

## PUZZLE 2 (LCD)

```
Python
import tkinter as tk
from tkinter import scrolledtext
from rpi_lcd import LCD # Asegúrate de tener tu clase LCD aquí
import sys

class LCDApp:
    def __init__(self, root):
        self.root = root
        self.root.title("LCD Display Control")

        # Crear un widget de Text para introducir múltiples líneas de texto
        self.text_input = scrolledtext.ScrolledText(root, width=20,
height=4, wrap=tk.WORD, bg='blue', fg='white', insertbackground='white')
        self.text_input.grid(column=0, row=0, padx=10, pady=10)

        # Crear un botón para mostrar el texto en el display LCD
        self.display_button = tk.Button(root, text="Display",
command=self.display_text)
        self.display_button.grid(column=0, row=1, padx=10, pady=10)

        # Inicializar el LCD (asumiendo que ya tienes la clase LCD
configurada)
        self.lcd = LCD()

    def validate_input(self, event):
        # Obtener el texto actual
        current_text = self.text_input.get("1.0", tk.END).strip()

        # Limitar a 80 caracteres
        if len(current_text) > 80:
            # Si excede, eliminar el exceso
            self.text_input.delete("1.0", tk.END) # Limpiar el TextView
            self.text_input.insert("1.0", current_text[:80]) # Reinsertar
solo los primeros 80 caracteres

    def display_text(self):
        # Obtener el texto ingresado en el TextView (multilínea)
        user_input = self.text_input.get("1.0", tk.END).strip() # Obtener
desde la línea 1 a la última

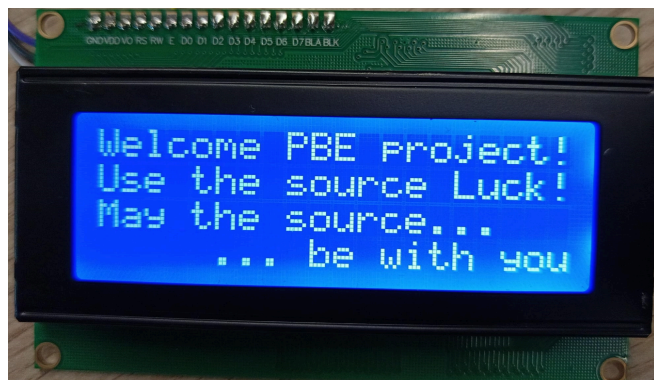
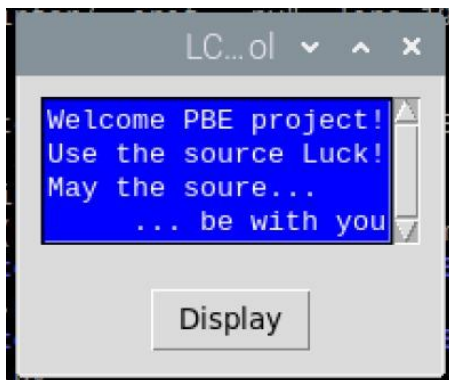
        # Limpiar el display antes de mostrar el nuevo texto
        self.lcd.clear()
```

```

        # Separar las líneas para imprimirlas en el LCD (máximo 4 líneas en
        el LCD 20x4)
        input_lines = user_input.split('\n')
        for i in range(min(len(input_lines), 4)): # Solo imprimimos 4
líneas como máximo
            line_text = input_lines[i]
            self.lcd.text(line_text, i + 1) # Mostrar cada línea en la fila
correspondiente (1-4)

# Función principal para correr la app
if __name__ == "__main__":
    root = tk.Tk()
    app = LCDApp(root)
    root.mainloop()

```



## Explicació del codi

He fet servir la llibreria tkinter per tenir un widget d'una barra de desplaçament vertical per si el text és molt llarg.

### **INICIALITZACIÓ**

He fet servir la funció `ScrolledText`, la qual permet a l'usuari ingressar varietat de línies de text. A part, aquesta funció també té l'opció de personalitzar el tamany del requadre i ajustar-lo al tamany del LCD. He posat el fons de color a blau i les lletres blanques per simular la pantalla del LCD.

Em vaig trobar amb el problema que si el text era de més de 20 caràcters de llarg i hi havia una paraula just a l'extrem, es partia la paraula. Aleshores, he fet servir la funció `WORD` que justament serveix per ajustar el text per paraules.

Pel botó de display, simplement he creat un botó amb la funció `display_button`, la qual crida a `display_text`.

### **validate\_input**

Aquesta funció és l'encarregada de limitar el text a 80 caràcters (20x4) i amb la funció `text_input.get("1.0", tk.END).strip()` obté el text complet que l'usuari ha escrit en el requadre. Si el text és més gran de 80 caràcters, es borra l'excés.

### **display\_text**

Aquesta funció és la que escriu el text introduït per l'usuari, a la pantalla del LCD.

Primer he fet un `.strip()` per eliminar els possibles espais en blanc addicionals. Després un `.clear()` per borrar el possible missatge que hi a la pantalla del LCD.

El text s'obté tot seguit, per tant he fet servir la funció `split` per separar el text cada '\n' i com que el display té 4 files, he fet un `for` per limitar a 4 línies el missatge i es va enviant al LCD.

Finalment en el main, he fet servir `app = LCDApp()` per generar tot l'entorn gràfic.