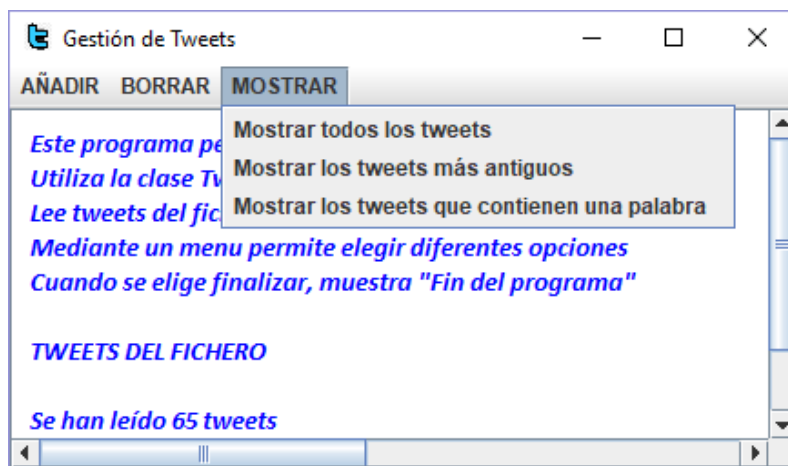


## Funcionalidad

- La ventana principal muestra los diferentes menús, el área de texto con scroll y el mensaje de presentación.  
El scroll sale cuando se ejecuta alguna opción.



- Las opciones "Cargar fichero", "Mostrar todos los tweets", "Borrar tweets de un usuario", "Mostrar los tweets que contienen una palabra" y "Mostrar los tweets más antiguos" funcionan correctamente.
- La opción "Añadir un tweet" funciona y se valida que la fecha introducida sea correcta.

```
// Creamos una subclase de InputVerifier
private class Verificador extends InputVerifier {

// Sobreescribimos el método verify
// Comprobamos que el componente es una caja de texto y leemos el texto
// Si la fecha tiene el formato requerido devuelve true
// Si se ha producido una excepción abre una ventana de error y devuelve
false

@Override
public boolean verify(JComponent componente) {
    if (componente instanceof JTextField) {
        JTextField cajaTextoFecha = (JTextField) componente;
        String texto = cajaTextoFecha.getText();
        // 2019-12-12T22:15:12
// Comprobamos la fecha que sea del formato correcto mediante una
expresión regular.
```

```

Pattern p = Pattern.compile("^\\d{4}-\\d{2}-\\d{2}T\\d{2}:\\d{2}:\\d{2}$");
Matcher m = p.matcher(texto);
if (m.matches()) {
    return true;
}

// Si el patrón no coincide devolvemos una caja de texto con
el patron que ha de reibir la caja
JOptionPane.showMessageDialog(cajaTextoFecha, "Introduce una fecha
valida (aaaa-MM-ddThh:mm:ss)");
}
return false;
}
}

```

Además comprobamos que no tengan campos vacíos.

```

// Si culaquiera de las cajas de texto está vacía dará un mensaje y
obligará a introducir el campo.
if ((usuarioCaja != null) && (usuarioCaja.equals("")) ||
(fecharCaja != null) && (fecharCaja.equals("")) || (mensajeCaja != null)
&& (mensajeCaja.equals("")))) {
    // cadena no está vacía
    JOptionPane.showMessageDialog(null, "Rellena todos los
campos", "Error!", JOptionPane.ERROR_MESSAGE);
    return;
}

```

- El programa realiza al menos 2 de las mejoras propuestas.
2. que al finalizar la opción se cierre automáticamente el JDialog

```

// Cuando se finaliza se cierra la ventana
dialogo.setVisible(false);

```

4. JOptionPane de confirmación solo muestre las opciones "Si" y "No", sin la opción "Cancelar"



5. modificar la configuración de algún componente: color, tipo de texto, se le ha cambiado el logo a la ventana principal y se le ha añadido padding al area de texto.

```
// Creamos una fuente nueva para añadirla al area de texto
Font fuente = new Font("Calibri", 3, 15);
areaTexto.setFont(fuente);
areaTexto.setForeground(Color.BLUE);

// colocamos el logo de twiter en la ventana principal
ventana.setIconImage (new ImageIcon("icono.jpg").getImage());

// Definimos un padding al area de texto
UIManager.put("TextArea.margin", new Insets(10,10,10,10));
```

- Se ha generado el fichero .jar.

Nota: 6,5

## Comentarios

Están bien escritos, en los lugares apropiados y ayudan a entender el funcionamiento del programa. Hay comentarios que no se han puesto por obvios.

Nota: 1,25

## Legibilidad

- El código está bien organizado, sigue la guía de estilo y es fácil de leer.

Nota: 1.5

Nota de la autoevaluación 9,25