

Memòria puzzle 2

Primer de tot vaig comprobar que tingués la versió completa de Ruby al sistema, i si no era així, instal·lar-la amb el comandament:

```
pablo@raspberrypi:~ $ sudo apt install ruby-full
```

Seguidament vaig instal·lar les biblioteques necessàries per desenvolupar aplicacions que utilitzen GTK i WebKit amb el comandament:

```
pablo@raspberrypi:~ $ sudo apt install libgtk-3-dev libgirepository1.0-dev libwebkit2gtk-4.0-dev
```

Una vegada instal·lat tot, vaig anar fent diferents codis per entendre com funciona i familiaritzar-me amb l'entorn.

El primer que vaig provar va ser un codi que ja tingui el text predeterminat i només el mostrés al terminal. El codi i el resultat van ser els següents:

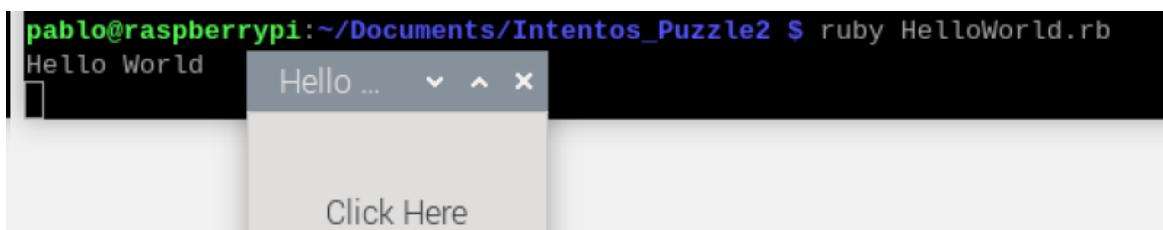
```
require 'gtk3'

app = Gtk::Application.new("org.gtk.example", :flags_none)

app.signal_connect "activate" do |application|
  window = Gtk::ApplicationWindow.new(application)
  window.set_title("Hello World")
  window.set_default_size(100, 100)

  button = Gtk::Button.new(label: "Click Here")
  button.signal_connect "clicked" do
    puts "Hello World"
  end
  window.add(button)
end

window.show_all
app.run
```



Després, vaig provar a introduir jo el text i que es mostri a la propia graella amb el següent codi i resultat:

```
require 'gtk3'

app = Gtk::Application.new("org.gtk.example", :flags_none)

app.signal_connect "activate" do |application|
  window = Gtk::ApplicationWindow.new(application)
  window.set_title("Texto Personalizado")
  window.set_default_size(300, 200)

  # Campo de texto para ingresar el texto
  entry = Gtk::Entry.new
  entry.set_placeholder_text("Ingresa tu texto aquí...")

  # Etiqueta para mostrar el texto ingresado
  label = Gtk::Label.new("")

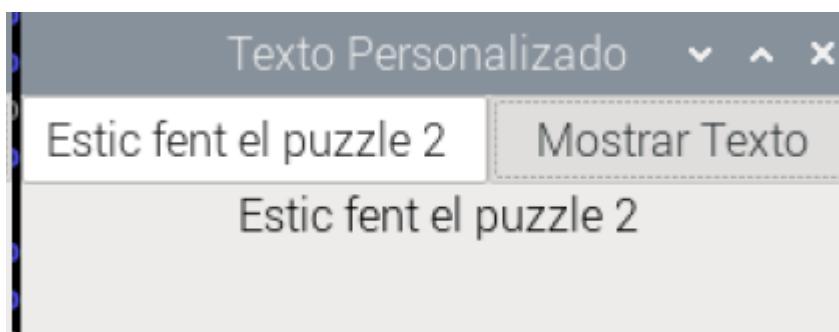
  # Botón para mostrar el texto
  button = Gtk::Button.new(label: "Mostrar Texto")amb el comandament:
  button.signal_connect "clicked" doamb el comandament:
    # Obtener el texto del campo de texto y establecerlo en la etiquetaamb el comandament:
    text = entry.text
    label.set_text(text)
  end

  # Crear un cuadro de rejilla para organizar los widgets
  grid = Gtk::Grid.new
  grid.attach(entry, 0, 0, 1, 1)
  grid.attach(button, 1, 0, 1, 1)
  grid.attach(label, 0, 1, 2, 1)

  window.add(grid)

  window.show_all
end

app.run
```



I finalment, vaig fer que el text s'escrigui a l'LCD. Vaig copiar el codi del puzzle 1 de la classe LCD aquí:

```
require 'i2c/drivers/lcd'
require 'gtk3'

class LCD
    def initialize(address)
        @display = I2C::Drivers::LCD::Display.new('/dev/i2c-1', address, rows=20, cols=4)
        @address = address
    end

    def show_string(string)
        @display.clear
        lines = string.split("\n")

        if lines.length>4
            @display.text(" Demasiadas lineas",1)
            @display.text(" Maximo de lineas:4",2)
        else
            i = 0
            while i < lines.length do
                if lines[i].length>20
                    lines[i] = "#{lines[i][0...17]}..."
                end
                @display.text(lines[i],i)
                i = i + 1
            end
        end
    end

end

end|
```

I tot seguit creava l'app amb els tamans i posicions dels botons que volia i e que feia era que quan es toqués el botó de "Mostrar Texto", cridi al codi del puzzle 1 perquè es mostri a l'LCD. Aquesta és la part que falta de codi:

```
lcd = LCD.new(0x27)

app = Gtk::Application.new("org.gtk.example", :flags_none)
app.signal_connect "activate" do |application|
    window = Gtk::ApplicationWindow.new(application)
    window.set_title("Puzzle 2")
    window.set_default_size(300, 100)

    # Campo de texto para ingresar el texto
    text_view = Gtk::TextView.new
    text_view.set_wrap_mode(Gtk::WrapMode::WORD) # Ajuste de palabras
    text_view.set_size_request(300, 100) # Tamaño fijo del área de texto

    # Botón para enviar el texto
    button = Gtk::Button.new(label: "Mostrar Texto")

    button.signal_connect "clicked" do
        buffer = text_view.buffer
        text = buffer.text
        # message = text.join("")
        lcd.show_string(text)
    end

    # Crear un cuadro de rejilla para organizar los widgets
    grid = Gtk::Grid.new
    grid.attach(text_view, 0, 0, 2, 1)
    grid.attach(button, 0, 1, 2, 1)

    window.add(grid)
    window.show_all
end
app.run
```

I aquests són els resultats:

