

# DOCUMENTACIÓN TÉCNICA

## SISTEMA DE CONVERSIÓN DE COLOR AVANZADO

Ing. Juan Sandoval

### 1. Introducción

Aplicación desarrollada en Python para conversión entre espacios de color (RGB, CMYKW, HSL), integrando:

Captura de imágenes (cámara/archivos)

Comunicación industrial vía Modbus TCP/Serial

Interfaz gráfica intuitiva

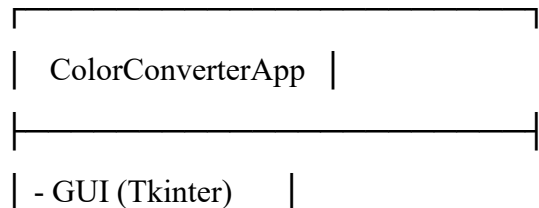
### 2. Especificaciones Técnicas

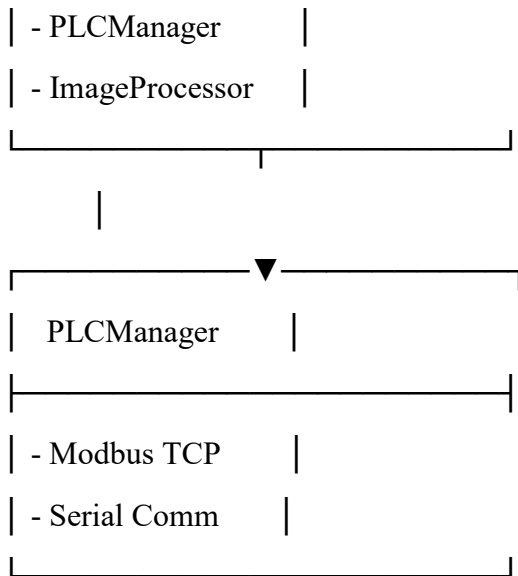
#### 2.1. Requisitos del Sistema

Componente	Versión
Python	3.8+
OpenCV (cv2)	4.5+
Pillow (PIL)	9.0+
pymodbus (opcional)	3.0+
pyserial (opcional)	3.5+

### 3. Arquitectura del Sistema

#### 3.1. Diagrama de Componentes





### 3.2. Flujo de Datos

Entrada: Imagen/Cámara/Selectores → RGB

Conversión: RGB ↔ CMYKW ↔ HSL

Salida: Envío a PLC/Visualización

## 4. Algoritmos Clave

### 4.1. RGB → CMYKW

python

```

def rgb_to_cmykw(r, g, b):
    c = 1 - r/255
    m = 1 - g/255
    y = 1 - b/255
    k = min(c, m, y)
    # Ajuste componentes
    return c, m, y, k, 0 if k < 1 else 100
  
```

### 4.2. HSL → RGB

python

```

def hsl_to_rgb(h, s, l):
    h /= 360; s /= 100; l /= 100
    q = l*(1 + s) if l < 0.5 else l + s - l*s
  
```

```
p = 2*1 - q  
  
# Conversión a RGB via hue  
  
return r, g, b
```

---

## 5. Comunicación Industrial

### 5.1. Protocolos Soportados

Protocolo	Configuración	Mensaje Ejemplo
Modbus TCP	IP: 192.168.0.10, Port:502	WriteRegisters(addr=0, [C,M,Y,K,W])
Serial RS232	9600 baud, 8N1	"C:100 M:050 Y:000 K:000\n"

### 5.2. Control de Errores

Timeout: 1 seg

Reintentos: 0 (gestión manual)

Throttling: 100ms entre envíos

---

## 6. Interfaz Gráfica

### 6.1. Componentes Principales

Canvas: Visualización imagen/cámara (450x350 px)

Selectores:

Sliders CMYKW (0-100%)

Círculo cromático HSL

Paletas: Colores predefinidos (RGB)

Historial: Últimos 10 colores seleccionados

### 6.2. Modos de Operación

Modo	Descripción
Cámara	Captura en tiempo real
Imagen	Carga desde archivo

Modo	Descripción
Selector HSL	Ajuste manual en círculo HSL

---

## 7. Configuración y Persistencia

### 7.1. Archivo JSON (color\_app\_config.json)

json

```
{
  "dark_mode": false,
  "plc_ip": "192.168.0.10",
  "serial_port": "COM3",
  "baudrate": 9600
}
```

## 8. Consideraciones de Rendimiento

Hilos: Comunicación PLC en segundo plano

Optimizaciones:

Redimensión de imágenes manteniendo aspect ratio

Caching de conversiones de color

## 9. Manejo de Errores

Código	Descripción	Solución
CAM-001	Cámara no accesible	Verificar dispositivos USB
PLC-002	Fallo conexión Modbus	Revisar IP/puerto/firewall
SER-003	Error puerto serial	Chequear baudrate/cableado

---

## 10. Conclusiones y Recomendaciones

Ventajas:

Soporte multiplataforma (Windows/Linux)

## Integración con sistemas industriales

### Mejoras Futuras:

## Añadir soporte para perfiles ICC

## Implementar comunicación MQTT

## Anexo A: Instalación de Dependencias

bash

```
pip install opencv-python pillow numpy pymodbus pyserial
```

## Anexo B: Capturas de Pantalla