



Centro de Investigación en
Tecnologías para la Sociedad
Facultad de Ingeniería

Requerimientos para el Sistema de Captura de Imágenes Foto UART Drop-in

(código: FOTO-UART-DROPIN-RS01-REQ)

Autor principal:
Facultad de Ingeniería

Director:
Dra. Zoë Fleming (UDD)

Investigadores:
Alejandro Rebolledo.

Índice

Introducción	5
1.1 Propósito	5
1.2 Ámbito del sistema	5
1.3 Definiciones, Acrónimos y Abreviaturas	5
1.4 Referencias	5
1.5 Visión general del documento	6
Descripción general	6
2.1 Perspectiva del producto	6
2.2 Funciones del producto	6
2.3 Usuarios	6
2.4 Restricciones	7
Requisitos específicos	7
3.1 Priorización y estabilidad	7
3.2 Interfaces externas	7
3.3 Funciones	8
3.4 Requisitos de rendimiento	8
3.5 Restricciones de diseño	8
3.6 Atributos del sistema	9
3.7 Otros requisitos	9
3.8 Entradas inválidas y manejo de errores	9
3.9 Matriz de trazabilidad (REQ a TEST)	9
Anexos	10
4.1 Protocolo de Comunicación UART	10
4.2 Entorno de pruebas sin ESP32 (USB/micro-USB)	10
4.3 Configuración JSON (ejemplo)	10

4.4 Casos de Prueba (resumen)	11
---	----

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
FOTO-UART-RS01-REQ	Creación del documento basado en especificaciones técnicas	31 de agosto de 2025

1. Introducción

1.1. Propósito

El objetivo principal de este Documento de Especificación de Requisitos del Sistema (ERS) es definir los requisitos técnicos necesarios para el desarrollo de un sistema de captura y transmisión de imágenes compatible con el protocolo UART existente. La solución propuesta debe funcionar como un “drop-in replacement” que permita capturar, procesar y transmitir imágenes manteniendo compatibilidad con sistemas ESP32 existentes, sin requerir modificaciones en el hardware receptor.

1.2. Ámbito del sistema

- El software se denomina **FOTO-UART-DROPIN** (Sistema de Captura de Imágenes Compatible por UART).
- El sistema capturará imágenes, aplicará mejoras opcionales y transmitirá por UART usando un protocolo de *handshake*. Mantiene compatibilidad con receptores ESP32 existentes.
- El sistema no incluye mantenimiento post-implementación, actualizaciones de firmware remotas, ni soporte técnico continuo.

1.3. Definiciones, Acrónimos y Abreviaturas

Abreviatura	Descripción
ACK	Confirmación de recepción correcta de un chunk (“ACK\p”).
AF	<i>Auto Focus</i> (autofoco) de la cámara.
CLAHE	<i>Contrast Limited Adaptive Histogram Equalization</i> .
CM3	Camera Module 3 (IMX708), cámara oficial de Raspberry Pi con AF.
CSI-2	MIPI <i>Camera Serial Interface 2</i> (bus de cámara en Raspberry Pi).
JPEG	Formato de compresión de imágenes Joint Photographic Experts Group.
UART	Puerto serie asíncrono (8N1 @ 115 200 Bd).
USB	Universal Serial Bus; usado en pruebas como gadget CDC-ACM.

1.4. Referencias

1. IEEE Std 830-1998. *IEEE Recommended Practice for Software Requirements Specifications*.
2. Raspberry Pi Foundation. (2025). *Camera Module 3 (docs generales)*.
3. Espressif Systems. (2024). *ESP32 Technical Reference Manual (PDF)*.
4. ISO/IEC 8859-1:1998. *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*.
5. RFC 3986. (2005). *Uniform Resource Identifier (URI): Generic Syntax*.
6. Alejandro Rebolledo. (2025). *fotoForward* (Repositorio GitHub). Accedido el 31-08-2025.

1.5. Visión general del documento

Este documento sigue el espíritu del estándar IEEE 830 e introduce requisitos verificables y trazables para un sistema de captura de imágenes compatible con UART.

2. Descripción general

2.1. Perspectiva del producto

Interoperabilidad:

Mantiene compatibilidad total con receptores ESP32 respetando el protocolo UART.

Reemplazo modular:

Sustituye sistemas existentes sin cambios en el receptor.

Independencia:

Autonomía en captura, procesamiento y transmisión.

Escalabilidad:

Permite nuevos sensores/algoritmos sin afectar la interfaz UART.

2.2. Funciones del producto

Captura de imágenes:

Con cámaras soportadas (p.ej., CM3 IMX708 o webcam USB).

Procesamiento:

Mejoras opcionales (CLAHE, *unsharp*), redimensionado con AR.

Comunicación UART:

Protocolo exacto READY/ACK/DONE con chunks de 256 B.

Almacenamiento local:

Guarda original (full-res) y procesada (enhanced).

Errores: Timeouts y un reintento total automático.

2.3. Usuarios

Desarrolladores:

Integración y ampliación.

Integradores:

Puesta en marcha y diagnóstico.

Operadores:

Uso mediante comandos simples.

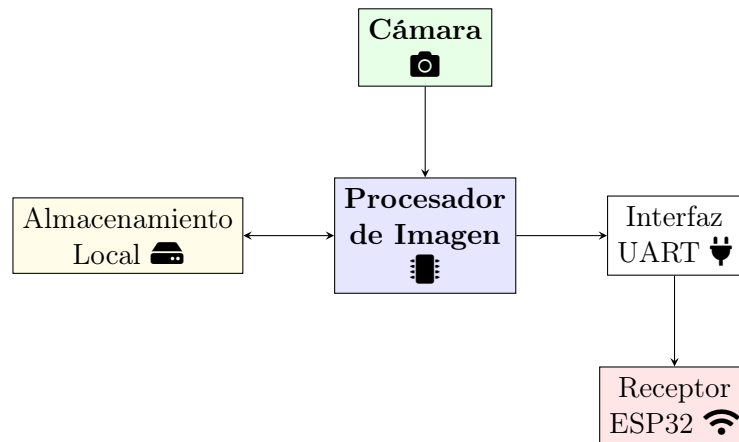


Figura 1. Esquema de bloques del sistema de captura de imágenes.

2.4. Restricciones

Protocolo:

UART 115 200 Bd 8N1, comandos exactos y secuencias de *handshake*.

Memoria:

Procesamiento por chunks para reducir RAM.

Hardware:

Soporta distintos tipos de cámara con misma salida UART.

Tiempo: SLA definido en §3.4.

Lenguaje:

Implementación en C/C++/Python según plataforma.

3. Requisitos específicos

3.1. Priorización y estabilidad

Cada requisito **RS01-REQxx** se clasifica como {Esencial, Condicional, Opcional} y estabilidad {Alta, Media, Baja}. Se incluye en línea al final de cada requisito.

3.2. Interfaces externas

[FOTO-UART-DROPIN-RS01-REQ01]

UART a 115 200 Bd (8N1). *Prioridad: Esencial; Estabilidad: Alta.*

[FOTO-UART-DROPIN-RS01-REQ02]

Comando de inicio: “foto\n” con variantes “foto [ancho]\n” y “foto [ancho] [calidad]\n”. Codificación UTF-8, terminador LF. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ03]

Header exacto: “YYYYMMDD_HHMMSS|tamaño_bytes\n”. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ04]

Handshake: esperar “READY\n”; transmitir en chunks de 256 B; esperar “ACK\n” por cada chunk; finalizar esperando “DONE\n”. *Esencial; Alta.*

3.3. Funciones

[FOTO-UART-DROPIN-RS01-REQ05]

Captura full-res y generación de versión procesada/redimensionada para UART. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ06]

Mejoras opcionales (CLAHE, *unsharp*) configurables. *Condicional; Media.*

[FOTO-UART-DROPIN-RS01-REQ07]

Redimensionado manteniendo AR; ancho por defecto 1024 px. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ08]

Almacenamiento local: original en **fullres/** y procesada en **enhanced/**, ambos con timestamp. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ09]

Reintento automático: un único reintento total ante timeout/NACK_TIMEOUT. *Esencial; Alta.*

3.4. Requisitos de rendimiento

[FOTO-UART-DROPIN-RS01-REQ10]

SLA de tiempo (≤ 110 KB): desde foto hasta header ≤ 5 s; tiempo total hasta DONE ≤ 10 s a 115 200 Bd. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ11]

Chunks de 256 B exactos (último puede ser menor). Timeout por ACK=10 s. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ12]

Compatibilidad con tamaños mayores por chunks (sin garantía de SLA de 10 s). *Opcional; Media.*

[FOTO-UART-DROPIN-RS01-REQ13]

Parámetros: ancho [100, 4608] px; calidad [1, 10] mapeada a JPEG [10, 100]. *Esencial; Alta.*

3.5. Restricciones de diseño

[FOTO-UART-DROPIN-RS01-REQ14]

Timestamps: YYYYMMDD_HHMMSS. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ15]

Imágenes en JPEG con calidad configurable; opción progresivo habilitable. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ16]

Mapeo de calidad: `jpeg_quality = clamp(quality_param,1,10) * 10`. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ17]

Estructura de directorios: `storage/fullres`, `storage/enhanced`, `storage/logs`. *Esencial; Alta.*

3.6. Atributos del sistema

[FOTO-UART-DROPIN-RS01-REQ18]

Logging con rotación: operaciones, timestamps, errores, reintentos. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ19]

Robustez ante desconexiones: timeouts y recuperación automática. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ20]

Validación de comandos y respuesta a malformados (`ERR_CMD\n`). *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ21]

Indicadores de estado durante captura y transmisión. *Condicional; Media.*

3.7. Otros requisitos

[FOTO-UART-DROPIN-RS01-REQ22]

Configuración por JSON: puerto, rutas y cámara/perfiles. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ23]

Diagnóstico: verificación de cámara y puerto UART. *Esencial; Alta.*

[FOTO-UART-DROPIN-RS01-REQ24]

Plataformas: Raspberry Pi OS, Ubuntu, Debian con libcamera/OpenCV. *Esencial; Alta.*

3.8. Entradas inválidas y manejo de errores

Comandos malformados (p.ej., “foto abc” o calidad fuera de rango) generan `ERR_CMD\n`, se registra en log y el sistema permanece a la espera del siguiente comando.

3.9. Matriz de trazabilidad (REQ a TEST)

REQ	Descripción	Caso de prueba
RS01-REQ01	UART 115200 8N1 exacto	CP-PROTO-01
RS01-REQ04	READY/ACK/DONE exactos	CP-PROTO-01, CP-TIMEOUT-02
SLA	≤ 10 s hasta 110 KB	CP-REN-01
Chunks	256 B exactos	CP-CHNK-01
Almacenamiento	fullres + enhanced	CP-IO-02

4. Anexos

4.1. Protocolo de Comunicación UART

Secuencia completa

1. **Comando inicial:** Receptor envía “foto\n”
2. **Preparación:** Sistema captura y procesa imagen
3. **Header:** Sistema envía “YYYYMMDD_HHMMSS|tamaño_bytes\n”
4. **Confirmación:** Receptor responde “READY\n”
5. **Transmisión:** Sistema envía chunks de 256 B
6. **ACK:** Receptor confirma cada chunk con “ACK\n”
7. **Finalización:** Receptor envía “DONE\n”

4.2. Entorno de pruebas sin ESP32 (USB/micro-USB)

1. **USB Gadget (CDC-ACM):** habilitar dwc2,g_serial en /boot/cmdline.txt; RPi expone /dev/ttyGS0 y el PC ve /dev/ttyACM0.
2. **UART TTL:** dongle USB-TTL (3.3 V) en /dev/serial0 (RPi) y /dev/ttyUSB0 (PC).
3. **Simulador Host (PC):** envía foto[[ancho][calidad]]\n, espera header, responde READY\n, ACK\n por chunk y DONE\n. Soporta NACK_TIMEOUT.

4.3. Configuración JSON (ejemplo)

```
{
  "serial": { "puerto": "/dev/ttyAMA0", "baudrate": 115200, "timeout": 1 },
  "imagen": {
    "ancho_default": 1024,
    "calidad_default": 5,
    "chunk_size": 256,
    "ack_timeout": 10,
    "jpeg_progressive": true
  },
  "almacenamiento": {
    "directorio_fullres": "storage/fullres",
    "directorio_enhanced": "storage/enhanced",
    "mantener originales": true,
    "logs_dir": "storage/logs"
  },
  "procesamiento": { "aplicar_mejoras": true, "unsharp_mask": true, "clahe_enabled": true },
  "limites": { "max_jpeg_bytes": 112640, "fallback_quality_drop": 10 }
}
```

4.4. Casos de Prueba (resumen)

Cuadro 1. Casos de prueba obligatorios

Caso de Prueba	Entrada	Resultado Esperado
CP-PROTO-01	"foto\n"	Header + transmisión completa
CP-REN-01	"foto 1024 5\n"	$t_{\text{header}} \leq 5\text{ s}$, $t_{\text{total}} \leq 10\text{ s}$
CP-CHNK-01	Flujo normal	256 B exactos por chunk
CP-TIMEOUT-02	Sin READY o sin ACK	Reintento total único + nuevo header
CP-IO-02	N/A	Guardado en fullres/enhanced