

CAMBIOS EN EL CÓDIGO CON IA

En este documento vamos a detallar los cambios que le pedimos a ChatGPT que nos hiciera para así poder mejorar nuestra GUI. Vamos a poner partes del código que habian antes y como ha quedado ahora para mostrar la diferencia.

1. Organizacion por pestañas

Pensamos que sería una buena idea que nuestra GUI estuviera organizada por pestañas según el tema que quisieras tocar. Por ejemplo una pestaña para las funciones que son de procesar imágenes, una pestaña para las funciones que son de procesar videos y una pestaña con otras funcionalidades. Antes de esto en nuestra Gui simplemente teníamos que todas las funciones aparecían con el nombre, le dabas y te pedía los atributos que necesitabas para realizar la función.

```
tk.Button(self.root, text="Convertir RGB a YUV",
command=self.convert_rgb_to_yuv).pack(pady=5)
tk.Button(self.root, text="Redimensionar Imagen",
command=self.resize_image).pack(pady=5)
```

Código anterior

```
def create_image_tab(self):
    """Crea la pestaña de procesamiento de imágenes."""
    self.image_tab = tk.Frame(self.notebook, bg="#ADD8E6")
    self.notebook.add(self.image_tab, text="Procesar Imágenes")

    ttk.Button(self.image_tab, text="Convertir RGB a YUV", style="Image.TButton", command=self.convert_rgb_to_yuv).pack(pady=5)
    ttk.Button(self.image_tab, text="Redimensionar Imagen", style="Image.TButton", command=self.resize_image).pack(pady=5)
    ttk.Button(self.image_tab, text="Convertir a Blanco y Negro y Comprimir", style="Image.TButton", command=self.convert_bw_compress).pack(pady=5)
```

*Código para crear la pestaña de procesamiento de imágenes.
(Código generado por ChatGPT)*

2. Colores de la GUI

Quisimos ponerle un color diferente a cada pestaña y ponerle también los colores acordes al fondo a los botones. Quisimos hacer un pequeño gesto a nuestro equipo favorito, y poner los colores de este en cada una de las pestañas. En la imagen anterior del código nuevo se ve una pequeña parte de la mejora que es la línea de `self.image.tab` donde le ponemos el color a cada pestaña. Luego en los botones hemos añadido un color diferente para cada pestaña. Y el código lo vemos a continuación. De esta parte el código antiguo simplemente era el básico

```
tk.Button(self.root, text="Convertir RGB a YUV", command=self.convert_rgb_to_yuv).pack(pady=5)
```

Esta es la línea de código que teníamos para cada botón, en la imagen de abajo veremos que le hemos añadido los colores

```
style = ttk.Style()
style.configure("Image.TButton", font=("Pacífico", 10, "bold"), foreground="white", background="#004d98") # Azul oscuro
style.map("Image.TButton", background=[("active", "#5A9BD5")]) # Azul más claro al hacer clic
```

Código para ponerle color a los botones (Código generado por ChatGPT)

En esta imagen también vemos que podemos modificar el tipo de letra, que hemos modificado de 'Arial' a 'Pacífico' , y el tamaño que este si que lo hemos dejado igual porque ya estaba bien.

3. Barra de progreso

Por último hemos añadido una barra de progreso para cada vez que se esté ejecutando una función salga la barra conforme se está haciendo. Esto no tiene código anterior, porque directamente no existía, pero ahora el código que hemos añadido es el siguiente. Aparte de esto en cada función se llama a `send_request_with_progress` para que salga cuando se utiliza.

```
def send_request_with_progress(self, endpoint, data):
    """Envía una solicitud y muestra una barra de progreso mientras se procesa."""
    progress = Progressbar(self.root, orient="horizontal", mode="indeterminate", length=300)
    progress.pack(pady=10)
    progress.start() # Inicia la animación de la barra de progreso

    # Enviar la solicitud en un hilo separado para no bloquear la interfaz
    threading.Thread(target=self.send_request, args=(endpoint, data, progress)).start()
```

Código para añadir una barra de progreso(Código generado por ChatGPT)

3. Mensaje de bienvenida

Quisimos pedirle también que nos hiciera un mensaje de bienvenida aprovechando que teníamos el primer get hecho y así usarlo de mensaje de bienvenida, no hemos cambiado nada ya que antes no teníamos mensaje y hemos añadido código para que nos saliera la pestaña con el mensaje, es el siguiente

```
def show_initial_message(self):
    try:
        url = "http://127.0.0.1:8000/"
        response = requests.get(url)
        response.raise_for_status()
        mensaje = response.json().get("mensaje", "¡Bienvenido a Monster API GUI!")
        messagebox.showinfo("Mensaje Inicial", mensaje)
    except requests.exceptions.RequestException as e:
        messagebox.showerror("Error", f"No se pudo obtener el mensaje inicial: {str(e)}")
```

Código para añadir mensaje de bienvenida(Código generado por ChatGPT)

Nos hubiera gustado jugar mucho más con la GUI y poder modificar muchas cosas mas, pero por falta de tiempo porque tenemos los exámenes muy cercanos no hemos podido.