4 DE DICIEMBRE DE 2025

PyQt

vs

PySide

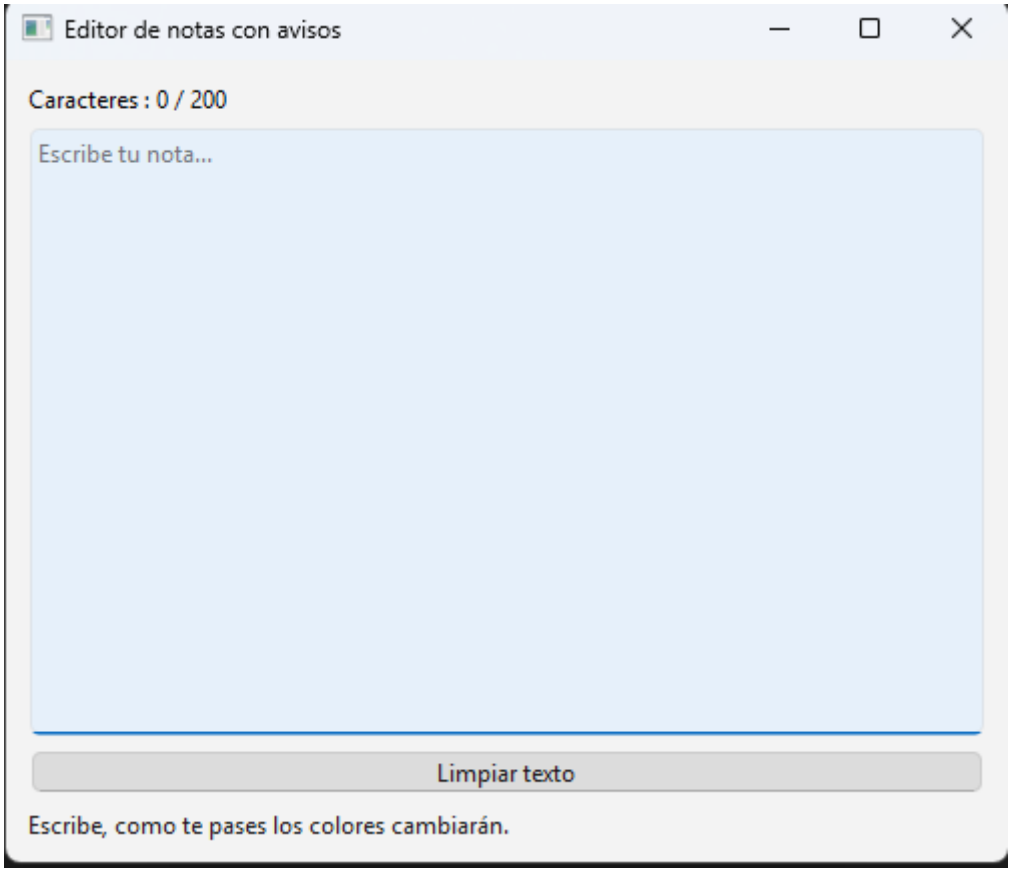
Qt for Python

TAREA 3.2 – EDITOR DE NOTAS CON AVISOS

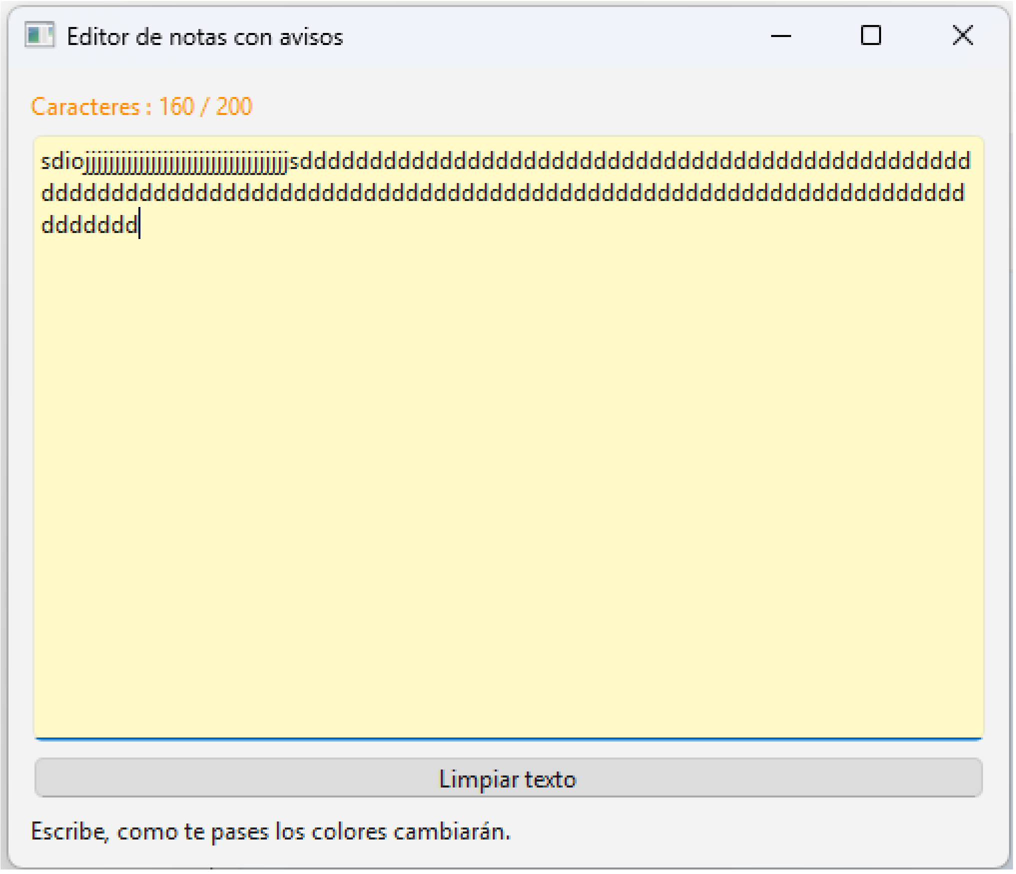
Contenido

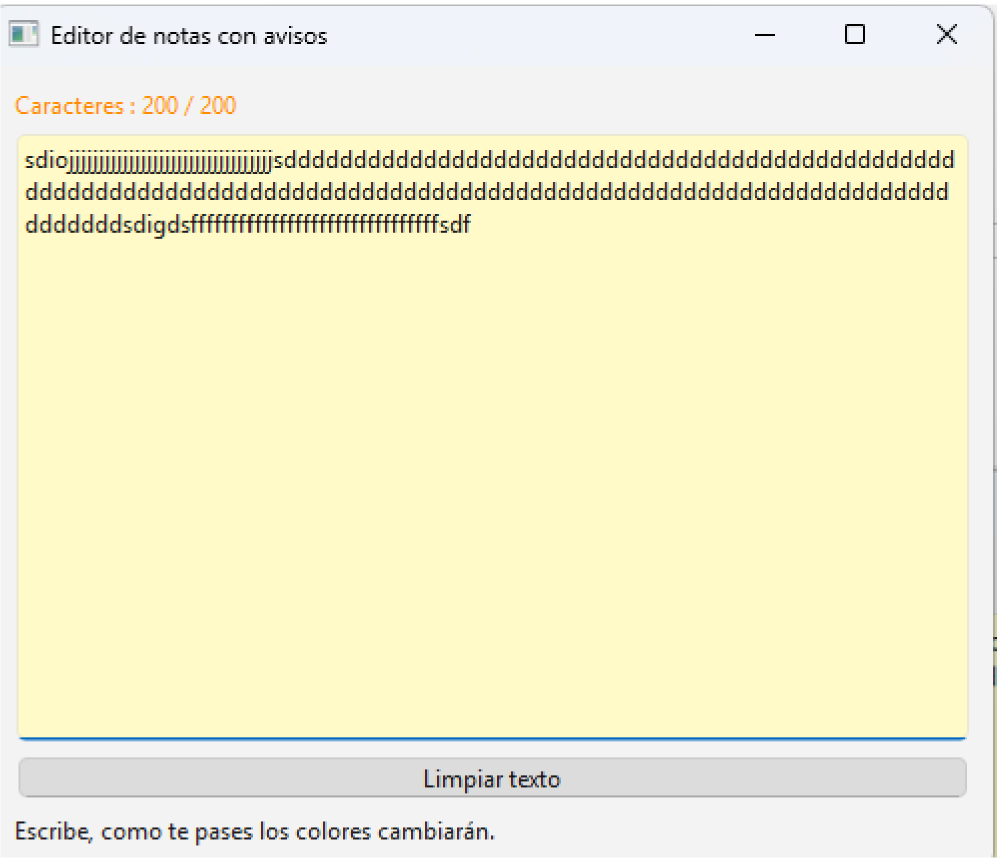
Explicación con capturas

Al inicio sale el fondo en azul clarito con algo de transparencia, luego un contador de caracteres y un botón de limpiar texto en gris.

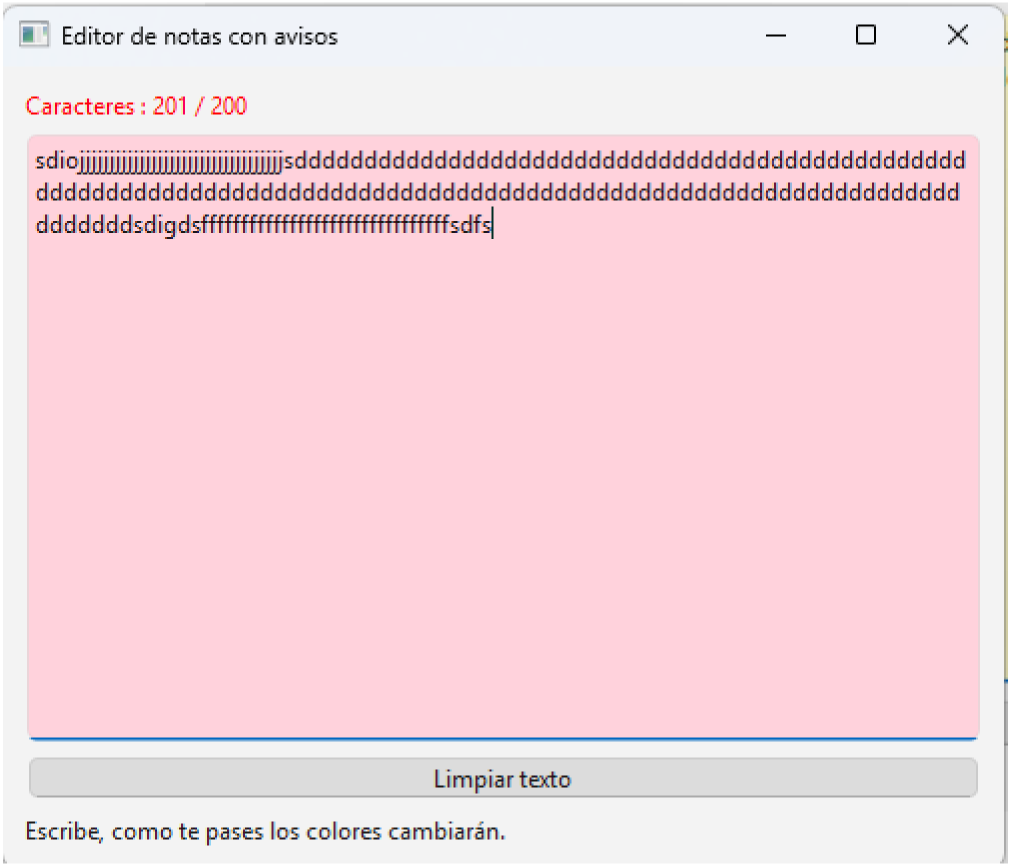


Si escribes 160 o 200 caracteres, el color de fondo de la nota se cambia a amarillo, como puedes ver en esta captura

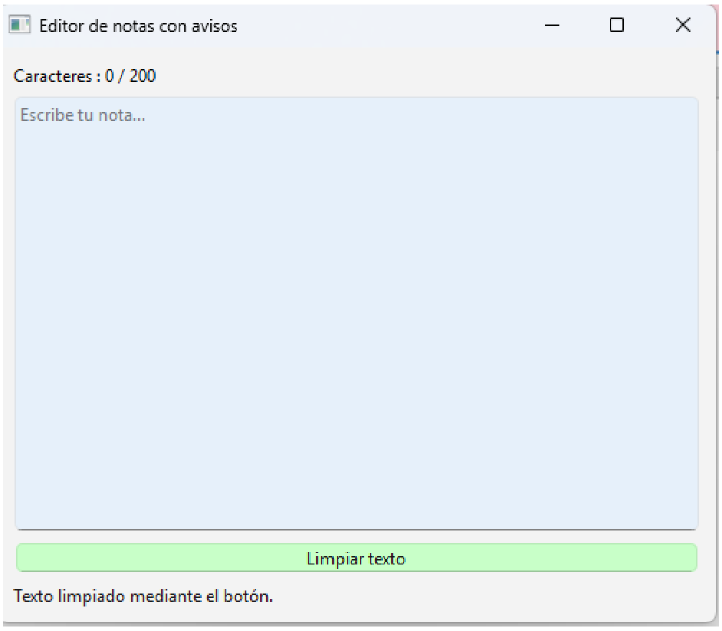

Editor de notas con avisos
Caracteres : 160 / 200
Limpiar texto
Escribe, como te pases los colores cambiarán.


Editor de notas con avisos
Caracteres : 200 / 200
Limpiar texto
Escribe, como te pases los colores cambiarán.

Pero si te pasas de 200, el color se cambia a rojo



Por último, si pulsamos el botón, este cambia de color a verde y el contenido se borra mostrando un mensaje de limpiado en el pie.



Explicación de derivación de cada widget

AreaTextoLimitada (QTextEdit)

He derivado este widget porque necesitaba agregarle un comportamiento que el QTextEdit no incluye:

- Establecer un límite de caracteres.
- emitir señales personalizadas cuando cambia la longitud de los caracteres.
- Modificar el fondo de forma dinámica con palette según el tamaño de los caracteres.

EtiquetaContadorCaracteres (QLabel)

La derivé porque el QLabel estándar solo permite ver el texto, pero yo quería:

- Método que permite actualizar el contador según los caracteres introducidos.
- Cambiar de forma dinámica el color de texto según la cantidad de caracteres introducidos.
- Actuar como componente independiente que reacciona a la señal del cuerpo de texto.

BotonLimpiarAviso (QPushButton)

Lo derivé porque el botón debía:

- Resetear el contenido del cuerpo del texto.
- Emitir señal personalizada para avisar de la acción.
- Cambiar el color con el uso de palette() después de resetear el texto.

2. Señales personalizadas implementadas

En AreaTextoLimitada

- **longitudCambiada (int)**
  Se emite cada vez que cambia el texto, enviando el número de caracteres actual al contador.
- **limiteSuperado (bool)**
  Informa si el límite de 200 caracteres ha sido superado.

En BotonLimpiarAviso

- **texto_limpiado ()**
  Se emite justo después de borrar el texto, para que la ventana principal pueda mostrar un aviso al usuario por el pie del documento de texto.

Funcionamiento del aviso visual

El aviso visual se basa en el uso de **palette()**, que permite cambiar colores de fondo y texto de cada widget.

AreaTextoLimitada: color de fondo

- **0% – 80% del límite (0–159 caracteres)** → fondo blanco
- **80% – 100% del límite (160–200)** → fondo amarillo suave
- **Más de 200 caracteres** → fondo rojo claro

Esto informa de forma inmediata al usuario sobre qué tan cerca está del límite.

EtiquetaContadorCaracteres: color del texto (Caracteres)

- **Normal (0–159)** → texto negro
- **Cerca del límite (160–200)** → texto naranja
- **Superado el límite (>200)** → texto rojo

Así el contador acompaña visualmente al área de texto.

BotonLimpiarAviso: color del botón

- **Estado inicial** → gris claro
- **Después de limpiar** → verde suave

Además, al limpiar se restablece el área a fondo blanco y aparece el mensaje “Texto limpiado correctamente”.