 horas

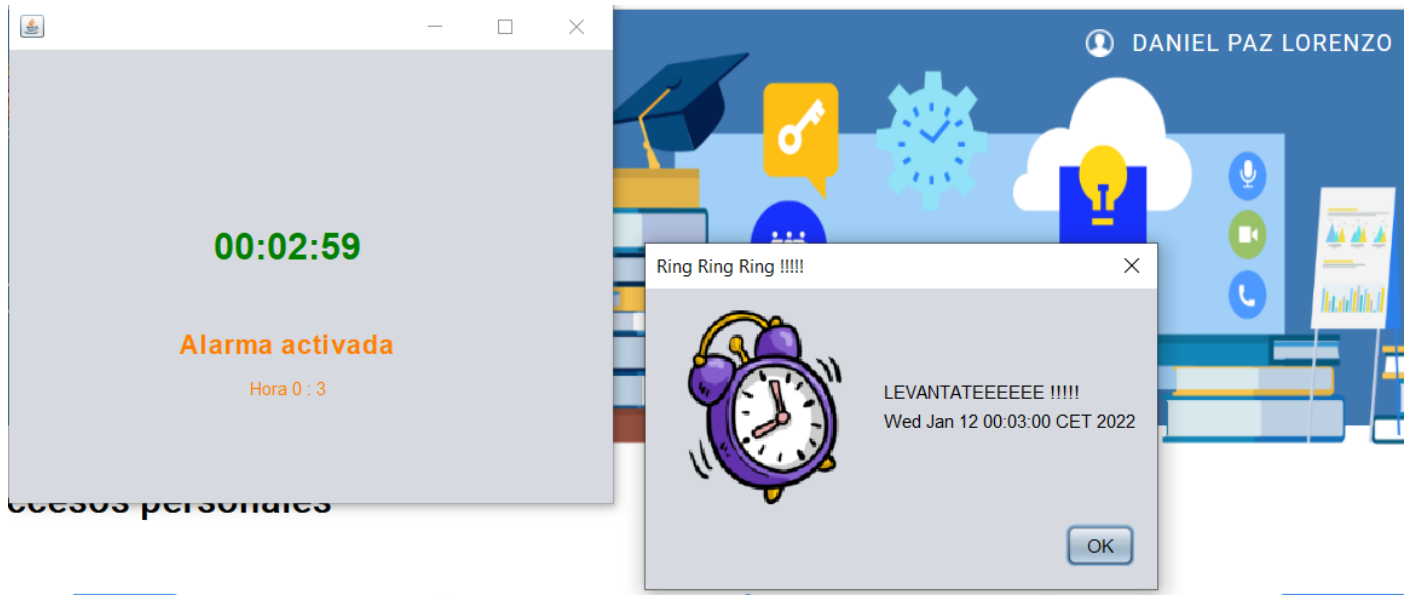


Alarma activada y formato de 24 horas



Modificamos la propiedad de hora y minutos de alarma

Ya vemos que las propiedades de nuestro componente funcionan, ahora nos queda probar si funciona el evento, es decir si cuando llega a la hora que hemos elegido como alarma se dispara el evento y muestra un panel con el mensaje que hemos puesto en la propiedad de mensaje alarma, en este caso “LEVANTATEEEEEEE !!!!!!!”. Nos disponemos entonces a ejecutar la aplicación de pruebas que hemos creado:



Vemos que se nos muestra el panel con el mensaje dentro de un panel con título, un icono de despertador y un texto a mayores que indica la fecha y hora a la que se activa la alarma

Como se puede ver en el resultado final a mayores de lo que nos piden y para hacer más visual la tarea he ido modificando el color de las letras, poniendo título al panel de alarma, añadiendo algunos componentes como el icono del despertador, añadiendo la hora de selección de la alarma, etc...