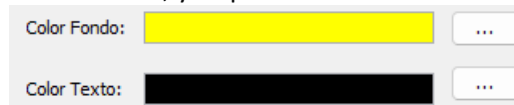


## **Ejercicio 2: Campo Texto que cambia según el número de caracteres introducido**

Crea un componente en NetBeans, que cuando se inserte, éste cambie su color de letra y el color de fondo a los que el usuario establezca, y ocurrirá cuando la longitud del texto exceda de un número de caracteres. El límite de caracteres también se establece como propiedad.

Se debe tener una **única propiedad para establecer el color de la letra y el color del fondo**, es decir deberá establecerse mediante un editor personalizado (panel personalizado). Los colores se podrán escoger mediante un selector de colores a partir del botón "...", y se podrá ver el color seleccionado.



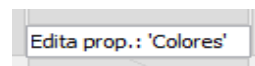
Recuerda que debes gestionar los posibles errores, ya que el usuario del componente puede pulsar "OK" y "Cancelar".

El **número límite de caracteres** que **dispara el cambio en los colores** del componente deberá ser una propiedad simple, y **debe tener un valor inicial por defecto inferior o igual a 25**, aunque luego el usuario pueda modificar a su antojo el límite.

Cuando el número de caracteres vuelva a ser inferior, se establecerán los colores iniciales del componente de forma automática, es decir los colores por defecto.

### **A tener en cuenta:**

- El nombre del componente será **"LimitedJTextField"**
- Cuando el componente sea arrastrado para ser colocado, éste deberá lanzar un mensaje de aviso indicando que se debe rellenar la propiedad correspondiente y que es recomendable establecer un número de caracteres personalizado (que por defecto es de 25).
- El componente al ser colocado tendrá un texto inicial corto directamente indicando la propiedad que se debe editar. Por ejemplo:



Por último, crea un ejemplo usando el componente para alguna utilidad que se te ocurra, como por ejemplo que se avise de que el campo de texto tiene un límite de 5 caracteres y que cuando se exceda el fondo cambie a rojo y las letras a blanco (por ejemplo).



### **AYUDA:**

Para controlar los eventos del documento dentro del componente, se puede acceder y controlar el texto del componente `JTextField` mediante el método `getDocument()` que devuelve un objeto de la clase ***Document***.

Añade un "oyente" (*Listener*) a esta clase (objeto *Document*) y sobrescribe aquellos métodos que controlen la actualización de la inserción, eliminación y cambios haciendo uso del tipo de evento de la clase ***DocumentEvent***.