

# Uso de una librería de chat en Swing

Con el fin de crear un chat en swing se ha diseñado una librería `chatWindowList`, Para poder utilizarla en Eclipse se debe añadir el archivo ***chatWindowLib.jar*** a las librerías del proyecto (build path).

**Importante:** debido a las librerías internas que utiliza el componente, sólo funcionará con una **versión Java 1.8.0\_181 o superior**.

## Uso de la librería

La librería `chatWindowList` proporciona la clase `BubbleText`, que se puede utilizar como cualquier otro componente de Swing. Dicha clase nos permite crear una burbuja de texto con un mensaje que se puede añadir a un contenedor `JPanel` que representará el panel del chat.

Para poder utilizar la clase `BubbleText` es necesario añadir la siguiente sentencia `import` a nuestra clase:

```
import tds.BubbleText;
```

## Creando una burbuja de texto

La burbuja de texto necesita conocer el panel del chat para ajustarse a su tamaño de forma automática, por lo que se le debe pasar una referencia a dicho `JPanel` como parámetro en el constructor.

Para crear una burbuja se puede utilizar el siguiente constructor:

```
new BubbleText(chat, texto, color, nombre, tipo);
```

El significado de los parámetros es el siguiente:

- **chat.** Es el `JPanel` que representa el chat donde tiene lugar la conversación. Se recomienda que este `JPanel` tenga un layout de distribución vertical para los componentes. Al añadir una burbuja de texto al chat, la burbuja ocupará una fila completa del chat, aunque aparecerá alineada junto a uno de los bordes, dejando el resto del espacio vacío. Es recomendable que el tamaño del `JPanel` del chat sea un tamaño fijo que no se pueda variar.
- **texto.** Es un `String` conteniendo el texto que aparecerá escrito en la burbuja como parte de una conversación. No se deben incluir caracteres de salto de línea (`'\n'`) o el texto no se podrá visualizar bien. La burbuja ajusta el texto para que no se corten las palabras y lo distribuye en varias líneas si es necesario.
- **color.** Especifica el color de fondo de la burbuja. Se trata de un objeto de la clase `Color` de `awt` (`java.awt.Color`). Se puede utilizar alguna de las constantes definidas en dicha clase o crear un color propio.

- **nombre.** Se trata de un String que especifica el nombre del contacto, es decir, el nombre de la persona que escribe el mensaje. Este nombre aparecerá en la parte superior de la burbuja, con el mismo tamaño de fuente.
- **tipo.** Es un valor entero que especifica si se trata de un mensaje enviado o recibido. La clase `BubbleText` tiene definidas unas constantes para dicho tipo (`BubbleText.SENT` para un mensaje enviado y `BubbleText.RECEIVED` para un mensaje recibido). El tipo de mensaje determina la forma de dibujarse la burbuja y su posición.

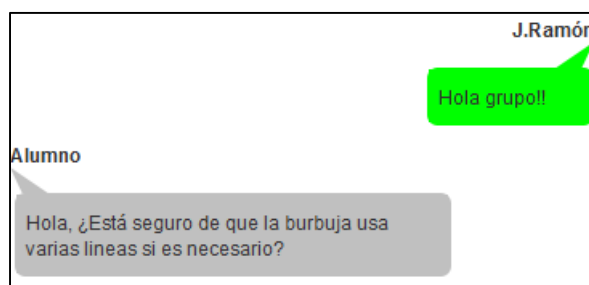
Ejemplo:

```
JPanel chat=new JPanel();
chat.setLayout(new BorderLayout(chat,BoxLayout.Y_AXIS));
chat.setSize(400,700);
chat.setMinimumSize(new Dimension(400,700));
chat.setMaximumSize(new Dimension(400,700));
chat.setPreferredSize(new Dimension(400,700));
```

```
BubbleText burbuja;
burbuja=new BubbleText(chat,"Hola grupo!!", Color.GREEN, "J.Ramón", BubbleText.SENT);
chat.add(burbuja);
```



```
BubbleText burbuja2;
burbuja2=new BubbleText(chat,
    "Hola, ¿Está seguro de que la burbuja usa varias lineas si es necesario?",
    Color.LIGHT_GRAY, "Alumno", BubbleText.RECEIVED);
chat.add(burbuja2);
```

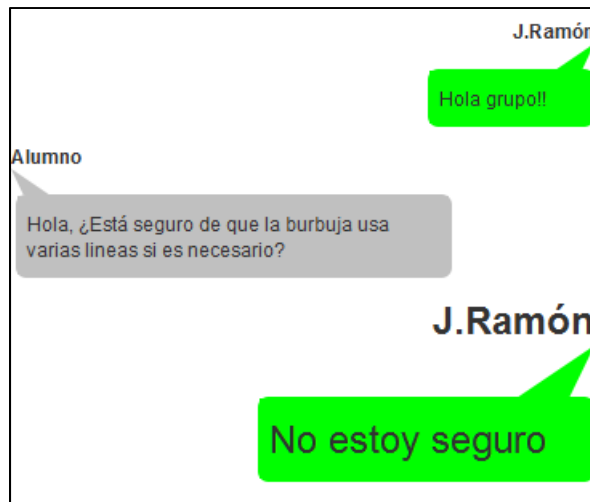


## Cambiando el tamaño de la fuente

El tamaño de la fuente por defecto para el texto de la burbuja es de 12 puntos, aunque se puede personalizar al tamaño que queramos si lo especificamos como un parámetro más al final del constructor. El cambio en el tamaño de la fuente afecta no solo al texto de la burbuja sino también al tamaño de la misma y a la fuente del nombre.

Ejemplo:

```
BubbleText burbuja3;  
burbuja3=new BubbleText(chat,"No estoy seguro",  
    Color.GREEN, "J.Ramón", BubbleText.SENT, 24);  
chat.add(burbuja3);
```



## Creando emoticonos

La librería permite el uso de emoticonos (emojis) en las burbujas del chat, con la limitación de que una burbuja puede contener un único emoticono y siempre que no incluya texto.

Para crear una burbuja con un emoticono se utiliza la siguiente variante del constructor:

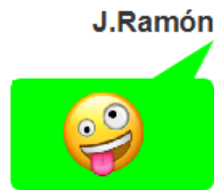
```
new BubbleText(chat, emoticono, color, nombre, tipo, tamaño);
```

Este constructor cambia el parámetro "texto" de la burbuja por el parámetro "emoticono" cuyo significado es el siguiente:

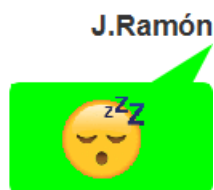
- **emoticono.** Se trata de un valor de tipo **int**, que indica el identificador del emoticono que se quiere representar en el mensaje. Dicho valor puede variar entre 0 y BubbleText.**MAXICONO**. Aunque este tipo de burbuja no tiene texto, el parámetro "tamaño" de la fuente sigue influyendo en el tamaño de la burbuja y del emoticono que contiene. Algunos emoticonos contienen animaciones.

Ejemplos:

```
BubbleText burbuja=new BubbleText(chat, 0, Color.GREEN, "J.Ramón", BubbleText.SENT,18);  
chat.add(burbuja);
```



```
BubbleText burbuja=new BubbleText(chat, 1, Color.GREEN, "J.Ramón", BubbleText.SENT,18);  
chat.add(burbuja);
```



## Consultando la librería de emoticonos

La clase BubbleText tiene un método estático que nos permite obtener una miniatura del emoticono que le pidamos. La imagen es una instancia de la clase javax.swing.ImageIcon., que se puede asignar fácilmente a una etiqueta JLabel. El siguiente método devuelve una imagen en miniatura del emoticono con identificador número 3.

```
ImageIcon miniatura=BubbleText.getEmoji(3);
```

Ejemplo:

```
JLabel x=new JLabel();  
x.setIcon(BubbleText.getEmoji(3));
```



A continuación se muestran las miniaturas de los primeros 8 emoticonos obtenidos por este método:



## Consideraciones para monitores 4K o de resoluciones mayores

Algunas versiones modernas de Java, al ejecutarse en ordenadores con pantallas 4K o de High DPI, escalan automáticamente el interfaz gráfico de las aplicaciones por un factor de 150% o 200% dependiendo del tipo de pantalla (lo que se conoce como soporte HiDPI). En el caso de esta librería el escalado automático "pixela" y degrada las imágenes del chat.

Es posible notificar a la aplicación Java que estamos desarrollando que no deseamos que se realice ese ajuste automático en monitores High DPI, (por ejemplo para gestionarlo nosotros dando un tamaño mayor al chat y a la fuente de las burbujas de texto).

Para ello basta con ejecutar el siguiente método al principio de nuestro programa, antes de empezar a crear las ventanas:

```
BubbleText.noZoom();
```

Ejemplo:

```
BubbleText.noZoom();  
ventana=new JFrame("CHAT Prototipe");
```

## Obtener la versión y el copyright del componente.

Se puede consultar mediante el método getVersion() que devuelve un String. Ejemplo:

```
System.out.println(BubbleText.getVersion());
```

## Licencia de uso

Esta librería es de libre uso siempre que se mencione a la Universidad de Murcia y al autor de la misma en la distribución o en el programa en el que se incluya. Dicha información se puede consultar invocando al método getVersion() como se explicó con anterioridad.

```
chatWindowLib version 1.1 - Jose R. Hoyos - (c) 2019 Universidad Murcia
```

## Ejemplo de interfaz y recomendaciones

Se recomienda que el JPanel del chat esté incluido dentro de un JScrollPane, para poder realizar un desplazamiento (scroll) de las conversaciones que excedan del tamaño del panel.

