

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



TESIS

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
DETECCIÓN PRELIMINAR NO INVASIVA DE ANEMIA
MEDIANTE CLASIFICACIÓN DE LA CONJUNTIVA OCULAR
POR APRENDIZAJE PROFUNDO**

PRESENTADO POR:

Br. JOSE HUMBERTO FUENTES BEINGOLEA

**PARA OPTAR AL TÍTULO PROFESIONAL
DE INGENIERO ELECTRÓNICO**

ASESOR:

Dr. Ing. FACUNDO PALOMINO QUISPE

CUSCO - PERÚ

2026

Universidad Nacional de San Antonio Abad del Cusco
INFORME DE SIMILITUD
(Aprobado por Resolución Nro.CU-321-2025-UNSAAC)

El que suscribe, el Asesor FACUNDO PALOMINO QUISPE quien aplica el software de detección de similitud al trabajo de investigación/tesis titulada: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN PRELIMINAR NO INVASIVA DE ANEMIA MEDIANTE CLASIFICACIÓN DE LA CONJUNTIVA OCULAR POR APRENDIZAJE PROFUNDO.

Presentado por: JOSE HUMBERTO FUENTES BEINGOLEA con DNI N° 70676674. Para optar el título Profesional/Grado Académico de INGENIERO ELECTRÓNICO.

Informo que el trabajo de investigación ha sido sometido a revisión por 2... veces, mediante el Software de Similitud, conforme al Art. 6° del Reglamento para Uso del Sistema Detección de Similitud en la UNSAAC y de la evaluación de originalidad se tiene un porcentaje de 3...%.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10 %	No sobrepasa el porcentaje aceptado de similitud.	X
Del 11 al 30 %	Devolver al usuario para las subsanaciones.	
Mayor a 31 %	El responsable de la revisión del documento emite un informe al inmediato jerárquico, conforme al reglamento, quien a su vez eleva el informe al Vicerrectorado de Investigación para que tome las acciones correspondientes; Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y adjunto las primeras páginas del reporte del Sistema de Detección de Similitud.

Cusco, 20 de Mayo de 2026



Firma

Post firma: FACUNDO PALOMINO QUISPE

Nro. de DNI: 00435194

ORCID del Asesor: 0000-0002-5947-6682

Se adjunta:

1. Reporte generado por el Sistema Antiplagio.
2. Enlace del Reporte Generado por el Sistema de Detección de Similitud: trn:oid::27259:593326807

TESIS_post17.pdf

 Universidad Nacional San Antonio Abad del Cusco

Detalles del documento

Identificador de la entrega

trn:oid:::27259:593326807

Fecha de entrega

20 may 2026, 4:39 p.m. GMT-5

Fecha de descarga

20 may 2026, 4:45 p.m. GMT-5

Nombre del archivo

TESIS_post17.pdf

Tamaño del archivo

32.6 MB

170 páginas

39.530 palabras

214.677 caracteres

8% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...




Filtrado desde el informe

- ▶ Bibliografía
- ▶ Texto citado
- ▶ Texto mencionado
- ▶ Coincidencias menores (menos de 8 palabras)

Exclusiones


- ▶ N.º de coincidencias excluidas

Fuentes principales

- 6%  Fuentes de Internet
- 2%  Publicaciones
- 6%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

-  **Texto oculto**
438 caracteres sospechosos en N.º de páginas
El texto es alterado para mezclarse con el fondo blanco del documento.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Dedicatoria

A mi madre, quien me dio la vida más de una vez.

A Madoka.

Agradecimientos

Mis mayores gratitudes hacia mis padres, Maritza y Vicente, quienes depositaron su confianza en mí desde el primer día, y cuyo ejemplo y valores fueron guía para mi trabajo y mi vida; así como hacia mi familia, por su apoyo incondicional que me permitió continuar en un camino que, en ocasiones, presentó dificultades y desafíos.

Asimismo, extendo agradecimientos al Dr. Ing. Facundo Palomino Quispe, mi asesor, por su orientación y paciencia para el correcto desarrollo de este trabajo. Agradezco a aquellos voluntarios anónimos que sin esperar retribución dieron su permiso para el tratamiento de sus datos y, finalmente, a los buenos amigos que conocí durante mis años de estudio.

*Por el futuro de una nueva era
yo intentaré
cambiar mi mundo...
...cambiaré al mundo.*

– «New Genesis» - Ado.

Resumen

La anemia constituye un problema crítico de salud pública cuya detección oportuna enfrenta severas limitaciones logísticas en zonas alejadas de países en desarrollo como Perú. Para abordar esta problemática, se presenta un sistema electrónico prototipo para la estimación de hemoglobina y detección preliminar *in-situ* de anemia de manera no invasiva, mediante el análisis de imágenes de la conjuntiva ocular. La validación del sistema se realizó en una población de altura (Cusco, 3,339 m.s.n.m.), empleando un hardware basado en la plataforma embebida *Orange Pi 5 Max* con iluminación controlada. Para mitigar la variabilidad entre *datasets*, se aplicó una estrategia de adaptación de dominio mediante la técnica *Reinhard*, logrando reducir la brecha entre conjuntos públicos y locales en un 73.01 %. En el ámbito del aprendizaje profundo, se implementaron arquitecturas optimizadas: una red *U-Net* para la segmentación semántica de conjuntiva, y el modelo propuesto *Hemo-ConViT* (basado en *Vision Transformers*) con función de pérdida híbrida. Los resultados experimentales en entorno controlado demostraron un Error Absoluto Medio (MAE) de 0.6075 g/dL y un coeficiente de determinación (R^2) de 0.7486 frente al método de referencia *HemoCue*. La variabilidad estadística fue validada mediante pruebas de *Bland-Altman*, mostrando concordancia significativa. Para la tarea de clasificación binaria en datos locales, con umbral de 15.3 g/dL, el sistema alcanzó una exactitud global del 96.20 % y una sensibilidad del 89.47 %. Estos hallazgos corroboran la viabilidad técnica de esta prueba de concepto y prototipo como herramientas de bajo costo para entornos de recursos limitados.

Palabras clave: Anemia; Clasificación de conjuntiva; Estimación de hemoglobina; Vision transformers; Sistemas embebidos.

Abstract

Anemia constitutes a critical public health problem whose timely detection faces severe logistical limitations in remote areas of developing countries such as Peru. To address this issue, a prototype electronic system for non-invasive hemoglobin estimation and preliminary *in-situ* anemia detection is presented through the analysis of ocular conjunctiva images. System validation was conducted in a high-altitude population (Cusco, 3,339 m.a.s.l.), employing hardware based on the embedded *Orange Pi 5 Max* platform with controlled illumination. To mitigate variability between *datasets*, a domain adaptation strategy using the *Reinhard* technique was applied, achieving a 73.01 % reduction in the gap between public and local datasets. In the field of deep learning, optimized architectures were implemented: a *U-Net* network for conjunctiva semantic segmentation, and the proposed *Hemo-ConViT* model (based on *Vision Transformers*) with a hybrid loss function. Experimental results in a controlled environment demonstrated a Mean Absolute Error (MAE) of 0.6075 g/dL and a coefficient of determination (R^2) of 0.7486 compared to the reference *HemoCue* method. Statistical variability was validated through *Bland-Altman* tests, showing significant agreement. For the binary classification task on local data, with a threshold of 15.3 g/dL, the system achieved an overall accuracy of 96.20 % and a sensitivity of 89.47 %. These findings corroborate the technical feasibility of this proof of concept and prototype as low-cost tools for resource-limited environments.

Keywords: Anemia; Conjunctiva classification; Hemoglobin estimation; Vision transformers; Embedded systems.

Índice general

Índice general	VII
Índice de tablas	XII
Índice de figuras	XIV
Introducción	1
1. Aspectos generales	3
1.1. Descripción de la realidad problemática	3
1.2. Formulación del problema	6
1.2.1. Problema general	6
1.2.2. Problemas específicos	6
1.3. Justificación	7
1.3.1. Justificación Social	7
1.3.2. Justificación Económica	7
1.3.3. Justificación Tecnológica	8
1.3.4. Justificación Académica	8
1.4. Objetivos	8
1.4.1. Objetivo general	8
1.4.2. Objetivos Específicos	8
1.5. Alcances	9
1.6. Limitaciones	10

2. Marco teórico	11
2.1. Antecedentes	11
2.1.1. Tesis Internacionales	11
2.1.2. Artículos científicos relacionados	13
2.2. Bases Teóricas	19
2.2.1. Fisiopatología de la Anemia y Adaptación a la Altura	19
2.2.2. Enfoque de Detección: Tamizaje vs. Diagnóstico	21
2.2.3. Métodos de Cuantificación de Hemoglobina	21
2.2.4. Anatomía de la Conjuntiva Ocular como Región de Interés	23
2.2.5. Fundamentos de Visión Computacional e Imagen Digital	24
2.2.6. Algoritmos de Adaptación de Dominio	26
2.2.7. Arquitecturas de Aprendizaje Profundo	27
2.2.8. Estrategias de Aprendizaje y Optimización	30
2.2.9. Computación en el Borde (<i>Edge AI</i>)	31
2.2.10. Métricas de Validación y Desempeño Clínico	31
2.2.11. Sensores de Imagen y Adquisición Óptica	34
2.2.12. Sistemas Embebidos y Computación en el Borde	36
2.2.13. Herramientas Computacionales para el Desarrollo	39
3. Diseño Metodológico y Lógica del Sistema	41
3.1. Metodología de la Investigación	41
3.1.1. Enfoque y Tipo de Investigación	41
3.1.2. Nivel de Investigación	42
3.1.3. Población y Muestra	42
3.2. Lógica General del Trabajo	43
3.3. Adquisición de Datos	44
3.3.1. Recolección y Descripción de Datos	44
3.4. Preprocesamiento y Acondicionamiento de Datos	46

3.4.1.	División de Subconjuntos (Data Split)	46
3.4.2.	Estrategia de Aumento de Datos (Data Augmentation)	47
3.4.3.	Adaptación de Dominio mediante Transferencia de Color	48
3.5.	Diseño de Redes de Aprendizaje Profundo	51
3.5.1.	Subsistema de Segmentación (U-Net)	51
3.5.2.	Subsistema de Clasificación por Regresión (Hemo-ConViT)	55
3.6.	Lógica de Alerta y Clasificación Clínica	61
3.6.1.	Definición de Umbrales y Corrección por Altitud	61
4.	Implementación del Sistema Prototipo	63
4.1.	Componentes del Sistema	63
4.1.1.	Módulo de Adquisición Óptica	64
4.1.2.	Sistema de Iluminación Controlada	65
4.1.3.	Unidad de Procesamiento (Sistema Embebido)	66
4.1.4.	Interfaz de Usuario y Visualización	67
4.1.5.	Sistema de Alimentación y Autonomía	67
4.2.	Criterios de Implementación en el Sistema Embebido	69
4.2.1.	Estandarización de Parámetros de Captura	69
4.2.2.	Garantía de Coherencia con el Dominio de Entrenamiento	70
4.2.3.	Optimización de Modelos para la NPU	70
4.3.	Lógica de Funcionamiento del Prototipo	71
4.3.1.	Máquina de Estados General	71
4.3.2.	Lógica de Captura y Procesamiento Secuencial	72
4.3.3.	Monitoreo Energético y Gestión de Recursos	74
4.3.4.	Integración de Servicios y Buses de Comunicación	76
4.4.	Validación de Consumo Energético	78
4.4.1.	Justificación del Sistema de Alimentación y Autonomía	80
4.5.	Costos de Implementación	84

5. Experimentos y Resultados	85
5.1. Fase 1: Evaluación Base (In-Silico)	85
5.1.1. Desempeño en Segmentación	85
5.1.2. Desempeño en Estimación de Hemoglobina	89
5.2. Análisis de la Reducción de la Brecha de Dominio	93
5.3. Fase 2: Evaluación con Adaptación de Dominio (Dataset Mixto)	94
5.3.1. Mejora en la Segmentación Local	94
5.3.2. Precisión en la Estimación de Hemoglobina	98
5.4. Fase 3: Comportamiento del Sistema en Entorno Real	101
5.4.1. Interpretabilidad del Modelo de Estimación	102
5.4.2. Evaluación del Desempeño Computacional (NPU)	103
5.4.3. Correlación Clínica: HemoCue vs. Prototipo	103
5.4.4. Validación de Clasificación (Screening de Anemia)	105
6. Discusión	107
6.1. Interpretación de los Resultados del Sistema	107
6.1.1. Desempeño de la Segmentación Automática	107
6.1.2. Estimación de Hemoglobina y Adaptación a la Altura	108
6.2. Comparativa con la Literatura	109
6.3. Discusión sobre el Prototipo Implementado	110
6.3.1. Viabilidad Técnica y Desempeño Computacional	110
6.3.2. Capacidad de Tamizaje Clínico	111
6.3.3. Limitaciones de la implementación	112
Conclusiones	113
Recomendaciones	115
Bibliografía	116

Índice de tablas

2.1. <i>Análisis comparativo de las plataformas embebidas evaluadas para procesamiento en el borde.</i>	39
3.1. <i>Hiperparámetros y configuración del entrenamiento para la red de segmentación U-Net.</i>	55
3.2. <i>Hiperparámetros y configuración del entrenamiento para el modelo Hemo-ConViT.</i>	60
4.1. <i>Parámetros de control configurados para el sistema de iluminación y modulación PWM.</i>	66
4.2. <i>Mapa de conexiones e interfaces físicas del sistema (Pinout).</i>	68
4.3. <i>Parámetros fijos de configuración del sensor óptico para la estandarización de captura.</i>	69
4.4. <i>Parámetros de conversión y cuantización para la NPU RK3588.</i>	71
4.5. <i>Estados operativos y consumo de potencia del sistema.</i>	81
4.6. <i>Costos de manufacturación del dispositivo implementado.</i>	84
5.1. <i>Análisis estadístico del desempeño del modelo de segmentación en el conjunto de prueba (dataset público).</i>	86
5.2. <i>Análisis de incertidumbre del modelo de segmentación mediante bootstrapping (dataset público).</i>	86
5.3. <i>Parámetros de concordancia en el dataset público.</i>	92
5.4. <i>Métricas de rendimiento diagnóstico (dataset público).</i>	93
5.5. <i>Análisis estadístico del desempeño del modelo de segmentación en el conjunto de prueba (dataset mixto, público + local).</i>	95

5.6.	<i>Análisis de incertidumbre del modelo de segmentación mediante bootstrapping (dataset mixto, público + local).</i>	95
5.7.	<i>Parámetros de concordancia (Bland-Altman).</i>	100
5.8.	<i>Métricas de rendimiento diagnóstico del mejor modelo para prototipo.</i>	101
5.9.	<i>Resumen General de Métricas de Rendimiento</i>	103
5.10.	<i>Resumen estadístico del análisis de concordancia.</i>	105
5.11.	<i>Métricas de rendimiento diagnóstico del prototipo.</i>	106
6.1.	<i>Comparativa del desempeño del prototipo implementado frente al estado del arte.</i>	109
6.2.	<i>Datos demográficos y niveles de hemoglobina de los voluntarios locales.</i>	122

Índice de figuras

1.1.	<i>Anemia en niños y niñas de 6 a 59 meses de edad según residencia.</i>	5
2.1.	<i>Dispositivo HemoCue Hb 201+ utilizado como referencia.</i>	22
2.2.	<i>Regiones anatómicas comunes para la detección no invasiva de anemia.</i>	23
2.3.	<i>Anatomía de la conjuntiva ocular: regiones forniceal y palpebral.</i>	24
2.4.	<i>Flujo general de un sistema de visión computacional.</i>	25
2.5.	<i>Esquema de conexiones de salto en arquitecturas tipo U-Net.</i>	29
2.6.	<i>Versiones de Arducam 16MP Autofocus.</i>	35
2.7.	<i>Raspberry Pi High Quality IMX477.</i>	36
2.8.	<i>Raspberry Pi 4 model B.</i>	37
2.9.	<i>Orange Pi 5.</i>	38
3.1.	<i>Diagrama de bloques general del sistema propuesto.</i>	43
3.2.	<i>Protocolo de adquisición de datos en campaña local (Cusco).</i>	46
3.3.	<i>Estrategia de división de datos y aumento sintético (Data Augmentation).</i>	48
3.4.	<i>Diagrama de flujo del algoritmo Masked Reinhard implementado.</i>	50
3.5.	<i>Flujo de diseño del subsistema de segmentación.</i>	52
3.6.	<i>Arquitectura Encoder-Decoder de la red U-Net propuesta.</i>	54
3.7.	<i>Flujo de procesamiento del subsistema de estimación de hemoglobina.</i>	56
3.8.	<i>Arquitectura del modelo Hemo-ConViT basada en Vision Transformer.</i>	58
3.9.	<i>Diagrama de pérdidas contrastiva y regresiva.</i>	59

4.1.	<i>Vista preliminar 3D del ensamblaje de hardware del dispositivo prototipo.</i>	64
4.2.	<i>Módulo de acoplamiento óptico.</i>	65
4.3.	<i>Funcionamiento de la pantalla OLED mostrando el estado del sistema en tiempo real.</i>	67
4.4.	<i>Esquemático de conexiones de periféricos para Orange Pi 5 Max.</i>	68
4.5.	<i>Diagrama general del firmware (Ciclo de vida).</i>	72
4.6.	<i>Diagrama de flujo de la subrutina de iluminación controlada y captura de imagen.</i>	73
4.7.	<i>Flujo lógico de la inferencia de Inteligencia Artificial en la NPU del sistema.</i>	74
4.8.	<i>Lógica general del sistema de gestión de energía y estados de batería.</i>	75
4.9.	<i>Detalle del algoritmo de decisión para las alertas de monitoreo energético.</i>	76
4.10.	<i>Verificación de estado de pines GPIO en consola del Orange Pi 5 Max.</i>	77
4.11.	<i>Detección del dispositivo OLED en el bus I2C (dirección 0x3C).</i>	77
4.12.	<i>Configuración del servicio systemd para el lanzamiento automático de la aplicación.</i>	78
4.13.	<i>Medición del consumo de corriente en estado de reposo (Idle).</i>	79
4.14.	<i>Medición del consumo de corriente durante la inferencia de modelos IA.</i>	79
4.15.	<i>Medición del consumo del sistema de iluminación activado.</i>	80
5.1.	<i>Evolución de métricas de entrenamiento y validación (Segmentación).</i>	87
5.2.	<i>Matriz de confusión normalizada en el conjunto de prueba.</i>	87
5.3.	<i>Distribución de probabilidad de las métricas de segmentación.</i>	88
5.4.	<i>Resultados cualitativos de segmentación automática.</i>	89
5.5.	<i>Evolución de las métricas Loss y MAE en el dataset público (Eye-Defy-Anemia).</i>	90
5.6.	<i>Análisis de regresión lineal en el dataset público: Referencia vs. Hemo-ConViT.</i>	91
5.7.	<i>Análisis de Bland-Altman para el dataset público.</i>	92
5.8.	<i>Matriz de confusión obtenida en el dataset público.</i>	93
5.9.	<i>Análisis de densidad de probabilidad (KDE) de los canales de color.</i>	94

5.10. Evolución de métricas de entrenamiento y validación (Dataset Mixto).	96
5.11. Matriz de confusión normalizada en el conjunto de prueba mixto.	96
5.12. Distribución de probabilidad de las métricas de segmentación.	97
5.13. Resultados cualitativos de segmentación en el dataset mixto.	98
5.14. Evolución de las métricas Loss y MAE en el conjunto de datos mixto.	99
5.15. Análisis de regresión lineal en el dataset mixto: Referencia vs. Hemo-ConViT.	99
5.16. Análisis de Bland-Altman para el conjunto de datos mixto.	100
5.17. Matriz de confusión del modelo en el conjunto de datos mixto.	101
5.18. Visualización de la atención del modelo (Attention Rollout) sobre la región de interés.	102
5.19. Análisis de regresión lineal en el dataset local: Referencia vs. Prototipo.	104
5.20. Gráfico de Bland-Altman para la evaluación de concordancia (en dataset local).	105
5.21. Matriz de confusión del diagnóstico de anemia en el dataset local.	106
6.1. Vistas del dispositivo implementado.	111

Introducción

La presente investigación se centra en el desarrollo de una solución tecnológica orientada al cribado no invasivo de la anemia, una condición de salud pública que afecta el desarrollo integral de poblaciones vulnerables. El trabajo propone el diseño e implementación de un sistema electrónico portátil que, mediante el uso de inteligencia artificial y procesamiento digital de imágenes, analiza las características visuales de la conjuntiva ocular para estimar niveles de hemoglobina. A través de la ejecución de algoritmos de aprendizaje profundo en una plataforma embebida y el uso de técnicas de adaptación de dominio, se busca ofrecer una herramienta de tamizaje de bajo costo, capaz de operar en zonas alejadas donde el acceso a laboratorios convencionales es limitado. La tesis se estructura en seis capítulos que detallan el proceso de desarrollo, desde la concepción teórica hasta la validación experimental.

El **Capítulo 1** expone la realidad problemática de la anemia en el contexto nacional y regional, evidenciando la necesidad de contar con métodos de detección accesibles. Se establecen los objetivos generales y específicos que guían la investigación, así como la justificación técnica, social y económica del estudio, delimitando los alcances y las restricciones operativas de la propuesta tecnológica.

En el **Capítulo 2** se desarrolla el marco teórico fundamental, abarcando desde la fisiopatología de la anemia y las características anatómicas de la conjuntiva, hasta los principios de visión computacional. Se revisan los antecedentes del estado del arte, las arquitecturas de redes neuronales —tanto convolucionales como basadas en *Transformers*— y las especificaciones téc-

nicas de los sistemas embebidos y sensores ópticos necesarios para el procesamiento en el borde.

El **Capítulo 3** detalla el diseño metodológico y la arquitectura lógica del sistema. Se describe el proceso de adquisición de datos, las técnicas de preprocesamiento y la estrategia de adaptación de dominio para mitigar la variabilidad entre fuentes de imágenes. Asimismo, se profundiza en el diseño y configuración de los modelos de aprendizaje profundo: la red U-Net para la segmentación semántica y la arquitectura propuesta *Hemo-ConViT* para la estimación de hemoglobina.

El **Capítulo 4** aborda la implementación física y lógica del prototipo final. Se describen la integración de los componentes de hardware, del sistema de iluminación controlada y la optimización de los modelos para su ejecución en la Unidad de Procesamiento Neuronal (NPU) RK3588. Además, se detalla la lógica del *firmware* que gestiona la captura, el procesamiento secuencial y la gestión energética del dispositivo.

En el **Capítulo 5** se presentan los experimentos y los resultados cuantitativos obtenidos. Se evalúa el desempeño de los modelos tanto en etapas de validación *in-silico* como en pruebas de campo con voluntarios locales. Se analizan las métricas de precisión, la efectividad de la transferencia de color y la concordancia estadística de las mediciones frente a métodos de referencia clínica.

Finalmente, el **Capítulo 6** ofrece una discusión integral de los hallazgos, interpretando la relevancia clínica de los resultados obtenidos en una población de altura. Se contrasta el desempeño del prototipo implementado frente a la literatura existente, analizando la viabilidad técnica del sistema como herramienta de tamizaje y reflexionando sobre las limitaciones identificadas durante el estudio.

Capítulo 1

Aspectos generales

1.1. Descripción de la realidad problemática

La anemia es una enfermedad caracterizada por una disminución en el número de glóbulos rojos o en la concentración de hemoglobina dentro de estos glóbulos, situándose por debajo de los niveles considerados normales. Esta condición afecta principalmente a mujeres, así como a niños y niñas. La anemia se origina cuando el organismo no cuenta con la cantidad adecuada de hemoglobina para transportar oxígeno de manera eficiente hacia los órganos y tejidos corporales. Existen diversas causas que pueden dar lugar a la anemia, que incluyen una alimentación deficiente, infecciones, enfermedades crónicas, complicaciones durante el embarazo y antecedentes familiares. La carencia de hierro en la sangre suele ser la causa subyacente más frecuente. En muchos entornos con ingresos bajos y medianos bajos, las causas más comunes de anemia son la deficiencia de hierro y la malaria (Organización Mundial de la Salud, 2023).

Según informes de la Organización Mundial de la Salud, la anemia afecta de manera significativa a ciertos grupos de población, como los menores de 5 años, especialmente los lactantes y niños menores de 2 años. A nivel mundial, se estima que aproximadamente 269 millones de niños y niñas de 6 a 59 meses sufren de anemia. En el año 2019, aproximadamente el 30 % de las mujeres en edad fértil (539 millones) y el 37 % de las mujeres embarazadas (32 millones) entre 15 y 49 años padecían anemia (Organización Mundial de la Salud, 2023).

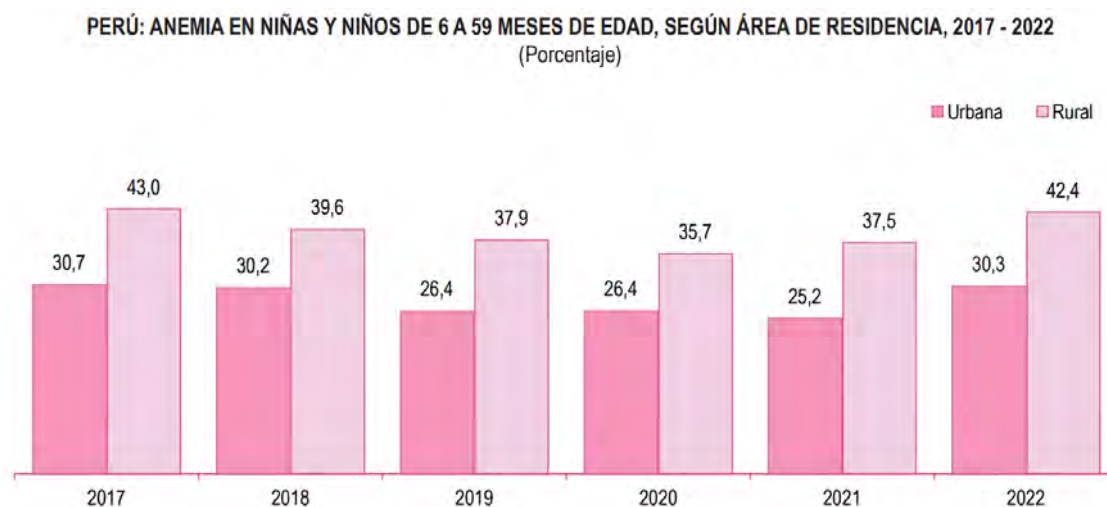
Según la Encuesta Demográfica y de Salud Familiar ENDES 2022, en Perú, en el año 2022, el 33,6 % de los niños y niñas de entre 6 y 59 meses de edad presentaron anemia. En relación con el año anterior, se observó un aumento del 5,2 % en la prevalencia de anemia en este grupo de edad. Además, la anemia afectó de manera más pronunciada a los niños y niñas que residían en áreas rurales, con una prevalencia del 42,4 % (Instituto Nacional de Estadística e Informática, 2022).

En Cusco, según el informe ENDES 2022, el índice de prevalencia de anemia en niños de entre 6 y 59 meses de edad se encuentra dentro del margen del 50 y 59,1 %, no habiendo cambios significativos en el rango determinado entre los años 2017 y 2022 (Instituto Nacional de Estadística e Informática, 2022). Debido a la lejanía y las complicaciones geográficas características de la región, los centros poblados y sitios alejados como el Valle de los Ríos Apúrame, Ene y Mantero (VRAEM) poseen índices elevados de destrucción y anemia debido a la débil presencia estatal que agrava la visibilización de estos problemas, con índices de hasta 24,8 % de anemia en menores de entre 6 a 35 meses, mayor al promedio nacional de 21,8 %, durante el año 2023, según estudio del Instituto Nacional de Salud en base a datos de los pacientes que acuden a centros de salud (Montaño, 2024).

Para la obtención de datos de la Encuesta Demográfica y de Salud Familiar se tomó en cuenta el método recomendado por el Ministerio de Salud, consistente en la toma de una muestra de sangre capilar para la detección de nivel de hemoglobina por medio de un sistema de detección fotométrica de hemoglobina con el instrumento HemoCue, técnica ampliamente empleada para la detección de anemia en diversos países (Ministerio de Salud del Perú, 2024). En el caso del Perú, esta técnica ha sido seleccionada como el método principal para la medición de hemoglobina. Sin embargo, según el informe ENDES 2022, *"determinar anemia en mujeres, niñas y niños constituye uno de los mayores esfuerzos que la Encuesta Demográfica y de Salud Familiar viene realizando desde 1996"* (Instituto Nacional de Estadística e Informática, 2022).

Figura 1.1

Anemia en niños y niñas de 6 a 59 meses de edad según residencia.



Nota: Extraído de Instituto Nacional de Estadística e Informática (2022).

A pesar de la fiabilidad clínica del método fotométrico invasivo, su naturaleza representa una barrera significativa, especialmente en poblaciones infantiles. La extracción de sangre capilar genera dolor y estrés en los niños, lo que a menudo provoca rechazo y renuencia por parte de los padres a someter a sus hijos a tamizajes de rutina en entornos no clínicos (Parker et al., 2018). A esto se suman las severas limitaciones logísticas y económicas: el método requiere de personal de salud capacitado, un manejo riguroso de residuos biocontaminados y un suministro constante de insumos fungibles, como lancetas y microcubetas reactivas.

Frente a estas barreras, la evaluación de la palidez en la conjuntiva palpebral ha surgido históricamente como un biomarcador clínico para la detección de anemia. Recientemente, el avance exponencial de la Inteligencia Artificial (IA), de manera específica mediante técnicas de Visión por Computadora y Aprendizaje Profundo (*Deep Learning*), ha demostrado que es posible automatizar el análisis de imágenes digitales de la conjuntiva. Diversos estudios recientes acreditan que modelos basados en aprendizaje profundo logran extraer características espaciales de la microvasculatura ocular para clasificar la presencia de anemia con niveles de precisión, sensibilidad y especificidad superiores al 85 %, operando exitosamente a través de un enfoque no invasivo utilizando hardware de cámaras de dispositivos móviles (Sehar et al.,

2024)

De allí que el desarrollo de un prototipo funcional de detección no invasiva de anemia, de costo mínimo por prueba, basado en el análisis de imágenes de la conjuntiva palpebral y forniceal mediante técnicas de procesamiento de imágenes y técnicas de aprendizaje profundo representa un enfoque innovador y prometedor para abordar un problema de salud pública significativo. Este proyecto tiene el potencial de mejorar la accesibilidad y la eficacia de la detección preliminar de anemia, lo que posibilita un impacto positivo en la salud y el bienestar de las poblaciones vulnerables que no pueden acceder a pruebas especializadas de laboratorio.

1.2. Formulación del problema

1.2.1. Problema general

¿De qué manera el diseño e implementación de un sistema electrónico basado en aprendizaje profundo permite la detección preliminar no invasiva de anemia?

1.2.2. Problemas específicos

1. ¿Cuál es el método óptimo para la adquisición no invasiva de imágenes de la conjuntiva ocular inferior y qué disponibilidad de conjuntos de datos existe para el entrenamiento efectivo de los modelos computacionales?
2. ¿Cómo desarrollar una arquitectura de aprendizaje profundo que integre eficientemente la segmentación de la región de interés (ROI) y la posterior clasificación automatizada de la conjuntiva para determinar la presencia de anemia?
3. ¿Qué métricas de evaluación son necesarias para garantizar la viabilidad técnica, sensibilidad y especificidad del modelo de clasificación antes de su despliegue físico?

4. ¿De qué manera se pueden integrar los algoritmos de procesamiento visual y la red neuronal en una plataforma de sistema embebido que asegure el bajo consumo energético y la operatividad del dispositivo sin dependencia de conexión a internet?
5. ¿Cuál es el nivel de fiabilidad y validez diagnóstica del sistema electrónico implementado al ser sometido a pruebas de validación en entorno controlado, y cómo se comparan sus resultados frente a los métodos clínicos invasivos de referencia?

1.3. Justificación

1.3.1. Justificación Social

La detección de anemia es fundamental para garantizar el correcto desarrollo cognitivo y motor de las personas en el marco del tiempo. Con la implementación del sistema propuesto, se contribuirá al avance tecnológico regional de herramientas de apoyo de diagnóstico con un prototipo que permita en instancias preliminares la detección de anemia. Además, por las cualidades de trabajo en entornos sin conexión a internet y detección *in situ* del prototipo a implementar, se busca aportar al desarrollo de herramientas replicables que aseguren su funcionamiento incluso en entornos remotos y con acceso limitado a laboratorios de diagnóstico especializados.

1.3.2. Justificación Económica

La implementación del prototipo del sistema podrá concurrir en una reducción de gastos de diagnóstico y pruebas de laboratorio bajo sospecha de anemia. Además, una detección correcta de anemia en poblaciones vulnerables contribuirá a evitar problemas futuros de salud que interrumpan el correcto desarrollo del menor, que se reflejan en gastos de cuidados médicos y hospitalarios posteriores.

1.3.3. Justificación Tecnológica

El diseño y desarrollo de un prototipo de un sistema de estimación de hemoglobina para la detección de anemia representaría una contribución significativa al campo del diagnóstico asistido y de la implementación de tecnologías de inteligencia artificial a la salud. Además, el uso de nuevas tecnologías de aprendizaje profundo destinado a sistemas embebidos y de bajo consumo supone un nuevo campo de investigación con múltiples posibilidades de aplicación.

1.3.4. Justificación Académica

La presente investigación se justifica desde el ámbito académico en tanto se pretende aplicar los conocimientos adquiridos durante la estancia en la Escuela Profesional de Ingeniería Electrónica, como son los que se imparten en las asignaturas de Inteligencia Artificial, Procesamiento Digital de Imágenes, Fundamentos de Bioingeniería, Sistemas Embebidos y Sistemas Electrónicos, que pertenecen a la malla curricular.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar e implementar un sistema electrónico que permita la estimación de hemoglobina para la detección *in situ* de anemia de manera no invasiva usando técnicas de procesamiento de imágenes y aprendizaje profundo.

1.4.2. Objetivos Específicos

1. Establecer el método de adquisición de imágenes no invasivas de la conjuntiva ocular humana y recopilar conjuntos de datos públicos y propios disponibles para el entrenamiento

de los algoritmos de aprendizaje profundo.

2. Diseñar y desarrollar modelos de aprendizaje profundo para la segmentación de la región de interés y la clasificación por regresión de niveles de hemoglobina, empleando estrategias de procesamiento digital de imágenes y arquitecturas de redes neuronales.
3. Evaluar el desempeño de los modelos computacionales desarrollados aplicando criterios métricos de segmentación y clasificación para asegurar su viabilidad técnica.
4. Implementar los modelos de procesamiento y redes neuronales en una plataforma de sistema embebido, integrando el hardware y software necesarios.
5. Validar el funcionamiento integral del sistema implementado mediante pruebas controladas de campo, determinando las métricas de desempeño resultantes en la detección de anemia en comparación con métodos de referencia.

1.5. Alcances

- La implementación del sistema de estimación de hemoglobina de manera no invasiva utilizará la detección de patrones con redes neuronales en imágenes de la conjuntiva palpebral y forniceal inferior.
- Mediante aprendizaje profundo y métodos de segmentación automática de imágenes, se entrenará una red neuronal convolucional capaz de extraer y clasificar las características de las imágenes del área que abarca la conjuntiva palpebral y forniceal inferior del ojo humano.
- Con la finalidad de realizar el procesamiento del sistema de forma local en un prototipo portátil en tiempo real, se precisa del uso de un sistema embebido con características de software y hardware capaces de soportar algoritmos de visión computacional y aprendizaje profundo.

1.6. Limitaciones

- El sistema es una herramienta de *screening* (cribado) y no sustituye al diagnóstico clínico de laboratorio. Los casos detectados como positivos requieren confirmación mediante hemograma.
- La precisión puede verse afectada en pacientes con condiciones oculares patológicas (conjuntivitis, ictericia) que alteren la coloración de la región de interés.
- El dataset local para calibración es limitado (25-50 sujetos), lo cual es suficiente para una prueba de concepto y ajuste de dominio, pero se requeriría una muestra mayor para una certificación médica comercial.

Capítulo 2

Marco teórico

2.1. Antecedentes

2.1.1. Tesis Internacionales

- Mannino, Robert G. (2018). **A NONINVASIVE, IMAGE-BASED SMARTPHONE APP FOR DIAGNOSING ANEMIA**. [Tesis de grado para optar el título profesional de Doctor en Filosofía en Ingeniería Biomédica, Georgia Institute of Technology and Emory University].

El autor propone sistema basado en teléfonos inteligentes, que requiera únicamente la tecnología nativa del teléfono, con ese fin se enfoca en el diseño de una aplicación de teléfono móvil que permita la detección de anemia sintomática mediante imágenes. Los enfoques de diagnóstico existentes para medir los niveles de hemoglobina (Hb) requieren equipos especializados y representan compensaciones entre la invasividad, la precisión, las necesidades de infraestructura y el costo. Además de ser prohibitivamente caros para ciertos sectores de población económica, el necesario muestreo invasivo de sangre para medir los niveles de hemoglobina causa incomodidad y trauma en pacientes pediátricos más jóvenes. Al examinar la palidez clínica, un síntoma común de la anemia, se desarrolla una metodología que analiza cuantitativamente las fotos proporcionadas por los pacientes utilizando algoritmos en teléfonos

inteligentes para permitir un diagnóstico de anemia no invasivo, preciso y accesible. En este enfoque se usan las imágenes de uñas tomadas con un teléfono inteligente, y el algoritmo de análisis de imágenes analiza los datos de color y metadatos de la imagen para medir el nivel de hemoglobina correspondiente. Al cuantificar la palidez clínica, este sistema mide de manera no invasiva los niveles de hemoglobina dentro de un margen de error clínicamente significativo y bien aceptado (± 2.6 g/dL) de la herramienta de medición de hemoglobina estándar con una sensibilidad y especificidad del 0.90 y 0.82, respectivamente, para predecir la anemia (definida como Hb menor a 11.0g/dL) en 100 pacientes pediátricos en Children's Healthcare of Atlanta con anemia de cualquier etiología mezclados con sujetos sanos. En la tesis en discusión se desarrolla un algoritmo de análisis de imágenes que para teléfonos inteligentes que es capaz de superar a los hematologistas entrenados en la medición de hemoglobina basada en exámenes físicos que permita cambiar, según el autor, el paradigma de los métodos de diagnóstico de anemia (Mannino, 2018).

- Nakib Rahman N. and Abdullah Sadik S. and Biswas A. & Bin Tazul R. (2020). **Point of Care Detection of Anemia in Non-Invasive Manner by Using Image Processing and Convolutional Neural Network with Mobile Devices**. [Tesis para obtener el grado de Bachiller en Ciencias de la Computación, BRAC University].

Tradicionalmente, la detección de anemia se realiza mediante pruebas de sangre, lo que puede ser costoso y difícil de realizar en áreas rurales de países en desarrollo donde la infraestructura médica es limitada. Esta tesis propone una solución de punto de atención no invasiva utilizando imágenes capturadas de videos de teléfonos móviles. La metodología consistió en la recolección de datos, el procesamiento, y clasificación mediante procesamiento de imágenes con una red neuronal convolucional diseñada para examinar la tonalidad de la sangre, dado que, por lo general, la sangre con anemia posee una mayor capacidad de transmitir luz en comparación con la sangre en condiciones normales. Utilizando la linterna de un teléfono celular, se logra capturar el color de la sangre en la punta de los dedos de los pacientes sin necesidad de extracción de sangre. La obtención de la base de datos se realizó manualmente

con 85 personas de entre 12 y 68 años, con una proporción buscada de 50-50 entre la relación de hombres y mujeres. La mitad de las muestras fueron de pacientes anémicos. En el análisis de datos se dividieron los vídeos obtenidos hasta obtener alrededor de 300 imágenes por paciente, luego se procesaron los datos obtenidos utilizando OpenCV. Para el diseño del modelo se hizo uso del entorno Keras en TensorFlow, dividiendo los datos en dos grupos: 80 % para datos de entrenamiento y 20 % para datos de prueba. Se documenta que el estudio logra una precisión del 94.55 % y una sensibilidad de 92.88 % en la detección de anemia utilizando vídeos cortos de diez segundos de la punta del dedo, sin embargo, se recomienda realizar más investigaciones para mejorar y validar el método propuesto. También se sugiere explorar el uso de sensores adecuados para estudiar nuevas posibilidades de detección de anemia. Este estudio representa un paso adelante en la detección no invasiva de anemia y tiene el potencial de mejorar la atención médica en áreas donde los recursos son limitados (Rahman et al., 2020).

2.1.2. Artículos científicos relacionados

- Dimauro Giovanni, Baldari Luigi, Caivano Danilo, Colucci G, Girardi Francesco, (2018) **Automatic Segmentation of Relevant Sections of the Conjunctiva for Non-Invasive Anemia Detection** (Dimauro et al., 2018).

Resumen- Las aplicaciones de detección de anemia no invasivas requieren una segmentación manual o un procedimiento de recorte manual de la región de interés como paso preliminar, por ello el artículo busca identificar un procedimiento para la segmentación automática y optimización de secciones de la conjuntiva para este fin. Se aplicaron algoritmos de análisis de imágenes para optimizar el área de interés en relación con el valor estimado de hemoglobina (Hb). Se evaluó la influencia del brillo de la imagen en la estimación correcta de Hb. Se realizaron dos fases: En la primera fase, se segmentó automáticamente la conjuntiva en cada imagen en el espacio de color CIELAB y se calcularon los valores de a^* . Luego, se correlacionaron con los valores de Hb, obteniendo una correlación de Pearson de 0.63 para el promedio de a^* y 0.71 para el promedio ponderado de a^* . En la segunda fase, se estudió el efecto de filtrar

píxeles según la cantidad de áreas claras y oscuras en las imágenes. Se observó que eliminar áreas oscuras mejoró significativamente la correlación con los valores de Hb, obteniendo una correlación de 0.735 al eliminar el 26 % de los píxeles más oscuros. Se concluye que la luminosidad de cada píxel debe considerarse, y se sugiere calcular un umbral global para la selección de píxeles. Además, se propone desarrollar métodos para descartar automáticamente imágenes inadecuadas para el análisis (Dimauro et al., 2018).

- Kasiviswanathan Sivachandar, Vijayan Thulasi Bai, Simone Lorenzo, Dimauro Giovanni (2020) **Semantic Segmentation of Conjunctiva Region for Non-Invasive Anemia Detection Applications** (Kasiviswanathan et al., 2020).

Resumen- El objetivo en este artículo es realizar la segmentación de la región de la conjuntiva para aplicaciones de detección de anemia no invasiva utilizando aprendizaje profundo, debido a la existencia de correlación entre los parámetros de píxeles de la región de la conjuntiva y el valor de hemoglobina, la capacidad de segmentación de este modelo ayudará en un diagnóstico preciso de la anemia. El Modelo de Segmentación de Conjuntiva Basado en U-Net propuesto (UNBCSM) utiliza una arquitectura U-Net afinada para una segmentación semántica de la conjuntiva a partir de imágenes digitales del ojo capturadas por cámaras de consumo en un entorno no controlado. La verdad fundamental (*ground truth*) para este aprendizaje supervisado se proporcionó como máscaras de Pascal obtenidas mediante la selección manual de píxeles de conjuntiva. Se realizaron aumentos y preprocesamiento de imágenes para aumentar el tamaño de los datos y el rendimiento del modelo. El trabajo propuesto identificó la idoneidad de un modelo U-Net afinado (UNBCSM) para la segmentación de imágenes de conjuntiva con una puntuación de IoU promedio del 84.5 % para el conjunto de validación. Dado que este modelo no descarta ninguna información que pueda estar correlacionada con el nivel de hemoglobina (patrón nervioso, el color de los nervios en la región de la conjuntiva, etc.), se pueden derivar modelos más precisos a partir del trabajo de investigación existente utilizando esto como un paso de preprocesamiento. UNBCSM mostró buenos resultados de segmentación y exhibió un valor comparable de puntuación de Intersección sobre Unión (IoU) entre la verdad fundamental y la máscara segmentada del 96 % y 85.7 % para entrenamiento y validación, respectivamente. Este

trabajo de investigación abre una serie de oportunidades para utilizar un espectro más amplio de las características de la imagen para la detección del nivel de hemoglobina en lugar de solo promedios de color en espacios de color (RGB, CIE Lab, YCrCb, etc.) (Kasiviswanathan et al., 2020).

- Dimauro Giovanni, Griseta Maria Elena, Camporeale Mauro Giuseppe, Clemente Felice, Guarini Attilio, Maglietta Rosalia, (2023). **An intelligent non-invasive system for automated diagnosis of anemia exploiting a novel dataset** (Dimauro et al., 2023).

Resumen- Los autores abordan los desafíos del diseño de un sistema inteligente basado en aprendizaje automático que respalde el diagnóstico automatizado de la anemia. Específicamente, ha sido entrenado en un conjunto de datos que contiene fotos de la conjuntiva ocular de pacientes indios e italianos. El sistema propuesto utiliza un dispositivo de adquisición de datos de bajo costo, lo que lo hace adecuado para un uso generalizado. Luego, se discute el rendimiento de los algoritmos de aprendizaje utilizando dos áreas diferentes de la membrana mucosa del ojo. En particular, el algoritmo RUSBoost, cuando se entrena adecuadamente en imágenes de la conjuntiva palpebral, muestra un buen rendimiento en la clasificación de pacientes anémicos y no anémicos, con resultados consistentes incluso al considerar diferentes etnias. El sistema fue entrenado utilizando la conjuntiva seleccionada de las imágenes del ojo recopiladas de estos conjuntos de datos. De hecho, el alto número de microvasos hace que la conjuntiva sea un candidato idóneo para estimar la palidez. En particular, la conjuntiva palpebral resalta la vascularización del área subyacente en comparación con la conjuntiva fórnix, lo que permite apreciar mejor incluso los tonos más leves del color de la sangre. Se utilizó la conjuntiva palpebral y completa para entrenar el sistema de apoyo a la toma de decisiones propuesto, y los resultados destacaron que la conjuntiva palpebral tuvo el mejor rendimiento en los dos conjuntos de datos. El sistema propuesto utiliza como umbral de anemia la concentración de hemoglobina con valor 10.5 g/dL, y es capaz de tratar con éxito un desequilibrio en el número de ejemplos disponibles de pacientes anémicos en comparación con los controles, mostrando como resultados una precisión del 88 % (sensibilidad = 0,66, especificidad = 0,91) en el conjunto de datos italiano y del 75 % (sensibilidad = 0,79, especificidad = 0,74) en el conjunto de datos

indio (Dimauro et al., 2023).

- Appiahene Peter, Arthur Enoch Justice, Korankye Stephen, Afrifa Stephen, Asare Justice Williams, Donkoh Emmanuel Timmy. (2022). **Detection of anemia using conjunctiva images: A smartphone application approach** (Appiahene et al., 2023).

Resumen- Los autores proponen un enfoque no invasivo para detección de anemia mediante el análisis del palidez en las imágenes de la conjuntiva ocular. Utilizaron técnicas de aprendizaje automático y una combinación de modelos como Redes Neuronales Convolucionales (CNN), Regresión Logística y Algoritmo de Desenfoque Gaussiano para desarrollar un modelo de detección de anemia y otro para detectar la conjuntiva. La metodología empleada comenzó con el uso del algoritmo YOLOv5 para detectar y extraer automáticamente la conjuntiva de imágenes oculares y crear la ROI. Este modelo se entrenó en Google Collab con imágenes expuestas de conjuntiva e imágenes no oculares, posteriormente, se entrenó, validó y probó el modelo con 1844, 264 y 527 imágenes de ojos respectivamente, representando el 70 %, 10 % y 20 % del conjunto de datos total. Luego, se evaluaron los modelos en 153 imágenes de ojos y 48 imágenes no oculares (datos en tiempo real). Los resultados mostraron que el modelo identificó correctamente 150 de las 153 imágenes de prueba con conjuntiva y todas las 48 imágenes sin ojos. Para reducir el ruido en las imágenes, se aplicó un desenfoque gaussiano, y luego se extrajeron los componentes RGB y se calcularon sus valores medios. La mejor detección se logró integrando características de tres componentes en total: a^* , b^* y el valor G derivado de las fotos de componentes RGB, mapeando el RGB al espacio de color CIE Lab* (RGB→CIELab*). Se calculó cuidadosamente los valores medios de las características a^* , b^* y G considerando solo los píxeles de imagen con $\min < L$ y $\min < R = G = B < \max$, garantizando una evaluación precisa del palidez de las fotos. Finalmente, implementada la aplicación también ofrece características únicas, como un módulo de gestión de datos que permite a los usuarios mantener un registro de sus resultados de pruebas y compartírlas con médicos para obtener orientación experta. Se usó procesamiento de imágenes y modelos de aprendizaje automático para desarrollar una aplicación móvil que analiza cuantitativamente el palidez en imágenes de la conjuntiva para predecir la anemia con una precisión del 92.50 %, una sensibilidad del 90 % y una especificidad

del 95 %, con un tiempo de ejecución de 50 segundos (Appiahene et al., 2023).

- Delgado-Rivera, Gerson and Roman-Gonzalez, Avid and Alva-Mantari, Alicia and Saldivar-Espinoza, Bryan and Zimic, Mirko and Barrientos-Porras, Franklin and Salgado-Bohorquez, Mario. (2018). **Method for the Automatic Segmentation of the Palpebral Conjunctiva using Image Processing** (Delgado-Rivera et al., 2018).

Resumen- En el artículo, los autores desarrollan un método para la segmentación automática de la conjuntiva palpebral utilizando una aplicación de Android y técnicas de procesamiento de imágenes. El método logra un éxito del 92.2 % en la segmentación de la conjuntiva palpebral. Se realizaron pruebas con una muestra de 115 fotografías de 25 adultos, evaluando la comparación entre la segmentación manual y automática de la conjuntiva palpebral. La técnica de segmentación descrita en el artículo incluye las siguientes etapas: Localización automática de la conjuntiva palpebral vía algoritmos de Viola-Jones en Matlab; Ecuilibración de la imagen RGB, pasando la imagen al espacio de color HSI, mejorando los componentes de tono y saturación y volviendo al espacio RGB; Aplicación de Máscara de Enfoque No Nitidez; Obtención de máscara del canal Rojo y finalmente un proceso de Convulación entre la imagen original y la máscara de canal rojo. Concluyéndose que el método automático de segmentación de la conjuntiva palpebral es práctico y efectivo en condiciones aceptables de imágenes (Delgado-Rivera et al., 2018).

- Zimic, Mirko. (2018). **Sistema Portátil para Diagnóstico de Anemia Basado en Análisis de la Conjuntiva Ocular Utilizando Smartphone e Inteligencia Artificial** (Zimic, 2018).

Resumen- Se expone para la Semana de Evidencia del MIDIS la presentación del Prof. Mirko Zimic, Jefe del Laboratorio de Bioinformática y Biología Molecular, que presenta los desarrollos del proyecto ganador para la Primera Convocatoria del Concurso de Proyectos de Innovación Social. En esta presentación se detalla la construcción de una aplicación para teléfono móvil que permita la medición de anemia de forma no invasiva, motivado por la dificultad del

tamizaje de hemoglobina de manera rutinaria. Para ello se utilizó un teléfono Huawei Y6 II, incluido su sensor de cámara para la toma de fotografías. El algoritmo de identificación toma como condición de anemia un nivel de concentración de hemoglobina menor o igual a 10 g/dL, además de la identificación de vascularizaciones en la zona de la conjuntiva, debido al índice de pacientes con anemia que presentan vascularidad, siendo de 34.62 %, respecto al 23.33 % de pacientes sin anemia que también presentan vascularidad. Para la clasificación se usa métodos de *Deep Learning*, como *Cross-Validation 5-Fold* usando el índice de eritema conjuntival para predecir el nivel de anemia (leve, moderado, grave) obteniéndose una sensibilidad de 0.7 y una especificidad de 0.75, con desviación estándar de 0.1, demostrando los avances de desarrollo del proyecto. (Zimic, 2018).

- Heredia-Menor Keico, & Mamani-Ticona Wilfredo. (2021). **Comparación de técnicas basadas en visión computacional y machine learning para la detección temprana de anemia a partir del análisis de uñas** (Heredia-Menor y Mamani-Ticona, 2021).

Resumen- En el Perú, la anemia afecta frecuentemente adolescentes, mujeres embarazadas y niños, comprometiendo seriamente su desarrollo integral. De allí que surja la necesidad de este artículo, de comparar técnicas basadas en visión computacional y aprendizaje automático para detectar anemia temprana a través del análisis de las uñas, proporcionando a los médicos una herramienta de apoyo para una detección temprana. Este artículo muestra el comportamiento de distintos extractores de características y clasificadores en tres escenarios, demostrando *in silico* que con la aplicación de técnicas de visión computacional descriptoras como SIFT, SURF y HOG se obtienen resultados inferiores a 95 %, por lo que, para criterio de los autores, no son las idóneas para este tipo de estudio porque las pruebas de laboratorio superan ese porcentaje de confiabilidad. Finalmente, se concluye que utilizando la arquitectura VGG19 como extractor de características en combinación con el clasificador Support Vector Machines (SVM), se lograron altos niveles de precisión, con un *accuracy* de 0,989, *precision* y *recall* de 0,98, y un *F1-score* de 0,98. Esto demuestra la viabilidad de detectar la anemia de manera rápida y confiable sin la necesidad de análisis de sangre tradicionales (Heredia-Menor y Mamani-Ticona, 2021).

- Fuentes-Beingolea, J. H., & Palomino-Quispe, F. (2025). **Adaptive Contrastive Learning in Vision Transformer for Non-invasive Hemoglobin Estimation from Conjunctival Images**. *IEEE Access*.

Resumen- En este artículo los autores presentan “Hemo-ConViT”, un marco de aprendizaje profundo novedoso para la estimación no invasiva de hemoglobina. La investigación aborda las limitaciones de las redes neuronales convolucionales (CNN) tradicionales mediante la implementación de un *Vision Transformer* (ViT) eficiente en datos (DeiT-Small). El aporte principal radica en la propuesta de una función de pérdida híbrida que combina la regresión con una pérdida contrastiva adaptativa, diseñada específicamente para manejar la naturaleza continua de los valores de hemoglobina. El modelo fue entrenado y validado con el dataset público *EYES-DEFY-ANEMIA*, logrando un Error Absoluto Medio (MAE) de 0.870 g/dL y una exactitud de clasificación de anemia del 95.0 %. Además, el estudio incluye un análisis de interpretabilidad mediante mapas de atención (*Attention Rollout*), demostrando que el modelo se enfoca correctamente en la vascularidad de la conjuntiva. Este trabajo sienta las bases algorítmicas para el desarrollo de sistemas de cribado portátiles y de bajo costo (Fuentes-Beingolea y Palomino-Quispe, 2025).

2.2. Bases Teóricas

2.2.1. Fisiopatología de la Anemia y Adaptación a la Altura

La anemia se define como una condición clínica caracterizada por una disminución en el número de glóbulos rojos o en la concentración de hemoglobina, lo cual compromete la capacidad de transporte de oxígeno hacia los tejidos periféricos. Esta deficiencia genera hipoxia tisular, manifestándose en sintomatología diversa como fatiga, debilidad muscular y disnea. Si bien las etiologías son múltiples, abarcando desde deficiencias nutricionales (ferropenia) hasta trastornos hereditarios, su impacto en la salud pública es significativo, afectando preponde-

rantemente a poblaciones vulnerables como infantes y gestantes (Organización Mundial de la Salud, 2023).

Manifestaciones Clínicas y Palidez Cutaneomucosa

Entre los signos físicos más consistentes de la anemia moderada a severa se encuentra la palidez de la piel y las mucosas. Fisiológicamente, esto responde a una redistribución del flujo sanguíneo y a una reducción en la concentración de oxihemoglobina, la cual otorga la coloración rojiza característica a los tejidos vascularizados. La evaluación clínica tradicional prioriza la inspección de regiones anatómicas donde la microvasculatura es superficial y la interferencia de la melanina es mínima, tales como la palma de la mano, el lecho ungueal y, con mayor relevancia para esta investigación, la conjuntiva palpebral (Ochoa Salazar, 2021).

Fisiología de Altura y Poliglobulia Compensatoria

Un factor crítico considerado en el diseño de este sistema es el efecto de la altitud sobre los niveles basales de hemoglobina. En poblaciones que residen a gran altitud, como es el caso de la ciudad del Cusco (3,339 m.s.n.m.), la menor presión parcial de oxígeno atmosférico estimula la producción de eritropoyetina, lo que conlleva a un incremento en la masa de glóbulos rojos, fenómeno conocido como policitemia o poliglobulia fisiológica.

En consecuencia, los umbrales de diagnóstico para anemia a nivel del mar (p.ej., 11 g/dL) resultan insuficientes en zonas de altura, ya que un paciente podría presentar niveles "normales" para el estándar global pero ser funcionalmente anémico en su contexto hipóxico. Por ello, las normativas técnicas nacionales establecen factores de corrección positivos que deben sustraerse al valor observado para estandarizar el diagnóstico (Ministerio de Salud, 2017).

2.2.2. Enfoque de Detección: Tamizaje vs. Diagnóstico

Es fundamental establecer la distinción clínica entre el diagnóstico definitivo y el tamizaje (*screening*). El diagnóstico clínico de anemia se fundamenta en pruebas de laboratorio invasivas de alta precisión, como el hemograma completo, que cuantifican directamente los componentes sanguíneos.

Por el contrario, el sistema propuesto en esta investigación se enmarca en el concepto de **detección preliminar o tamizaje**. Su objetivo no es sustituir las pruebas de laboratorio, sino actuar como una herramienta de triaje masivo, rápida y no invasiva, capaz de identificar con alta sensibilidad a individuos con alta probabilidad de padecer anemia para su posterior derivación y confirmación clínica. Este enfoque es especialmente valioso en entornos de recursos limitados donde el acceso a laboratorios es restringido (Asare et al., 2023).

2.2.3. Métodos de Cuantificación de Hemoglobina

Métodos Estándar Invasivos

Conteo Sanguíneo Completo (CSC): Considerado el estándar de oro (*Gold Standard*), este análisis automatizado realizado en laboratorio cuantifica con exactitud los índices eritrocitarios mediante citometría de flujo. Requiere extracción de sangre venosa y personal especializado (MedlinePlus, 2022).

Sistema HemoCue: Es el método de punto de atención (*Point-of-Care*) predominante en campañas de salud pública (como la ENDES en Perú). Funciona mediante la fotometría de una muestra de sangre capilar obtenida por punción digital, la cual reacciona químicamente en una microcubeta desechable. Aunque portátil, su carácter invasivo genera residuos biocontaminados y dolor, factores limitantes en población pediátrica (HemoCue, 2024).

Figura 2.1

Dispositivo HemoCue Hb 201+ utilizado como referencia.



Nota: Extraído de Organización Mundial de la Salud (2023).

Métodos No Invasivos Basados en Imágenes

La visión computacional aplicada a la salud busca correlacionar características visuales con biomarcadores fisiológicos. A diferencia de la espectrofotometría de oclusión (que requiere hardware dedicado costoso), el análisis de imágenes digitales aprovecha sensores RGB convencionales para cuantificar el grado de eritema o palidez en regiones de interés específicas.

Figura 2.2

Regiones anatómicas comunes para la detección no invasiva de anemia.



(A)



(B)

Nota: Comparativa visual entre pacientes sanos (A) y anémicos (B) en conjuntiva, palma y lecho ungueal. Adaptado de Asare et al. (2023).

2.2.4. Anatomía de la Conjuntiva Ocular como Región de Interés

La conjuntiva ocular es una membrana mucosa transparente y vascularizada que recubre la superficie posterior de los párpados y la cara anterior de la esclerótica. Desde la perspectiva de la ingeniería biomédica y la óptica tisular, la conjuntiva palpebral inferior se considera una "ventana biológica ideal" debido a dos factores:

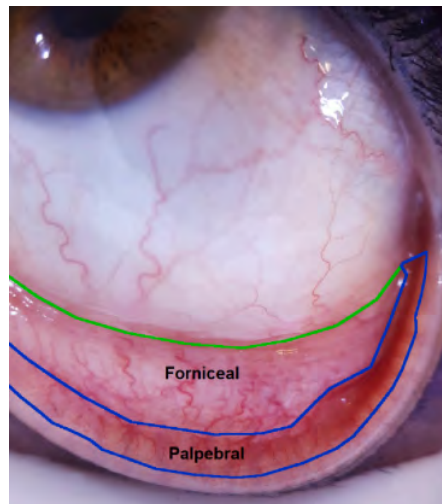
1. **Microvascularización superficial:** La red capilar es densa y se encuentra muy próxima a la superficie, facilitando la absorción de luz por la hemoglobina.
2. **Ausencia de melanina y queratina:** A diferencia de la piel, la conjuntiva carece de pigmentación significativa y de capa córnea gruesa, lo que reduce el ruido espectral y permite una correlación más directa entre el color percibido y la concentración de hemoglobina sanguínea (Brunzini et al., 2022).

Para fines de procesamiento digital, se distingue anatómicamente la **conjuntiva palpebral**

(región rojiza interna del párpado) de la **conjuntiva bulbar** (sobre la esclerótica blanca) y del **fórnix** (pliegue de transición), siendo la primera la que contiene la mayor densidad de información útil para los algoritmos de regresión (Dimauro et al., 2023).

Figura 2.3

Anatomía de la conjuntiva ocular: regiones forniceal y palpebral.



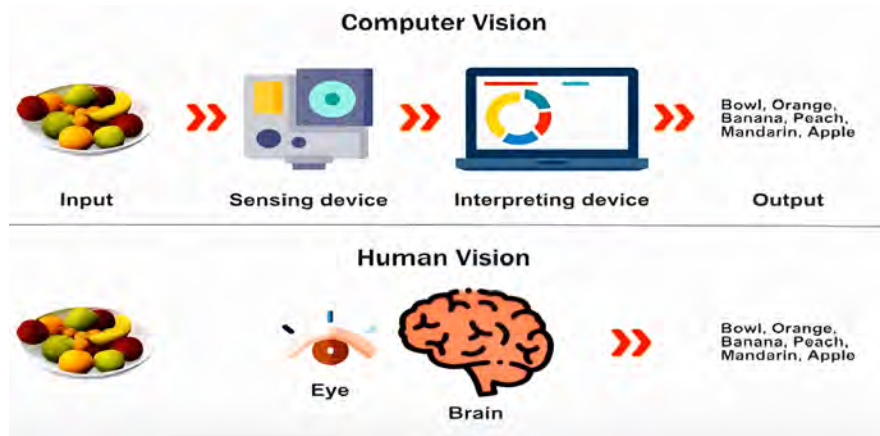
Nota: La segmentación precisa de la región palpebral es crítica para evitar el ruido introducido por la esclera. Extraído de Dimauro et al. (2023).

2.2.5. Fundamentos de Visión Computacional e Imagen Digital

La Visión Computacional es la disciplina que permite a los sistemas informáticos extraer información significativa de entradas visuales. En el contexto de este trabajo, se emplea para automatizar la detección de la región de interés y cuantificar patrones cromáticos sutiles imperceptibles para el ojo humano inexperto (Szeliski, 2022).

Figura 2.4

Flujo general de un sistema de visión computacional.



Nota: El flujo abarca desde la adquisición sensorial hasta la interpretación semántica. Adaptado de Szeliski (2022).

Una imagen digital se define matemáticamente como una función bidimensional de intensidad $f(x, y)$, donde x e y representan coordenadas espaciales y f la amplitud (brillo o color) en dicho punto. Tras los procesos de muestreo y cuantización, esta función se convierte en una matriz discreta I de dimensiones $M \times N$:

$$I(x, y) = \begin{pmatrix} I(0, 0) & I(0, 1) & \cdots & I(0, N - 1) \\ I(1, 0) & I(1, 1) & \cdots & I(1, N - 1) \\ \vdots & \vdots & \ddots & \vdots \\ I(M - 1, 0) & I(M - 1, 1) & \cdots & I(M - 1, N - 1) \end{pmatrix} \quad (2.1)$$

Espacios de Color

La representación numérica del color es determinante para el desempeño de los algoritmos de aprendizaje profundo, especialmente en tareas donde la iluminación es variable.

Espacio RGB: Es el modelo aditivo estándar (Rojo, Verde, Azul) utilizado por los sensores de cámara. Sin embargo, presenta una alta correlación entre sus canales, lo que implica que las variaciones en la intensidad de la luz afectan a los tres componentes simultáneamente,

dificultando la segmentación basada exclusivamente en color bajo iluminación no controlada (Gonzalez & Woods, 2018).

Espacio CIELAB ($L^*a^*b^*$): A diferencia del RGB, el espacio CIELAB está diseñado para aproximarse a la percepción visual humana y, crucialmente, desacopla la información de luminancia de la cromática:

- L^* (**Luminancia**): Representa el brillo de la imagen (0 a 100).
- a^* (**Componente Verde-Rojo**): Fundamental para la detección de anemia, ya que correlaciona directamente con el grado de eritema de la conjuntiva.
- b^* (**Componente Azul-Amarillo**): Representa la variación en el espectro azul-amarillo.

Esta separación es la base teórica que justifica el uso del algoritmo de transferencia de color (Reinhard) en el Capítulo 3, permitiendo normalizar las estadísticas de iluminación (L^*) sin alterar la información biológica contenida en los canales cromáticos (a^* , b^*) (Reinhard et al., 2001).

2.2.6. Algoritmos de Adaptación de Dominio

Dado que la variabilidad en la temperatura de color de la iluminación ambiental afecta drásticamente la percepción de la palidez en la conjuntiva, se requiere una estrategia matemática para estandarizar las imágenes antes de su procesamiento por redes neuronales.

Transferencia de Color Estadística (Método Reinhard)

El método propuesto por Reinhard et al. (2001) constituye una técnica de normalización de color que impone las características estadísticas de una imagen de referencia (destino) sobre una imagen de entrada (fuente). Esta técnica se fundamenta en la transformación del espacio de color RGB al espacio $L\alpha\beta$ de Ruderman, el cual minimiza la correlación entre sus

canales, permitiendo tratar la luminancia (L) y la crominancia (α, β) como variables aleatorias independientes.

Bajo la premisa de que la distribución de color en cada canal puede aproximarse a una distribución normal, el algoritmo alinea las estadísticas de primer orden (media) y segundo orden (desviación estándar) de la imagen fuente con las de la imagen objetivo. Matemáticamente, para cada píxel en la posición (x, y) y para cada canal $k \in \{L, \alpha, \beta\}$, la transformación se define formalmente en la Ecuación 2.2:

$$I_{out}^k(x, y) = \left(\frac{I_{src}^k(x, y) - \mu_{src}^k}{\sigma_{src}^k} \right) \cdot \sigma_{tgt}^k + \mu_{tgt}^k \quad (2.2)$$

Donde los términos estadísticos se definen de la siguiente manera: $I_{src}^k(x, y)$ y $I_{out}^k(x, y)$ representan la intensidad del píxel en el canal k de la imagen original y la procesada, respectivamente. μ_{src}^k y μ_{tgt}^k son las medias de la distribución de color de la imagen fuente y destino. σ_{src}^k y σ_{tgt}^k corresponden a las desviaciones estándar de la imagen fuente y destino.

La operación descrita en 2.2 realiza implícitamente tres pasos: (1) sustracción de la media para centrar los datos en el origen, (2) escalado por el cociente de las desviaciones estándar ($\frac{\sigma_{tgt}}{\sigma_{src}}$) para igualar la varianza, y (3) adición de la media del destino para desplazar la distribución al nuevo punto de operación cromática.

En el contexto específico de esta tesis, se implementa una variante denominada *Masked Reinhard*. A diferencia de la formulación original que calcula μ y σ sobre la totalidad de la imagen ($N \times M$ píxeles), esta adaptación restringe el cálculo estadístico exclusivamente a los píxeles pertenecientes a la región de interés (ROI), ignorando el fondo.

2.2.7. Arquitecturas de Aprendizaje Profundo

El aprendizaje profundo (*Deep Learning*) ha revolucionado el análisis de imágenes médicas al permitir la extracción automática de características jerárquicas. Para este sistema, se

fundamentan dos paradigmas arquitectónicos distintos según la naturaleza de la tarea: local (segmentación) y global (regresión).

Redes Convolucionales y Arquitectura U-Net

Las Redes Neuronales Convolucionales (CNN) son el estándar para el procesamiento de imágenes debido a su invariancia a la traslación y capacidad para capturar patrones locales. Para la tarea de segmentación semántica, la arquitectura **U-Net**, propuesta por Ronneberger et al. (2015), es la referencia en el ámbito biomédico.

Su diseño se caracteriza por una estructura simétrica en forma de U compuesta por dos rutas:

- **Ruta de contracción (Encoder):** Captura el contexto de la imagen mediante capas de convolución y reducción de dimensionalidad (*pooling*).
- **Ruta de expansión (Decoder):** Permite una localización precisa mediante convoluciones transpuestas.

El elemento distintivo de U-Net son las **conexiones de salto** (*skip connections*), las cuales concatenan mapas de características de alta resolución del encoder directamente con el decoder. Esto permite recuperar información espacial fina (bordes de la conjuntiva) que suele perderse en las operaciones de reducción, resultando en máscaras de segmentación de alta precisión incluso con datasets limitados (Kasiviswanathan et al., 2020).

Figura 2.5

Esquema de conexiones de salto en arquitecturas tipo U-Net.



Nota: Extraído de (NVIDIA, 2024).

Vision Transformers (ViT) y Mecanismos de Atención

A diferencia de las CNN, que procesan la información de manera local, los *Vision Transformers* (ViT) introducen el mecanismo de **Auto-atención (Self-Attention)** para modelar dependencias globales a larga distancia dentro de una imagen (Dosovitskiy et al., 2020).

En un ViT, la imagen se divide en una secuencia de parches (x_p) de tamaño fijo, los cuales son proyectados linealmente a vectores de características (z_0) a los que se suma información posicional (E_{pos}), tal como se describe en la Ecuación 2.3:

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \dots; x_p^N E] + E_{pos} \quad (2.3)$$

Para la estimación de hemoglobina, esta capacidad de atención global permite al modelo ponderar y relacionar patrones de vascularización dispersos en toda la conjuntiva, en lugar de depender únicamente de texturas locales, lo que resulta crítico para diferenciar matices sutiles de palidez en casos limítrofes de anemia. Específicamente, se utiliza la variante **DeiT** (*Data-efficient Image Transformer*), la cual integra estrategias de destilación de conocimiento

para optimizar el entrenamiento con volúmenes de datos reducidos, típicos del sector médico (Fuentes-Beingolea & Palomino-Quispe, 2025).

2.2.8. Estrategias de Aprendizaje y Optimización

Aprendizaje Contrastivo Adaptativo

El aprendizaje contrastivo es un paradigma que busca aprender representaciones en las que muestras similares se agrupan en el espacio latente, mientras que muestras disímiles se separen. Tradicionalmente binario, este enfoque ha sido adaptado en esta investigación para tareas de regresión continua. Se define una función de pérdida que pondera la similitud entre imágenes basándose en la distancia euclídea de sus valores de hemoglobina real, forzando al modelo a aprender un espacio de características estructurado y continuo (Fuentes-Beingolea & Palomino-Quispe, 2025).

Funciones de Pérdida (*Loss Functions*)

La optimización de los modelos se guía mediante la minimización de funciones de costo específicas:

- **Dice Loss (Segmentación):** Utilizada en la U-Net para manejar el desbalance de clases entre el fondo (mayoría de píxeles) y la conjuntiva (región pequeña). Maximiza la superposición entre la predicción y la verdad fundamental.
- **Smooth L1 Loss (Regresión):** Empleada en Hemo-ConViT para la estimación de hemoglobina. Es menos sensible a valores atípicos (*outliers*) que el error cuadrático medio (MSE), proporcionando estabilidad numérica durante el entrenamiento.

2.2.9. Computación en el Borde (*Edge AI*)

Para la implementación del prototipo portátil autónomo, se recurre al paradigma de computación en el borde, donde el procesamiento de IA se realiza localmente en el dispositivo en lugar de la nube.

Unidades de Procesamiento Neuronal (NPU)

A diferencia de las CPU (procesamiento serial) o GPU (procesamiento gráfico general), las NPU son aceleradores de hardware diseñados específicamente para ejecutar operaciones tensoriales y matriciales de redes neuronales con alta eficiencia energética. El uso de la NPU (como la integrada en el SoC RK3588) permite inferencias en tiempo real con un consumo de potencia mínimo, vital para dispositivos alimentados por batería.

Cuantización de Modelos

La cuantización es el proceso de reducir la precisión numérica de los pesos y activaciones de la red neuronal, pasando de punto flotante de 32 bits (FP32) a formatos reducidos como FP16 o INT8. Esta técnica reduce significativamente el tamaño del modelo y el ancho de banda de memoria requerido, acelerando la inferencia con una pérdida despreciable de precisión en la tarea de diagnóstico.

2.2.10. Métricas de Validación y Desempeño Clínico

Para evaluar la eficacia del sistema propuesto, se emplean métricas estándar en visión computacional y estadística clínica.

Métricas de Segmentación

- **Intersección sobre Unión (IoU o Índice de Jaccard):** Mide el grado de superposición entre la máscara predicha (A) y la real (B). Es la métrica principal para validar la precisión de la U-Net.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (2.4)$$

- **Coficiente Dice:** Similar al IoU, pero otorga mayor peso a la intersección, siendo estándar en literatura médica.

Métricas de Regresión

- **Error Absoluto Medio (MAE):** Cuantifica la magnitud promedio de los errores en las predicciones de hemoglobina (en g/dL), sin considerar su dirección.
- **Coficiente de Determinación (R^2):** Indica la proporción de la varianza en la hemoglobina real que es predecible a partir de las imágenes.

Métricas de Clasificación (Matriz de Confusión)

Para evaluar la capacidad del sistema como herramienta de tamizaje, los valores continuos de hemoglobina se binarizan según el umbral clínico. A partir de la matriz de confusión (Verdaderos Positivos TP , Falsos Positivos FP , Verdaderos Negativos TN , Falsos Negativos FN), se calculan las siguientes métricas (Kasiviswanathan et al., 2020):

- **Exactitud (Accuracy):** Proporción total de predicciones correctas.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.5)$$

- **Sensibilidad (Recall):** Capacidad del sistema para detectar correctamente a los pacientes

anémicos. Es crítica para evitar falsos negativos en tamizaje.

$$Recall = \frac{TP}{TP + FN} \quad (2.6)$$

- **Precisión (*Precision*):** Proporción de diagnósticos positivos que fueron correctos.

$$Precision = \frac{TP}{TP + FP} \quad (2.7)$$

- **F1-Score:** Media armónica entre precisión y sensibilidad, útil cuando existe desbalance de clases.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.8)$$

Análisis de Concordancia de Bland-Altman

El **Análisis de Bland-Altman** es un método gráfico y estadístico utilizado para evaluar la concordancia entre dos técnicas de medición cuantitativas (en este caso, el prototipo propuesto y el dispositivo de referencia HemoCue). A diferencia de las métricas de correlación, este análisis permite identificar sesgos sistemáticos y cuantificar la variabilidad de las discrepancias entre métodos.

Sea X_i la medición obtenida por el prototipo y Y_i la medición de referencia para el i -ésimo sujeto. Se definen las siguientes variables fundamentales:

$$d_i = X_i - Y_i \quad (2.9)$$

$$m_i = \frac{X_i + Y_i}{2} \quad (2.10)$$

donde d_i representa la diferencia entre métodos y m_i la media de las mediciones. El gráfico de Bland-Altman se construye representando d_i en función de m_i , lo que permite evaluar la

presencia de sesgos dependientes de la magnitud.

A partir de las diferencias d_i , se calculan las siguientes métricas:

- **Sesgo (*Bias*):** Corresponde a la media de las diferencias, indicando la tendencia sistemática entre métodos:

$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i \quad (2.11)$$

- **Desviación estándar de las diferencias:**

$$s_d = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \bar{d})^2} \quad (2.12)$$

- **Límites de Acuerdo (LoA):** Definen el intervalo dentro del cual se espera que se encuentren aproximadamente el 95 % de las diferencias, bajo el supuesto de normalidad:

$$\text{LoA} = \bar{d} \pm 1.96 \cdot s_d \quad (2.13)$$

El factor 1.96 proviene de la distribución normal estándar y corresponde al intervalo de confianza bilateral del 95 %, es decir, $P(-1.96 \leq Z \leq 1.96) \approx 0.95$.

La interpretación clínica de los límites de acuerdo es fundamental: si el intervalo definido por los LoA se encuentra dentro de un margen de error clínicamente aceptable, ambos métodos pueden considerarse intercambiables para fines de tamizaje.

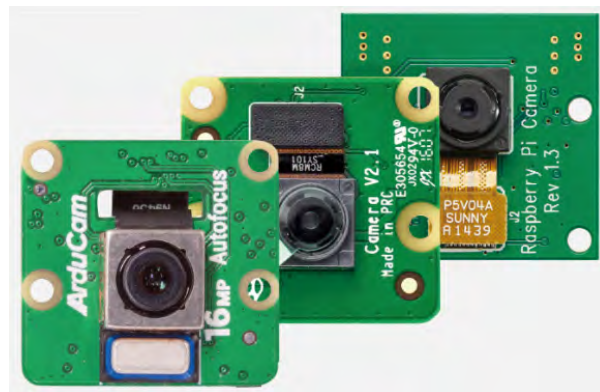
2.2.11. Sensores de Imagen y Adquisición Óptica

Para la obtención de datos en tiempo real, se requiere la integración de un sensor óptico capaz de capturar imágenes de alta resolución de la conjuntiva palpebral inferior. La calidad de la imagen es crítica, ya que la segmentación y el análisis cromático dependen de la nitidez de los capilares y la fidelidad del color. A continuación, se describen las tecnologías evaluadas:

Arducam 16MP Autofocus Camera: Módulo de cámara de la empresa Arducam que integra el sensor Sony IMX519 de 16 megapíxeles. Posee una longitud focal de 4.28 mm y capacidad de enfoque automático. Esta característica es relevante para compensar los micro-movimientos del paciente durante la captura, asegurando que la región de interés se mantenga nítida (Arducam, 2022).

Figura 2.6

Versiones de Arducam 16MP Autofocus.



Fuente: Arducam, 2022.

Raspberry Pi High Quality Camera: Dispositivo de captura de alta fidelidad basado en el sensor Sony IMX477 de 12.3 megapíxeles. Se caracteriza por una diagonal de imagen de 7.9 mm y una arquitectura de sensor retroiluminado, lo que mejora la sensibilidad a la luz y reduce el ruido en la imagen. Además, permite el montaje de lentes intercambiables con enfoque trasero ajustable, lo cual es ideal para configurar una distancia focal fija en entornos controlados de tamizaje (Raspberry Pi, 2023).

Figura 2.7

Raspberry Pi High Quality IMX477.



Fuente: Raspberry Pi, 2023.

2.2.12. Sistemas Embebidos y Computación en el Borde

Un sistema embebido se define como un sistema de computación diseñado para realizar funciones dedicadas en tiempo real, con restricciones de potencia y tamaño, donde la mayoría de sus componentes se integran en una placa base única (Aguirre & Giraldo, 2014).

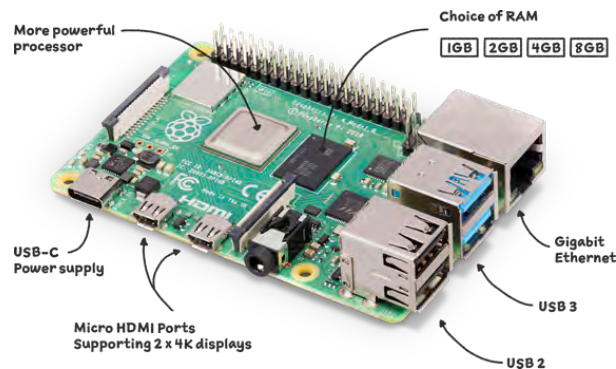
En el contexto de la salud digital, la tendencia hacia la **computación en el borde (Edge AI)** ha convertido a estos sistemas en herramientas imprescindibles. Permiten la ejecución de algoritmos de visión computacional y aprendizaje profundo directamente en el dispositivo, eliminando la dependencia de conexión a internet y garantizando la privacidad de los datos del paciente. La migración de algoritmos a plataformas integradas de alto rendimiento facilita la portabilidad y reduce los costos operativos de los sistemas de seguridad y diagnóstico (Pérez et al., 2018).

Entre las plataformas de desarrollo evaluadas para la implementación del prototipo se encuentran:

Raspberry Pi 4 Model B: Computador de placa única (*Single Board Computer - SBC*) desarrollado por la Fundación Raspberry Pi. Cuenta con un procesador ARM Cortex-A72 y diversas interfaces de entrada/salida (GPIO, CSI, USB). Es ampliamente utilizada por su soporte comunitario y versatilidad en proyectos de electrónica. Sin embargo, carece de una unidad de procesamiento neuronal (NPU) dedicada, lo que limita su velocidad en la inferencia de modelos profundos complejos (Raspberry Pi Foundation, 2020).

Figura 2.8

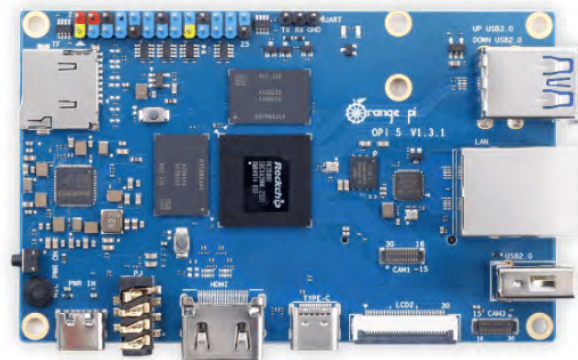
Raspberry Pi 4 model B.



Fuente: Raspberry Pi Foundation, 2020.

Orange Pi 5: Placa de desarrollo de código abierto basada en el procesador Rockchip RK3588S (ARM Cortex-A76/A55). Su característica distintiva es la integración de una **NPU** (*Neural Processing Unit*) con una potencia de cómputo de hasta 6 TOPS (*Tera Operations Per Second*). Esta aceleración por hardware permite la ejecución eficiente de modelos de aprendizaje profundo cuantizados (INT8/FP16), resultando en tiempos de inferencia significativamente menores en comparación con el procesamiento por CPU estándar. Por estas razones, se considera una alternativa idónea para prototipos de tamizaje que requieren respuesta en tiempo real (Shenzhen Xunlong Software, 2024).

Figura 2.9
Orange Pi 5.



Fuente: Shenzhen Xunlong Software, 2024.

Orange Pi 5 Max (Edición 16GB RAM): Representa la evolución de la familia Orange Pi, impulsada por el avanzado SoC Rockchip RK3588. Además de mantener la potente **NPU de 6 TOPS**, esta versión integra de fábrica módulos inalámbricos de última generación (Wi-Fi 6E y Bluetooth 5.3), un puerto de red Ethernet de 2.5 Gbps y un bus de almacenamiento PCIe 3.0 para alta velocidad de lectura (Shenzhen Xunlong Software Co., Ltd, 2024).

El Orange Pi 5 Max resulta superior para proyectos de este tipo debido a que la incorporación de 16 GB de memoria de última generación (LPDDR5) aumenta drásticamente el ancho de banda y la velocidad de acceso, eliminando los cuellos de botella en la memoria caché que se generan al cargar modelos pesados como los *Vision Transformers* (ViT) y redes convolucionales simultáneamente; esto, sumado a su conectividad inalámbrica nativa y su capacidad de procesamiento en paralelo, permite un despliegue verdaderamente autónomo (*Edge IoT*) con tiempos de latencia mínimos, volviéndolo idóneo para el tamizaje clínico en tiempo real.

A fin de discernir objetivamente entre las alternativas, la Tabla 2.1 expone el análisis comparativo de hardware que motivó la selección definitiva de la placa Orange Pi 5 Max.

Tabla 2.1

Análisis comparativo de las plataformas embebidas evaluadas para procesamiento en el borde.

Característica	Raspberry Pi 4	Orange Pi 5	Orange Pi 5 Max
Procesador (SoC)	Broadcom BCM2711	Rockchip RK3588S	Rockchip RK3588
Memoria RAM	Hasta 8 GB LPDDR4	8 GB LPDDR4X	16 GB LPDDR5
Aceleración IA (NPU)	No disponible (0 TOPS)	6 TOPS	6 TOPS
Bus NVMe (M.2)	No nativo	PCIe 2.0	PCIe 3.0
Conectividad de Red	1 GbE, Wi-Fi 5, BT 5.0	1 GbE (Sin Wi-Fi nativo)	2.5 GbE, Wi-Fi 6E, BT 5.3

2.2.13. Herramientas Computacionales para el Desarrollo

El desarrollo del software y el entrenamiento de los modelos se sustentan en un ecosistema de herramientas de código abierto ampliamente validadas en la comunidad científica.

Jupyter Notebook: Aplicación web que permite la creación de documentos interactivos con código ejecutable, visualizaciones y texto narrativo. Es fundamental en la etapa de experimentación y análisis de datos (*in-silico*), facilitando la iteración rápida en el preprocesamiento de imágenes y la evaluación de métricas (Project Jupyter, 2024).

Python: Lenguaje de programación interpretado de alto nivel, consolidado como el estándar en inteligencia artificial debido a su sintaxis clara y su extenso ecosistema de bibliotecas. Su naturaleza multiplataforma y de código abierto permite la integración fluida entre el entrenamiento en servidores y el despliegue en sistemas embebidos (Coutinho Menezes, 2017; Python Software Foundation, 2023).

TensorFlow y PyTorch: Bibliotecas de código abierto para computación numérica y aprendizaje automático. Permiten la definición, entrenamiento y despliegue de redes neuronales profundas (como CNNs y Transformers). Para entornos embebidos, se utilizan versiones optimizadas como **TensorFlow Lite** o conversores a formatos específicos de hardware (como

RKNN para Rockchip), diseñados para ejecutar modelos con recursos limitados de memoria y energía sin sacrificar significativamente la precisión (TensorFlow, 2023).

OpenCV: Biblioteca de visión computacional de código abierto (*Open Source Computer Vision Library*). Proporciona la infraestructura necesaria para el procesamiento de imágenes en tiempo real, incluyendo la lectura de sensores, manipulación de matrices de píxeles, transformaciones de espacio de color y operaciones morfológicas previas a la inferencia neuronal (OpenCV, 2023).

Capítulo 3

Diseño Metodológico y Lógica del Sistema

En el presente capítulo se detalla la arquitectura lógica y los procedimientos metodológicos establecidos para la construcción del sistema de detección no invasiva de anemia. Se describe el flujo de trabajo integral, abarcando desde la adquisición y el acondicionamiento de los datos clínicos y sintéticos, hasta el diseño de las arquitecturas de aprendizaje profundo para la segmentación semántica (U-Net) y la estimación de regresión (Hemo-ConViT).

3.1. Metodología de la Investigación

3.1.1. Enfoque y Tipo de Investigación

La presente investigación se enmarca dentro de un **enfoque cuantitativo**, puesto que se fundamenta en la recolección de datos numéricos y la medición objetiva de parámetros fisiológicos (niveles de hemoglobina) e informáticos (métricas de error como MAE y exactitud) para probar hipótesis mediante análisis estadístico (Hernández-Sampieri et al., 2014).

En cuanto a su tipología, el estudio se clasifica como **investigación de desarrollo tecnológico**. Según los lineamientos del *Manual de Frascati* (OCDE, 2015), el desarrollo tecnológico consiste en trabajos sistemáticos que aprovechan los conocimientos existentes obtenidos de la

investigación empírica para la producción de nuevos materiales, dispositivos o sistemas. Desde la perspectiva epistemológica de Bunge (2014), se trata de una investigación tecnológica, dado que emplea el conocimiento científico de la ingeniería electrónica, la visión computacional y el aprendizaje profundo no solo para explicar la realidad, sino para transformarla mediante el diseño, implementación y validación de un prototipo funcional para la detección de anemia.

3.1.2. Nivel de Investigación

El nivel de investigación es aplicativo, debido a que integra teorías, modelos y tecnologías existentes —como la Inteligencia Artificial, los Sistemas Embebidos y el Procesamiento Digital de Imágenes— en el desarrollo de un sistema de ingeniería orientado a un fin específico.

3.1.3. Población y Muestra

La población objetivo de la presente investigación está conformada por estudiantes de la Universidad Nacional de San Antonio Abad del Cusco, quienes participaron de manera voluntaria en el estudio. La selección de la muestra se realiza mediante un muestreo no probabilístico por conveniencia, considerando como criterios de inclusión la mayoría de edad y la aceptación expresa de participación mediante consentimiento informado.

La muestra se constituye por un total de 32 pacientes voluntarios, entre hombres y mujeres, de los cuales se adquirieron 79 fotografías digitales de la conjuntiva ocular. Estas imágenes representan las unidades de análisis empleadas para el entrenamiento y evaluación de los modelos de procesamiento de imágenes y aprendizaje profundo desarrollados en la investigación.

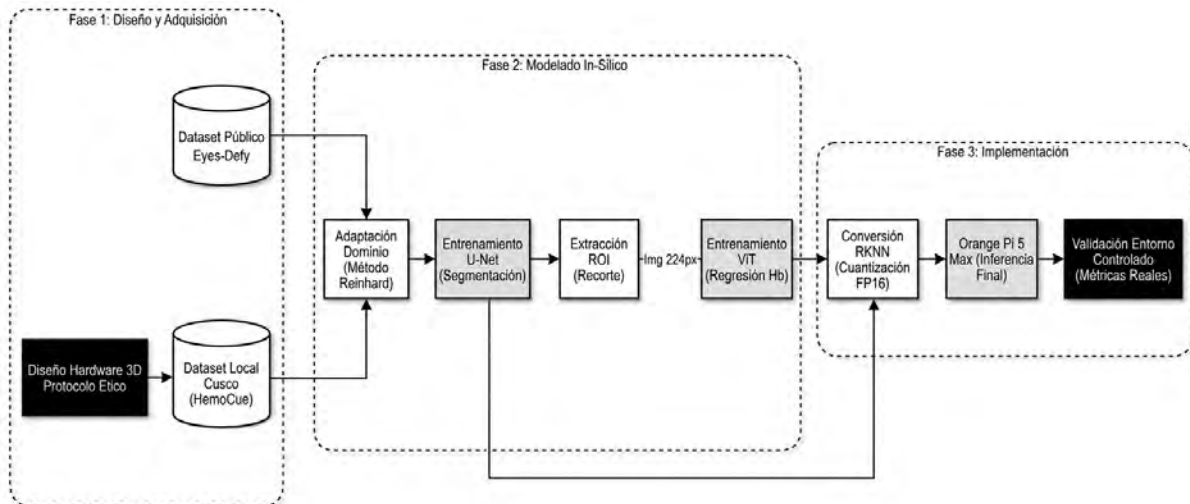
3.2. Lógica General del Trabajo

La metodología propuesta fue estructurada en un flujo secuencial de procesamiento de información diseñado para emular el criterio clínico estándar: la localización precisa de la zona anatómica y el posterior análisis de su coloración.

Como se ilustra en la **Figura 3.1**, el sistema opera bajo un esquema híbrido robusto ante variaciones de iluminación, fundamentado en estrategias de preprocesamiento validadas previamente (Fuentes-Beingolea et al., 2025). En una primera instancia, se aplica una red neuronal convolucional para aislar la región de interés (conjuntiva palpebral), con el fin de eliminar el ruido del entorno (piel, pestañas, esclerótica). Posteriormente, esta región segmentada es procesada por un modelo basado en mecanismos de atención (*Transformer*) que correlaciona las características espectrales con los niveles de hemoglobina. Finalmente, el valor estimado es sometido a un umbral de decisión ajustado por altitud (ver Anexo C) para emitir una alerta de tamizaje.

Figura 3.1

Diagrama de bloques general del sistema propuesto.



Nota: El esquema abarca las etapas principales del proyecto: desde la adquisición de datos y el modelado in-silico, hasta la implementación y validación en el prototipo físico.

3.3. Adquisición de Datos

3.3.1. Recolección y Descripción de Datos

Para el desarrollo y la validación del sistema, fueron utilizados dos conjuntos de datos distintos. El primero, de acceso público, permitió el entrenamiento base (*Pre-training*) de los modelos; mientras que el segundo, recolectado específicamente en la ciudad del Cusco, facilitó el ajuste fino (*Fine-tuning*) a las condiciones fisiológicas de altura y a las características del sensor del prototipo.

Dataset de Referencia: Eyes-Defy-Anemia

Se empleó el repositorio público *Eyes-Defy-Anemia*, el cual consta de 218 imágenes de regiones oculares centradas en la conjuntiva (con 217 imágenes válidas) tomadas en las ciudades costeras de Bari (Italia) y Chennai (India). Este conjunto incluye máscaras de segmentación manual, insumo fundamental para el entrenamiento supervisado de algoritmos de segmentación semántica, tal como se sugiere en estudios previos sobre regiones de interés biomédicas (Dimauro et al., 2023). El proceso de adquisición original de este dataset fue estandarizado mediante un *smartphone* Samsung S6 con iluminación LED controlada. Cada imagen se encuentra vinculada a metadatos clínicos (edad, sexo, nivel de hemoglobina).

Dataset Experimental Local (Cusco)

Con el objetivo de validar el sistema en una población residente a un promedio de 3,339 m.s.n.m. (Wanchaq, Cusco), se construyó un conjunto de datos propio. La recolección de datos fue realizada en una sesión controlada de tres horas, siguiendo estrictamente el protocolo de investigación aprobado por el Comité de Ética Institucional (véase el documento de aprobación en el Anexo A). Los detalles específicos de la demografía y distribución de los datos locales se

presentan en el Anexo B.

- **Protocolo de Adquisición:** Para garantizar la estabilidad de las tomas, se diseñó y fabricó un dispositivo de captura mediante manufactura aditiva (impresión 3D). Este prototipo integra una estructura rígida tipo cilindro para aislar la luz ambiental, una cámara con sensor CMOS 4K (Modelo: MAIRUIGE) y un sistema de iluminación interna mediante LEDs regulados, asegurando condiciones lumínicas constantes.
- **Medición de Referencia:** De manera simultánea a la captura de imagen, se midió la hemoglobina real mediante el método de azida metahemoglobina utilizando un hemoglobímetro portátil (HemoCue® Hb 201+). Este procedimiento fue realizado en estricto cumplimiento con la Norma Técnica de Salud N° 134-MINSA/2017 para el manejo terapéutico y preventivo de la anemia (Ministerio de Salud, 2017).
- **Características de la Población:**
 - **Muestra:** 32 pacientes voluntarios, obteniéndose un total de 79 imágenes válidas tras el filtrado de calidad.
 - **Demografía:** La población presentó una media de edad de 25 años (Rango: 20-30), compuesta por 23 hombres y 9 mujeres.
 - **Hemoglobina:** Se registró una media de 16.65 g/dL ($\sigma = 1.96$), con un rango entre 12.2 y 19.9 g/dL. Estos valores reflejan la poliglobulia compensatoria característica de los habitantes de altura.

El proceso completo, desde la toma de muestra sanguínea hasta la captura digital, se detalla gráficamente en la **Figura 3.2**.

Figura 3.2

Protocolo de adquisición de datos en campaña local (Cusco).



(a) Toma de muestra.

(b) Medición referencial.

(c) Captura de imagen.

Nota: Se muestra la secuencia del procedimiento experimental: (a) punción capilar, (b) lectura con hemoglobinómetro estándar y (c) adquisición digital con el prototipo.

3.4. Preprocesamiento y Acondicionamiento de Datos

Antes de ser ingresados a las redes neuronales, los datos crudos requieren un acondicionamiento riguroso para garantizar la convergencia del entrenamiento y la capacidad de generalización del modelo frente a variaciones intra-clase. El preprocesamiento robusto es crítico para reducir la variabilidad causada por condiciones de iluminación no uniformes (Fuentes-Beingolea et al., 2025).

3.4.1. División de Subconjuntos (Data Split)

Para asegurar una evaluación objetiva y evitar el sesgo de evaluación (conocido como *data leakage*), se unificaron ambos datasets y se procedió a una partición aleatoria estratificada manteniendo la siguiente proporción estándar, comúnmente aceptada en la literatura de aprendizaje profundo (Goodfellow et al., 2016):

- **Entrenamiento (Train - 80 %):** Utilizado para el ajuste de los pesos sinápticos mediante el algoritmo de retropropagación.

- **Validación (Val - 10 %):** Utilizado para el monitoreo de métricas durante el entrenamiento y la optimización de hiperparámetros (ajuste de *Learning Rate* y *Early Stopping*).
- **Prueba (Test - 10 %):** Conjunto totalmente aislado, utilizado únicamente para la evaluación final de las métricas de desempeño del sistema.

Además, para garantizar que el modelo no solo aprenda del dataset público adaptado, sino que generalice correctamente sobre la población objetivo, se diseñó una estrategia de Validación Estratificada Mixta. Del conjunto total de datos locales ($n=79$), se aislaron aleatoriamente 14 muestras para el subconjunto de Validación (*Validation*) y 14 muestras para el subconjunto de Prueba (*Test*). Estas muestras locales se combinaron con sus contrapartes del dataset público para conformar los conjuntos finales, asegurando que el modelo final sea evaluado siempre con una proporción representativa de datos reales del entorno de despliegue.

3.4.2. Estrategia de Aumento de Datos (Data Augmentation)

Dado que el número total de imágenes médicas disponibles es limitado para el entrenamiento de arquitecturas profundas, se implementó una estrategia de aumento de datos sintético aplicada **exclusivamente al subconjunto de entrenamiento**. El objetivo fue aumentar la variabilidad en un factor de $\times 4$, permitiendo que el modelo sea robusto ante pequeñas variaciones de posicionamiento del paciente frente al dispositivo, una técnica validada para evitar el sobreajuste (Shorten & Khoshgoftaar, 2019).

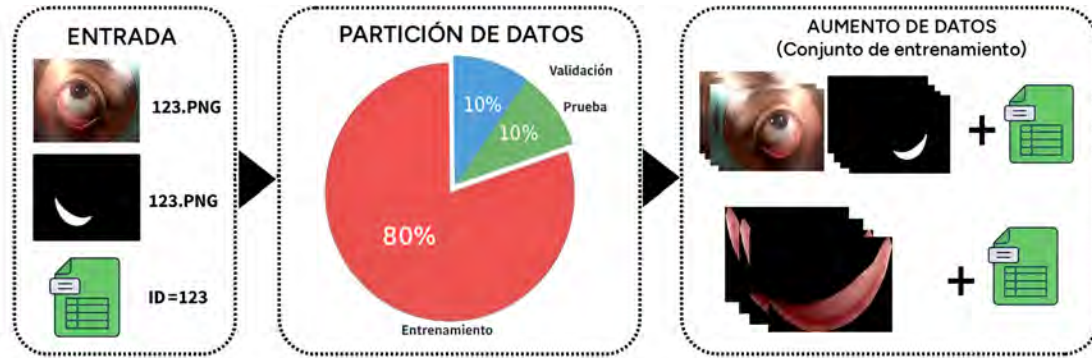
Las transformaciones aleatorias aplicadas, visualizadas en la **Figura 3.3**, incluyeron:

1. **Rotación:** Giros aleatorios en el rango de $\pm 10^\circ$.
2. **Desplazamiento:** Traslaciones horizontales y verticales de hasta un 10 %.
3. **Zoom:** Acercamiento o alejamiento aleatorio del 10 % para simular variaciones en la distancia del ojo a la cámara.

4. **Volteo (Flip):** Espejado horizontal para duplicar la diversidad anatómica (ojo izquierdo vs. derecho).

Figura 3.3

Estrategia de división de datos y aumento sintético (Data Augmentation).



3.4.3. Adaptación de Dominio mediante Transferencia de Color

Dado que el modelo de regresión (Hemo-ConViT) es sensible a las variaciones cromáticas, la discrepancia entre el dominio fuente (imágenes de *smartphones* del dataset *Eyes-Defy-Anemia*) y el dominio objetivo (imágenes capturadas por el prototipo con sensor CMOS 4K en Cusco) representa un obstáculo significativo para la generalización. Para mitigar este efecto, se implementó una variante del algoritmo de transferencia de color propuesto por Reinhard et al. (2001). Esta adaptación fue diseñada específicamente para trabajar sobre las Regiones de Interés (ROI) segmentadas, tal como se detalla en el algoritmo de código incluido en el Anexo D.

El algoritmo opera en el espacio de color $L\alpha\beta$ (CIELAB), elegido por su capacidad para descorrelacionar la luminancia (L) de la información cromática (α y β). A diferencia de la implementación estándar, se diseñó un procedimiento de *enmascaramiento lógico* para excluir los píxeles del fondo (valor 0) durante el cálculo de las estadísticas, evitando que el área negra circundante altere la media y la desviación estándar representativas de la conjuntiva.

El proceso matemático se define de la siguiente manera:

1. **Cálculo de Estadísticas del Dominio Objetivo (Cusco):** Se computan la media (μ_{tgt}) y

la desviación estándar (σ_{tgt}) de los canales L, α, β únicamente sobre los píxeles pertenecientes a la mucosa (M_{roi}) del dataset local:

$$\mu_{tgt}^k = \frac{1}{N} \sum_{x \in M_{roi}} I_{tgt}^k(x), \quad \sigma_{tgt}^k = \sqrt{\frac{1}{N} \sum_{x \in M_{roi}} (I_{tgt}^k(x) - \mu_{tgt}^k)^2} \quad (3.1)$$

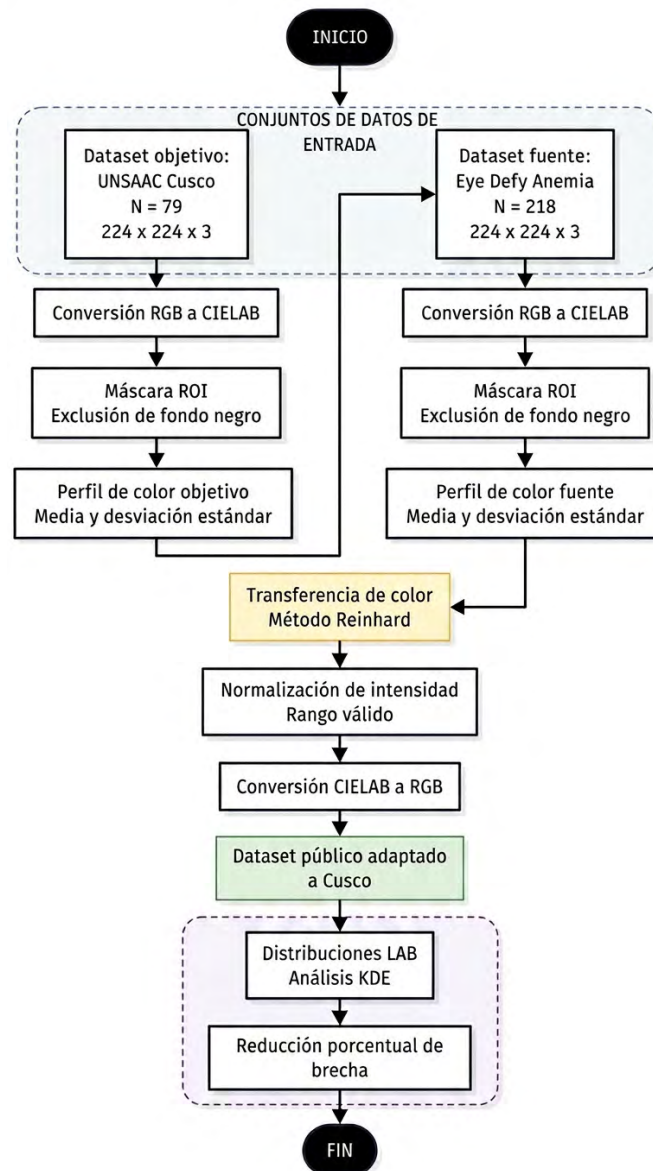
Donde $k \in \{L, \alpha, \beta\}$ y N es el número de píxeles válidos.

2. **Transferencia de Color:** Cada imagen del dataset público (fuente) es transformada píxel a píxel para alinear su distribución estadística con la del prototipo local (ver Ecuación (2.2)).

Esta transformación garantiza que el modelo de aprendizaje profundo sea entrenado con la variabilidad anatómica del dataset masivo, pero bajo las condiciones de iluminación y respuesta espectral específicas del hardware desarrollado en esta investigación. El flujo de este proceso se resume en la **Figura 3.4**.

Figura 3.4

Diagrama de flujo del algoritmo Masked Reinhard implementado.



Al aplicar esta técnica, se transforma la apariencia de la totalidad del dataset público para que mimetice las características visuales (iluminación, contraste y tono) capturadas por el dispositivo en Cusco. Esto permitió generar un modelo robusto y generalizable para los pacientes locales, aprovechando la cantidad de datos del repositorio externo y alineando los dominios para el correcto funcionamiento de Hemo-ConViT (Fuentes-Beingolea & Palomino-Quispe, 2025).

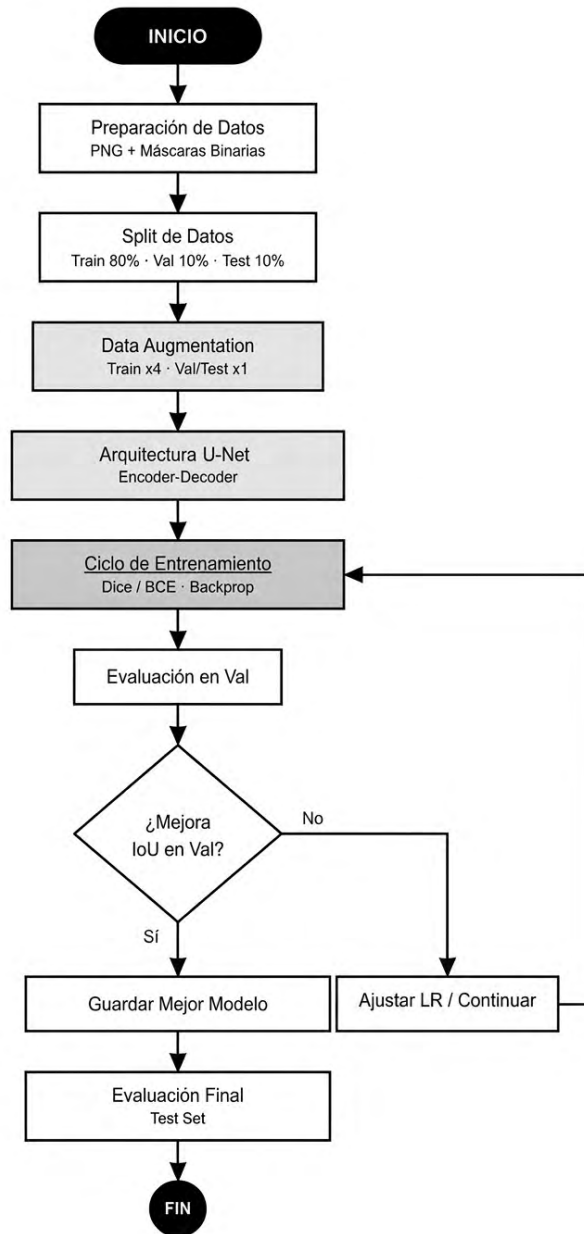
3.5. Diseño de Redes de Aprendizaje Profundo

Para el procesamiento de las imágenes adquiridas y preprocesadas, se diseñó un esquema en cascada compuesto por dos modelos especializados: uno para la segmentación semántica y otro para la estimación de regresión.

3.5.1. Subsistema de Segmentación (U-Net)

Para la extracción automática de la región de la conjuntiva palpebral, se implementó una arquitectura basada en U-Net. La elección de este modelo se fundamenta en su capacidad probada para operar con conjuntos de datos limitados y su precisión en la delimitación de bordes en imágenes médicas, validada en nuestro estudio previo sobre preprocesamiento robusto ante iluminación variable (Fuentes-Beingolea et al., 2025). El flujo metodológico seguido para el diseño y validación de este subsistema se detalla en la **Figura 3.5**.

Figura 3.5
Flujo de diseño del subsistema de segmentación.



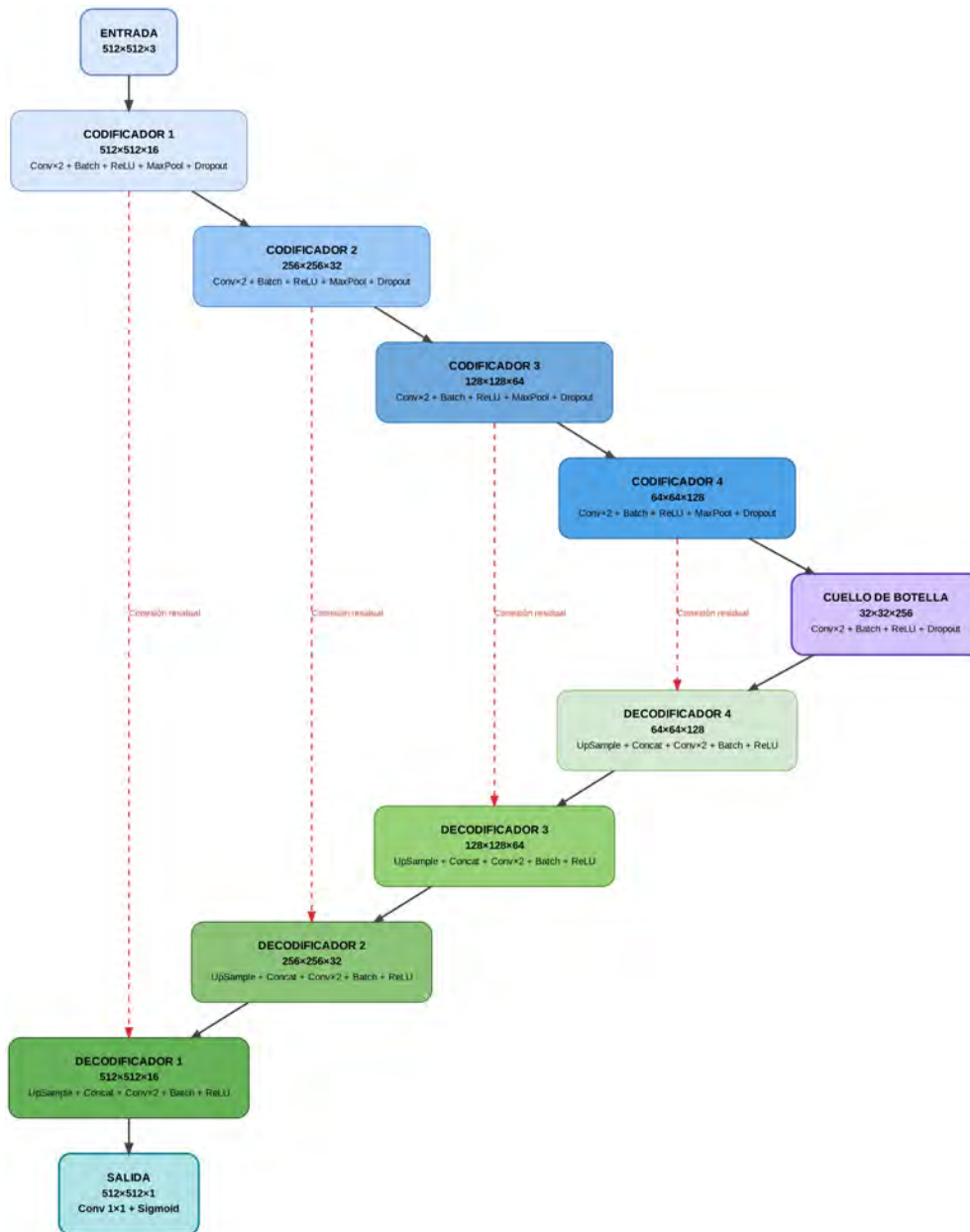
Arquitectura Encoder-Decoder

La red neuronal recibe como entrada imágenes de 512×512 píxeles. La estructura, ilustrada en la **Figura 3.6**, sigue el paradigma clásico de codificador-decodificador (*Encoder-Decoder*) propuesto por Ronneberger et al. (2015).

En la fase de codificación (*Encoder*), la dimensionalidad espacial es reducida progresi-

vamente mediante operaciones de convolución y *max-pooling*, lo cual permite la extracción de características de alto nivel como textura y forma. Posteriormente, en la fase de decodificación (*Decoder*), la resolución espacial es recuperada mediante *up-sampling*. Un componente crítico de este diseño son las conexiones residuales o *skip connections*, las cuales concatenan la información de las capas de contracción con las de expansión, permitiendo una delimitación precisa de los bordes de la mucosa. Para asegurar la viabilidad del despliegue en el sistema embebido, el número de filtros base fue ajustado a 16, escalando hasta un máximo de 256 en el cuello de botella.

Figura 3.6
Arquitectura Encoder-Decoder de la red U-Net propuesta.



Nota: El diagrama muestra la ruta de contracción (izquierda) y expansión (derecha), unidas por las conexiones de salto (skip connections).

Configuración de Hiperparámetros

Para determinar la configuración óptima de la red, se realizó un proceso de búsqueda exhaustiva (*Grid Search*). Se evaluaron distintas tasas de aprendizaje y optimizadores, seleccionándose aquellos parámetros que garantizaron la mejor convergencia y estabilidad durante la etapa de validación, siguiendo las recomendaciones de optimización para segmentación

biomédica (Fuentes-Beingolea et al., 2025).

La **Tabla 3.1** presenta la configuración final establecida. Los detalles del código de implementación y las curvas de entrenamiento se encuentran disponibles en el **Anexo D.1**.

Tabla 3.1

Hiperparámetros y configuración del entrenamiento para la red de segmentación U-Net.

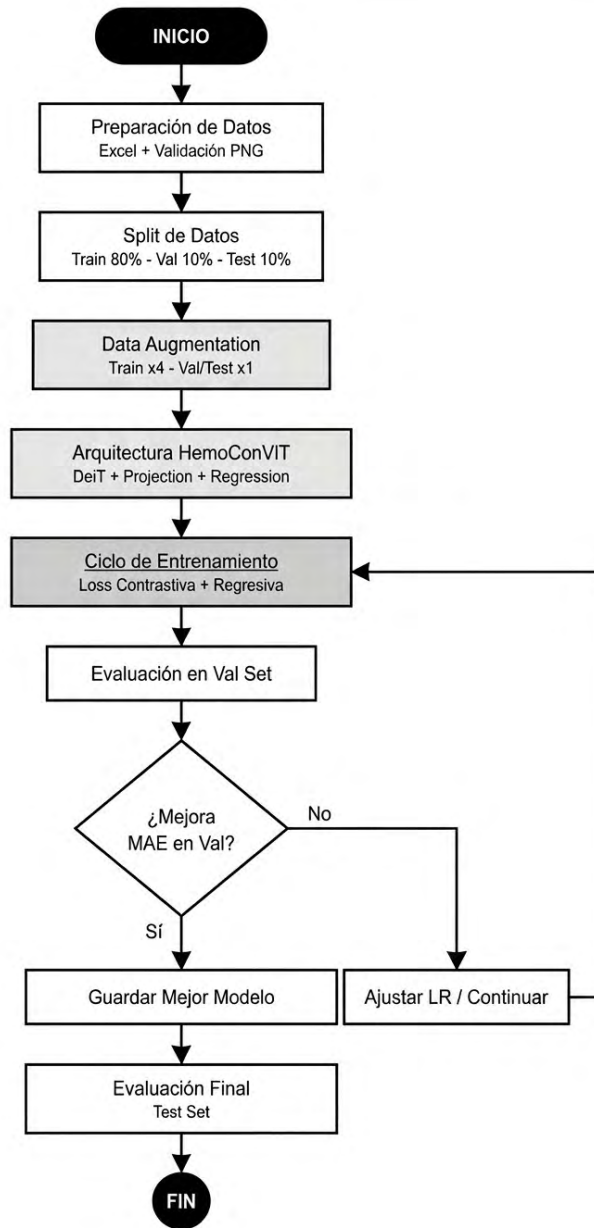
Parámetro	Configuración / Valor
Arquitectura	U-Net convolucional
Resolución de Entrada	512×512 píxeles (RGB)
Tamaño de Lote	2 imágenes
Optimizador	Adam (Learning Rate = 1×10^{-4})
Función de Pérdida	Híbrida: Binary Crossentropy + Dice Loss
Épocas	40 (Guardado del mejor checkpoint según IoU)
Semilla Aleatoria	45 (Reproducibilidad experimental)
Aumento de Datos	Rotación ($\pm 5\%$), Zoom ($\pm 10\%$), Flip horizontal

3.5.2. Subsistema de Clasificación por Regresión (Hemo-ConViT)

Una vez obtenida la máscara binaria, la región de interés (ROI) es recortada y redimensionada a 224×224 píxeles para ser procesada por el modelo de estimación denominado **Hemo-ConViT** (*Hemoglobin Conjunctiva Vision Transformer*). En este modelo se implementa una estrategia de aprendizaje contrastivo adaptativo diseñada específicamente para la regresión de valores continuos de hemoglobina, metodología fundamentada en nuestra investigación reciente (Fuentes-Beingolea & Palomino-Quispe, 2025). El flujo secuencial de este procesamiento se detalla en la **Figura 3.7**.

Figura 3.7

Flujo de procesamiento del subsistema de estimación de hemoglobina.



Nota: El esquema detalla la secuencia desde la preparación de la ROI y el aumento de datos, hasta la inferencia y obtención del valor numérico.

Visual Transformer Ligero

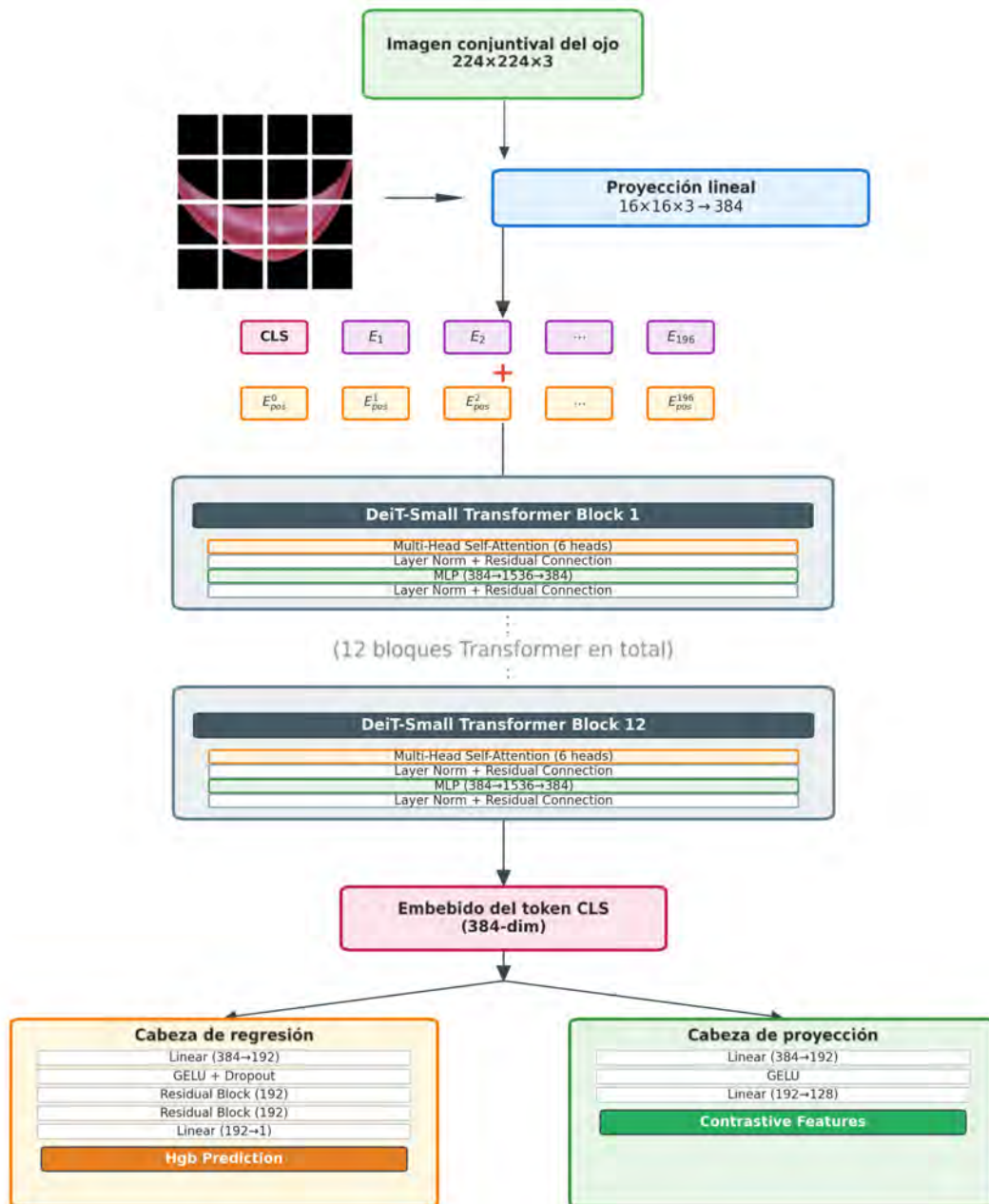
A diferencia de las Redes Neuronales Convolucionales (CNN) tradicionales que operan mediante filtros locales, la arquitectura seleccionada (detallada en la **Figura 3.8**) divide la imagen de la ROI en una secuencia de parches de 16×16 píxeles, los cuales son procesados

como una secuencia de vectores embebidos.

Fue incorporado el mecanismo de auto-atención (*Self-Attention*), característico de los *Vision Transformers* (Dosovitskiy et al., 2020), lo cual permite al modelo ponderar globalmente la relación entre las diferentes zonas de la conjuntiva. Esta capacidad resulta fundamental para mitigar el impacto de zonas pálidas causadas por brillo excesivo o artefactos lumínicos, forzando a la red a enfocar su atención en la vascularización real del tejido dispersa en la imagen. La elección de la arquitectura DeiT (Data-efficient Image Transformer) es crítica para la viabilidad de este estudio con la muestra disponible. A diferencia de los Vision Transformers (ViT) estándar que requieren cientos de millones de imágenes para converger, DeiT integra estrategias de destilación que le permiten generalizar eficazmente con volúmenes de datos reducidos. Esto, sumado al pre-entrenamiento en ImageNet-1k, mitiga el riesgo de sobreajuste (*overfitting*) en el dataset local, validando técnicamente el uso de las 79 imágenes para el ajuste final.

Figura 3.8

Arquitectura del modelo Hemo-ConViT basada en Vision Transformer.



Nota: El diagrama ilustra el flujo de los parches de imagen a través del Encoder y los cabezas de regresión y proyección.

Para optimizar el aprendizaje de características sutiles asociadas a la anemia y manejar la naturaleza continua de la variable objetivo, se definió una función de pérdida híbrida \mathcal{L}_{total} . Esta función combina la precisión de la regresión con la estructuración del espacio latente, tal como se visualiza en la **Figura 3.9**. La minimización se rige por la siguiente ecuación:

$$\mathcal{L}_{total} = \mathcal{L}_{reg} + \lambda \mathcal{L}_{con} \quad (3.2)$$

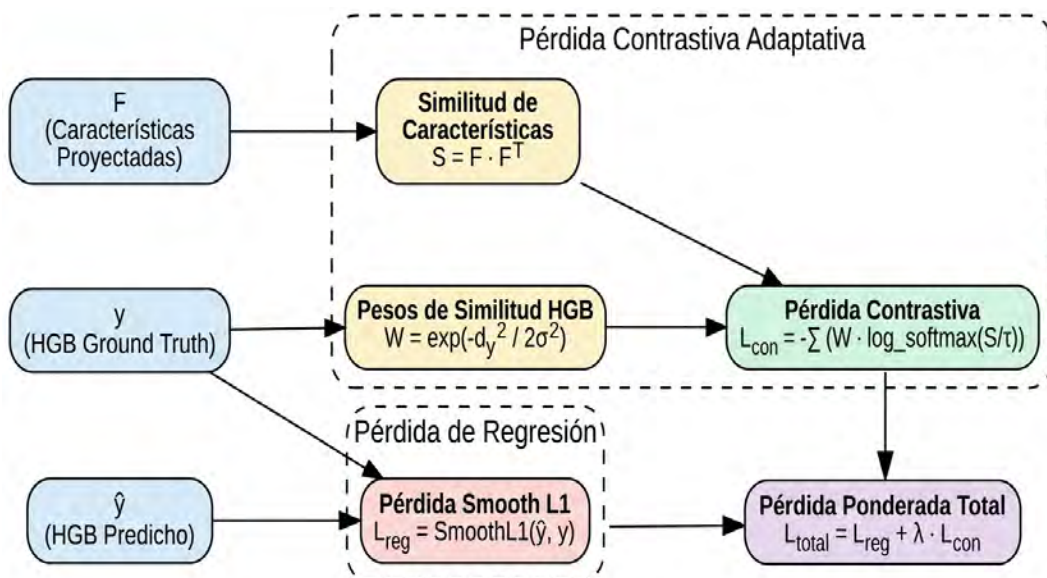
Donde \mathcal{L}_{reg} corresponde a la pérdida de regresión *Smooth L1*, seleccionada por su robustez ante valores atípicos, y λ es un hiperparámetro de ponderación. La innovación principal radica en el componente contrastivo adaptativo \mathcal{L}_{con} , el cual no utiliza clases binarias, sino que pondera la similitud entre dos muestras i y j basándose en la cercanía de sus valores reales de hemoglobina (y_i, y_j). El peso de similitud W_{ij} se calcula mediante un núcleo Gaussiano:

$$W_{ij} = \exp\left(-\frac{(y_i - y_j)^2}{2\sigma^2}\right) \quad (3.3)$$

Siendo σ el parámetro de sensibilidad que regula qué tan cercanos deben estar los valores de hemoglobina para ser considerados similares en el espacio de características. De esta forma, se fuerza al modelo a agrupar representaciones vectoriales de pacientes con niveles de hemoglobina clínicamente equivalentes.

Figura 3.9

Diagrama de pérdidas contrastiva y regresiva.



Nota: El esquema representa la función de pérdida híbrida utilizada, la cual combina la minimización del error de regresión con la optimización del espacio latente mediante aprendizaje contrastivo.

Cabezal de Regresión

La arquitectura concluye con un Perceptrón Multicapa (MLP) diseñado para proyectar las características extraídas a un único valor escalar continuo: la concentración estimada de hemoglobina en g/dL. Cabe destacar que en esta etapa se prescinde de una clasificación binaria directa, con el objetivo de preservar la riqueza de los datos continuos y permitir un ajuste más preciso a la variabilidad fisiológica del paciente.

Configuración de Hiperparámetros

Los hiperparámetros para Hemo-ConViT fueron seleccionados basándose en los resultados experimentales reportados en (Fuentes-Beingolea & Palomino-Quispe, 2025), priorizando la minimización del error absoluto medio (MAE) y la eficiencia computacional para sistemas embebidos. Los valores seleccionados se detallan en la **Tabla 3.2**, mientras que el código de entrenamiento correspondiente se adjunta en el Anexo D.2.

Tabla 3.2

Hiperparámetros y configuración del entrenamiento para el modelo Hemo-ConViT.

Parámetro	Configuración / Valor
Arquitectura Base	DeiT-Small (Data-efficient Image Transformer)
Pre-entrenamiento	ImageNet-1k (Transfer Learning)
Resolución de Entrada	224 × 224 píxeles
Tamaño de Lote	16 imágenes
Optimizador	AdamW (Weight Decay = 0.05)
Tasa de Aprendizaje	1×10^{-4} (con Cosine Annealing)
Función de Pérdida	Smooth L1 Loss (Regresión Robusta)
Factor de contraste (λ)	0.15
Sensibilidad Gaussiana (σ)	1.0
Épocas	50 (Guardado del mejor checkpoint por MAE)

3.6. Lógica de Alerta y Clasificación Clínica

Finalmente, la salida numérica generada por el modelo Hemo-ConViT es procesada por un algoritmo de decisión lógica diseñado para el tamizaje clínico. Este enfoque se enmarca en el paradigma de **clasificación basada en umbrales sobre una salida de regresión continua**, una decisión metodológica fundamental para la validez clínica del sistema.

Aunque la arquitectura de aprendizaje profundo opera como un regresor que estima la concentración de hemoglobina ($Y \in \mathbb{R}$), el sistema global actúa funcionalmente como un clasificador binario. Se optó por esta estrategia en lugar de un clasificador directo (p.ej., entrenado con entropía cruzada) basándose en dos principios teóricos. Primero, la regresión preserva la **magnitud de la condición clínica**; no solo indica la probabilidad de anemia, sino que cuantifica la distancia del paciente al umbral diagnóstico, información que se pierde en una clasificación puramente binaria. Segundo, este enfoque se alinea con la recomendación estadística de evitar la dicotomización prematura de variables continuas, ya que esto conlleva una pérdida de información y potencia estadística (Altman & Royston, 2006).

Desde la perspectiva del aprendizaje automático, la estrategia de modelar primero una densidad de probabilidad o una función de valor real para luego aplicar una regla de decisión es teóricamente más robusta, ya que separa el proceso de inferencia de la política de decisión (Bishop, 2006). En este sistema, los umbrales clínicos descritos a continuación actúan como dicha regla de decisión.

3.6.1. Definición de Umbrales y Corrección por Altitud

La definición del umbral de corte (*Threshold*) para la alerta de "Posible Anemia" varía según el entorno de despliegue y la procedencia de los datos, ajustándose a las normativas fisiológicas vigentes.

Escenario 1: Validación In-Silico (Nivel del Mar)

Para las etapas de validación utilizando exclusivamente el conjunto de datos público *Eye-Defy-Anemia*, recolectado en entornos a nivel del mar (Dimauro et al., 2023), se estableció un umbral conservador de **13.0 g/dL**. Este valor, correspondiente al límite inferior para hombres adultos según la OMS, fue seleccionado para maximizar la sensibilidad del sistema en un dataset que no discrimina por género en sus metadatos, asegurando así la cobertura de casos masculinos y femeninos sin riesgo de sub-diagnóstico en esta etapa experimental.

Escenario 2: Prototipo en Entorno Local (Adaptación de dominio)

Considerando que el sistema prototipo opera en la ciudad del Cusco, distrito de Wanchaq (3,399 m.s.n.m.), el umbral estándar resulta insuficiente debido al fenómeno de poliglobulia compensatoria (aumento de glóbulos rojos por hipoxia). En cumplimiento con la Norma Técnica de Salud N° 134-MINSA (Ministerio de Salud, 2017), se aplicó un factor de ajuste fisiológico.

Así, el algoritmo del prototipo debe elevar el umbral de decisión, manteniendo inalterado el valor estimado por la red neuronal. Basándose en el factor de ajuste de +2.3 g/dL correspondiente a la altitud local (ver Anexo C), el nuevo umbral de corte se define como:

$$Umbral_{Cusco} = Umbral_{Base} + Factor_{Ajuste} = 13.0 + 2.3 = \mathbf{15.3 \text{ g/dL}} \quad (3.4)$$

En consecuencia, para la evaluación del modelo con el conjunto de datos mixto (imágenes públicas + locales), el sistema clasifica como "Posible Anemia" a cualquier sujeto cuya estimación sea inferior a **15.3 g/dL**. Esta adaptación asegura que la herramienta no genere falsos negativos en la población local, respetando tanto la fisiología de altura como la diversidad del conjunto de entrenamiento.

Capítulo 4

Implementación del Sistema Prototipo

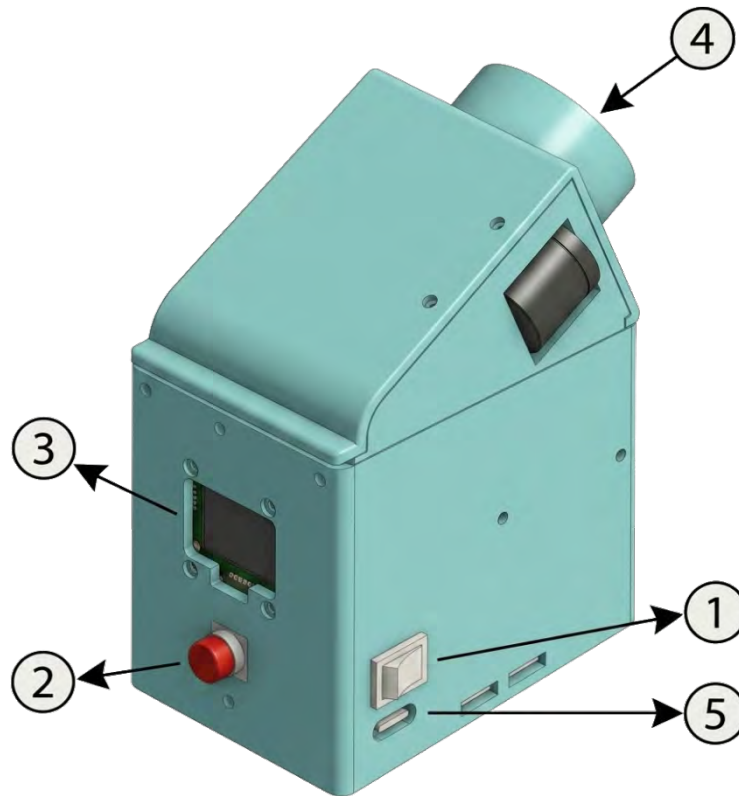
En este capítulo se detalla la integración física y lógica del sistema de detección de anemia. Se describen las especificaciones técnicas de los componentes de hardware seleccionados, los criterios de configuración necesarios para garantizar la coherencia con los modelos de Inteligencia Artificial entrenados y la lógica de programación del firmware que gobierna el dispositivo autónomo.

4.1. Componentes del Sistema

Esta sección describe la arquitectura de hardware del dispositivo, dividida en los módulos de procesamiento, adquisición óptica e interfaz de usuario. La **Figura 4.1** muestra el ensamblaje final del sistema, diseñado para ser ergonómico y portátil.

Figura 4.1

Vista preliminar 3D del ensamblaje de hardware del dispositivo prototipo.



Nota: En: 1. Interruptor de energía. 2. Botón de captura. 3. Pantalla OLED I2C. 4. Módulo de adquisición óptica (módulo 3D + cámara + lente macro + diodos LEDs difusos). 5. Conector de carga USB-C.

4.1.1. Módulo de Adquisición Óptica

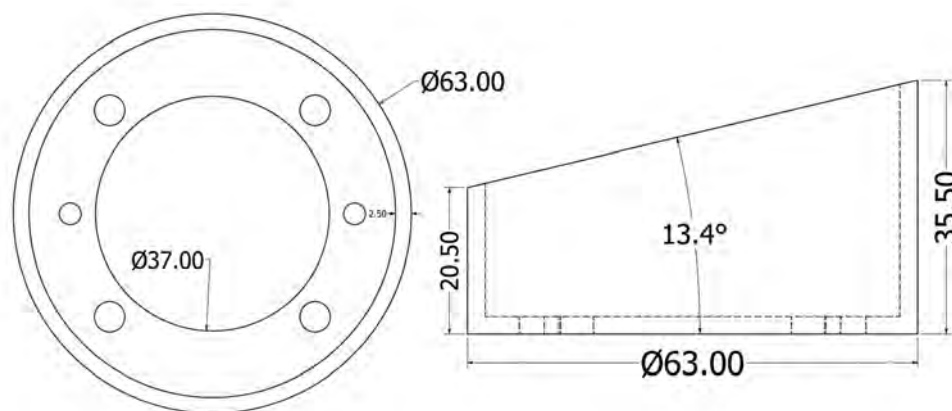
Se detalla el diseño del recinto oscuro, fabricado mediante manufactura aditiva (impresión 3D) utilizando ácido poliláctico (PLA) con coloración *Holly green mate*. Este material fue seleccionado por sus propiedades de opacidad para asegurar el aislamiento de la luz ambiental, condición crítica para la estabilidad del algoritmo de color. El diseño mantiene una distancia focal fija de **20 mm** entre el sensor y la superficie de la conjuntiva, estándar utilizado en fotografía macro para oftalmología básica.

Se integra una cámara USB con sensor CMOS 4K (Modelo MAIRUIGE), seleccionada por su capacidad de enfoque macro y control manual de parámetros de exposición (ver ficha técnica en el **Anexo E.2**). El sistema óptico se complementa con un lente macro genérico con

magnificación de x0.6, optimizado para la distancia de trabajo establecida.

Figura 4.2

Módulo de acoplamiento óptico.



Nota: Módulo óptico de impresión 3D diseñado para albergar el sensor de webcam, la iluminación controlada y adaptarse al rostro del paciente.

4.1.2. Sistema de Iluminación Controlada

El sistema de iluminación consta de un arreglo de diodos LED de luz blanca difusa, dispuestos geoméricamente para proporcionar una iluminación uniforme sobre la conjuntiva tarsal. La disposición física se justifica para evitar reflejos especulares directos (brillos) sobre la mucosa húmeda, los cuales constituyen el principal artefacto de ruido en la segmentación automática.

Para garantizar la estabilidad lumínica y la seguridad ocular, se implementó un circuito con resistencias limitadoras de 220Ω y un control por Modulación de Ancho de Pulso (PWM). Se programó una rutina de encendido suave (*Soft-Start*) para evitar deslumbramiento repentino en el paciente. La intensidad se limitó al 60 % del ciclo de trabajo nominal, asegurando un nivel de iluminancia adecuado para el sensor sin exceder los umbrales de confort visual para exposiciones cortas (< 1 segundo). Los parámetros de configuración se detallan en la **Tabla 4.1** y las especificaciones de los emisores en el **Anexo E.3**.

Tabla 4.1

Parámetros de control configurados para el sistema de iluminación y modulación PWM.

Parámetro	Valor	Función en el Sistema
Frecuencia PWM	1000 Hz	Frecuencia alta establecida para evitar el parpadeo visible (<i>flicker</i>) y el efecto de bandas en el sensor CMOS (<i>rolling shutter</i>).
Ciclo de Trabajo	60 % (0.6)	Intensidad lumínica nominal definida para equilibrar la exposición y el confort del paciente.
Pines de Control	GPIO 15, 16	Salidas digitales moduladas para los LEDs difusos.
Pasos de Rampa	25 pasos	Incrementos graduales para evitar picos de corriente (<i>Inrush current</i>).
Tiempo Estabilización	0.4 s	Retardo previo a la captura para permitir la estabilización del balance de blancos automático del sensor.

4.1.3. Unidad de Procesamiento (Sistema Embebido)

El núcleo de procesamiento seleccionado es el **Orange Pi 5 Max**, basado en el SoC Rockchip RK3588. Esta placa fue elegida por su arquitectura ARM de bajo consumo y, fundamentalmente, por la presencia de una Unidad de Procesamiento Neuronal (NPU) integrada con una potencia de 6 TOPS (ver **Anexo E.1**). Esta característica es necesaria para ejecutar la inferencia de los modelos U-Net y Vision Transformer en tiempo real (*Edge Computing*), eliminando la dependencia de conexión a internet para el procesamiento. La elección se sustenta en la comparación de la **Tabla 2.1**.

4.1.4. Interfaz de Usuario y Visualización

La interacción con el usuario se gestiona a través de una pantalla OLED de 0.96 pulgadas (comunicación vía protocolo I2C, ver **Anexo E.3**) y un botón físico para la captura. La interfaz gráfica fue diseñada bajo un criterio minimalista, mostrando el estado del sistema, el nivel de batería y el resultado del diagnóstico en tiempo real, facilitando su uso por personal no especializado (ver **Figura 4.3**).

Figura 4.3

Funcionamiento de la pantalla OLED mostrando el estado del sistema en tiempo real.

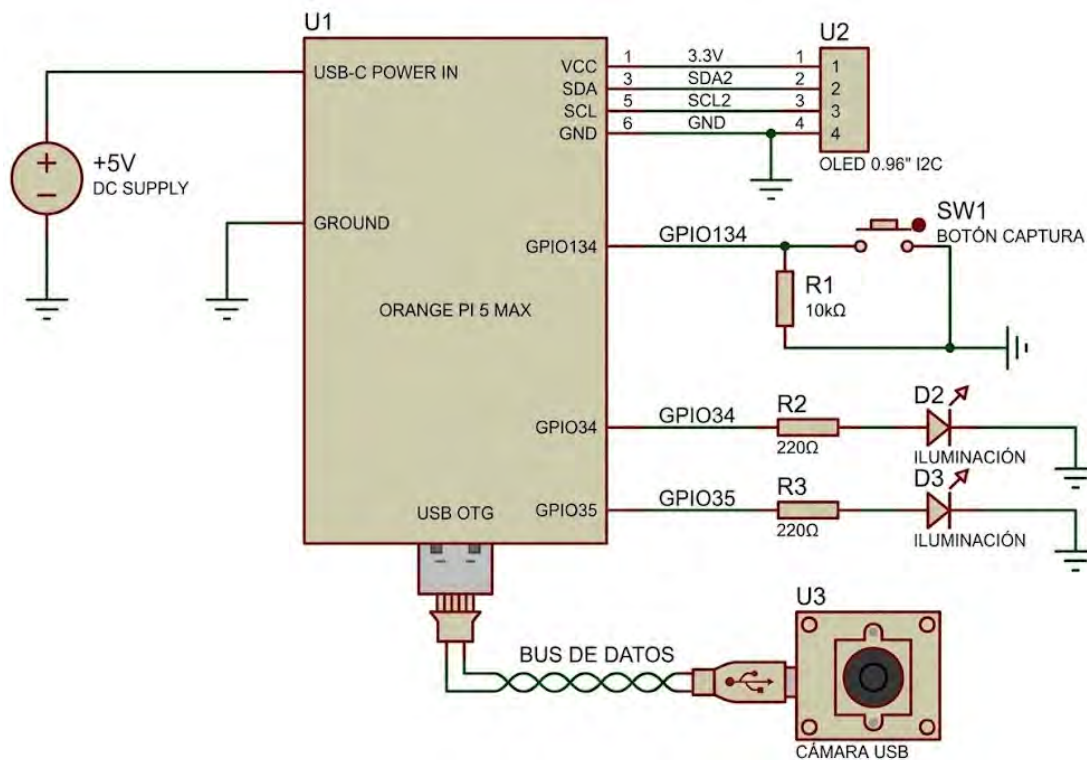


4.1.5. Sistema de Alimentación y Autonomía

La energía es suministrada por un banco de baterías (*Powerbank*) comercial modelo J134 de hasta 5000 mAh (ver **Anexo E.4**). Se optó por una solución comercial certificada (cumplimiento con estándares de seguridad para baterías de litio como IEC 62133) para garantizar la estabilidad del voltaje de 5V y la seguridad térmica del dispositivo portátil, en cumplimiento de los requisitos energéticos detallados en **4.4.1**.

Figura 4.4

Esquemático de conexiones de periféricos para Orange Pi 5 Max.



Finalmente, la interconexión de todos los periféricos con el sistema embebido se resume en el mapa de pines (*Pinout*) presentado en la **Tabla 4.2**.

Tabla 4.2

Mapa de conexiones e interfaces físicas del sistema (Pinout).

Periférico	Interfaz	Pin Físico	ID Lógico
Botón de Captura	Digital (Pull-Down)	Pin 12	GPIO 134
LEDs Iluminación	Salida PWM	Pin 15, 16	GPIO 34, 35
Pantalla OLED	Bus I2C	Pin 3, 5	Bus 2, 0x3C
Cámara	MIPI/USB	Puerto USB	<i>video0</i>

4.2. Criterios de Implementación en el Sistema Embebido

Para garantizar que los modelos de Inteligencia Artificial funcionen correctamente en el hardware final y repliquen las condiciones del entrenamiento, se establecieron criterios estrictos de configuración y optimización.

4.2.1. Estandarización de Parámetros de Captura

Se implementó una configuración fija en el controlador de la cámara, bloqueando los algoritmos automáticos de exposición (AE) y balance de blancos (AWB). Esta medida es crítica para asegurar que las imágenes capturadas en campo posean las mismas propiedades estadísticas de histograma que las imágenes utilizadas durante el entrenamiento. Los valores establecidos se presentan en la **Tabla 4.3**.

Tabla 4.3

Parámetros fijos de configuración del sensor óptico para la estandarización de captura.

Parámetro	Valor	Descripción / Justificación
Resolución	3264 × 2448 px	Resolución nativa máxima (4K) seleccionada para preservar el detalle microvascular de la conjuntiva.
Enfoque	230 (Fijo)	Valor calibrado para la distancia focal fija de 20 mm del cilindro impreso.
Exposición	7500 μ s	Tiempo de integración fijo para evitar saturación en la mucosa bajo la iluminación LED controlada.
Ganancia	100 (ISO Base)	Ganancia mínima para reducir el ruido electrónico (grano) en la imagen.
Balance Blancos	5300 K	Temperatura de color fijada para coincidir con la emisión espectral de los LEDs integrados.
Auto-Exposición	Desactivado	Bloqueo necesario para cumplir con el criterio de consistencia del algoritmo Post-Reinhard.

4.2.2. Garantía de Coherencia con el Dominio de Entrenamiento

Si bien el algoritmo de transferencia de color de Reinhard se aplicó en la etapa de preprocesamiento para adaptar el dataset de entrenamiento, su efectividad en la fase de inferencia depende estrictamente de la estabilidad del sistema de adquisición físico.

Por consiguiente, para la implementación en el sistema embebido, se estableció como requisito la **inmutabilidad de los parámetros ópticos**. Al fijar la ganancia electrónica del sensor y el tiempo de exposición, se garantiza que los valores de píxel capturados en tiempo real sean estadísticamente congruentes con los datos procesados mediante la técnica Reinhard durante el entrenamiento. Esta configuración de hardware actúa como la contraparte física del algoritmo de adaptación de dominio.

4.2.3. Optimización de Modelos para la NPU

Para aprovechar la aceleración por hardware del SoC RK3588, los modelos entrenados en PyTorch (U-Net y Hemo-ConViT) fueron convertidos al formato propietario RKNN. Este proceso incluyó la cuantización de pesos a precisión media (FP16), lo cual permite reducir el uso de memoria RAM y acelerar el tiempo de inferencia sin una pérdida significativa de precisión. El código de conversión utilizado se encuentra detallado en el **Anexo C.3**.

Tabla 4.4

Parámetros de conversión y cuantización para la NPU RK3588.

Parámetro	Configuración
Plataforma Objetivo	RK3588
Formato de Entrada	ONNX (Opset 15)
Formato de Datos	NCHW (Canales primero, optimizado para NPU)
Tipo de Cuantización	FP16 (Punto flotante de 16 bits)
Normalización (Mean)	[0, 0, 0] (Preprocesamiento en CPU)
Normalización (Std)	[255, 255, 255] (Normalización a rango 0-1)
Nivel de Optimización	3 (Máximo rendimiento)

4.3. Lógica de Funcionamiento del Prototipo

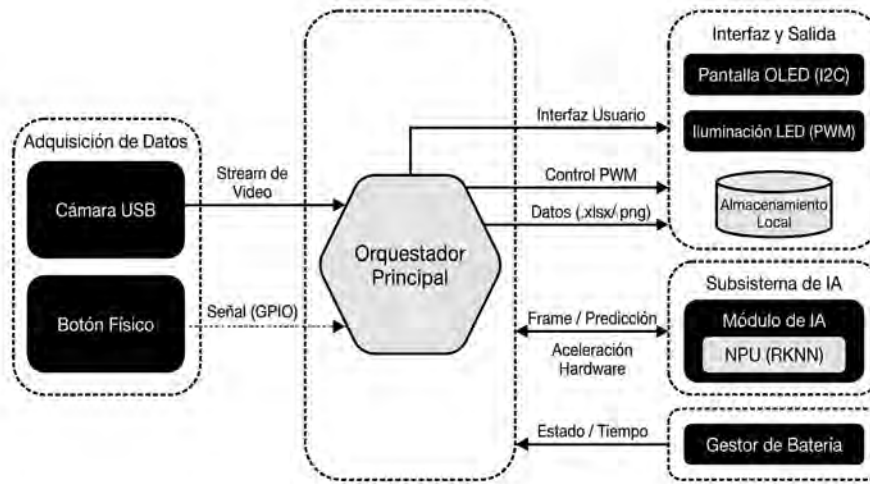
Esta sección describe la arquitectura del software (*Firmware*) desarrollado en Python 3.10, el cual orquesta la interacción entre el hardware, la adquisición de imágenes y la inferencia de los modelos de inteligencia artificial en la NPU. El sistema opera de manera autónoma, iniciándose automáticamente al energizar la placa mediante un servicio de sistema (*systemd*). La estructura completa de los códigos y algoritmos implementados se encuentra documentada en el **Anexo C.4**.

4.3.1. Máquina de Estados General

Para garantizar la estabilidad operativa y evitar comportamientos indefinidos, el software se diseñó siguiendo el patrón de una máquina de estados finitos. El flujo de control gestiona las transiciones deterministas entre los estados de *Inicialización*, *Reposo (Idle)*, *Captura*, *Procesamiento* y *Visualización*, tal como se ilustra en la **Figura 4.5**.

Figura 4.5

Diagrama general del firmware (Ciclo de vida).



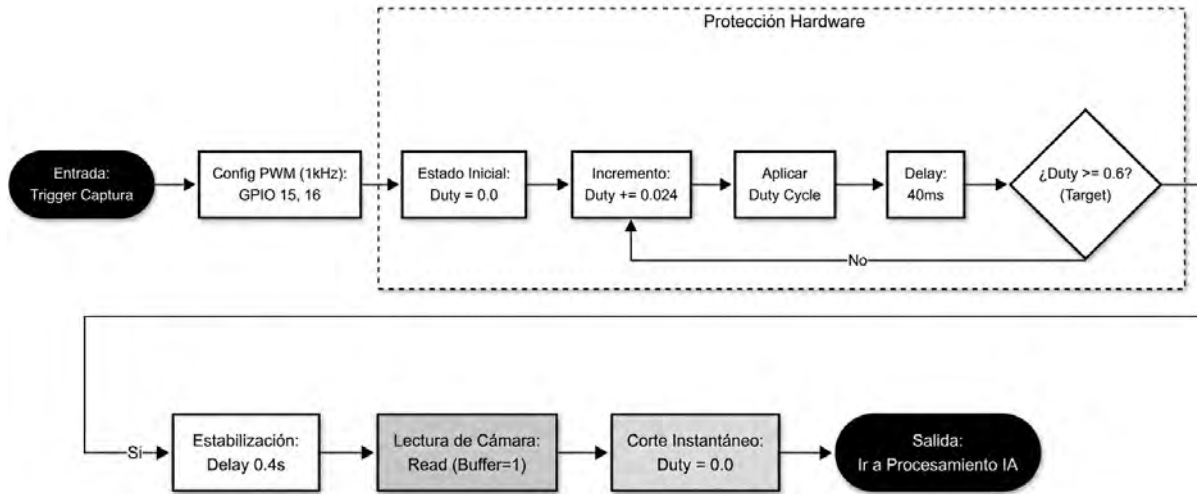
En el estado de **Inicialización**, se cargan los modelos RKNN en la memoria de la NPU y se verifican los periféricos conectados.

4.3.2. Lógica de Captura y Procesamiento Secuencial

La rutina de adquisición sigue una secuencia estricta para sincronizar la iluminación con la exposición del sensor, minimizando la variabilidad lumínica externa. Al detectar la interrupción física en el GPIO 134 (botón), el sistema ejecuta la subrutina de iluminación controlada (**Figura 4.6**) que aplica una rampa PWM (*Soft-Start*) para evitar picos de corriente, captura el fotograma en formato RAW y apaga los LEDs inmediatamente para confort del paciente.

Figura 4.6

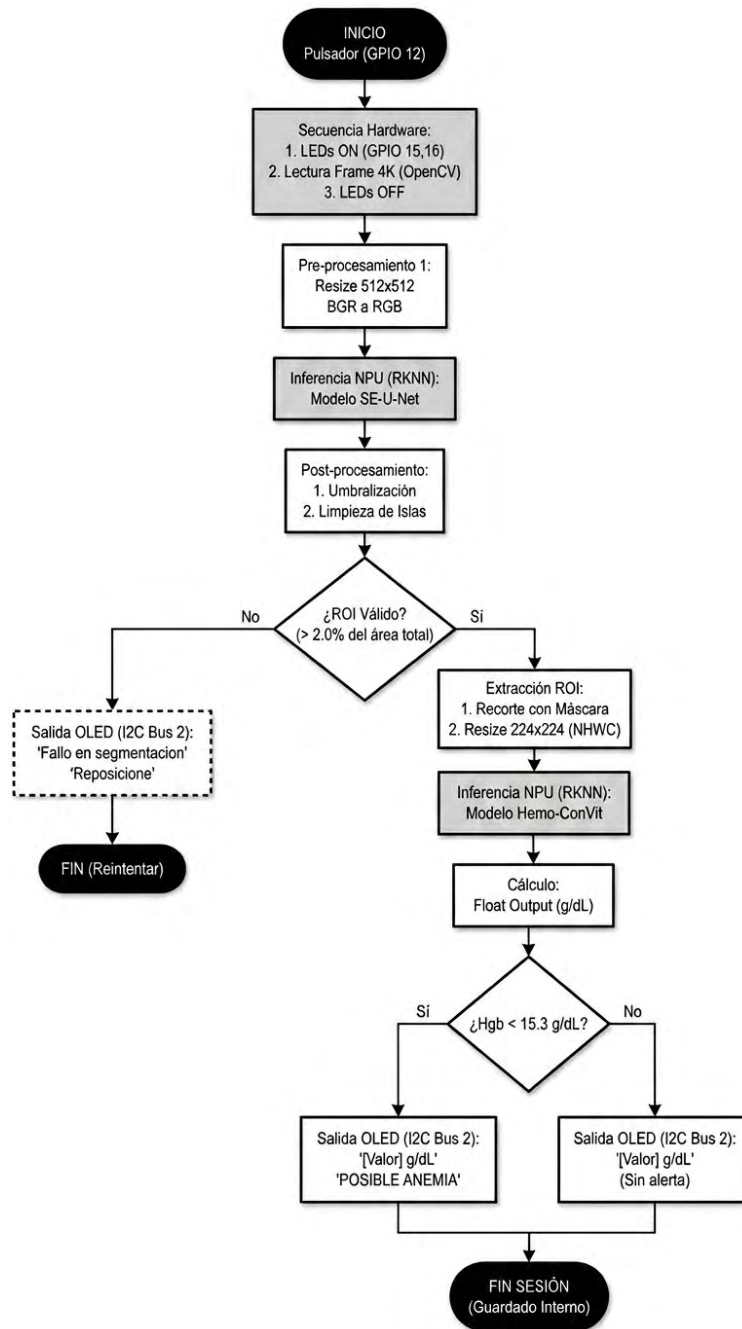
Diagrama de flujo de la subrutina de iluminación controlada y captura de imagen.



Posteriormente, la imagen ingresa al flujo de inferencia acelerada (**Figura 4.7**). Se realiza el preprocesamiento (normalización y redimensionamiento) antes de ejecutar secuencialmente los modelos de segmentación (U-Net) y regresión (Hemo-ConViT), cuyos resultados se muestran en la pantalla OLED. Esta arquitectura secuencial permite una gestión eficiente de los recursos de la NPU.

Figura 4.7

Flujo lógico de la inferencia de Inteligencia Artificial en la NPU del sistema.



4.3.3. Monitoreo Energético y Gestión de Recursos

Dada la naturaleza portátil del dispositivo, se implementó un módulo de software dedicado a la gestión energética (ver **Anexo C.4.5**). Este script monitorea el tiempo de uso acumulado para estimar la carga restante de la batería y gestiona las alertas visuales en la pantalla OLED,

asegurando que el dispositivo no se apague abruptamente.

La **Figura 4.8** muestra la lógica general del gestor, mientras que la **Figura 4.9** detalla los umbrales de decisión para emitir advertencias de "Batería Baja" o ejecutar un apagado seguro del sistema operativo para proteger la integridad de los archivos y la tarjeta de memoria.

Figura 4.8

Lógica general del sistema de gestión de energía y estados de batería.

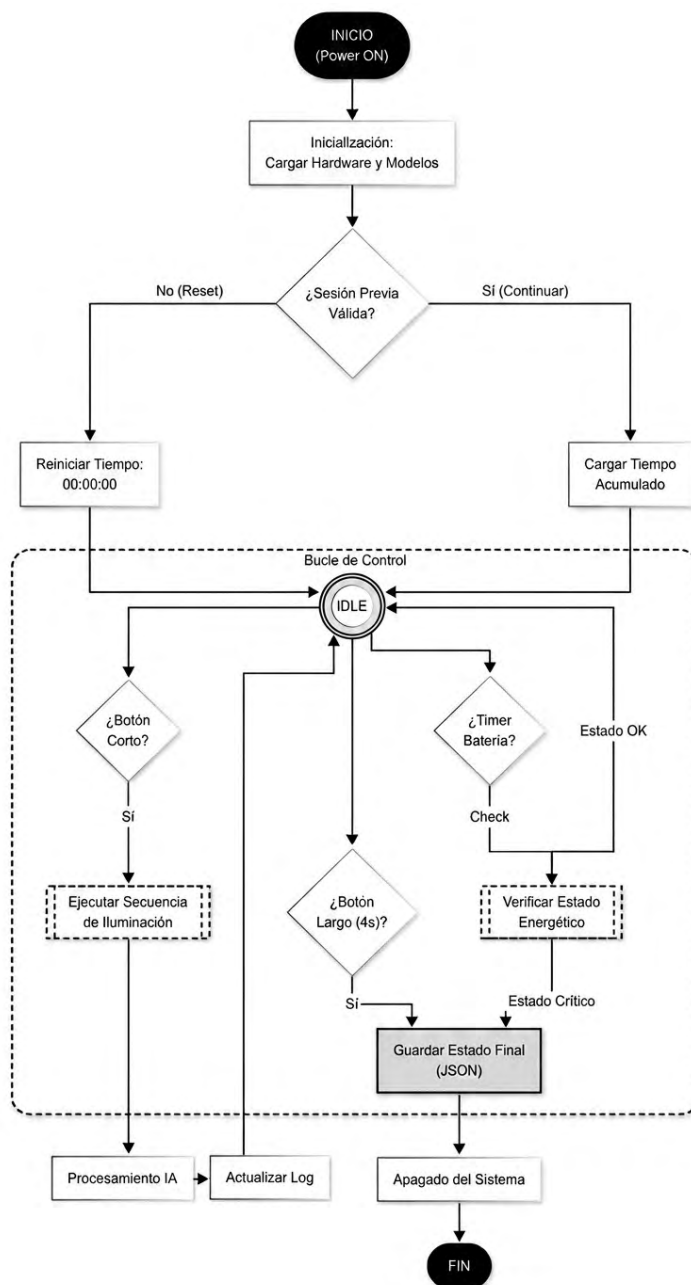
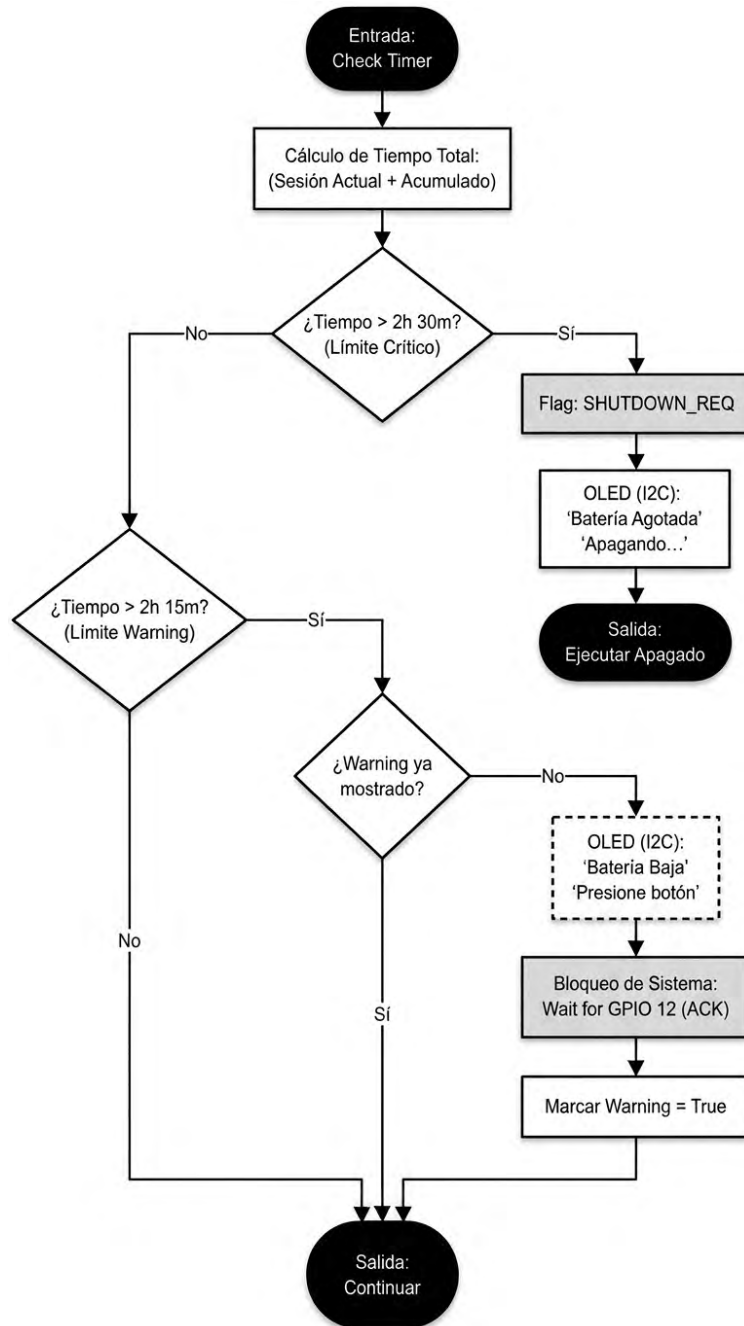


Figura 4.9

Detalle del algoritmo de decisión para las alertas de monitoreo energético.



4.3.4. Integración de Servicios y Buses de Comunicación

Para la correcta ejecución del hardware, se verificó la disponibilidad de los buses de comunicación y el acceso a los pines GPIO mediante herramientas de consola en el sistema operativo Linux (Ubuntu OS).

La **Figura 4.10** muestra el estado de los pines GPIO configurados, mientras que la **Figura 4.11** confirma la detección de la pantalla OLED en la dirección 0x3C del bus I2C-2. Finalmente, para garantizar la operatividad autónoma, se configuró un servicio *systemd* que lanza la aplicación principal al arranque del sistema operativo (**Figura 4.12**).

Figura 4.10

Verificación de estado de pines GPIO en consola del Orange Pi 5 Max.

```

facundo@facundo-liecar: ~
facundo@facundo-liecar:~$ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GPIO | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | GPIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 16 | 0 | 3.3V | | | 1 | 2 | | | 5V | | |
| 15 | 1 | SDA.2 | ALT9 | 1 | 3 | 4 | | | 5V | | |
| 39 | 2 | SCL.2 | ALT9 | 1 | 5 | 6 | | | GND | | |
| | | PWM3 | IN | 1 | 7 | 8 | 1 | ALT10 | TXD.2 | 3 | 13 |
| | | GND | | | 9 | 10 | 1 | ALT10 | RXD.2 | 4 | 14 |
| 32 | 5 | RXD.6 | IN | 0 | 11 | 12 | 0 | IN | GPIO4_A6 | 6 | 134 |
| 33 | 7 | TXD.6 | OUT | 0 | 13 | 14 | | | GND | | |
| 34 | 8 | GPIO1_A2 | OUT | 0 | 15 | 16 | 0 | OUT | GPIO1_A3 | 9 | 35 |
| | | 3.3V | | | 17 | 18 | 0 | IN | GPIO1_A4 | 10 | 36 |
| 42 | 11 | SPI0_TXD | IN | 0 | 19 | 20 | | | GND | | |
| 41 | 12 | SPI0_RXD | IN | 0 | 21 | 22 | 1 | IN | GPIO1_B0 | 13 | 40 |
| 43 | 14 | SPI0_CLK | IN | 0 | 23 | 24 | 1 | IN | SPI0_CS0 | 15 | 44 |
| | | GND | | | 25 | 26 | 1 | IN | SPI0_CS1 | 16 | 45 |
| 47 | 17 | RXD.1 | IN | 1 | 27 | 28 | 1 | IN | TXD.1 | 18 | 46 |
| 113 | 19 | GPIO3_C1 | IN | 1 | 29 | 30 | | | GND | | |
| 109 | 20 | CAN1_RX | IN | 1 | 31 | 32 | 1 | IN | PWM14 | 21 | 62 |
| 110 | 22 | CAN1_TX | IN | 1 | 33 | 34 | | | GND | | |
| 114 | 23 | GPIO3_C2 | IN | 1 | 35 | 36 | 1 | IN | GPIO3_D7 | 24 | 63 |
| 135 | 25 | GPIO4_A7 | IN | 0 | 37 | 38 | 1 | IN | GPIO3_C0 | 26 | 112 |
| | | GND | | | 39 | 40 | 1 | IN | GPIO3_B7 | 27 | 111 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| GPIO | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | GPIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
facundo@facundo-liecar:~$ 3-

```

Figura 4.11

Detección del dispositivo OLED en el bus I2C (dirección 0x3C).

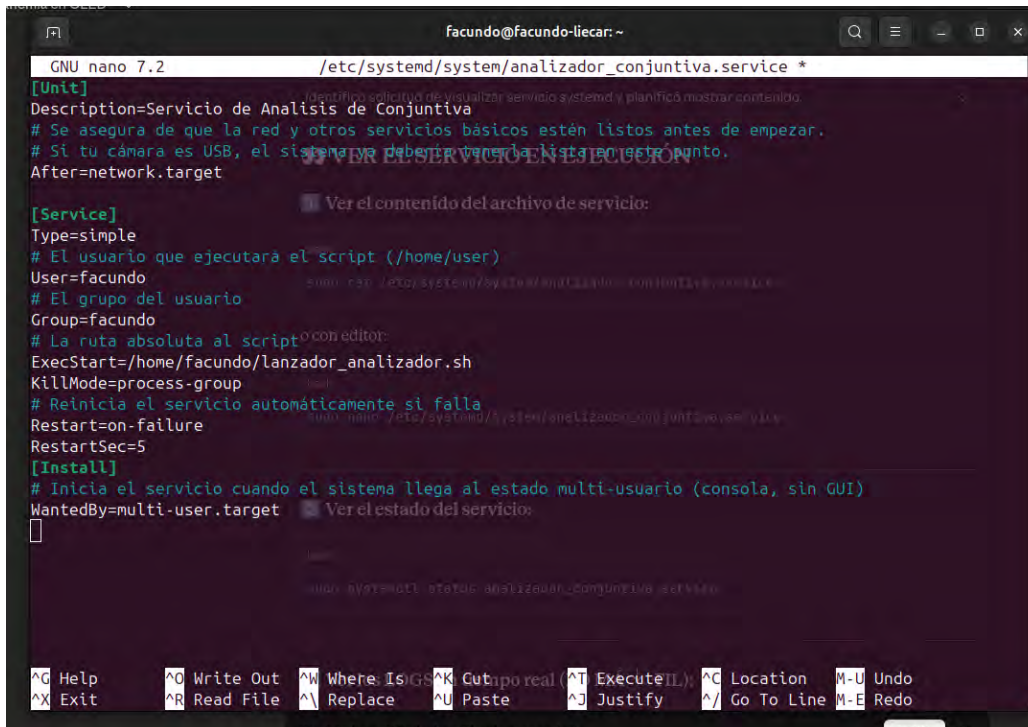
```

facundo@facundo-liecar: ~
facundo@facundo-liecar:~$ sudo systemctl start analizador_conjuntiva.service
[sudo] password for facundo:
facundo@facundo-liecar:~$ sudo systemctl stop analizador_conjuntiva.service
facundo@facundo-liecar:~$ sudo systemctl stop analizador_conjuntiva.service
facundo@facundo-liecar:~$ sudo systemctl start analizador_conjuntiva.service
facundo@facundo-liecar:~$ sudo i2cdetect -y 2
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- 3c -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
facundo@facundo-liecar:~$

```

Figura 4.12

Configuración del servicio systemd para el lanzamiento automático de la aplicación.



```
GNU nano 7.2 /etc/systemd/system/analizador_conjuntiva.service *
[Unit]
Description=Servicio de Analisis de Conjuntiva
# Se asegura de que la red y otros servicios básicos estén listos antes de empezar.
# Si tu cámara es USB, el sistema va a tener que estar en ese punto.
After=network.target

[Service]
Type=simple
# El usuario que ejecutará el script (/home/user)
User=facundo
# El grupo del usuario
Group=facundo
# La ruta absoluta al script
ExecStart=/home/facundo/lanzador_analizador.sh
KillMode=process-group
# Reinicia el servicio automáticamente si falla
Restart=on-failure
RestartSec=5

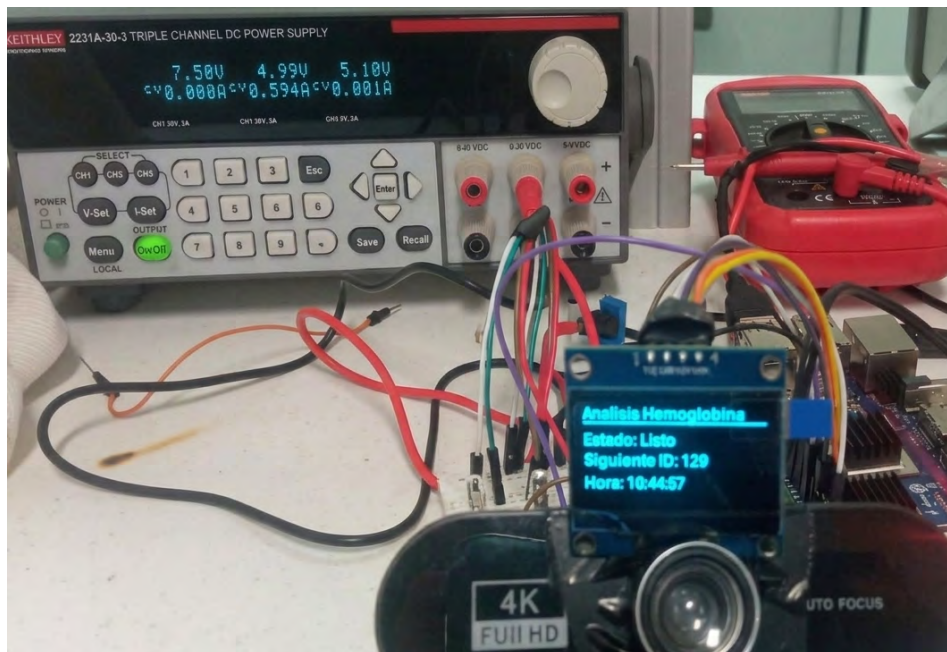
[Install]
# Inicia el servicio cuando el sistema llega al estado multi-usuario (consola, sin GUI)
WantedBy=multi-user.target
```

4.4. Validación de Consumo Energético

Se realizaron pruebas de medición de corriente y potencia para validar la autonomía del sistema bajo diferentes estados operativos. Las mediciones se efectuaron utilizando una fuente de alimentación de laboratorio simulando la salida del *powerbank* elegido a 5V que energiza directamente al dispositivo (ver especificaciones de instrumentación en **Anexo E.4**).

Figura 4.13

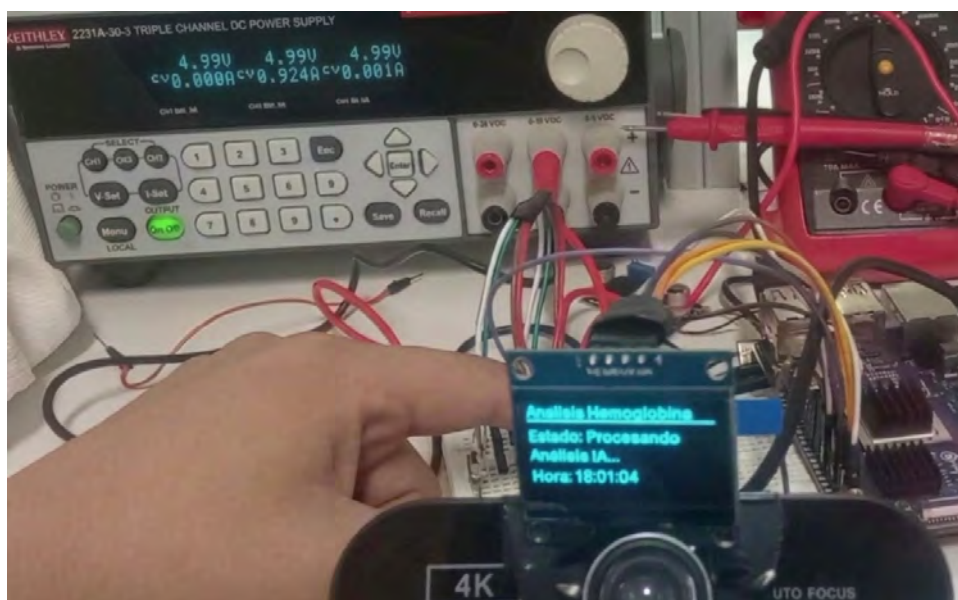
Medición del consumo de corriente en estado de reposo (Idle).



Nota: Consumo simulado en estado de espera. La potencia registrada es de aproximadamente 2.97 W, lo cual garantiza una autonomía prolongada en modo standby.

Figura 4.14

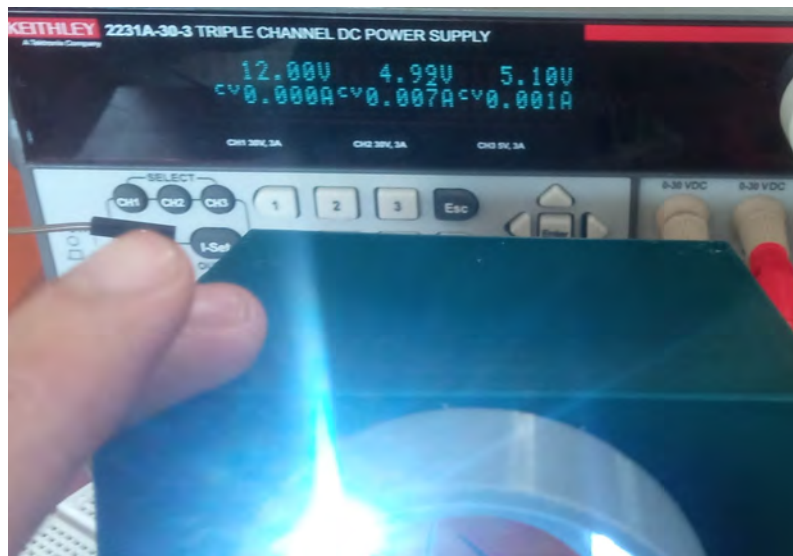
Medición del consumo de corriente durante la inferencia de modelos IA.



Nota: Durante el procesamiento intensivo en la NPU, la potencia asciende a 4.62 W. Este pico de consumo es breve (<1 segundo por captura).

Figura 4.15

Medición del consumo del sistema de iluminación activado.



Nota: El consumo adicional de los LEDs difusos es marginal (0.035 W).

4.4.1. Justificación del Sistema de Alimentación y Autonomía

Requisito de Autonomía

El sistema está concebido para operar en campañas de tamizaje de salud en entornos de punto de atención (*Point-of-Care*), en las cuales una sesión clínica típica tiene una duración de entre dos y tres horas (McCormack et al., 2018). Esto es consistente con la sesión de recolección de datos realizada en esta investigación, que tuvo una duración de tres horas (véase la Sección 3.3). Por lo tanto, se establece como requisito de diseño una autonomía operativa mínima de:

$$t_{\text{objetivo}} = 2.5 \text{ horas} \quad (4.1)$$

Este valor garantiza la cobertura de una sesión completa y otorga un margen de seguridad de 30 minutos frente a la duración máxima esperada.

Caracterización del Perfil de Consumo

A partir de las mediciones realizadas sobre el sistema ensamblado (Sección 4.4), se identificaron tres estados operativos:

Tabla 4.5

Estados operativos y consumo de potencia del sistema.

Estado	Descripción	Potencia (W)
Reposo (<i>Idle</i>)	Sistema en espera de captura	2.97
Iluminación sola	LEDs sin inferencia activa	0.035
Activo (NPU + LEDs)	Inferencia + iluminación LED	4.66

Para el cálculo de la capacidad de batería, se define un **perfil de carga conservador** basado en el uso clínico esperado. El tiempo de procesamiento total por captura es de 664.5 ms en promedio (Tabla 5.9), incluyendo preprocesamiento e inferencia de ambas redes neuronales. Considerando un escenario de alta demanda con una captura cada 60 segundos, la fracción del tiempo en estado activo es:

$$\alpha = \frac{t_{\text{inferencia}}}{t_{\text{ciclo}}} = \frac{0.665 \text{ s}}{60 \text{ s}} \approx 1.1 \% \quad (4.2)$$

Para dimensionar la batería con holgura, se adopta $\alpha = 10 \%$ como escenario conservador (equivalente a una captura cada ≈ 6.6 segundos, ritmo superior a cualquier uso clínico real). El consumo promedio ponderado resulta:

$$P_{\text{prom}} = P_{\text{idle}} \cdot (1 - \alpha) + P_{\text{activo}} \cdot \alpha \quad (4.3)$$

$$P_{\text{prom}} = 2.97 \cdot 0.90 + 4.66 \cdot 0.10 = 2.673 + 0.466 = 3.14 \text{ W} \quad (4.4)$$

Cálculo de la Capacidad Mínima Requerida

La energía mínima que debe entregar la batería durante la sesión operativa es:

$$E_{\min} = P_{\text{prom}} \cdot t_{\text{objetivo}} = 3.14 \text{ W} \times 2.5 \text{ h} = 7.85 \text{ Wh} \quad (4.5)$$

Sin embargo, no debe usarse el 100 % de la capacidad disponible. Para baterías de polímero de litio (LiPo), la descarga profunda acorta drásticamente la vida útil y puede causar inestabilidad de voltaje al final de la curva de descarga. Se aplican dos factores de corrección estándar (Reddy, 2011):

- **Profundidad de descarga máxima** (*Depth of Discharge, DoD*): $f_{\text{DoD}} = 0.80$, es decir, se utiliza como máximo el 80 % de la capacidad nominal, reservando el 20 % restante para proteger la química de la celda y garantizar tensión estable en los picos de demanda (hasta 4.66 W durante la inferencia).
- **Factor de envejecimiento** (*aging factor*): $f_{\text{age}} = 0.90$, considerando una reducción del 10 % en la capacidad efectiva tras los ciclos de carga/descarga propios de un prototipo en uso.

La fracción de energía aprovechable es entonces:

$$f_{\text{util}} = f_{\text{DoD}} \cdot f_{\text{age}} = 0.80 \times 0.90 = 0.72 \quad (4.6)$$

Por lo tanto, la energía total que debe poseer la batería para garantizar E_{\min} es:

$$E_{\text{req}} = \frac{E_{\min}}{f_{\text{util}}} = \frac{7.85 \text{ Wh}}{0.72} = 10.90 \text{ Wh} \quad (4.7)$$

Expresada como capacidad en la salida de 5 V:

$$C_{\min} = \frac{E_{\text{req}}}{V_{\text{salida}}} = \frac{10.90 \text{ Wh}}{5 \text{ V}} = 2.18 \text{ Ah} = \mathbf{2180 \text{ mAh @ 5 V}} \quad (4.8)$$

Selección de la Batería y Verificación

El banco de baterías comercial HOCO J134 fue seleccionado por satisfacer el requisito calculado. Su hoja de datos (Anexo E.4) especifica una *Rated Capacity* de $C_{\text{rated}} = 2600 \text{ mAh}$ a $5 \text{ V}/2 \text{ A}$, correspondiente a su capacidad real disponible en la salida de 5 V . Esta cifra debe distinguirse de la capacidad nominal de celda (5000 mAh a 3.7 V), que representa la carga total almacenada en la celda antes de la conversión interna de voltaje.

Se verifica que $C_{\text{rated}} = 2600 \text{ mAh} > C_{\min} = 2180 \text{ mAh}$, con un margen de capacidad de:

$$\text{Margen} = \frac{C_{\text{rated}} - C_{\min}}{C_{\min}} \times 100 = \frac{2600 - 2180}{2180} \times 100 \approx 19.3 \% \quad (4.9)$$

Este margen absorbe variaciones en el perfil de uso, la incertidumbre en los valores de f_{DoD} y f_{age} , y garantiza que el sistema alcance la autonomía requerida incluso en condiciones de batería parcialmente degradada.

Verificación del tiempo operativo disponible. Con la batería seleccionada, el tiempo máximo operativo bajo el perfil conservador ($\alpha = 10 \%$) es:

$$t_{\text{disp}} = \frac{C_{\text{rated}} \cdot f_{\text{util}}}{I_{\text{prom}}} = \frac{2600 \text{ mAh} \times 0.72}{P_{\text{prom}}/V_{\text{sal}}} = \frac{1872 \text{ mAh}}{628 \text{ mA}} \approx 2.98 \text{ h} \quad (4.10)$$

El algoritmo de gestión energética (Sección 4.3.3 y Anexo C.4.5) fuerza el apagado seguro del sistema (*graceful shutdown*) al cumplir $t_{\text{objetivo}} = 2.5 \text{ h}$, momento en que aún restan aproximadamente $2.98 - 2.50 = 0.48 \text{ h}$ de energía aprovechable en la batería. Esta reserva ($\approx 16 \%$ del tiempo operativo disponible) previene caídas de tensión bruscas (*brownouts*) que podrían corromper el sistema de archivos o interrumpir la inferencia activa en la NPU. La

advertencia visual se activa 15 minutos antes, a las 2,25 h (Figura 4.9).

4.5. Costos de Implementación

Se presenta el presupuesto detallado para la fabricación e implementación del prototipo desarrollado. El análisis de costos contempla los principales componentes electrónicos, los materiales empleados en la manufactura aditiva mediante impresión 3D, así como los accesorios necesarios para la integración y funcionamiento del sistema.

En la **Tabla 4.6** se detalla el desglose de costos de cada componente utilizado. Los valores presentados permiten estimar el costo global de implementación del dispositivo y evaluar su viabilidad económica frente a soluciones comerciales existentes, resultando en un costo total aproximado de S/. 720.00.

Tabla 4.6
Costos de manufacturación del dispositivo implementado.

Componente	Cantidad	Costo Unitario (S/.)	Costo Total (S/.)
Orange Pi 5 Max	1	520	520
Cámara web 4K MAIRUIGE	1	60	60
Lente macro	1	10	10
Pantalla OLED 0.96"	1	10	10
LEDs difusos	2	1	2
Power bank HOCO J134	1	45	45
Botón	1	1	1
Interruptor KCD2-201N-B	1	2	2
Resistencias varias	1 lote	5	5
Impresión 3D (carcasa)	5 piezas	–	65
Total			720

Nota: Los costos corresponden a precios referenciales de mercado al momento de la implementación del dispositivo.

Capítulo 5

Experimentos y Resultados

En este capítulo se presentan las métricas cuantitativas obtenidas tras el entrenamiento y validación de los modelos de inteligencia artificial, así como la evaluación del desempeño del prototipo final en condiciones reales de operación en la ciudad del Cusco. El análisis se divide en tres fases: validación base con datos públicos (estado del arte), validación de la estrategia de adaptación de dominio (dataset mixto con técnica Reinhard) y validación clínica del sistema embebido optimizado.

5.1. Fase 1: Evaluación Base (In-Silico)

Esta etapa evalúa la capacidad teórica de las arquitecturas propuestas utilizando únicamente el dataset público *Eyes-Defy-Anemia*, estableciendo una línea base de comparación antes de la adaptación al entorno local.

5.1.1. Desempeño en Segmentación

Se presenta la evaluación del desempeño del modelo de segmentación entrenado utilizando el conjunto de datos de referencia. Con el objetivo de cuantificar la incertidumbre de las métricas obtenidas y evaluar su significancia estadística, se aplicó un procedimiento de *bootstrapping* con 1000 iteraciones. Los detalles de la implementación y el código de entrenamiento correspondiente se encuentran disponibles en el **Anexo D.1**.

La Tabla 5.1 resume los valores promedio, la desviación estándar y los rangos obtenidos. Se destaca una exactitud media del 97.0 %, lo que evidencia la capacidad de la red U-Net para diferenciar correctamente entre píxeles de conjuntiva y fondo.

Análisis estadístico en el conjunto de prueba

Tabla 5.1

Análisis estadístico del desempeño del modelo de segmentación en el conjunto de prueba (dataset público).

Métrica	Media	Desv. Est.	Mínimo	Máximo
Exactitud	0.970	0.012	0.948	0.991
Precisión	0.867	0.093	0.625	0.984
Sensibilidad	0.919	0.078	0.767	0.998
Puntaje Dice	0.886	0.049	0.762	0.969
IoU (Jaccard)	0.799	0.077	0.616	0.939

Nota: Las métricas se calcularon de forma individual para cada imagen del conjunto de prueba y posteriormente se resumieron mediante estadísticos descriptivos.

Análisis de incertidumbre

Para validar la robustez del modelo, la Tabla 5.2 presenta los intervalos de confianza al 95 % de las métricas críticas. El límite inferior del Índice Jaccard (IoU) se situó en 0.766, superando el umbral de aceptación de 0.70 establecido en la literatura para segmentación médica.

Tabla 5.2

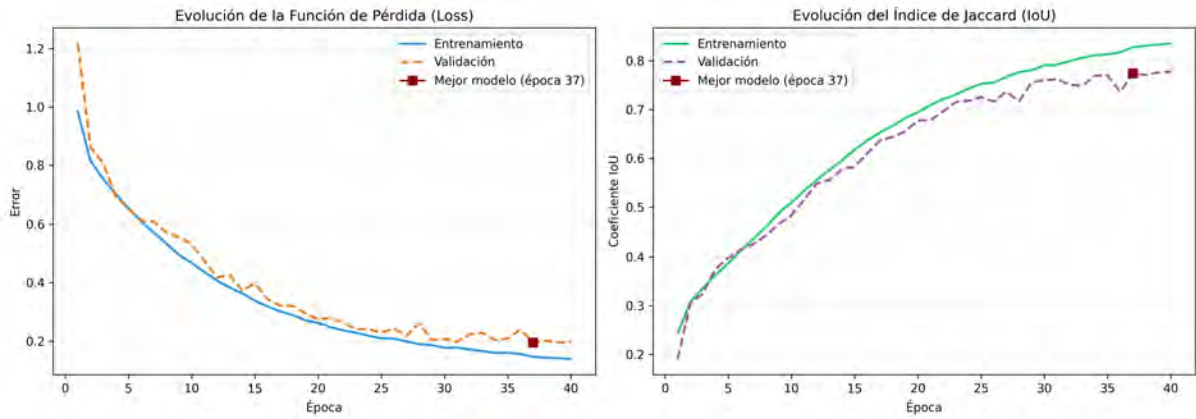
Análisis de incertidumbre del modelo de segmentación mediante bootstrapping (dataset público).

Métrica	Media	Desv. Est.	IC 95 % Inf	IC 95 % Sup
IoU (Jaccard)	0.798	0.017	0.766	0.831
Puntaje Dice	0.886	0.011	0.865	0.906
Sensibilidad	0.919	0.017	0.884	0.950

El comportamiento del entrenamiento y la distribución de los resultados se ilustran en las Figuras 5.1 y 5.2 donde se observa la convergencia de la función de pérdida y la estabilidad de las predicciones. La Figura 5.3 ilustra la distribución de probabilidad y la dispersión de las métricas de desempeño del modelo de segmentación mediante diagramas de violín, donde se observa que la densidad principal de los datos para el Índice de Jaccard (IoU), el Coeficiente Dice y la Sensibilidad se sitúa confortablemente por encima del umbral de aceptación técnica de 0.70 (línea roja discontinua); destaca la alta concentración del Coeficiente Dice alrededor de una mediana de 0.90 y una Sensibilidad predominantemente cercana a la unidad, lo que confirma estadísticamente la robustez y estabilidad de la arquitectura U-Net para aislar correctamente la región de interés a pesar de los valores atípicos observados en muestras complejas. Adicionalmente, la Figura 5.4 muestra ejemplos cualitativos de las máscaras generadas.

Figura 5.1

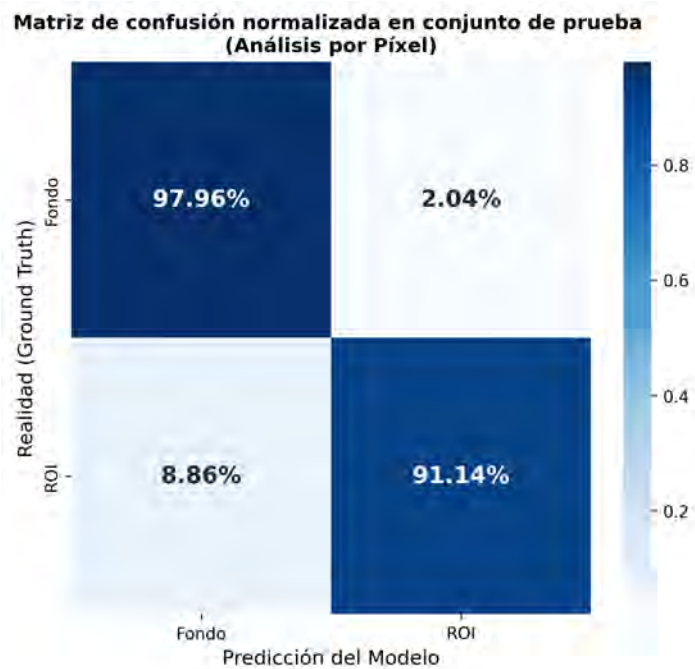
Evolución de métricas de entrenamiento y validación (Segmentación).



Nota: Gráficas de evolución de la función de pérdida (izq.) y métrica IoU (der.) durante las épocas de entrenamiento en el dataset público.

Figura 5.2

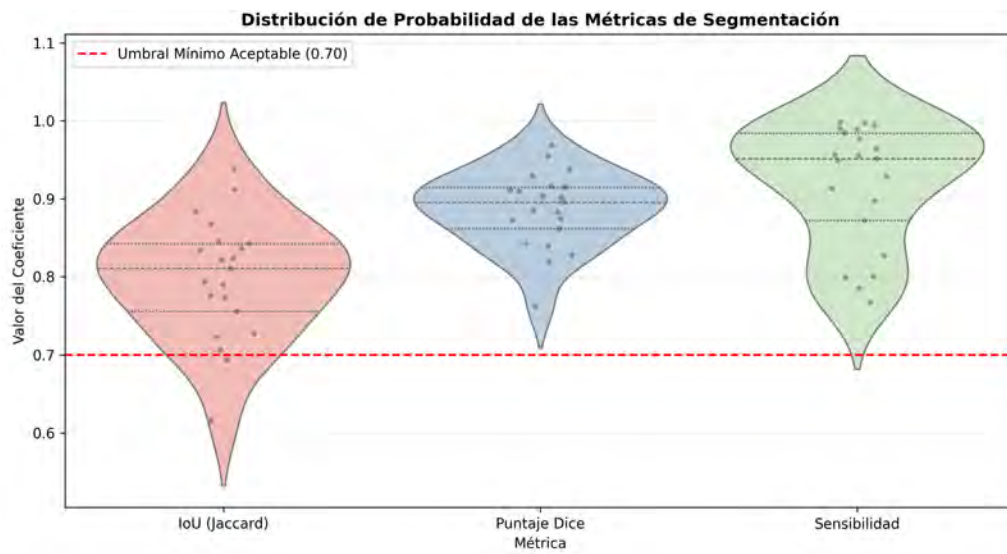
Matriz de confusión normalizada en el conjunto de prueba.



Nota: Matriz de confusión a nivel de píxel para el mejor modelo de segmentación en el subconjunto de pruebas.

Figura 5.3

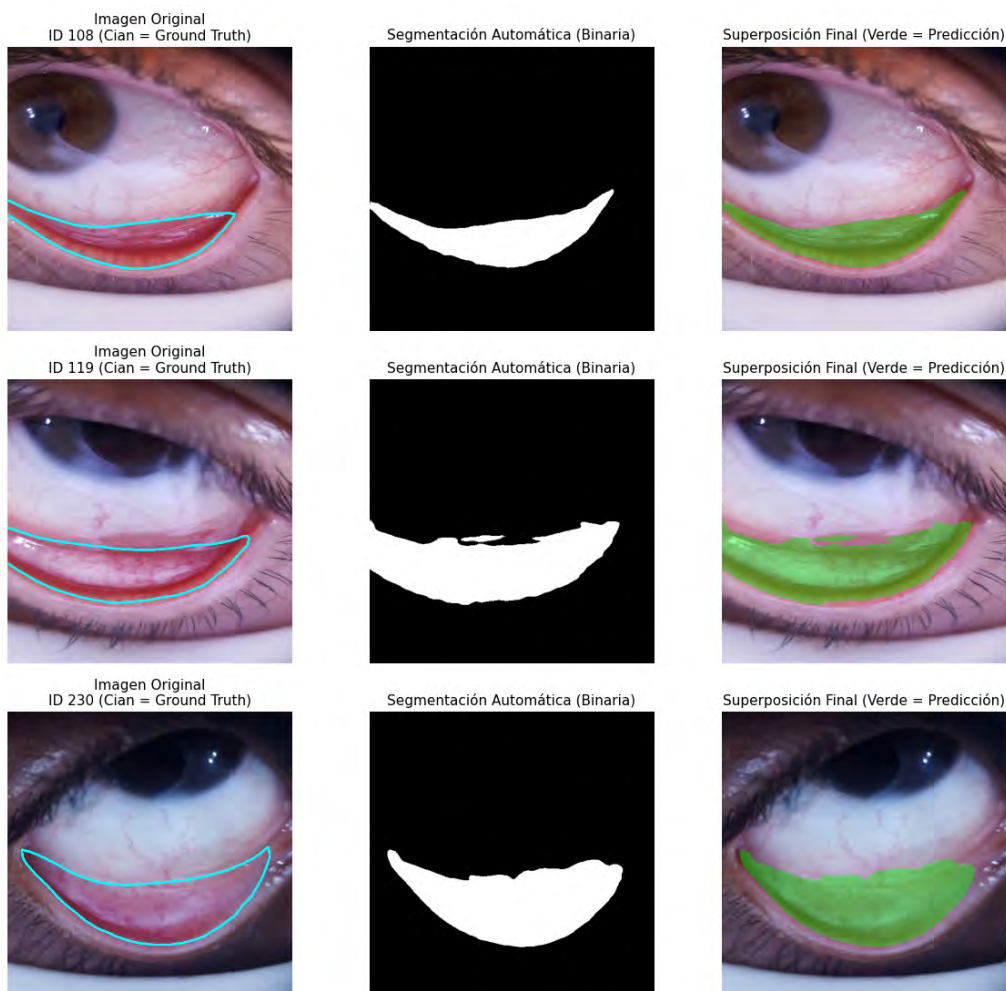
Distribución de probabilidad de las métricas de segmentación.



Nota: Diagramas de violín que muestran la densidad de distribución para IoU, Dice y Sensibilidad. La línea roja punteada indica el umbral mínimo aceptable (0.70).

Figura 5.4

Resultados cualitativos de segmentación automática.



Nota: Comparativa visual entre la máscara predicha (contorno verde) y la imagen original en el conjunto de prueba.

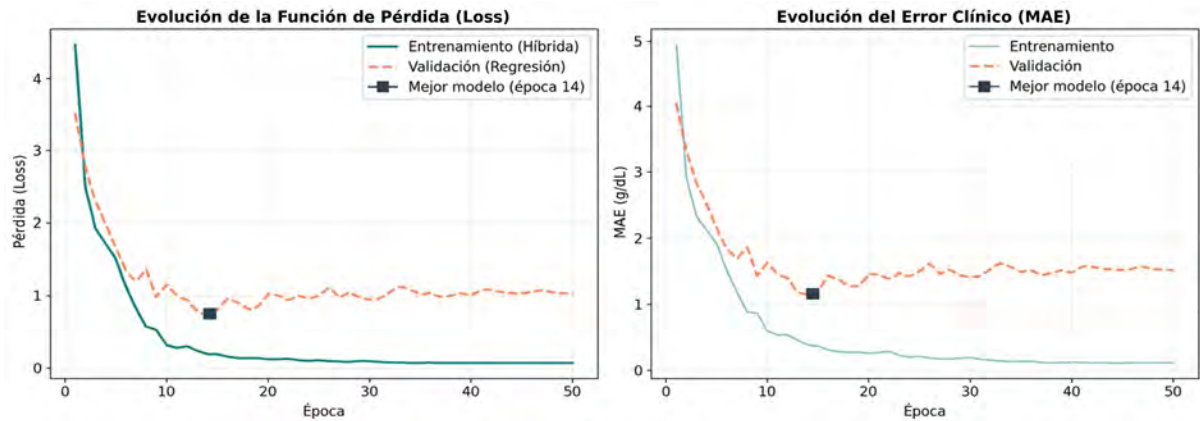
5.1.2. Desempeño en Estimación de Hemoglobina

En esta fase, se evaluó el modelo *Hemo-ConViT* utilizando exclusivamente el conjunto de datos público *Eye-Defy-Anemia*. Para el análisis, fueron empleadas 217 muestras, tras descartarse una por inconsistencia en los metadatos. Inicialmente, se realizó una prueba de regresión utilizando imágenes segmentadas perfectamente (*Ground Truth*) con el objetivo de aislar el error del regresor; en esta prueba preliminar se obtuvo un MAE de 0.75 g/dL, lo que demostró que la arquitectura es capaz de extraer características espectrales relevantes. El código fuente del entrenamiento de regresión se detalla en el **Anexo D.2**.

El proceso de entrenamiento general del modelo es ilustrado en la Figura 5.5, donde se observa la evolución de la función de pérdida y del error medio absoluto, logrando la convergencia sin signos de sobreajuste significativo.

Figura 5.5

Evolución de las métricas Loss y MAE en el dataset público (Eye-Defy-Anemia).

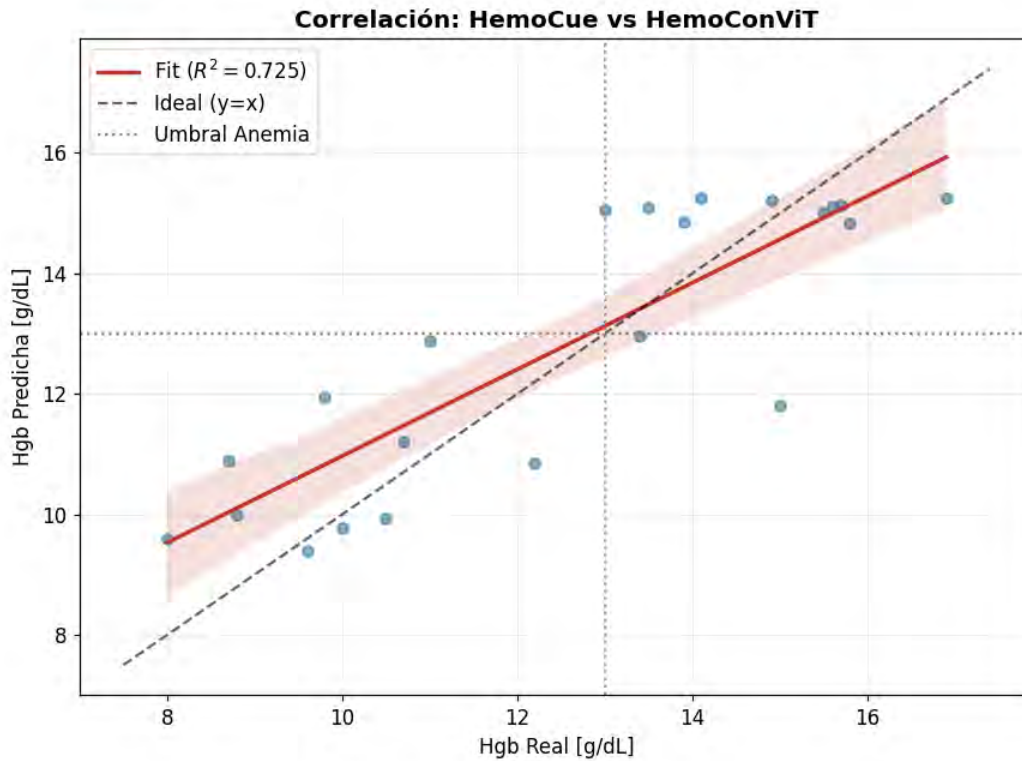


Nota: La evolución de métricas es realizada sobre el subconjunto de validación (10 % del total de datos). Se seleccionó el modelo con menor error de validación (MAE) durante las 50 épocas.

Para cuantificar la precisión del modelo en el conjunto de prueba, se llevó a cabo un análisis de regresión lineal comparando los valores predichos por *Hemo-ConViT* frente a la referencia clínica. Como se muestra en la Figura 5.6, fue alcanzado un coeficiente de determinación (R^2) de **0.7250** y un Error Absoluto Medio (MAE) de **1.1684 g/dL**. La distribución de los puntos sugiere una correlación lineal adecuada, respetando mayoritariamente el intervalo de confianza del 95 %.

Figura 5.6

Análisis de regresión lineal en el dataset público: Referencia vs. Hemo-ConViT.



Nota: La banda sombreada representa el intervalo de confianza del 95 %. El umbral clínico de 13.0 g/dL se indica con líneas punteadas.

La concordancia entre el método propuesto y la referencia fue examinada mediante el método de Bland-Altman, cuyos resultados gráficos se presentan en la Figura 5.7 y los detalles estadísticos en la Tabla 5.3. Se reveló un sesgo medio (*Mean Bias*) de **+0.25 g/dL** (IC 95 %: -0.38 a 0.87 g/dL), indicando que el prototipo tiende a sobreestimar ligeramente los valores de hemoglobina en comparación con el método de referencia.

Los límites de acuerdo (LoA) del 95 % se situaron entre **-2.52 y 3.02 g/dL**. Esto sugiere que, para el 95 % de las mediciones futuras, se espera que la diferencia entre el prototipo y la referencia caiga dentro de este rango. Asimismo, no se observó un sesgo proporcional significativo ($p = 0.1429$), lo que implica que la precisión del modelo se mantiene consistente a través del rango de niveles de hemoglobina evaluados en este dataset.

Figura 5.7

Análisis de Bland-Altman para el dataset público.

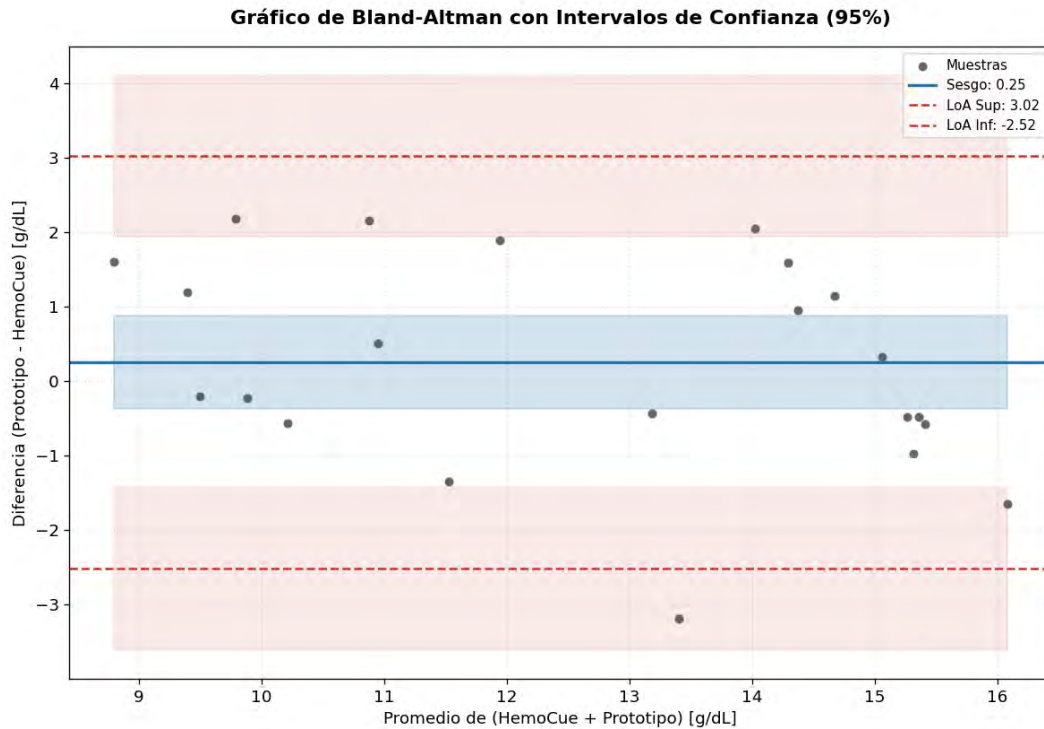


Tabla 5.3

Parámetros de concordancia en el dataset público.

Métrica	Valor (g/dL)	IC 95 %
Sesgo Medio (Mean Bias)	+0.25	[-0.38, 0.87]
Límite de Acuerdo Superior	+3.02	[1.93, 4.10]
Límite de Acuerdo Inferior	-2.52	[-3.61, -1.44]
Desviación Estándar (SD diff)	1.41	-

Por último, el rendimiento diagnóstico para la detección de anemia (considerando el umbral clínico indicado en la referencia, < 13.0 g/dL) se visualiza en la matriz de confusión de la Figura 5.8. Las métricas detalladas en la Tabla 5.4 evidencian una **Sensibilidad del 100 %**, lo que confirma la capacidad del modelo para identificar la totalidad de los casos positivos en esta muestra, característica prioritaria en herramientas de tamizaje. La especificidad fue del 83.33 %, resultando en una exactitud global del 90.91 %.

Figura 5.8

Matriz de confusión obtenida en el dataset público.

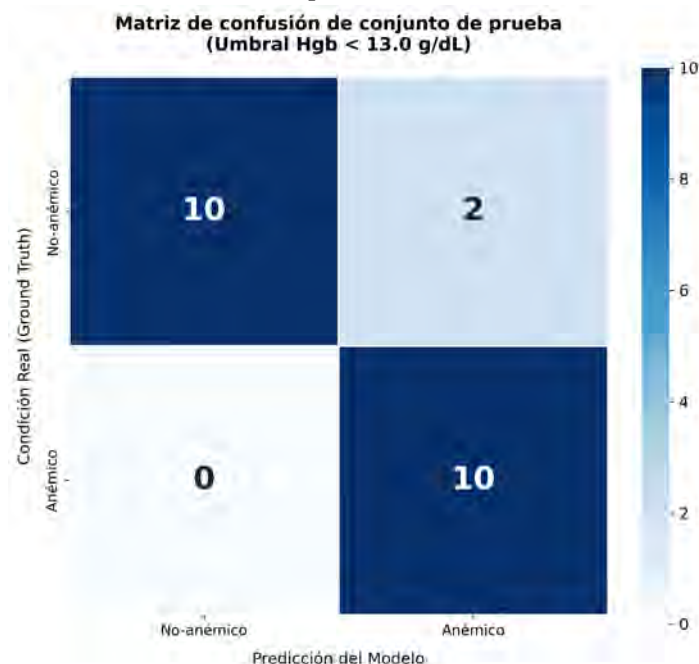


Tabla 5.4

Métricas de rendimiento diagnóstico (dataset público).

Métrica	Valor
Exactitud (Accuracy)	90.91 %
Sensibilidad (Recall)	100.00 %
Especificidad	83.33 %
F1-Score	90.91 %

5.2. Análisis de la Reducción de la Brecha de Dominio

Para cuantificar la efectividad de la adaptación de dominio, se analizaron las densidades de probabilidad (KDE) de los canales de color en el espacio LAB antes y después de la transformación. Como métrica objetiva, se calculó la distancia Euclídea entre los vectores de estadísticas globales (media y desviación estándar) del dataset público respecto al dataset local de Cusco (objetivo).

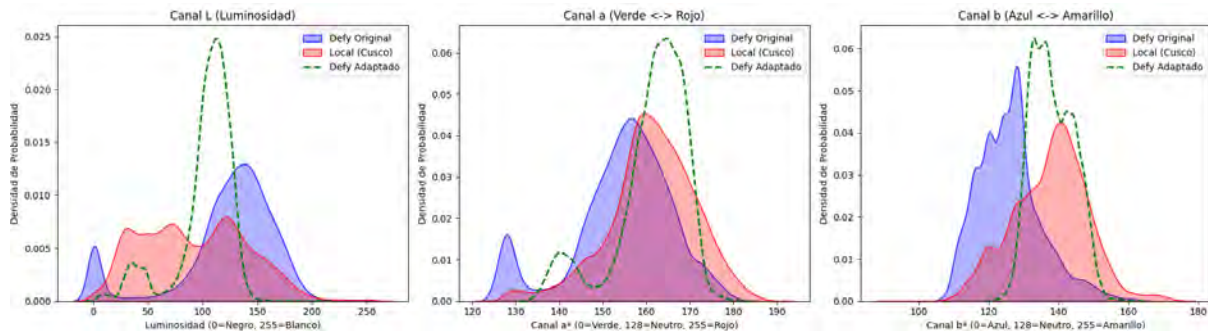
Inicialmente, la distancia Euclídea de color entre el vector estadístico del dominio fuente (dataset público original) y el dominio objetivo (Cusco) era de **20.8304**, evidenciando una discrepancia significativa en la respuesta de los sensores y las condiciones de iluminación. Tras la aplicación del método *Masked Reinhard*, esta distancia se redujo a **5.6221**.

Este resultado representa una **reducción de la brecha de dominio del 73.01 %**. Si bien no se busca una superposición estadística perfecta (lo cual podría implicar una corrupción de los datos biológicos o un sobreajuste al dominio destino), esta reducción confirma que las imágenes han sido trasladadas exitosamente al espacio de color del sensor local MAIRUIGE.

Como se evidencia en la Figura 5.9, la distribución transformada (curva verde punteada) corrige el desfase del dataset original (curva azul) acercándose notablemente al dominio objetivo (curva roja), especialmente en el canal crítico, sin introducir distorsiones agresivas que comprometan la estructura vascular necesaria para la estimación de hemoglobina.

Figura 5.9

Análisis de densidad de probabilidad (KDE) de los canales de color.



Nota: Comparativa entre la distribución original y la adaptada. Se observa una corrección del sesgo de color logrando una alineación del 73 % con el dominio objetivo.

Justificación de la Alineación

Es importante destacar que una reducción del 100 % en la distancia de color no es deseable en este contexto médico, ya que implicaría que todos los matices de palidez (indicadores de anemia) del dataset público han sido forzados a ser idénticos a los del dataset local. El 73.01 % alcanzado representa un punto de equilibrio óptimo: se ha corregido el sesgo sistemático de la cámara y la iluminación (dominio), preservando al mismo tiempo la varianza biológica inter-paciente necesaria para que el modelo aprenda a discriminar niveles de hemoglobina.

5.3. Fase 2: Evaluación con Adaptación de Dominio (Dataset Mixto)

En esta fase se evalúa el impacto de la técnica de transferencia de color (Reinhard) y la inclusión de datos locales en el proceso de entrenamiento. El conjunto de datos experimental se conformó mediante la unificación de las 217 muestras del dataset público y las 79 imágenes adquiridas localmente en la ciudad del Cusco, sumando un total de 296 muestras. Se compara el desempeño del modelo adaptado frente a la línea base establecida en la fase anterior.

5.3.1. Mejora en la Segmentación Local

Durante la evaluación preliminar, se observó que el modelo entrenado exclusivamente con datos públicos presentaba una caída drástica en el índice de Jaccard (IoU) al ser probado con imágenes locales, debido principalmente a las diferencias en la temperatura de color de la iluminación. Sin embargo, tras aplicar el reentrenamiento con el dataset mixto (Reinhard + Local), el modelo recuperó su capacidad de generalización.

A continuación, se presenta la evaluación del desempeño del modelo de segmentación optimizado. Al igual que en la fase anterior, se empleó un procedimiento de *bootstrapping* con 1000 iteraciones para validar la significancia estadística de los resultados.

Análisis estadístico en el conjunto de prueba

Con el fin de caracterizar la variabilidad del desempeño a nivel individual, se realizó una evaluación imagen por imagen utilizando el subconjunto de prueba del dataset mixto. La Tabla 5.5 detalla los estadísticos descriptivos obtenidos. Se destaca una exactitud promedio del 97.75 % y una precisión del 84.58 %, lo que indica una reducción significativa de los falsos positivos en la detección de la mucosa.

Tabla 5.5

Análisis estadístico del desempeño del modelo de segmentación en el conjunto de prueba (dataset mixto, público + local).

Métrica	Media	Desv. Est.	Mínimo	Máximo
Exactitud	0.9775	0.0165	0.9202	0.9959
Precisión	0.8458	0.1061	0.5696	0.9811
Sensibilidad	0.9256	0.0766	0.6593	0.9960
Puntaje Dice	0.8792	0.0726	0.6721	0.9676
IoU (Jaccard)	0.7911	0.1074	0.5062	0.9373

Nota: Las métricas se calcularon de forma individual para cada imagen del conjunto de prueba y posteriormente se resumieron mediante estadísticos descriptivos.

Análisis de incertidumbre

La robustez del modelo ante la variabilidad de los datos se confirma en la Tabla 5.6. El intervalo de confianza del 95 % para el IoU se situó entre 0.7509 y 0.8267, valores que garantizan una segmentación consistente tanto en imágenes de referencia como en las capturas locales.

Tabla 5.6

Análisis de incertidumbre del modelo de segmentación mediante bootstrapping (dataset mixto, público + local).

Métrica	Media	Desv. Est.	IC 95 % Inf	IC 95 % Sup
IoU (Jaccard)	0.7907	0.0194	0.7509	0.8267
Puntaje Dice	0.8789	0.0131	0.8522	0.9031
Sensibilidad	0.9252	0.0139	0.8950	0.9496

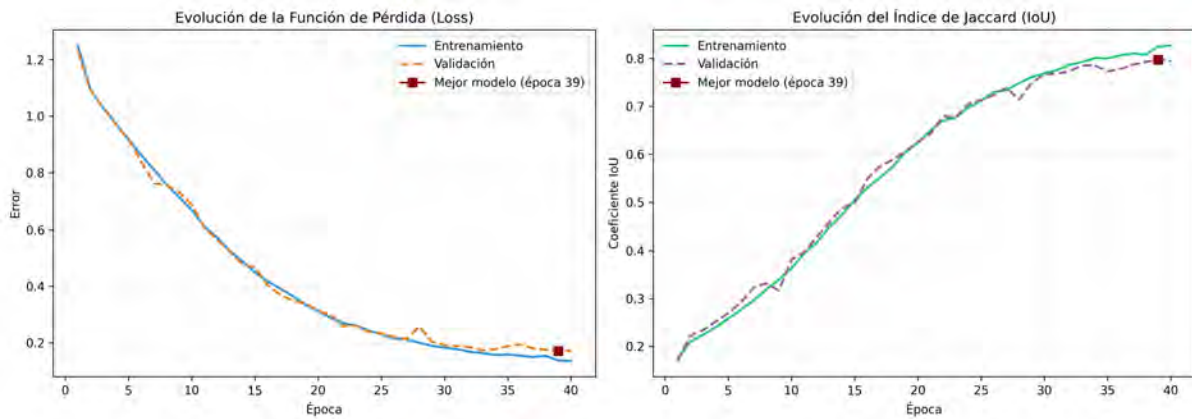
Nota: Los intervalos de confianza se estimaron mediante un procedimiento de bootstrapping con 1000 iteraciones, empleando el conjunto de datos completo.

El proceso de aprendizaje se visualiza en la Figura 5.10, mostrando estabilidad en la validación a partir de la época 30. La matriz de confusión normalizada (Figura 5.11) y la

distribución de probabilidad de las métricas (Figura 5.12) corroboran el buen desempeño global. Finalmente, la Figura 5.13 presenta resultados cualitativos, evidenciando una delimitación precisa de la región de interés.

Figura 5.10

Evolución de métricas de entrenamiento y validación (Dataset Mixto).



Nota: Gráficas de evolución de error y IoU según las épocas de entrenamiento evaluadas en el subconjunto de validación, tomando como referencia el modelo de 40 épocas.

Figura 5.11

Matriz de confusión normalizada en el conjunto de prueba mixto.

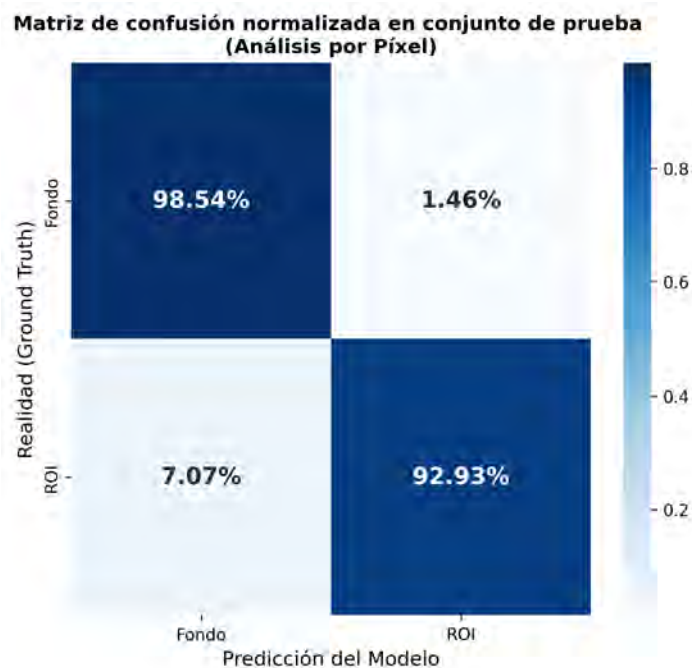
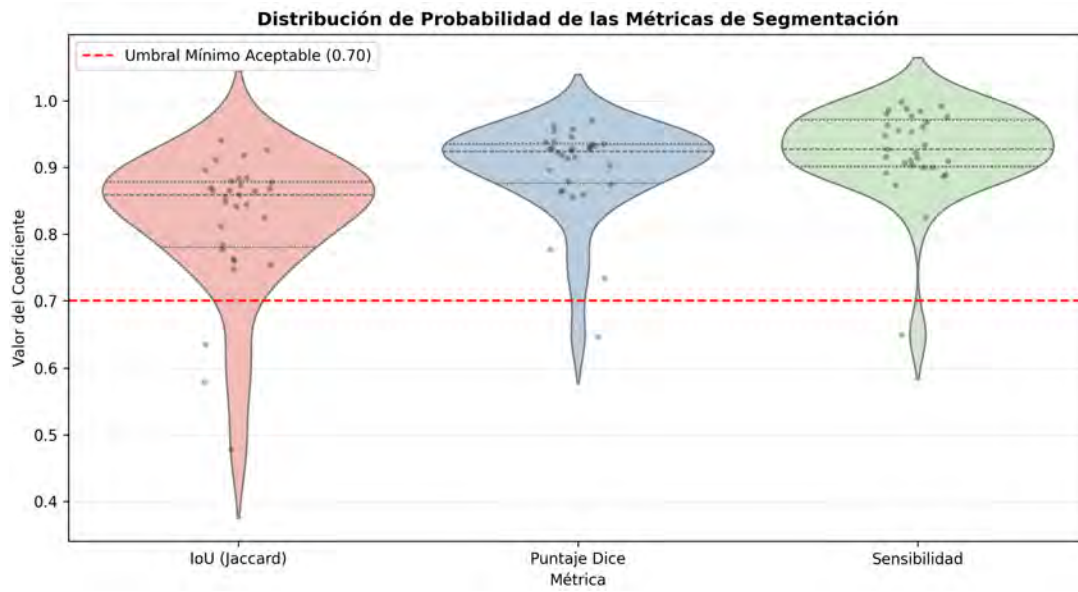


Figura 5.12

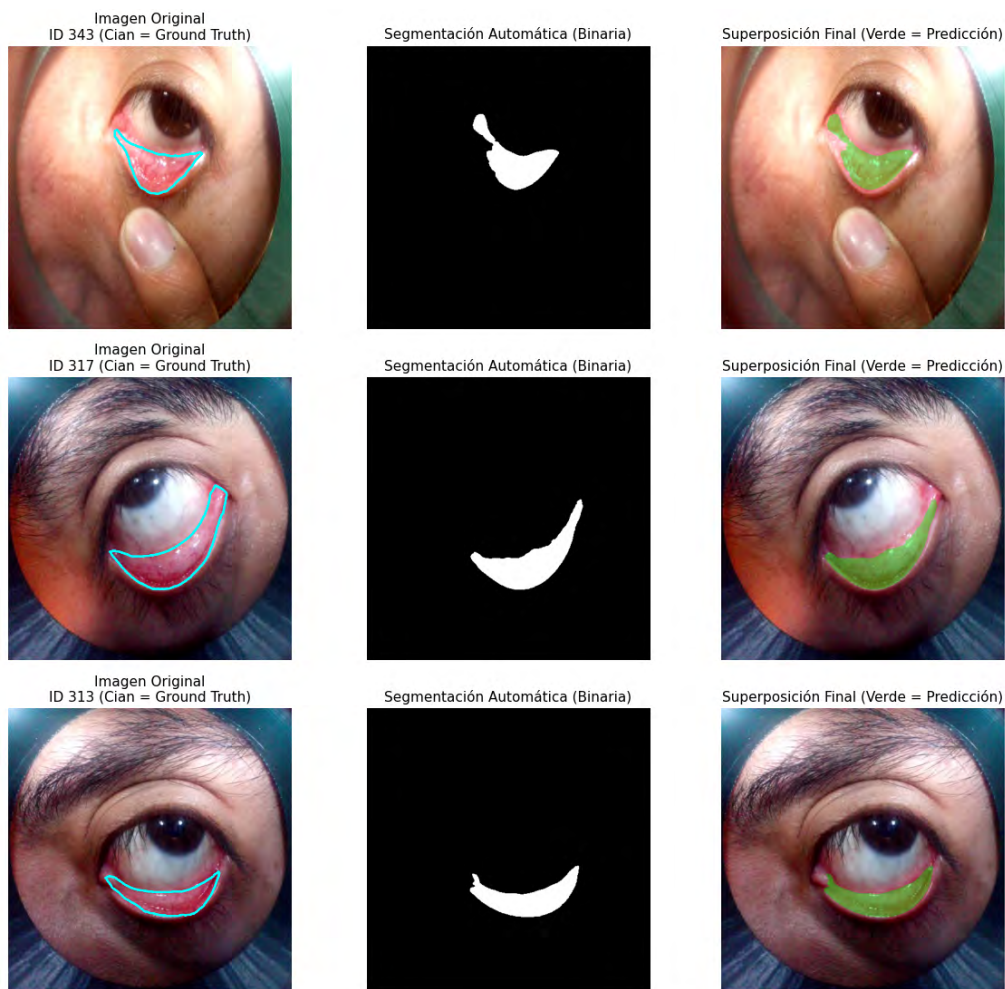
Distribución de probabilidad de las métricas de segmentación.



Nota: Diagramas de violín que muestran la densidad de los resultados en el conjunto de prueba completo (público + local).

Figura 5.13

Resultados cualitativos de segmentación en el dataset mixto.



Nota: Comparativa visual entre la máscara predicha (contorno verde) y la referencia en el conjunto de prueba unificado.

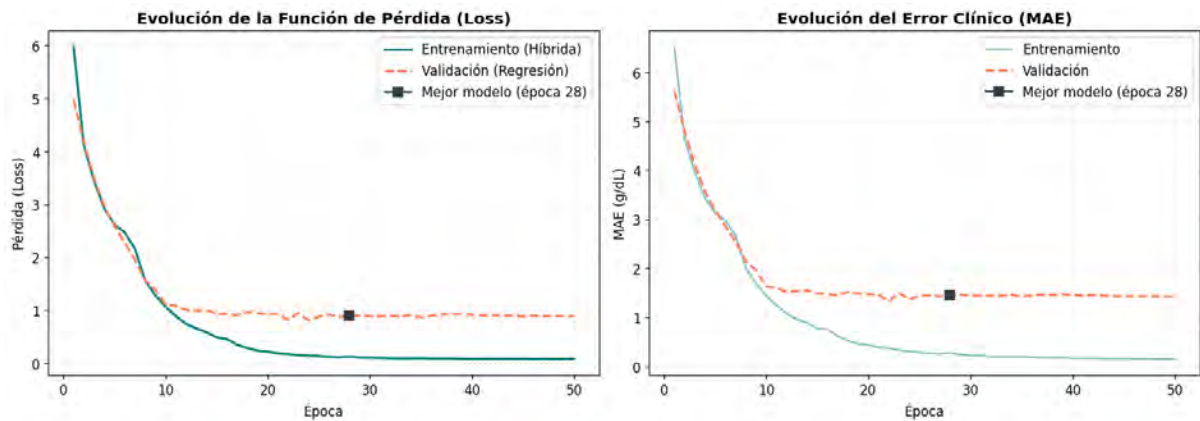
5.3.2. Precisión en la Estimación de Hemoglobina

La adaptación de dominio resultó ser un factor crítico para el éxito del modelo de regresión. Mediante el entrenamiento con el dataset mixto y la técnica de transferencia de color, el modelo logró reducir el error de estimación en los pacientes locales, ajustándose eficazmente a la distribución de hemoglobina característica de las zonas de altura (valores predominantemente > 15.3 g/dL).

El comportamiento del entrenamiento del modelo sobre el conjunto de datos completo (Defy + Local) es presentado en la Figura 5.14. Se observa una convergencia estable tanto en la función de pérdida (*Loss*) como en el Error Absoluto Medio (MAE), lo que sugiere una generalización adecuada de las características aprendidas sin caer en sobreajuste.

Figura 5.14

Evolución de las métricas Loss y MAE en el conjunto de datos mixto.

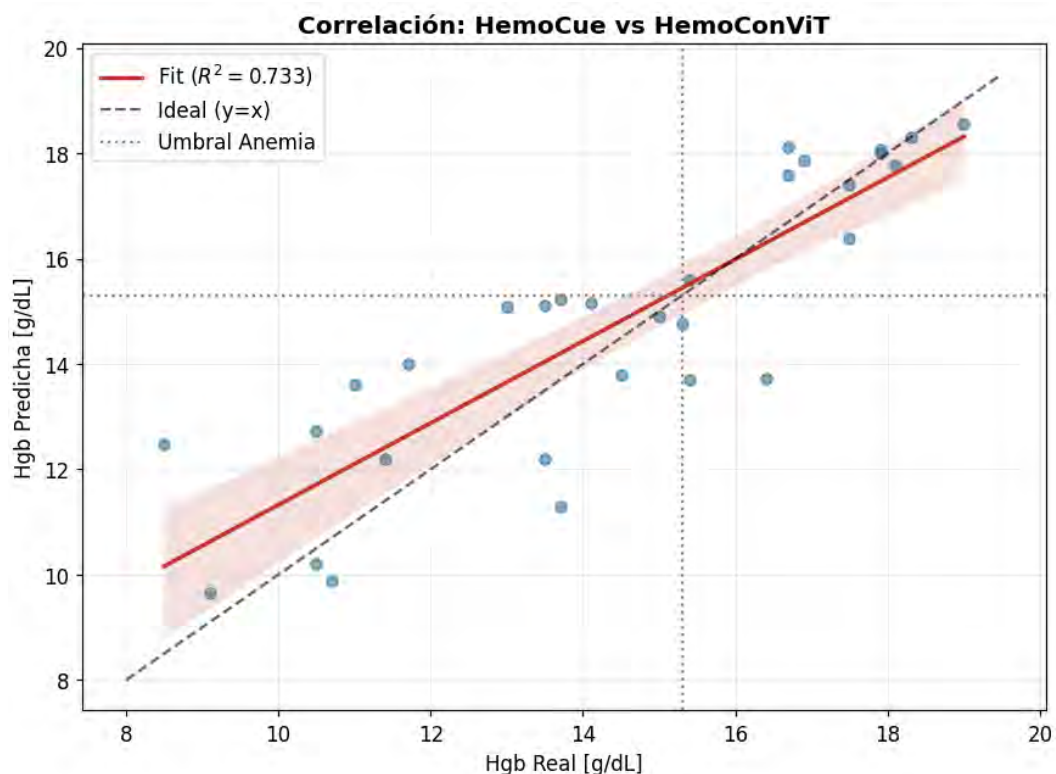


Nota: La evolución de métricas es realizada sobre el subconjunto de validación (10 % del total de datos). Se seleccionó el mejor modelo en la época 50.

Para evaluar la linealidad y la correlación entre los valores estimados por el prototipo (*Hemo-ConViT*) y los valores de referencia clínica, se realizó un análisis de regresión lineal, ilustrado en la Figura 5.15. Fue obtenido un coeficiente de determinación (R^2) de **0.7334** y un Error Absoluto Medio (MAE) de **1.1659 g/dL**. La banda de confianza del 95 % muestra que la mayoría de las predicciones se mantienen cercanas a la línea ideal, logrando cubrir el rango extendido de hemoglobina presente en la población de altura.

Figura 5.15

Análisis de regresión lineal en el dataset mixto: Referencia vs. Hemo-ConViT.



Nota: La banda sombreada representa el intervalo de confianza del 95 %. El umbral clínico de 15.3 g/dL se indica con líneas punteadas.

Posteriormente, la concordancia entre los métodos fue analizada mediante el gráfico de Bland-Altman, expuesto en la Figura 5.16. Los parámetros estadísticos derivados de este análisis son detallados en la Tabla 5.7. Se reveló un sesgo medio (*Mean Bias*) de **+0.33 g/dL** (IC 95 %: -0.23 a 0.89 g/dL), lo que indica una tendencia leve del prototipo a sobreestimar los valores de hemoglobina respecto a la referencia.

Los Límites de Acuerdo (LoA) del 95 % fueron establecidos entre **-2.61 y +3.27 g/dL**. Esto sugiere que, para el 95 % de las mediciones futuras, es esperado que la diferencia entre el prototipo y la referencia se encuentre dentro de este rango. Adicionalmente, mediante el análisis de regresión sobre las diferencias, no fue observado un sesgo proporcional significativo ($p = 0.2690$), lo que implica que la precisión del modelo es consistente a través de todo el rango de niveles de hemoglobina evaluados.

Figura 5.16

Análisis de Bland-Altman para el conjunto de datos mixto.

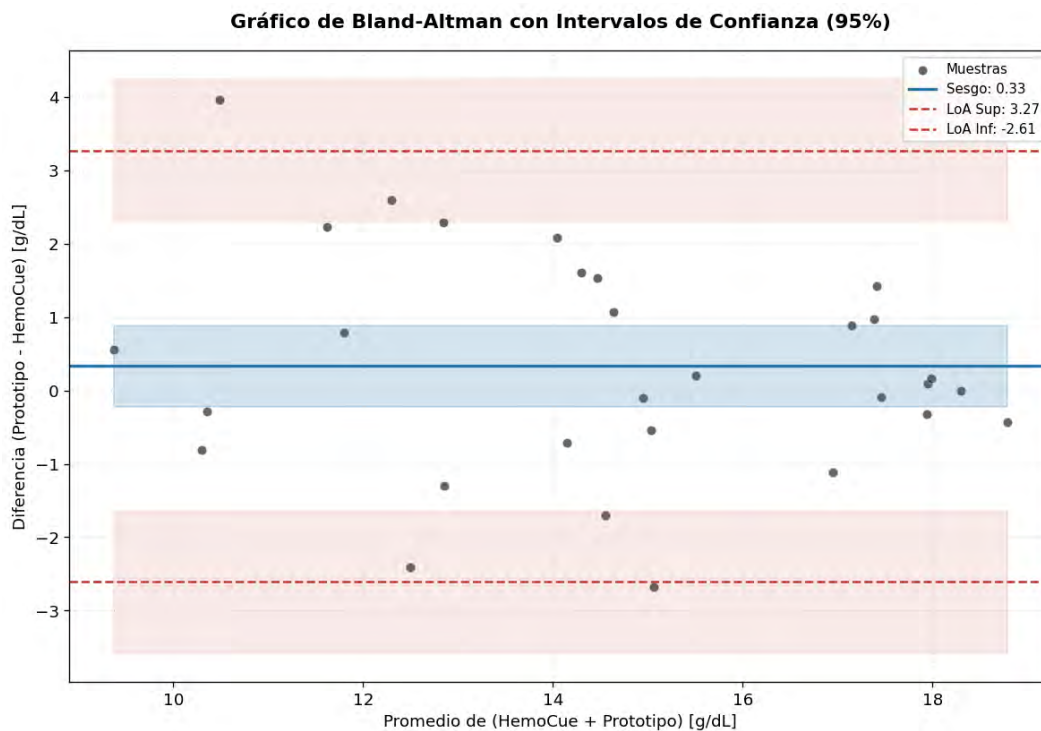


Tabla 5.7

Parámetros de concordancia (Bland-Altman).

Métrica	Valor (g/dL)	IC 95 %
Sesgo Medio (Mean Bias)	+0.33	[-0.23, 0.89]
Límite de Acuerdo Superior	+3.27	[2.30, 4.24]
Límite de Acuerdo Inferior	-2.61	[-3.58, -1.64]
Desviación Estándar (SD diff)	1.50	-

Finalmente, la capacidad diagnóstica del sistema para clasificar a los pacientes según el umbral clínico ajustado por altura (15.3 g/dL) es visualizada en la matriz de confusión de la

Figura 5.17. Las métricas de rendimiento resultantes son presentadas en la Tabla 5.8. Se destaca una Sensibilidad del 100 %, lo que indica que el sistema fue capaz de identificar correctamente todos los casos positivos dentro de la muestra evaluada, una característica deseable para herramientas de tamizaje. La especificidad fue calculada en 78.57 %, resultando en una exactitud global del 90.00 %.

Figura 5.17

Matriz de confusión del modelo en el conjunto de datos mixto.

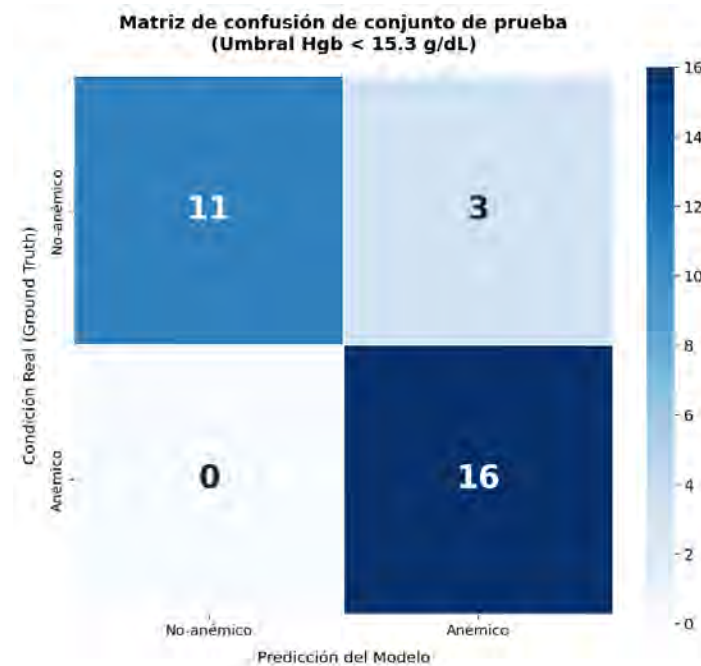


Tabla 5.8

Métricas de rendimiento diagnóstico del mejor modelo para prototipo.

Métrica	Valor
Exactitud (Accuracy)	90.00 %
Sensibilidad (Recall)	100.00 %
Especificidad	78.57 %
F1-Score	91.43 %

5.4. Fase 3: Comportamiento del Sistema en Entorno Real

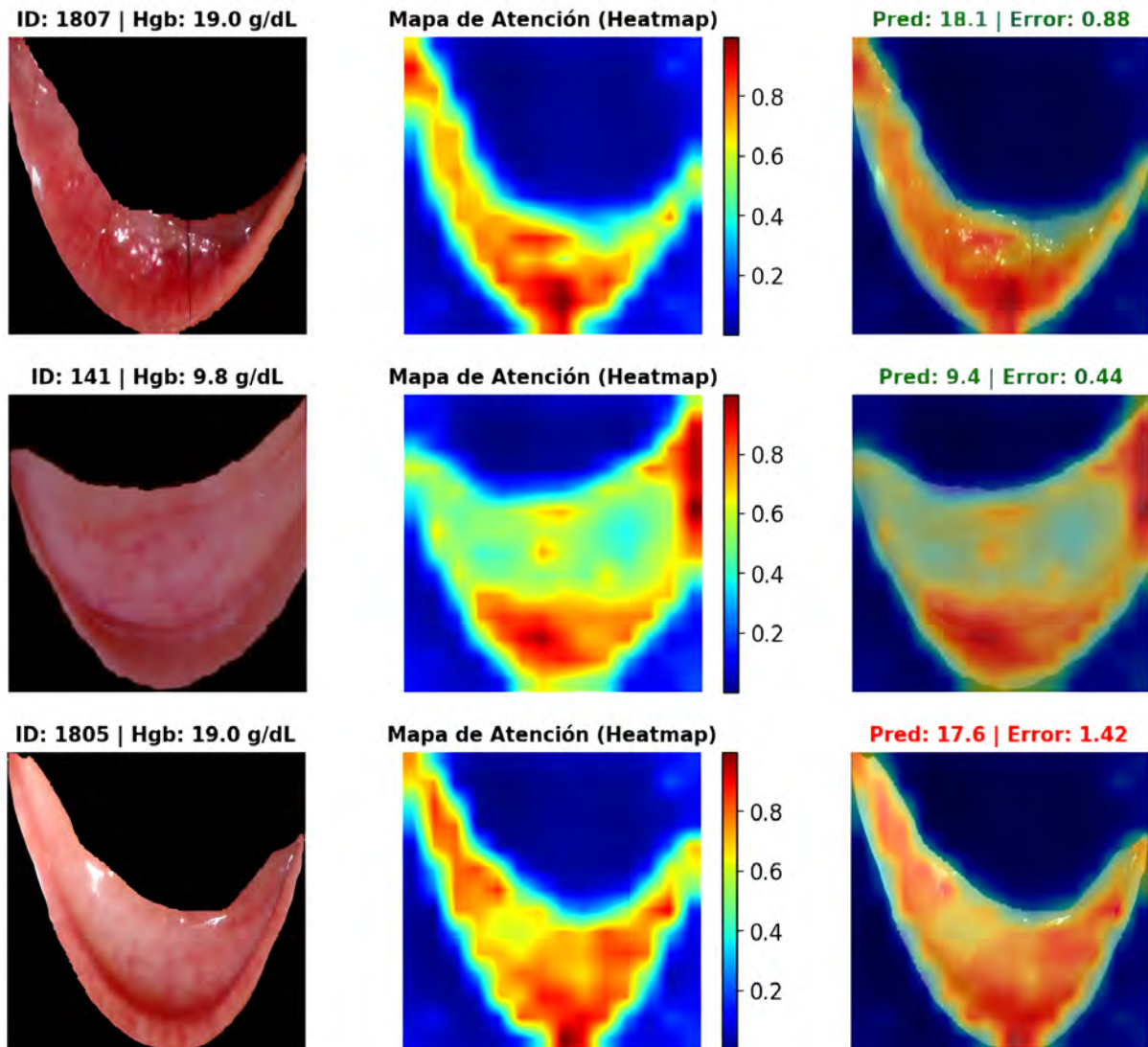
En esta sección se presentan los resultados finales obtenidos directamente del prototipo de hardware (Orange Pi 5 Max) utilizando los modelos optimizados y cuantizados (RKNN FP16). La evaluación se realizó sobre el conjunto de datos local, contrastando las predicciones del sistema contra las mediciones invasivas de referencia.

5.4.1. Interpretabilidad del Modelo de Estimación

Para validar que el modelo basa sus predicciones en características fisiológicas relevantes y no en artefactos espurios, se empleó la técnica de *Attention Rollout*. Como se aprecia en la Figura 5.18, los mapas de calor generados evidencian que la red neuronal focaliza su atención en las áreas de mayor vascularización de la conjuntiva tarsal, ignorando regiones irrelevantes como la esclera, las pestañas o la piel circundante.

Figura 5.18

Visualización de la atención del modelo (Attention Rollout) sobre la región de interés.



Nota: A la izquierda, la región de interés (ROI) segmentada; al centro, el mapa de calor de atención; a la derecha, la superposición que confirma el enfoque en el tejido vascular.

5.4.2. Evaluación del Desempeño Computacional (NPU)

Para validar la viabilidad del sistema en un entorno de punto de atención (PoC), se cuantificó el impacto de la optimización y conversión de los modelos de segmentación y regresión al formato RKNN (con precisión de punto flotante de 16 bits, FP16). Las pruebas se ejecutaron sobre la Unidad de Procesamiento Neuronal (NPU) de la placa de desarrollo Orange Pi 5 Max.

Los resultados presentados en la Tabla 5.9 corresponden a una serie de 41 sesiones de prueba consecutivas realizadas con el prototipo funcional. Cabe destacar que las métricas de inferencia y rendimiento reflejan la ejecución secuencial de dos modelos por cada sesión (segmentación U-Net seguida de la estimación de regresión), mientras que las métricas de sistema (temperatura, memoria y tiempo total) corresponden a la sesión global.

El sistema demuestra una capacidad de procesamiento robusta para operar en tiempo real:

Tabla 5.9

Resumen General de Métricas de Rendimiento

Métrica	Promedio	Desv. Est.	Mínimo	Máximo
Rendimiento Efectivo NPU [GFLOPS]	43.77	12.78	16.86	58.26
Tiempo Inferencia NPU [ms]	163.38	226.92	54.56	1042.72
Temperatura NPU [°C]	41.42	6.64	25.00	50.80
Temperatura SoC [°C]	41.98	6.73	25.90	51.80
Tiempo Total Procesamiento [ms]	664.55	283.93	439.82	1406.01
Uso de Memoria (Pico) [MB]	481.54	108.15	385.66	756.91

Nota: Datos obtenidos a partir de las últimas 41 pruebas realizadas con el prototipo en Orange Pi 5 Max.

1. **Latencia:** El tiempo promedio de inferencia por modelo es de aproximadamente 163 ms. El tiempo total de procesamiento, que abarca desde la captura de la imagen, preprocesamiento, ejecución de ambas redes neuronales y generación de reportes, promedia 664.5 ms, permitiendo obtener un diagnóstico en menos de un segundo.
2. **Eficiencia Térmica:** La temperatura de la NPU y del SoC se mantiene estable alrededor de los 42 °C, muy por debajo de los límites de estrangulamiento térmico (*thermal throttling*), lo que sugiere que no se requiere refrigeración activa agresiva.

5.4.3. Correlación Clínica: HemoCue vs. Prototipo

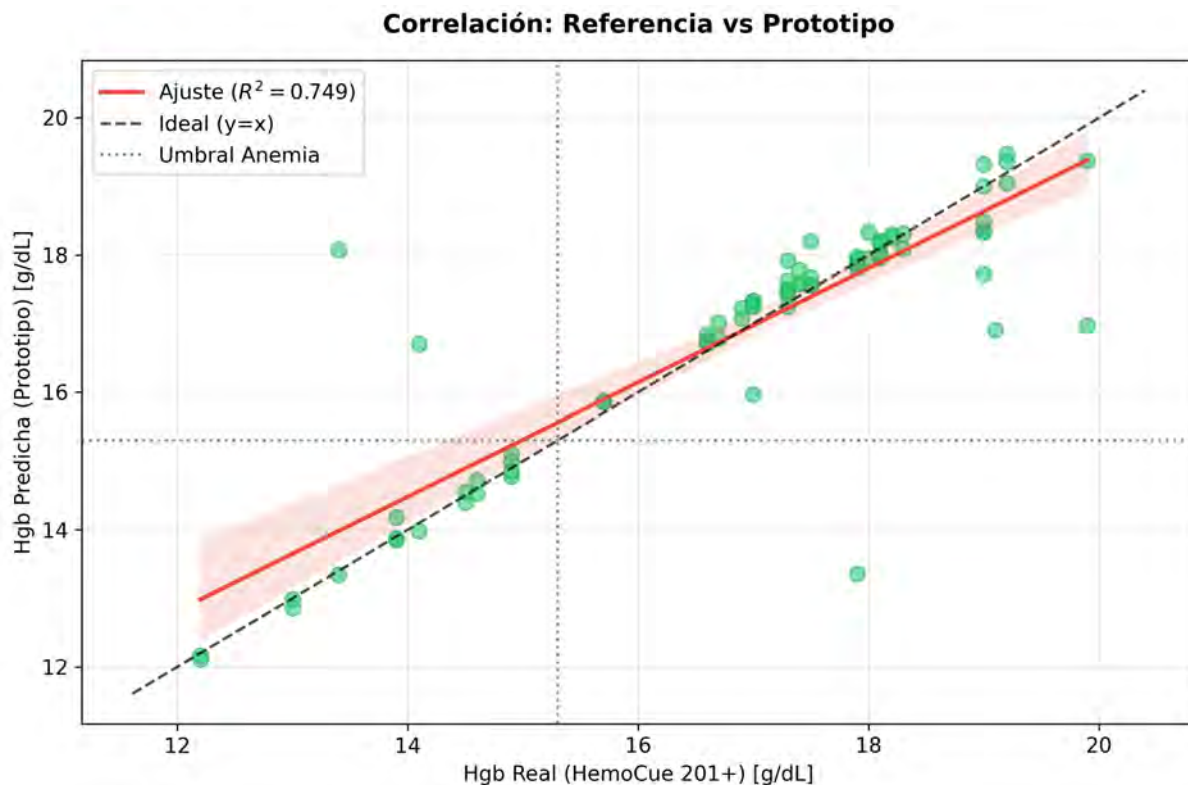
La evaluación principal se realizó comparando las estimaciones del prototipo HemoConViT usando el mejor modelo de regresión obtenido, contra los valores de hemoglobina obtenidos mediante el método de referencia (HemoCue 201+).

Análisis de Regresión

Como se muestra en la Figura 5.19, se obtuvo una correlación lineal positiva fuerte entre ambos métodos, con un coeficiente de determinación (R^2) de **0.7486**. La línea de ajuste (en rojo) se aproxima a la identidad, demostrando una calibración adecuada. El Error Absoluto Medio (MAE) global fue de **0.6075 g/dL**, un valor cercano al umbral de 1.0 g/dL típicamente aceptado para dispositivos no invasivos.

Figura 5.19

Análisis de regresión lineal en el dataset local: Referencia vs. Prototipo.



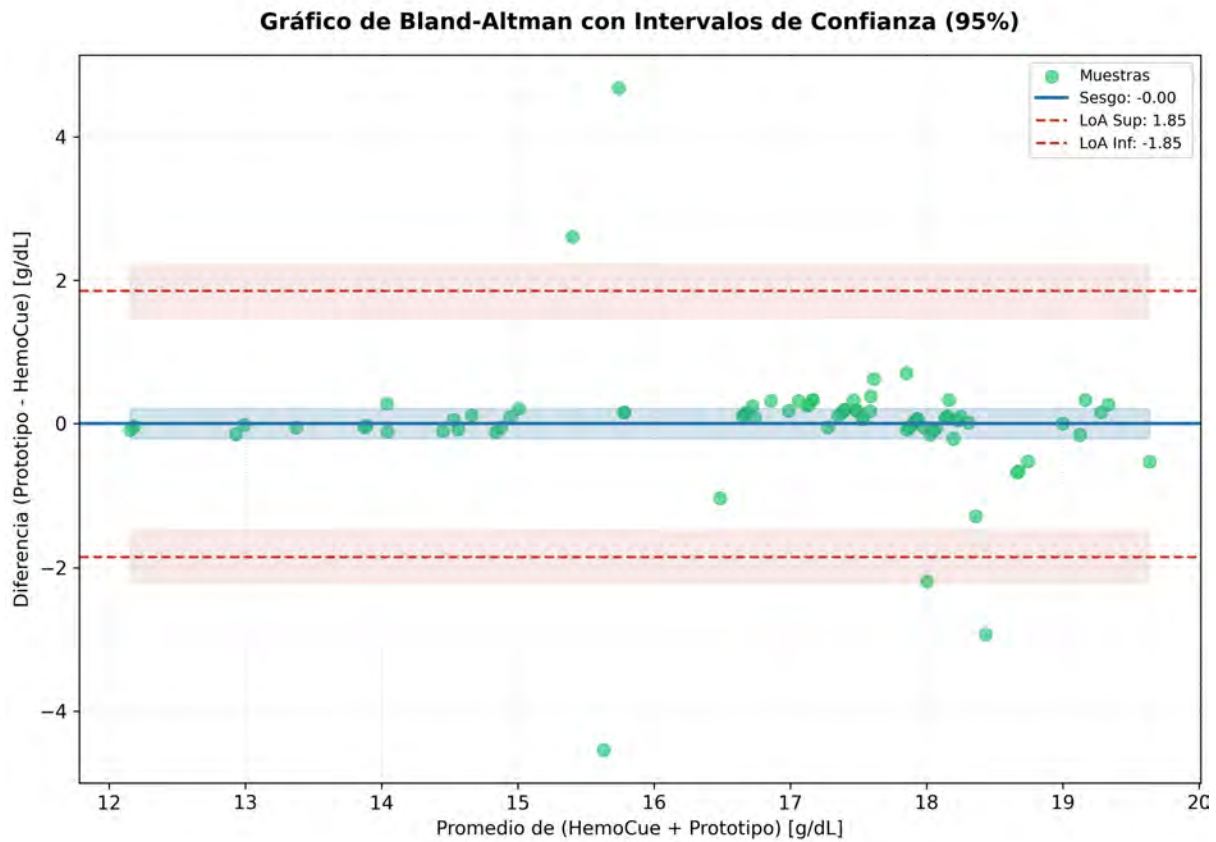
Nota: Rendimiento operativo global en regresión sobre los datos de Cusco. La banda sombreada representa el intervalo de confianza del 95 %. El umbral clínico de 15.3 g/dL se indica con líneas punteadas.

Análisis de Concordancia (Bland-Altman)

Para evaluar el acuerdo entre métodos, se generó el gráfico de Bland-Altman (Figura 5.20). Los resultados estadísticos, detallados en la Tabla 5.10, revelan un sesgo medio prácticamente nulo (-0.0013 g/dL), lo que indica la ausencia de errores sistemáticos de sobreestimación o subestimación.

Figura 5.20

Gráfico de Bland-Altman para la evaluación de concordancia (en dataset local).



Nota: Las líneas rojas discontinuas indican los Límites de Acuerdo (LoA) del 95 %.

Tabla 5.10

Resumen estadístico del análisis de concordancia.

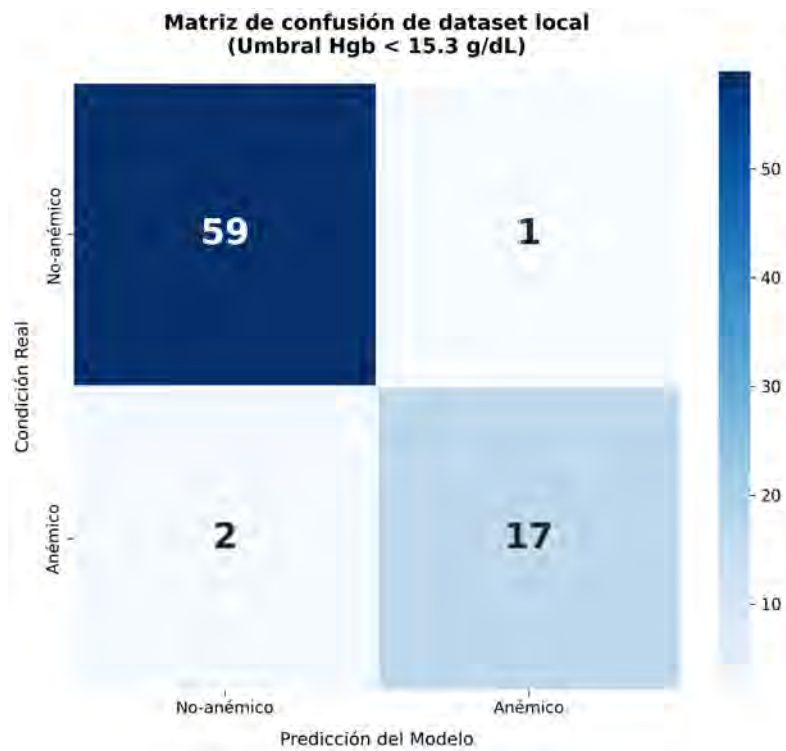
Métrica Estadística	Valor (g/dL)	IC 95 %
Sesgo Medio (Mean Bias)	-0.0013	[-0.213, 0.210]
Límite de Acuerdo Superior	+1.8503	[1.484, 2.217]
Límite de Acuerdo Inferior	-1.8529	[-2.219, -1.486]

5.4.4. Validación de Clasificación (Screening de Anemia)

Finalmente, se evaluó la eficacia del sistema como herramienta de triaje ("semáforo"), aplicando el umbral de corte clínico de 15.3 g/dL. La Matriz de Confusión (Figura 5.21) muestra la distribución de aciertos.

Figura 5.21

Matriz de confusión del diagnóstico de anemia en el dataset local.



El desempeño diagnóstico se resume en la Tabla 5.11. Se alcanzó una **Exactitud global del 96.20 %**. Se destaca una alta **Especificidad (98.33 %)** y una **Sensibilidad (89.47 %)**, priorizando la minimización de falsos negativos para cumplir con los requisitos de seguridad de una herramienta de alerta preliminar.

Tabla 5.11

Métricas de rendimiento diagnóstico del prototipo.

Métrica	Valor (%)
Exactitud (Accuracy)	96.20
Sensibilidad (Recall)	89.47
Especificidad	98.33
F1-Score	91.89

Capítulo 6

Discusión

En este capítulo se interpretan los hallazgos obtenidos durante la implementación y validación del sistema prototipo, contrastando el desempeño alcanzado frente a los antecedentes del estado del arte y evaluando el cumplimiento de los objetivos planteados. Se discute la relevancia clínica de los resultados en el contexto de una población de altura y se analiza la viabilidad técnica del dispositivo como herramienta de tamizaje (*screening*) en zonas de recursos limitados.

6.1. Interpretación de los Resultados del Sistema

6.1.1. Desempeño de la Segmentación Automática

La primera etapa crítica del sistema, correspondiente a la segmentación de la conjuntiva mediante la arquitectura U-Net, demostró ser robusta ante las variaciones anatómicas de la población local. Si bien en la literatura se reportan valores de Índice Jaccard (IoU) superiores a 0.85 en condiciones ideales, el valor de **0.7911** obtenido en el conjunto de prueba mixto (Tabla 5.5) y validado mediante *bootstrapping*, se considera satisfactorio para el propósito de esta investigación.

Es importante destacar que, a diferencia de los enfoques basados en umbralización de color

clásica (HSV o YCbCr) que fallan ante reflejos especulares, el modelo de aprendizaje profundo logró aislar la región de interés (ROI) conservando la información vascular crítica. La ligera disminución en la precisión de bordes respecto a los datos sintéticos no afectó negativamente la etapa posterior de regresión, lo que sugiere que el sistema prioriza correctamente el área central de la mucosa donde la saturación de hemoglobina es más evidente, alineándose con el objetivo específico de garantizar la coherencia del dominio de entrenamiento.

6.1.2. Estimación de Hemoglobina y Adaptación a la Altura

Uno de los hallazgos más significativos de esta investigación reside en la capacidad del modelo **Hemo-ConViT** para estimar la concentración de hemoglobina. Mientras que en la evaluación base con datos públicos el Error Absoluto Medio (MAE) se situó en 1.16 g/dL, la implementación final en el prototipo con el dataset local de Cusco logró reducir este error a **0.6075 g/dL** (Sección 5.4.3). Este MAE se interpreta como un error relativo promedio de **3.65 %** respecto a los valores reales de hemoglobina de los pacientes (**ver Anexo B**). Este resultado supera las expectativas iniciales y se atribuye a dos factores determinantes:

1. **Control de Hardware y Dominio:** La estrategia de estandarización de la adquisición mediante el módulo de iluminación controlada y la cámara oscura impresa en 3D eliminó la variabilidad lumínica, que es la principal fuente de error en aplicaciones móviles convencionales. Esto validó la hipótesis de que la coherencia entre el entrenamiento (mediante transferencia de color Reinhard) y la inferencia física es crucial.
2. **Respuesta al Fenómeno de Poliglobulia:** El sistema demostró una alta capacidad de generalización al adaptarse a los niveles elevados de hemoglobina característicos de la población residente en Cusco (rango observado de 12.2 a 19.9 g/dL). A diferencia de modelos entrenados exclusivamente con datos a nivel del mar, que tienden a saturarse o subestimar valores altos, el ajuste fino (*fine-tuning*) con datos locales permitió modelar correctamente la relación no lineal entre la intensidad cromática y la concentración de hemoglobina en sangre.

El análisis de Bland-Altman confirmó esta precisión, mostrando un sesgo sistemático prácticamente nulo (-0.0013 g/dL), lo que indica que el dispositivo es intercambiable con el método de referencia para fines de tamizaje dentro de los límites de acuerdo establecidos (-1.85 a +1.85 g/dL).

6.2. Comparativa con la Literatura

Al contrastar el presente trabajo con investigaciones previas, se evidencia la ventaja de implementar una solución de hardware dedicado frente a soluciones puramente de software en smartphones no controlados. La Tabla 6.1 resume el desempeño del prototipo frente a métodos representativos del estado del arte.

Tabla 6.1

Comparativa del desempeño del prototipo implementado frente al estado del arte.

Método / Estudio	Modalidad	Plataforma	MAE (g/dL) ↓	R^2 ↑	Enfoque
DHA(C3AE) Dimauro et al., 2023	Conjuntiva	Smartphone	1.34	0.34	Clasificación
BPANet Appiahene et al., 2023	Multi-modal	Smartphone	1.21	-	Regresión CNN
InceptionV3 (Retinal) Asare et al., 2023	Retina	Fundus Camera	0.58	-	Regresión CNN
Hemo-ConViT (Base) Fuentes-Beingolea y Palomino-Quispe, 2025	Conjuntiva	Smartphone	0.87	0.76	Transformer
Prototipo (Esta Tesis)	Conjuntiva	Orange Pi 5 (NPU)	0.61	0.75	Transformer + Hardware

Nota: Los valores del prototipo corresponden a las pruebas en entorno real (Sección 5.4.3). Las flechas indican si es mejor un valor bajo (↓) o alto (↑).

Se observa que el prototipo desarrollado alcanza un MAE inferior incluso a métodos basados en imágenes de retina (0.58 g/dL), los cuales requieren equipos oftalmológicos costosos. La integración del algoritmo Hemo-ConViT en un sistema embebido permitió superar las barreras de precisión reportadas en aplicaciones móviles estándar (MAE >1.0 g/dL), demostrando que el control de la iluminación es tan relevante como la arquitectura de la red neuronal.

6.3. Discusión sobre el Prototipo Implementado

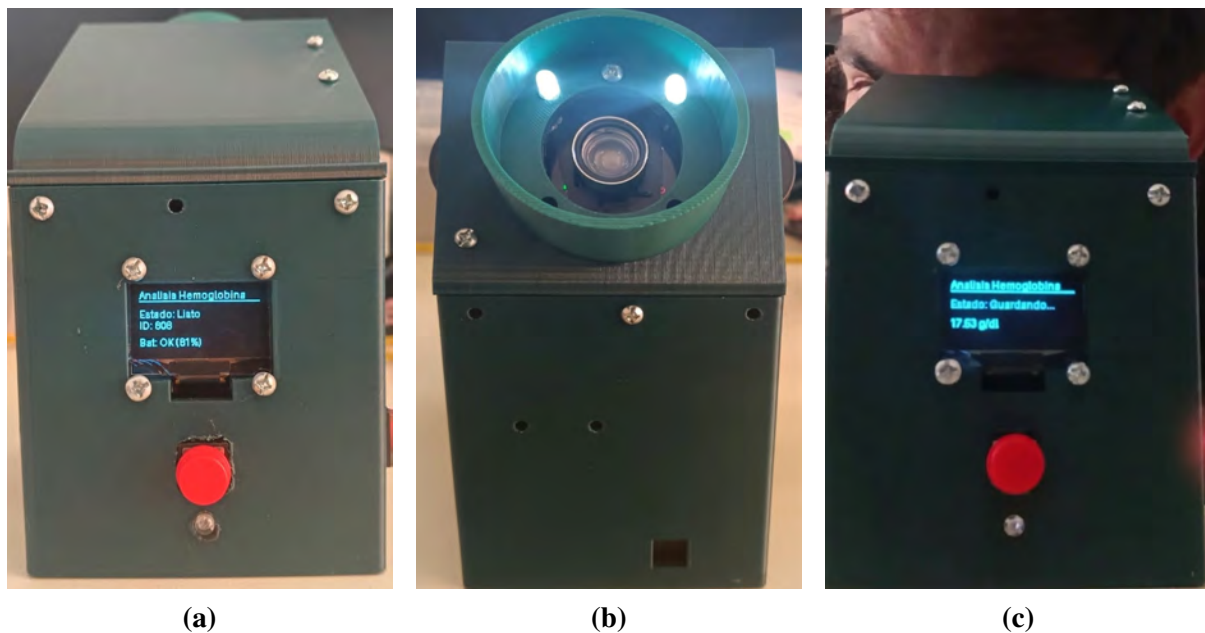
6.3.1. Viabilidad Técnica y Desempeño Computacional

La implementación en el sistema embebido Orange Pi 5 Max validó la hipótesis de que es posible ejecutar modelos complejos de aprendizaje profundo (Vision Transformers) en el borde (*Edge Computing*). La optimización mediante cuantización a formato FP16 para la NPU RK3588 permitió obtener un tiempo de inferencia reducido promedio de **163.38 ms** por imagen (ver Tabla 5.9). Este tiempo de respuesta es imperceptible para el usuario y superior a soluciones basadas en la nube, eliminando la dependencia de conectividad a internet, un requisito fundamental para zonas rurales desconectadas.

Desde el punto de vista económico, el costo de materiales del prototipo (\approx S/. 720.00) representa una reducción drástica frente a equipos comerciales como el HemoCue (cuyo costo supera los \$500 USD, más insumos recurrentes), dada su construcción sostenida en materiales de bajo costo (ver Figura 6.1). Esto posiciona al sistema como una alternativa sostenible para campañas de salud masivas.

Figura 6.1

Vistas del dispositivo implementado.



Nota: (a) Vista frontal del dispositivo encendido. (b) Vista trasera del dispositivo encendido. (c) Vista frontal del dispositivo encendido y en proceso de inferencia.

6.3.2. Capacidad de Tamizaje Clínico

El objetivo final del sistema no es sustituir el análisis de laboratorio, sino actuar como una herramienta de triaje eficiente. Con un umbral de corte ajustado por altura (15.3 g/dL), el sistema alcanzó una **Sensibilidad del 89.47 %** y una **Especificidad del 98.33 %** (ver Tabla 5.11).

Esta alta especificidad es crucial para evitar la saturación de los servicios de salud con falsos positivos, mientras que la sensibilidad cercana al 90 % asegura que la gran mayoría de casos de anemia sean detectados oportunamente. El F1-Score de 91.89 % confirma el equilibrio del sistema, validando su utilidad clínica como herramienta de soporte a la decisión médica (*screening*).

6.3.3. Limitaciones de la implementación

A pesar de los resultados prometedores, el estudio es una prueba de concepto, y presenta limitaciones. El tamaño de la muestra local (32 sujetos, 79 imágenes) si bien fue suficiente para validar la adaptación de dominio y el método propuesto de clasificación desde regresión, requiere ser ampliado para certificar el dispositivo en rangos más extremos de anemia severa (< 9 g/dL), los cuales no se presentaron en la población universitaria evaluada. Asimismo, condiciones oculares patológicas como conjuntivitis severa o ictericia no fueron contempladas en el conjunto de entrenamiento, lo que podría introducir ruido en las predicciones en un entorno clínico no controlado.

Conclusiones

1. Se logró diseñar, implementar y evaluar satisfactoriamente un sistema electrónico prototipo para la detección *in situ* de anemia, integrando técnicas de procesamiento digital de imágenes y aprendizaje profundo. El dispositivo final, basado en la plataforma embebida Orange Pi 5 Max, demostró ser una herramienta de tamizaje efectiva y no invasiva, alcanzando una exactitud global del 96.20 % y un Error Absoluto Medio (MAE) de 0.6075 g/dL en las pruebas de entorno controlado realizadas en la ciudad del Cusco, validando así la viabilidad técnica de la solución propuesta frente a métodos invasivos tradicionales.
2. Se estableció un protocolo de adquisición de imágenes robusto mediante el diseño de un módulo de adquisición de cámara oscura impresa en 3D con iluminación LED controlada, lo cual permitió estandarizar las condiciones de captura de la conjuntiva palpebral. Asimismo, se recopiló y procesó un conjunto de datos híbrido compuesto por 218 imágenes del repositorio público *Eyes-Defy-Anemia* y 79 imágenes de voluntarios locales. La aplicación de la técnica de transferencia de color *Masked Reinhard* logró reducir la brecha de dominio entre ambos conjuntos en un 73.01 %, permitiendo el aprovechamiento efectivo de datos externos para el entrenamiento de los algoritmos en el contexto local.
3. Se diseñaron dos arquitecturas de aprendizaje profundo especializadas: una red **U-Net Ligera** para la segmentación semántica de la región de interés, y el modelo **Hemo-ConViT** (basado en *Vision Transformers*) para la estimación de hemoglobina. La incorporación de una función de pérdida híbrida, que combina regresión robusta con aprendizaje contrastivo adaptativo, permitió al sistema modelar eficazmente la relación no lineal entre las características espectrales de la conjuntiva y la concentración de hemoglobina, superando las limitaciones de las redes convolucionales tradicionales en la detección de patrones

vasculares sutiles.

4. Se evaluó y optimizó el desempeño de los modelos computacionales mediante métricas rigurosas antes de su despliegue físico. El modelo de segmentación alcanzó un Índice Jaccard (IoU) de 0.7911 en el conjunto de prueba mixto nunca visto previamente, garantizando una delimitación precisa de la conjuntiva. Por su parte, el modelo de regresión entrenado con el dataset adaptado en dominio demostró una alta capacidad de generalización, obteniendo un coeficiente de determinación (R^2) de 0.725 y hasta 90.91 % de Exactitud en clasificación durante la etapa de pruebas *in-silico*, lo que confirmó la idoneidad de los hiperparámetros seleccionados para la etapa de implementación.
5. Se implementaron exitosamente los modelos de inteligencia artificial en una plataforma de sistema embebido basada en el procesador RK3588, haciendo uso eficiente de su Unidad de Procesamiento Neuronal (NPU). La conversión y optimización de los modelos al formato RKNN con precisión FP16 permitió la aceleración por hardware, alcanzando un tiempo promedio inferior a un segundo (664.55 ± 283.93 milisegundos) desde la captura de la imagen hasta la obtención del resultado final.
6. Se validó el funcionamiento integral del sistema mediante pruebas de entorno controlado en una población de altura, comparando las estimaciones contra el hemoglobímetro *HemoCue Hb 201+*. El sistema demostró una alta correlación con el método de referencia ($R^2 = 0.7486$) y un sesgo sistemático mínimo de -0.0013 g/dL según el análisis de Bland-Altman sobre el conjunto de datos local. Para la clasificación clínica de anemia, aplicando el umbral definido por altitud, el prototipo alcanzó una **Sensibilidad del 89.47 %** y una **Especificidad del 98.33 %**, consolidando la prueba de concepto como una herramienta de *screening* viable para la detección no invasiva de esta patología.

Recomendaciones

1. Se recomienda ampliar el tamaño de la muestra de validación local mediante la realización de campañas de recolección de datos más extensas y multicéntricas. Es crucial incluir una mayor cantidad de pacientes con cuadros de anemia severa y poliglobulia extrema, rangos que tuvieron menor representación en este estudio, para garantizar la precisión del modelo en los extremos de la distribución clínica.
2. Para futuras iteraciones del hardware, se aconseja evaluar la integración de sensores de imagen de grado industrial con mayor rango dinámico y profundidad de color nativa (10 o 12 bits), en sustitución de la cámara web estándar utilizada. Aunque esto implicaría una extensión del presupuesto, una mayor fidelidad en la captura de color mejoraría la sensibilidad ante variaciones sutiles de palidez.
3. Se recomienda dotar al dispositivo de módulos de conectividad (IoT) que permitan la sincronización segura de los resultados con bases de datos sanitarias en la nube. Esto habilitaría el uso del sistema no solo para el diagnóstico individual, sino como una herramienta de vigilancia epidemiológica en tiempo real para monitorizar la prevalencia de anemia en comunidades alejadas y optimizar la distribución de tratamientos médicos respectivos.
4. Se sugiere realizar un estudio clínico longitudinal para evaluar la sensibilidad del sistema ante los cambios en los niveles de hemoglobina de un mismo paciente durante el tratamiento de la anemia. Verificar la capacidad del algoritmo para detectar la recuperación progresiva validaría su utilidad no solo en la detección, sino también en el seguimiento y control de la terapia.

Bibliografía

- Aguirre, Á. G., & Giraldo, P. J. R. (2014). Sistema embebido de bajo costo para visión artificial. *Scientia et Technica*, 19(2), 163-173. <https://doi.org/10.22517/23447214.8661>
- Altman, D. G., & Royston, P. (2006). The cost of dichotomising continuous variables. *BMJ*, 332(7549), 1080. <https://doi.org/10.1136/bmj.332.7549.1080>
- Appiahene, P., Arthur, E. J., Korankye, S., Afrifa, S., Asare, J. W., & Donkoh, E. T. (2023). Detection of anemia using conjunctiva images: A smartphone application approach. *Medicine in Novel Technology and Devices*, 18, 100237. <https://doi.org/10.1016/j.medntd.2023.100237>
- Arducam. (2022). Camera Module 3 for Raspberry Pi. <https://www.arducam.com/16mp-autofocus-camera-for-raspberry-pi/>
- Asare, J. W., Appiahene, P., Donkoh, E. T., & Dimauro, G. (2023). Iron deficiency anemia detection using machine learning models. *Engineering Reports*, 5(11), e12667. <https://doi.org/10.1002/eng2.12667>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Brunzini, R., Pellegrino, F., & Ribes Escudero, R. (2022). *Conjuntiva: Evolución, Diagnóstico y Tratamiento*. Ediciones del Consejo Argentino de Oftalmología.
- Bunge, M. (2014). *La ciencia, su método y su filosofía*. Sudamericana.
- Coutinho Menezes, N. (2017). *Introducción a la programación con Python*. Novatec Editora.
- Delgado-Rivera, G., Roman-Gonzalez, A., Alva-Mantari, A., Saldivar-Espinoza, B., Zimic, M., Barrientos-Porras, F., & Salgado-Bohorquez, M. (2018). Method for the Automatic Segmentation of the Palpebral Conjunctiva. *IEEE International Conference on Automation / ICA-ACCA*, 1-4. <https://doi.org/10.1109/ICA-ACCA.2018.8609744>
- Dimauro, G., Baldari, L., Caivano, D., Colucci, G., & Girardi, F. (2018). Automatic Segmentation of Relevant Sections of the Conjunctiva. *Proceedings of SpliTech*, 1-5. <https://doi.org/10.23919/SpliTech.2018.8448321>

- Dimauro, G., Griseta, M. E., Camporeale, M. G., Clemente, F., Guarini, A., & Maglietta, R. (2023). An intelligent non-invasive system for automated diagnosis of anemia. *Artificial Intelligence in Medicine*, 136, 102477. <https://doi.org/10.1016/j.artmed.2022.102477>
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2020). An Image Is Worth 16x16 Words. *arXiv*. <https://arxiv.org/abs/2010.11929>
- Fuentes-Beingolea, J. H., & Palomino-Quispe, F. (2025). Adaptive Contrastive Learning in Vision Transformer. *IEEE Access*, 13, 210257-210270. <https://doi.org/10.1109/ACCESS.2025.3638640>
- Fuentes-Beingolea, J. H., Palomino-Quispe, F., Herrera-Levano, J. C., Florez, R., & Alvarez, A. B. (2025). Illumination-Robust Conjunctival Image Preprocessing. *International Journal of Online and Biomedical Engineering*, 21(7), 106-124. <https://doi.org/10.3991/ijoe.v21i07.54439>
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4.^a ed.). Pearson.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- HemoCue. (2024). HemoCue Hb 201+ System. <https://www.hemocue.com/es-es/soluciones/hematologia/hemocue-hb-201-dm-system>
- Heredia-Menor, K., & Mamani-Ticona, W. (2021). Comparación de técnicas basadas en visión computacional y machine learning para la detección temprana de anemia a partir del análisis de uñas. *Ingeniería Industrial*, 40, 159-178. <https://doi.org/10.26439/ing.ind2021.n040.5165>
- Hernández-Sampieri, R., Fernández-Collado, C., & Baptista-Lucio, M. d. I. P. (2014). *Metodología de la investigación* (6ta). McGraw-Hill.
- Instituto Nacional de Estadística e Informática. (2022). Encuesta Demográfica y de Salud Familiar - ENDES 2022.
- Kasiviswanathan, S., Vijayan, T. B., Simone, L., & Dimauro, G. (2020). Semantic Segmentation of Conjunctiva Region for Non-Invasive Anemia Detection Applications. *Electronics*, 9(8), 1309. <https://doi.org/10.3390/electronics9081309>
- Mannino, R. G. (2018). *A Noninvasive, Image-Based Smartphone App for Diagnosing Anemia* [Tesis doctoral, Georgia Institute of Technology y Emory University].
- McCormack, T., Ayub, R., & Aziz, F. (2018). Point-of-care testing facilitates screening and treatment for anaemia in women and children in rural Pakistan. *Australian Journal of Rural Health*, 26(3), 194-198. <https://doi.org/10.1111/ajr.12411>

- MedlinePlus. (2022). Conteo sanguíneo completo (CSC).
- Ministerio de Salud. (2017). *Norma Técnica de Salud para el manejo terapéutico y preventivo de la anemia en niños, adolescentes, mujeres gestantes y puérperas (NTS N° 134-MINSA/2017/DGIESP)* (inf. téc.). Ministerio de Salud del Perú. Lima, Perú.
- Ministerio de Salud del Perú. (2024). Qué es la anemia, tratamiento y diagnóstico.
- Montaño, F. (2024). Desnutrición y anemia crecen a la par de la hoja de coca en el Vraem. <https://ojo-publico.com/edicion-regional/desnutricion-y-anemia-crecen-la-par-la-hoja-coca-el-vraem>
- NVIDIA. (2024). U-Net for 2D Image Segmentation.
- OCDE. (2015). *Manual de Frascati 2015: Guía para la recopilación y presentación de información sobre la investigación y el desarrollo experimental*. <https://doi.org/10.1787/9789264239012-en>
- Ochoa Salazar, A. D. C. (2021). *Valor predictivo de la palidez cutaneomucosa en el diagnóstico de anemia en niños de 6 meses a 60 meses atendidos en el centro de salud Jesús Poderoso de San Juan de Miraflores–Lima durante el año 2020* [Tesis de maestría, Universidad Privada San Juan Bautista].
- OpenCV. (2023). OpenCV About.
- Organización Mundial de la Salud. (2023). Anemia.
- Parker, M., Han, Z., Abu-Haydar, E., Matsiko, E., Iyakaremye, D., Tuyisenge, L., Magaret, A., & Lyambabaje, A. (2018). An evaluation of hemoglobin measurement tools and their accuracy and reliability when screening for child anemia in Rwanda: A randomized study. *PLOS One*, *13*(1), e0187663. <https://doi.org/10.1371/journal.pone.0187663>
- Pérez, M., Nisso, G. A. C., & Buitrago, F. V. (2018). Sistema embebido de detección de movimiento mediante visión artificial. *Visión electrónica*, *12*(1), 97-101. <https://doi.org/10.14483/22484728.13410>
- Project Jupyter. (2024). About Us.
- Python Software Foundation. (2023). Python Introduction.
- Rahman, M. N. N., Sadik, S. A., Biswas, A., & Tazul, R. B. (2020). *Point of Care Detection of Anemia in Non-Invasive Manner by Using Image Processing and Convolutional Neural Network with Mobile Devices* [Tesis de maestría, BRAC University].
- Raspberry Pi. (2023). Raspberry Pi High Quality Camera.

Raspberry Pi Foundation. (2020). About us.

Reddy, T. B. (2011). *Linden's Handbook of Batteries* (4th). McGraw-Hill.

Reinhard, E., Adhikhmin, M., Gooch, B., & Shirley, P. (2001). Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5), 34-41. <https://doi.org/10.1109/38.946629>

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, 234-241.

Sehar, N., Krishnamoorthi, N., & Kumar, C. V. (2024). Deep Learning Model-Based Detection of Anemia from Conjunctiva Images. *Healthcare Informatics Research*, 30(4), 335-345. <https://doi.org/10.4258/hir.2024.30.4.335>

Shenzhen Xunlong Software. (2024). Orange Pi 5 Max User Manual.

Shenzhen Xunlong Software Co., Ltd. (2024). Orange Pi 5 Max - Especificaciones Técnicas y Hardware. <http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-5-Max.html>

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. <https://doi.org/10.1186/s40537-019-0197-0>

Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd). Springer. <https://doi.org/10.1007/978-3-030-34372-9>

TensorFlow. (2023). Create production-grade machine learning models with TensorFlow.

Zimic, M. (2018). Sistema Portátil para Diagnóstico de Anemia Basado en Análisis de la Conjuntiva Ocular Utilizando Smartphone e Inteligencia Artificial.

Anexos

Anexo A: Oficio de Aprobación del Comité de Ética



Comité de Bioética Institucional y Sub Comités de Bioética de la UNSAAC

“Año de la recuperación y consolidación de la economía peruana”.

Cusco, 03 de octubre del 2025

Oficio virtual N° 013-2025-CBI-VRIN-UNSAAC

Sr. FUENTES-BEINGOLEA-JOSE HUMBERTO - responsable

Facultad de Ingeniería Eléctrica, Electrónica, Informática y Mecánica de la Universidad Nacional de San Antonio Abad del Cusco.

ASUNTO : Informe sobre exención de aspectos bioéticos del proyecto de investigación.

Ref : **Exp. 831828**

De mi mayor consideración:

Previo un cordial y atento saludo, el presente es para remitir a usted el informe **CBI-UNSAAC2025-13** de la revisión de los aspectos bioéticos al trabajo de investigación “Protocolo de investigación y adquisición de datos - diseño e implementación de un sistema de detección preliminar no invasiva de anemia mediante clasificación de la conjuntiva ocular por aprendizaje profundo”.

En la revisión del trabajo de investigación han participado los miembros del CBI- UNSAAC y los miembros del Subcomité de bioética en ensayos clínicos, estudios clínicos epidemiológicos en seres humanos.

Sin otro particular, uso de la ocasión para expresar las consideraciones de nuestra estima personal.

Atentamente,

Fdo. Dra. Tatiana Del Castillo de Loayza.
Presidente del Comité de Bioética Institucional de la UNSAAC
comite.bioetica@unsaac.edu.pe

C.C. VRIN
/Archivo
/TDL



Comité de Bioética Institucional y Sub Comités de Bioética de la UNSAAC

Código: CBI-UNSAAC2025-013

INFORME FINAL DE ASPECTOS BIOÉTICOS DEL PROYECTO

Datos del Investigador que solicita la opinión del CBI-UNSAAC

Nombre: Jose Humberto Fuentes Beingolea.

Facultad: Facultad de Ingeniería Eléctrica, Electrónica, Informática y Mecánica

Escuela: Ingeniería Electrónica

Universidad: Universidad Nacional de San Antonio Abad del Cusco

Correo electrónico: 131358@unsaac.edu.pe

Datos del Proyecto de investigación

Título: "Protocolo de investigación y adquisición de datos - diseño e implementación de un sistema de detección preliminar no invasiva de anemia mediante clasificación de la conjuntiva ocular por aprendizaje profundo"

Fecha de ingreso: 2025-04-14

Fecha de emisión de informe final: 03-10-2025

Resultado de la evaluación a aspectos bioéticos: APROBADO.

Observaciones: El proyecto presentado a consideración del Comité de Bioética Institucional en Investigación de la Universidad Nacional de San Antonio Abad del Cusco (CBI-UNSAAC), ha sido evaluado por el CBI-UNSAAC y el Subcomité de bioética en ensayos clínicos, estudios clínicos epidemiológicos en seres humanos, quienes solicitaron observaciones, que fueron levantadas en el plazo establecido por el responsable del trabajo de investigación, dando su opinión favorable.

Atentamente,



Mg. Tatiana Del Castillo de Loayza.
Presidente del Comité de Bioética Institucional de la UNSAAC
comite.bioetica@unsaac.edu.pe

Anexo B: Conjuntos de datos local (Cusco)

Tabla 6.2

Datos demográficos y niveles de hemoglobina de los voluntarios locales.

ID	Hgb	Sexo	Edad	ID	Hgb	Sexo	Edad
257	17.5	Masculino	26	298	15.7	Masculino	22
258	17.5	Masculino	26	299	15.7	Masculino	22
259	18.2	Masculino	22	300	19.9	Masculino	30
260	18.2	Masculino	22	301	19.9	Masculino	30
261	17.3	Masculino	24	302	14.1	Femenino	24
262	17.3	Masculino	24	303	14.1	Femenino	24
263	16.6	Masculino	23	304	17.9	Masculino	27
264	16.6	Masculino	23	305	17.9	Masculino	27
265	18.3	Masculino	23	306	14.9	Femenino	29
266	18.3	Masculino	23	307	14.9	Femenino	29
268	19.1	Masculino	20	308	14.6	Femenino	21
269	18.1	Masculino	22	309	14.6	Femenino	21
270	18.1	Masculino	22	310	17.5	Masculino	27
271	14.9	Femenino	28	311	17.5	Masculino	27
272	14.9	Femenino	28	312	19.2	Masculino	30
273	13.0	Femenino	30	313	19.2	Masculino	30
274	13.0	Femenino	30	314	19.2	Masculino	30
275	14.5	Femenino	29	315	16.6	Masculino	21
276	14.5	Femenino	29	316	16.6	Masculino	21
277	16.7	Masculino	30	528	18.1	Masculino	28
278	16.7	Masculino	30	543	18.1	Masculino	28
279	16.9	Masculino	24	544	18.1	Masculino	28
280	16.9	Masculino	24	540	18.1	Masculino	28
281	17.0	Masculino	21	511	17.3	Masculino	26
282	17.0	Masculino	21	513	17.3	Masculino	26
283	18.0	Masculino	22	510	17.3	Masculino	26
284	18.0	Masculino	22	509	17.3	Masculino	26
285	12.2	Femenino	26	508	17.3	Masculino	26
286	12.2	Femenino	26	519	19.0	Masculino	29
287	13.4	Femenino	25	523	19.0	Masculino	29
288	13.4	Femenino	25	524	19.0	Masculino	29
289	17.0	Masculino	24	521	19.0	Masculino	29
290	17.0	Masculino	24	525	19.0	Masculino	29
291	17.0	Masculino	24	526	19.0	Masculino	29
292	13.9	Femenino	22	517	17.9	Masculino	29
293	13.9	Femenino	22	507	17.9	Masculino	29
294	13.9	Femenino	22	514	17.9	Masculino	29
295	17.4	Masculino	24	515	17.9	Masculino	29
296	17.4	Masculino	24	533	17.9	Masculino	29
297	17.4	Masculino	24	–	–	–	–

Anexo C: Ajuste de hemoglobina por altitud

GUÍA TÉCNICA: PROCEDIMIENTO PARA LA DETERMINACIÓN DE LA HEMOGLOBINA MEDIANTE HEMOGLOBINÓMETRO PORTÁTIL

ANEXO 1

AJUSTE DE HEMOGLOBINA SEGÚN LA ALTITUD⁽¹⁾

El ajuste para la evaluación de la determinación de la hemoglobina se realiza por cada persona evaluada y para ello se lleva el valor observado en el hemoglobinómetro al nivel del mar.

Esto se hace restando del valor observado el factor de ajuste que corresponde a la altura (msnm) en la que reside el paciente evaluado durante los últimos cuatro meses.

Para el ajuste por altura se utiliza la siguiente tabla:

Altura (msnm)		Factor de ajuste por	Calculo para hallar hemoglobina ajustada	Altura (msnm)		Factor de ajuste por	Calculo para hallar hemoglobina ajustada
Desde	Hasta			Desde	Hasta		
1000	1041	01	= Hb observada - 01	4183	4235	3.8	= Hb observada 3.8
1042	1265	02	= Hb observada - 02	4236	4286	3.9	= Hb observada 3.9
1266	1448	03	= Hb observada - 03	4287	4337	4.0	= Hb observada 4.0
1449	1608	04	= Hb observada - 04	4338	4388	4.1	= Hb observada 4.1
1609	1751	05	= Hb observada - 05	4389	4437	4.2	= Hb observada 4.2
1752	1882	06	= Hb observada - 06	4438	4487	4.3	= Hb observada 4.3
1883	2003	07	= Hb observada - 07	4488	4535	4.4	= Hb observada 4.4
2004	2116	08	= Hb observada - 08	4536	4583	4.5	= Hb observada 4.5
2117	2223	09	= Hb observada - 09	4584	4631	4.6	= Hb observada 4.6
2224	2325	1.0	= Hb observada - 1.0	4632	4678	4.7	= Hb observada 4.7
2326	2422	1.1	= Hb observada - 1.1	4679	4725	4.8	= Hb observada 4.8
2423	2515	1.2	= Hb observada - 1.2	4726	4771	4.9	= Hb observada 4.9
2516	2604	1.3	= Hb observada - 1.3	4772	4816	5.0	= Hb observada 5.0
2605	2690	1.4	= Hb observada - 1.4	4817	4861	5.1	= Hb observada 5.1
2691	2773	1.5	= Hb observada - 1.5	4862	4906	5.2	= Hb observada 5.2
2774	2853	1.6	= Hb observada - 1.6	4907	4851	5.3	= Hb observada 5.3
2854	2932	1.7	= Hb observada - 1.7	4952	4994	5.4	= Hb observada 5.4
2933	3007	1.8	= Hb observada - 1.8	4995	5038	5.5	= Hb observada 5.5
3008	3081	1.9	= Hb observada - 1.9	5039	5081	5.6	= Hb observada 5.6
3082	3153	2.0	= Hb observada - 2.0	5082	5124	5.7	= Hb observada 5.7
3154	3224	2.1	= Hb observada - 2.1	5124	5166	5.8	= Hb observada 5.8
3225	3292	2.2	= Hb observada - 2.2	5167	5208	5.9	= Hb observada 5.9
3293	3360	2.3	= Hb observada - 2.3	5209	5250	6.0	= Hb observada 6.0
3361	3425	2.4	= Hb observada - 2.4	5251	5291	6.1	= Hb observada 6.1
3426	3490	2.5	= Hb observada - 2.5	5292	5332	6.2	= Hb observada 6.2
3491	3553	2.6	= Hb observada - 2.6	5333	5373	6.3	= Hb observada 6.3
3554	3615	2.7	= Hb observada - 2.7	5374	5413	6.4	= Hb observada 6.4
3616	3676	2.8	= Hb observada - 2.8	5414	5153	6.5	= Hb observada 6.5
3677	3736	2.9	= Hb observada - 2.9	5454	5193	6.6	= Hb observada 6.6
3737	3795	3.0	= Hb observada - 3.0	5494	5532	6.7	= Hb observada 6.7
3796	3853	3.1	= Hb observada - 3.1	5533	5573	6.8	= Hb observada 6.8
3854	3910	3.2	= Hb observada - 3.2	5574	5610	6.9	= Hb observada 6.9
3911	3966	3.3	= Hb observada - 3.3	5611	5649	7.0	= Hb observada 7.0
3967	4021	3.4	= Hb observada - 3.4	5650	5687	7.1	= Hb observada 7.1
4022	4076	3.5	= Hb observada - 3.5	5688	5726	7.2	= Hb observada 7.2
4077	4129	3.6	= Hb observada - 3.6	5726	5763	7.3	= Hb observada 7.3
4130	4182	3.7	= Hb observada - 3.7	5764	5801	7.4	= Hb observada 7.4

Fuente: Adaptado de Hurtado A, Merino C, Delgado E. Influence of anorexia on haematopoietic activities. Archives of Internal Medicine, 1945,75(5): 284-323. Iron Deficiency Anaemia: Assessment, Prevention, and Control. A guide for programme managers. WHO - 2001 CDC Recommendations to prevent and control iron deficiency in the United States MMWR June 03, 1998/47(3); MMWR June 09, 1989/ 38(22); 400-404. Organización Mundial de la Salud. Concentraciones de hemoglobina para diagnosticar la anemia y evaluar su gravedad. Ginebra, Organización Mundial de la Salud, 2011 (WHO/NMH/NHD/MNM/11.1) (http://www.who.int/vmnis/indicators/haemoglob_in_es.pdf, consultado el 26/01/2022).

Anexo D: Códigos y algoritmos del Sistema

Esta sección contiene los scripts principales desarrollados para el entrenamiento, validación, conversión e implementación del sistema.

D.1. Entrenamiento y Validación: Segmentación (U-Net)

```
1 # @title 1. CONFIGURACION INICIAL Y PARTICION DE DATOS
2 import os
3 import random
4 import numpy as np
5 import tensorflow as tf
6 from sklearn.model_selection import train_test_split
7 # rutas y parametros generales
8 base_dir = 'ruta/hacia/datos'
9 img_dir = os.path.join(base_dir, 'images')
10 mask_dir = os.path.join(base_dir, 'masks')
11 img_size = (512, 512) #tamaño imagen 512x512
12 batch_size = 2
13 seed = 45 # reproducibilidad
14
15 # configurar semillas
16 os.environ['PYTHONHASHSEED'] = str(seed)
17 random.seed(seed)
18 np.random.seed(seed)
19 tf.random.set_seed(seed)
20 # listar y ordenar archivos
21 img_ids = sorted([f for f in os.listdir(img_dir) if f.endswith(('.jpg', '.png'))])
22 mask_ids = sorted([f for f in os.listdir(mask_dir) if f.endswith(('.jpg', '.png'))])
23 # crear pares de rutas
24 all_pairs = list(zip(
25     [os.path.join(img_dir, f) for f in img_ids],
26     [os.path.join(mask_dir, f) for f in mask_ids]
27 ))
28 random.shuffle(all_pairs) # mezclar antes de dividir
29 # division 80/10/10
30 train_val, test_pairs = train_test_split(all_pairs, test_size=0.1, random_state=seed)
31 train_pairs, val_pairs = train_test_split(train_val, test_size=0.1111, random_state=seed)
32 # 0.1111 de 0.9 es aprox 0.1 del total
33 # separar en listas individuales
34 train_imgs, train_masks = zip(*train_pairs)
35 val_imgs, val_masks = zip(*val_pairs)
36 test_imgs, test_masks = zip(*test_pairs)
37 print(f"datos cargados por jose. total: {len(all_pairs)}")
38 print(f"train: {len(train_imgs)}, val: {len(val_imgs)}, test: {len(test_imgs)}")
39
40 # @title 2. PIPELINE DE DATOS Y DATA AUG
41 # capas de aumento (rotacion +-5 %, zoom +-10 %, flip horizontal)
42 aug_layers = tf.keras.Sequential([
43     RandomRotation(0.05, seed=seed),
44     RandomZoom((-0.1, 0.1), seed=seed),
45     RandomFlip("horizontal", seed=seed)
46 ])
47 def read_file(img_path, mask_path):
48     # carga y normalizacion imagen
49     img = tf.io.read_file(img_path)
50     img = tf.image.decode_jpeg(img, channels=3)
51     img = tf.image.resize(img, img_size) / 255.0
52     # carga y normalizacion mascara
53     mask = tf.io.read_file(mask_path)
54     mask = tf.image.decode_jpeg(mask, channels=1)
55     mask = tf.image.resize(mask, img_size) / 255.0
56     mask = tf.round(mask) # binarizar
57     return img, mask
58 def augment(img, mask):
59     # aplicar aumentos sincronizados
60     combined = tf.concat([img, mask], axis=-1)
61     combined = aug_layers(combined, training=True)
62     return combined[:, :, :3], combined[:, :, 3:]
63 def create_ds(images, masks, train=False):
64     ds = tf.data.Dataset.from_tensor_slices((list(images), list(masks)))
65     ds = ds.map(read_file, num_parallel_calls=tf.data.AUTOTUNE)
66     if train:
67         ds = ds.map(augment, num_parallel_calls=tf.data.AUTOTUNE)
68         ds = ds.shuffle(buffer_size=100)
69     return ds.batch(batch_size).prefetch(tf.data.AUTOTUNE)
70 # instanciar datasets (jose fuentes)
71 train_ds = create_ds(train_imgs, train_masks, train=True)
```

```

72 val_ds = create_ds(val_imgs, val_masks, train=False)
73 test_ds = create_ds(test_imgs, test_masks, train=False)
74
75 # @title 3. ARQUITECTURA UNET Y ENTRENAMIENTO
76 from tensorflow.keras.models import Model
77 from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, UpSampling2D, concatenate, BatchNormalization, Activation, Dropout
78 from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
79 # metricas personalizadas
80 def dice_coef(y_true, y_pred, smooth=1e-6):
81     intersection = tf.reduce_sum(y_true * y_pred)
82     return (2. * intersection + smooth) / (tf.reduce_sum(y_true) + tf.reduce_sum(y_pred) + smooth)
83 def iou_coef(y_true, y_pred, smooth=1e-6):
84     intersection = tf.reduce_sum(y_true * y_pred)
85     union = tf.reduce_sum(y_true) + tf.reduce_sum(y_pred) - intersection
86     return (intersection + smooth) / (union + smooth)
87 def bce_dice_loss(y_true, y_pred):
88     return tf.keras.losses.binary_crossentropy(y_true, y_pred) + (1 - dice_coef(y_true, y_pred))
89 # bloque convolucional basico sin SE
90 def conv_block(input_tensor, filters):
91     x = Conv2D(filters, (3, 3), padding='same')(input_tensor)
92     x = BatchNormalization()(x)
93     x = Activation('relu')(x)
94     x = Conv2D(filters, (3, 3), padding='same')(x)
95     x = BatchNormalization()(x)
96     x = Activation('relu')(x)
97     return x
98
99 # construccion del modelo
100 def build_unet(input_shape):
101     inputs = Input(input_shape)
102     # encoder
103     c1 = conv_block(inputs, 16)
104     p1 = MaxPooling2D((2, 2))(c1)
105     p1 = Dropout(0.1)(p1)
106     c2 = conv_block(p1, 32)
107     p2 = MaxPooling2D((2, 2))(c2)
108     p2 = Dropout(0.1)(p2)
109     c3 = conv_block(p2, 64)
110     p3 = MaxPooling2D((2, 2))(c3)
111     p3 = Dropout(0.2)(p3)
112     c4 = conv_block(p3, 128)
113     p4 = MaxPooling2D((2, 2))(c4)
114     p4 = Dropout(0.2)(p4)
115     # cuello de botella
116     c5 = conv_block(p4, 256)
117     c5 = Dropout(0.3)(c5)
118
119     # decoder
120     u6 = UpSampling2D((2, 2))(c5)
121     u6 = concatenate([u6, c4])
122     c6 = conv_block(u6, 128)
123     u7 = UpSampling2D((2, 2))(c6)
124     u7 = concatenate([u7, c3])
125     c7 = conv_block(u7, 64)
126     u8 = UpSampling2D((2, 2))(c7)
127     u8 = concatenate([u8, c2])
128     c8 = conv_block(u8, 32)
129     u9 = UpSampling2D((2, 2))(c8)
130     u9 = concatenate([u9, c1])
131     c9 = conv_block(u9, 16)
132
133     outputs = Conv2D(1, (1, 1), activation='sigmoid')(c9)
134     return Model(inputs=[inputs], outputs=[outputs])
135 # compilar
136 model = build_unet((512, 512, 3))
137 model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4),
138              loss=bce_dice_loss,
139              metrics=[iou_coef, dice_coef, 'accuracy'])
140 # callbacks
141 best_model_path = 'ruta/hacia/mejor_modelo/best_unet_jose.keras'
142 callbacks = [
143     ModelCheckpoint(best_model_path, monitor='val_iou_coef', mode='max', save_best_only=True, verbose=1),
144     ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-7)
145 ]
146 # entrenar
147 history = model.fit(train_ds, validation_data=val_ds, epochs=40, callbacks=callbacks, verbose=1)
148
149 # @title 4. EVALUACION DEL MEJOR MODELO EN SET DE PRUEBAS
150 import pandas as pd
151 from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
152 # cargar mejor modelo
153 custom_objs = {'iou_coef': iou_coef, 'dice_coef': dice_coef, 'bce_dice_loss': bce_dice_loss}
154 best_model = tf.keras.models.load_model(best_model_path, custom_objects=custom_objs)
155 print(f"modelo cargado desde: {best_model_path}")

```

```

156 # preparar listas para metricas
157 test_metrics = []
158 # iterar sobre set de pruebas
159 for img_batch, mask_batch in test_ds:
160     preds = best_model.predict(img_batch, verbose=0)
161     preds_bin = (preds > 0.5).astype(np.uint8)
162     masks_bin = (mask_batch.numpy() > 0.5).astype(np.uint8)
163     # calcular metricas batch a batch (aplanando)
164     for i in range(len(preds_bin)):
165         y_true = masks_bin[i].flatten()
166         y_pred = preds_bin[i].flatten()
167         # calculos basicos
168         inter = np.sum(y_true * y_pred)
169         union = np.sum(y_true) + np.sum(y_pred) - inter
170         iou = inter / (union + 1e-7)
171         dice = (2 * inter) / (np.sum(y_true) + np.sum(y_pred) + 1e-7)
172         test_metrics.append({
173             'accuracy': accuracy_score(y_true, y_pred),
174             'precision': precision_score(y_true, y_pred, zero_division=0),
175             'recall': recall_score(y_true, y_pred, zero_division=0),
176             'f1_score': f1_score(y_true, y_pred, zero_division=0),
177             'iou': iou,
178             'dice': dice
179         })
180 # dataframe de resultados
181 df_res = pd.DataFrame(test_metrics)
182 print("\n----- resumen de metricas en test (promedio) -----")
183 print(df_res.mean().round(4))
184 # guardar resultados
185 df_res.to_csv('ruta/hacia/resultados_unet.csv', index=False)

```

C.3. Transformación y Cuantización RKNN

```

1 #CONVERSION DE MODELO A RKNN
2 import sys
3 from rknn.api import RKNN
4
5 def convert_to_rknn(onnx_path, rknn_path, input_shape, platform='rk3588'):
6     """
7     Convierte ONNX a RKNN unificado para entrada de imagenes NHWC (RGB).
8     Args:
9         onnx_path: Ruta al modelo ONNX.
10        rknn_path: Ruta de salida .rknn.
11        input_shape: Tupla (H, W, C) ej: (512, 512, 3) o (224, 224, 3).
12    """
13    print(f" Iniciando conversin: {onnx_path} -> {rknn_path}")
14    # Inicializar RKNN
15    rknn = RKNN(verbose=True)
16    # 1. CONFIGURACION
17    # mean_values y std_values permiten pasar uint8 (0-255) directamente a la NPU.
18    # La NPU normalizar internamente: (pixel - 0) / 255 = 0.0 a 1.0 (FP16)
19    rknn.config(
20        mean_values=[[0, 0, 0]],
21        std_values=[[255, 255, 255]],
22        target_platform=platform,
23        # optimization_level=3 activa optimizaciones agresivas del compilador
24        optimization_level=3
25    )
26    # 2. CARGAR ONNX
27    # RKNN detecta automaticamente si el ONNX es NCHW (PyTorch) o NHWC (TF).
28    # Si es PyTorch (NCHW), RKNN insertar automaticamente una capa de transposicion
29    print("--> Cargando modelo ONNX...")
30    ret = rknn.load_onnx(model=onnx_path)
31    if ret != 0:
32        print(" Error cargando ONNX")
33        sys.exit(1)
34    # 3. CONSTRUIR (BUILD)
35    # do_quantization=False para mantener FP16
36    print("--> Construyendo modelo RKNN (FP16)...")
37    ret = rknn.build(do_quantization=False)
38    if ret != 0:
39        print(" Error construyendo RKNN")
40        sys.exit(1)
41    # 4. EXPORTAR
42    print("--> Exportando .rknn...")
43    ret = rknn.export_rknn(rknn_path)
44    if ret != 0:
45        print(" Error exportando RKNN")
46        sys.exit(1)
47    print(" Conversin exitosa!")

```

C.4. Códigos embebidos (Firmware Orange Pi)

C.4.1 Orquestador principal

```
1 #CODIGO ORQUESTADOR PRINCIPAL
2 import cv2
3 import time
4 import datetime
5 import traceback
6 import signal
7 import os
8 import threading
9 import copy
10 from rknnlite.api import RKNNLite
11 from analizador import hardware, display, ia_processing, reporting, battery_manager
12
13 # ---- Constantes de Configuraci on ----
14 CAMERA_INDEX = 0
15 LONG_PRESS_DURATION_S = 4.0
16 FIXED_FRAME_WIDTH = 3264
17 FIXED_FRAME_HEIGHT = 2448
18 FIXED_FOCUS_VALUE = 230
19 # ---- Constantes de Energia (Soft Start) ----
20 LED_TARGET_DUTY = 0.6
21 LED_RAMP_STEPS = 25
22 LED_RAMP_DELAY = 0.04
23 STABILIZATION_DELAY = 0.4
24 # ---- Constantes de Loop (Optimizaci on) ----
25 IDLE_FPS_TARGET = 5.0
26 IDLE_FRAME_TIME = 1.0 / IDLE_FPS_TARGET
27 # ---- Variables Globales Compartidas (Thread-Safe) ----
28 app_status = "Iniciando"
29 shutdown_flag = False
30 manual_shutdown_triggered = False
31 system_busy = False
32 trigger_capture = False
33 trigger_shutdown = False
34 battery_warning_acknowledged = False
35 def graceful_shutdown(signum, frame):
36     global shutdown_flag
37     if not shutdown_flag:
38         print("\n[Sistema] Senial de terminaci on recibida...")
39         shutdown_flag = True
40 def button_listener_thread(gpio_interface, battery_mgr):
41     """
42     Hilo dedicado para escuchar eventos del boton.
43     Detecta:
44     - Click corto: Captura de imagen (o ACK de warning de bateria)
45     - Presi on larga (4s): Apagado del sistema
46     """
47     global shutdown_flag, system_busy, trigger_capture, trigger_shutdown, battery_warning_acknowledged
48     last_state = hardware.GPIO.LOW
49     press_start = 0.0
50     is_pressing = False
51     print("[Input] Hilo de boton iniciado correctamente.")
52     while not shutdown_flag:
53         if system_busy:
54             time.sleep(0.1)
55             is_pressing = False
56             last_state = hardware.GPIO.LOW
57             continue
58         try:
59             current_state = gpio_interface.input(hardware.BUTTON_PIN)
60             now = time.time()
61             # Detectar flanco de subida (bot on presionado)
62             if current_state == hardware.GPIO.HIGH and last_state == hardware.GPIO.LOW:
63                 press_start = now
64                 is_pressing = True
65             # Detectar flanco de bajada (bot on liberado)
66             elif current_state == hardware.GPIO.LOW and last_state == hardware.GPIO.HIGH:
67                 if is_pressing:
68                     duration = now - press_start
69                     # Presi on larga: APAGADO
70                     if duration >= LONG_PRESS_DURATION_S:
71                         print(f"[Input] APAGADO detectado ({duration:.2f}s)")
72                         trigger_shutdown = True
73                     # Presi on corta: CAPTURA o ACK
74                     elif duration > 0.05:
75                         # Si hay warning de bater ia pendiente, solo hacer ACK
76                         if battery_mgr.should_show_warning() and not battery_warning_acknowledged:
77                             print(f"[Input] ACK warning bateria")
```

```

78         battery_warning_acknowledged = True
79         battery_mgr.acknowledge_warning()
80     else:
81         print(f"[Input] CLICK detectado ({duration:.2f}s)")
82         trigger_capture = True
83         is_pressing = False
84         last_state = current_state
85         time.sleep(0.01)
86     except Exception as e:
87         print(f"[Input Thread Error] {e}")
88         time.sleep(1)
89
90 def main():
91     global shutdown_flag, system_busy, trigger_capture, trigger_shutdown
92     global manual_shutdown_triggered, battery_warning_acknowledged
93     signal.signal(signal.SIGINT, graceful_shutdown)
94     signal.signal(signal.SIGTERM, graceful_shutdown)
95     led_pwm = []
96     rknn_segenter = None
97     rknn_estimator = None
98     cap = None
99     gpio = None
100    battery_mgr = None
101    try:
102        # [0] INICIALIZAR GESTOR DE BATERIA
103        print("\n" + "="*60)
104        print(" INICIALIZANDO GESTOR DE BATERIA")
105        print("="*60)
106        battery_mgr = battery_manager.BatteryManager()
107        # Verificar si debe resetear por apagado previo
108        was_reset = battery_mgr.check_and_reset_if_needed()
109        if was_reset:
110            print(" Bateria considerada recargada – Iniciando desde 0:00")
111        else:
112            print(f" Continuando desde sesion anterior: {battery_mgr.get_formatted_time()}")
113
114        # [1] HARDWARE
115        print("\n" + "="*60)
116        print(" INICIALIZANDO HARDWARE")
117        print("="*60)
118        gpio = hardware.GPIO()
119        oled_ok = display.init_oled()
120        if oled_ok:
121            display.update_oled_display("Iniciando...", "Sistema OK")
122            time.sleep(STABILIZATION_DELAY)
123
124        # [2] MODELOS IA
125        display.update_oled_display("Cargando...", "Modelos IA")
126        print("\n" + "="*60)
127        print(" CARGANDO MODELOS DE IA EN NPU")
128        print("="*60)
129        # Modelo de Segmentacion (U-Net)
130        if not os.path.exists(ia_processing.SEG_MODEL_PATH):
131            raise FileNotFoundError(" Falta modelo de segmentacion")
132        print(" Cargando U-Net para segmentacion...")
133        rknn_segenter = RKNNLite(verbose=False)
134        if rknn_segenter.load_rknn(ia_processing.SEG_MODEL_PATH) != 0:
135            raise RuntimeError(" Error al cargar U-Net")
136        if rknn_segenter.init_runtime() != 0:
137            raise RuntimeError(" Error al inicializar U-Net")
138        print(" U-Net cargado exitosamente")
139        # Modelo de Estimacion de Hemoglobina
140        if not os.path.exists(ia_processing.HGB_MODEL_PATH):
141            raise FileNotFoundError(" Falta modelo de hemoglobina")
142        print(" Cargando modelo de estimacion de hemoglobina...")
143        rknn_estimator = RKNNLite(verbose=False)
144        if rknn_estimator.load_rknn(ia_processing.HGB_MODEL_PATH) != 0:
145            raise RuntimeError(" Error al cargar HemoConvit")
146        if rknn_estimator.init_runtime() != 0:
147            raise RuntimeError(" Error al inicializar HemoConvit")
148        print(" Modelo de hemoglobina cargado exitosamente")
149        time.sleep(STABILIZATION_DELAY)
150
151        # [3] LEDS
152        display.update_oled_display("Configurando...", "Luces")
153        print("\n" + "="*60)
154        print(" CONFIGURANDO SISTEMA DE ILUMINACION")
155        print("="*60)
156        gpio.setup(hardware.BUTTON_PIN, hardware.GPIO.IN)
157        print(" Boton configurado en GPIO 12")
158        for pin in hardware.LED_PINS:
159            pwm = hardware.PWM(gpio, pin, frequency_hz=hardware.LED_FREQUENCY_HZ)
160            pwm.start(0.0)
161            led_pwm.append(pwm)

```

```

162 print(f" {len(led_pwm)} LEDs inicializados")
163 print(f"Encendiendo LEDs con soft—start ({LED_TARGET_DUTY*100:.0f} %)...")
164 hardware.soft_start_leds(led_pwm, LED_TARGET_DUTY)
165 print(" LEDs listos")
166 time.sleep(STABILIZATION_DELAY)
167
168 # [4] C aMARA
169 display.update_oled_display("Iniciando...", "Camara...")
170 print("\n" + "="*60)
171 print(" INICIALIZANDO C aMARA")
172 print("="*60)
173 cap = cv2.VideoCapture(CAMERA_INDEX)
174 if not cap.isOpened():
175     raise RuntimeError(" Error: No se pudo acceder a la c amara")
176 print(f" Configurando resoluci on: {FIXED_FRAME_WIDTH}x{FIXED_FRAME_HEIGHT}")
177 cap.set(cv2.CAP_PROP_FRAME_WIDTH, FIXED_FRAME_WIDTH)
178 cap.set(cv2.CAP_PROP_FRAME_HEIGHT, FIXED_FRAME_HEIGHT)
179 cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
180 cap.set(cv2.CAP_PROP_AUTOFOCUS, 0)
181 cap.set(cv2.CAP_PROP_FOCUS, FIXED_FOCUS_VALUE)
182 print(f"Foco fijo configurado: {FIXED_FOCUS_VALUE}")
183 # Warm—up (Auto Exposure) (Calentar antes de bloquear exposici on)
184 print("Warm—up de auto—exposici on (1.5s)...")
185 cap.set(cv2.CAP_PROP_AUTO_EXPOSURE, 3)
186 start_t = time.time()
187 while (time.time() — start_t) < 1.5:
188     cap.grab()
189     time.sleep(0.05)
190 # Bloqueo de Exposici on (Parametros fijos)
191 print(" Bloqueando parametros de exposicion...")
192 cap.set(cv2.CAP_PROP_AUTO_EXPOSURE, 1)
193 cap.set(cv2.CAP_PROP_EXPOSURE, 7500)
194 cap.set(cv2.CAP_PROP_GAIN, 100)
195 cap.set(cv2.CAP_PROP_WB_TEMPERATURE, 5300)
196 print(" Camara configurada y lista")
197
198 # [5] PREPARACI oN FINAL
199 session_index = reporting.get_next_session_index()
200 # INICIAR SESI oN DE BATER IA
201 battery_mgr.start_session()
202 display.update_oled_with_battery("Listo", battery_mgr, f"ID: {session_index:03d}")
203 print("\n" + "="*60)
204 print(" SISTEMA COMPLETAMENTE OPERATIVO")
205 print("="*60)
206 print(f" ID de Sesion: {session_index:03d}")
207 print(f" Bateria: {battery_mgr.get_battery_status()}")
208 print(f" Tiempo acumulado: {battery_mgr.get_formatted_time()}")
209 print(f" Tiempo restante: {battery_mgr.get_remaining_time()}")
210 print("="*60)
211 print("\n Esperando eventos del usuario...\n")
212 # Iniciar hilo de escucha de bot on
213 input_thread = threading.Thread(
214     target=button_listener_thread,
215     args=(gpio, battery_mgr),
216     daemon=True
217 )
218 input_thread.start()
219 # Variables de control del loop
220 last_idle_check = 0.0
221
222 # [6] BUCLE PRINCIPAL
223 while not shutdown_flag:
224     # VERIFICAR ESTADO DE BATERIA (PRIORITARIO)
225     # CASO CR ITICO 1: Apagado autom atico por bateria agotada
226     if battery_mgr.should_shutdown():
227         print("\n" + "="*60)
228         print("BATERIA AGOTADA — APAGADO AUTOMaTICO")
229         print("="*60)
230         print(f"Tiempo total de uso: {battery_mgr.get_formatted_time()}")
231         print("El sistema se apagar a autom aticamente")
232         print("Al encender nuevamente, el contador se reiniciar a (asumiendo recarga)")
233         print("="*60)
234         display.update_oled_display("Bateria Agotada", "Apagando...")
235
236         # guardar estado (ser a detectado en pr oximo boot para reset)
237         battery_mgr.save_session_state()
238         manual_shutdown_triggered = True
239         for pwm in led_pwm: pwm.stop()
240         time.sleep(3.0)
241         os.system("sudo shutdown —h now")
242         shutdown_flag = True
243         break
244     # CASO CR ITICO 2: Mostrar advertencia de bateria baja (2:15)
245     if battery_mgr.should_show_warning() and not battery_warning_acknowledged:

```

```

246     print("\n" + "="*60)
247     print(" ADVERTENCIA: BATERIA BAJA")
248     print("="*60)
249     battery_mgr.mark_warning_shown()
250     system_busy = True # Bloquear capturas durante el warning
251     remaining = battery_mgr.get_remaining_time()
252     print(f" Tiempo restante estimado: {remaining}")
253     print(f" Tiempo transcurrido: {battery_mgr.get_formatted_time()}")
254     print(" El usuario debe reconocer este mensaje presionando el bot on")
255     print("="*60)
256     # Mostrar advertencia en pantalla
257     display.update_oled_display(" Bateria Baja!", f"Quedan ~{remaining}")
258     time.sleep(2.0)
259     display.update_oled_display("Presione para", "continuar")
260     # Esperar confirmacion del usuario
261     print(" Esperando confirmacion del usuario...")
262     while not battery_warning_acknowledged and not shutdown_flag:
263         time.sleep(0.1)
264     if not shutdown_flag:
265         print("Usuario confirmo el warning — Continuando operacion")
266         display.update_oled_with_battery("Listo", battery_mgr, f"ID: {session_index:03d}")
267         system_busy = False
268         last_idle_check = time.time()
269     # Actualizacion periodica del estado de bateria (cada 10 min)
270     if battery_mgr.needs_periodic_update():
271         battery_mgr.update_time()
272         battery_mgr.save_session_state()
273         print(f"\n[Bateria] Actualizacion periodica")
274         print(f" Tiempo: {battery_mgr.get_formatted_time()}")
275         print(f" Restante: {battery_mgr.get_remaining_time()}")
276         print(f" Estado: {battery_mgr.get_battery_status()}\n")
277     # CASO A: APAGADO MANUAL SOLICITADO
278     if trigger_shutdown:
279         print("\n" + "="*60)
280         print("APAGADO MANUAL SOLICITADO")
281         print("="*60)
282         # Siempre guardar estado actual
283         battery_mgr.save_session_state()
284         # Mensaje diferente segun el contexto
285         if battery_mgr.is_conscious_shutdown_after_warning():
286             print("Apagado manual tras warning de bateria")
287             print(" En el proximo arranque se reseteara a el contador (asumiendo recarga)")
288             display.update_oled_display("Recargue antes", "de usar")
289         else:
290             print(" Apagado manual normal")
291             print(f" Estado guardado: {battery_mgr.get_formatted_time()}")
292             display.update_oled_display("Apagando", "Sistema...")
293         print("="*60)
294         manual_shutdown_triggered = True
295         for pwm in led_pwm: pwm.stop()
296         time.sleep(2.0)
297         os.system("sudo shutdown -h now")
298         shutdown_flag = True
299         break
300     # CASO B: CAPTURA SOLICITADA
301     if trigger_capture:
302         trigger_capture = False
303         system_busy = True
304         battery_mgr.update_time()
305         display.update_oled_display("Capturando...", "")
306         try:
307             print("\n" + "="*60)
308             print(f" INICIANDO CAPTURA #{session_index:03d}")
309             print("="*60)
310             capture_start = time.time()
311             # 1. CAPTURA DE IMAGEN
312             print("Capturando imagen...", end="", flush=True)
313             ret_cap, frame_original = hardware.flash_and_capture(cap)
314             capture_time = time.time() - capture_start
315             if ret_cap:
316                 print(f" Hecho ({capture_time*1000:.1f}ms)")
317                 # 2. APAGAR LEDS INMEDIATAMENTE (Feedback visual)
318                 hardware.instant_leds_off(led_pwm)
319                 # 3. ACTUALIZAR PANTALLA
320                 display.update_oled_display("Capturado!", "Procesando...")
321                 # 4. PROCESAMIENTO CON IA
322                 print("Procesando con IA...")
323                 processing_start = time.time()
324                 frame_work = frame_original.copy()
325                 results = ia_processing.run_analysis(frame_work, rknn_segmenter, rknn_estimator)
326                 processing_time = time.time() - processing_start
327                 print(f" Tiempo de procesamiento: {processing_time*1000:.1f}ms")
328                 # 5. MANEJO DE RESULTADOS
329                 if not results['success']:

```

```

330         error_code = results.get('error_code')
331     if error_code in ['ROI_TOO_SMALL', 'NO_ROI_DETECTED']:
332         print(f"Fallo en segmentaci on: {results['error_message']}")
333         display.update_oled_display("Fallo en", "segmentacion")
334         time.sleep(3.0)
335         display.update_oled_display("Reposicione", "Presione boton")
336         time.sleep(2.0)
337     else:
338         err_msg = results.get('error_message', 'Error IA')
339         print(f" Error en procesamiento: {err_msg}")
340         display.update_oled_display("Fallo IA", err_msg[:16])
341         time.sleep(2.0)
342
343     else:
344         # AN aLISIS EXITOSO
345         val = results['metrics'].get('hemoglobin_g_dl', 0.0)
346         roi_pct = results['metrics'].get('roi_percentage_512', 0.0)
347         txt = f"{val:.2f} g/dl"
348         # DETERMINAR SI HAY ANEMIA
349         is_anemic = val < 15.3
350         print("="*60)
351         print(" ANALISIS COMPLETADO EXITOSAMENTE")
352         print("="*60)
353         print(f" Hemoglobina estimada: {txt}")
354         print(f"ROI valido: {roi_pct:.2f} % del area total")
355         # MOSTRAR DIAGN oSTICO SI APLICA
356         if is_anemic:
357             print(f"DIAGNOSTICO: POSIBLE ANEMIA (Hb < 15.3 g/dL)")
358             print("="*60)
359             # USAR LA FUNCION DE DISPLAY CON DIAGNOSTICO
360             display.update_oled_with_diagnosis(val, is_anemic)
361             # 6. GUARDADO DE RESULTADOS
362             print(" Guardando resultados...", end='', flush=True)
363             display.update_oled_display("Guardando...", txt)
364             save_start = time.time()
365             reporting.save_session_results(session_index, frame_original, results)
366             save_time = time.time() - save_start
367             print(f" Hecho ({save_time:.2f}s)")
368             # 7. ACTUALIZAR ESTADO DE BATERIA
369             battery_mgr.save_session_state()
370             print(f" Estado de bater ia actualizado: {battery_mgr.get_formatted_time()}")
371             # Incrementar indice de sesion
372             session_index += 1
373             total_time = time.time() - capture_start
374             print(f"\n Tiempo total de captura: {total_time:.2f}s")
375             print("-"*60 + "\n")
376             time.sleep(2.0)
377
378     else:
379         print(" ERROR")
380         hardware.instant_leds_off(led_pwm)
381         display.update_oled_display("Error Camara", "Reintentar")
382         time.sleep(2.0)
383
384     except Exception as e:
385         print(f"\n ERROR CRITICO EN CAPTURA: {e}")
386         traceback.print_exc()
387         hardware.instant_leds_off(led_pwm)
388         display.update_oled_display("Error", "Sistema")
389         time.sleep(2.0)
390
391     finally:
392         # REENCENDER LEDS Y DESBLOQUEAR SISTEMA
393         print("Reencendiendo LEDs con soft—start...")
394         display.update_oled_display("Preparando...", "")
395         hardware.soft_start_leds(led_pwm, LED_TARGET_DUTY)
396         display.update_oled_with_battery("Listo", battery_mgr, f"ID: {session_index:03d}")
397         system_busy = False
398         last_idle_check = time.time()
399         print(" Sistema listo para nueva captura\n")
400
401     # CASO C: IDLE (Mantenimiento de camara a 5 FPS)
402     if not system_busy:
403         now = time.time()
404         if (now - last_idle_check) > IDLE_FRAME_TIME:
405             cap.grab() # Mantener buffer fresco
406             last_idle_check = now
407             time.sleep(0.01)
408
409     except Exception as e:
410         print("\n" + "="*60)
411         print("ERROR CRITICO EN MAIN")
412         print("="*60)
413         print(f"Error: {e}")
414         traceback.print_exc()
415         print("="*60)
416         # Guardar estado de bateria en caso de crash
417         if battery_mgr:
418             try:

```

```

414         battery_mgr.save_session_state()
415         print("Estado de bater ia guardado tras error")
416     except:
417         pass
418     try:
419         display.init_oled()
420         display.update_oled_display("ERROR FATAL", "Reinicie")
421         time.sleep(5)
422     except:
423         pass
424
425     finally:
426         print("\n" + "="*60)
427         print(" FINALIZANDO APLICACION")
428         print("="*60)
429         shutdown_flag = True
430         # Guardar estado final de bateria
431         if battery_mgr:
432             try:
433                 battery_mgr.save_session_state()
434                 print(" Estado final de bateria guardado")
435             except Exception as e:
436                 print(f" Error al guardar estado de bateria: {e}")
437         # Limpiar recursos
438         if not manual_shutdown_triggered:
439             display.clear_oled()
440         print(" Apagando LEDs...", end='')
441         for pwm in led_pwm:
442             try:
443                 pwm.stop()
444             except:
445                 pass
446         print(" ")
447         print("Limpiando GPIO...", end='')
448         if gpio:
449             try:
450                 gpio.cleanup()
451             except:
452                 pass
453         print(" ")
454         print(" Liberando modelos de IA...", end='')
455         if rknn_segementer:
456             try:
457                 rknn_segementer.release()
458             except:
459                 pass
460         if rknn_estimator:
461             try:
462                 rknn_estimator.release()
463             except:
464                 pass
465         print(" ")
466         print(" Liberando camara...", end='')
467         if cap:
468             try:
469                 cap.release()
470             except:
471                 pass
472         print(" ")
473         print("="*60)
474         print(" Aplicacion finalizada correctamente")
475         print("="*60 + "\n")
476     if __name__ == '__main__':
477         main()

```

C.4.2 Hardware

```

1  import os, time, threading
2  # ----- Constantes de Hardware -----
3  BUTTON_PIN = 12
4  LED_PINS = [15, 16]
5  LED_FREQUENCY_HZ = 1000
6  # ----- Constantes Soft Start -----
7  LED_RAMP_STEPS = 25
8  LED_RAMP_DELAY = 0.04
9
10 class GPIO:
11     """Control unificado GPIO sysfs para Orange Pi 5."""
12     BOARD = {
13         15: 34, 16: 35, # LEDs
14         12: 134, # Boton
15         7: 139, 18: 133, 22: 135,

```

```

16 }
17 OUT, IN = "out", "in"
18 HIGH, LOW = 1, 0
19
20 def __init__(self):
21     self.exported = []
22 def setup(self, pin, direction):
23     gpio = self.BOARD.get(pin)
24     if gpio is None: raise ValueError(f"Pin {pin} no mapeado.")
25     path = f"/sys/class/gpio/gpio{gpio}"
26
27     if not os.path.exists(path):
28         try:
29             with open("/sys/class/gpio/export", "w") as f: f.write(str(gpio))
30             time.sleep(0.1)
31         except Exception as e:
32             if "busy" not in str(e): raise RuntimeError(f"Error export GPIO {gpio}: {e}")
33
34         try:
35             with open(f"{path}/direction", "w") as f: f.write(direction)
36         except Exception as e: raise RuntimeError(f"Error direccion GPIO {gpio}: {e}")
37
38     if (pin, gpio) not in self.exported: self.exported.append((pin, gpio))
39     return gpio
40 def output(self, pin, value):
41     gpio = self.BOARD.get(pin)
42     if gpio is None: raise ValueError(f"Pin {pin} no mapeado.")
43     try:
44         with open(f"/sys/class/gpio/gpio{gpio}/value", "w") as f: f.write(str(value))
45     except Exception as e: raise RuntimeError(f"Error escritura GPIO {gpio}: {e}")
46 def input(self, pin):
47     gpio = self.BOARD.get(pin)
48     if gpio is None: raise ValueError(f"Pin {pin} no mapeado.")
49     try:
50         with open(f"/sys/class/gpio/gpio{gpio}/value", "r") as f: return int(f.read().strip())
51     except Exception as e: raise RuntimeError(f"Error lectura GPIO {gpio}: {e}")
52 def cleanup(self):
53     for _, gpio in self.exported:
54         try:
55             if os.path.exists(f"/sys/class/gpio/gpio{gpio}/value"):
56                 with open(f"/sys/class/gpio/gpio{gpio}/value", "w") as f: f.write("0")
57                 with open("/sys/class/gpio/unexport", "w") as f: f.write(str(gpio))
58             except Exception: pass
59     self.exported = []
60
61 class PWM:
62     """PWM por software usando clase GPIO."""
63     def __init__(self, gpio_ctrl, pin, freq_hz):
64         self.gpio = gpio_ctrl
65         self.pin = pin
66         self.period = 1.0 / freq_hz
67         self.duty = 0.0
68         self.run = False
69         self.thread = None
70         self.gpio.setup(self.pin, GPIO.OUT)
71     def _loop(self):
72         while self.run:
73             d = self.duty
74             if d < 0.01: self.gpio.output(self.pin, 0); time.sleep(self.period); continue
75             if d > 0.99: self.gpio.output(self.pin, 1); time.sleep(self.period); continue
76             self.gpio.output(self.pin, 1); time.sleep(self.period * d)
77             self.gpio.output(self.pin, 0); time.sleep(self.period * (1.0 - d))
78     def start(self, init_duty=0.0):
79         if self.run: return
80         self.change_duty_cycle(init_duty)
81         self.run = True
82         self.thread = threading.Thread(target=self._loop, daemon=True)
83         self.thread.start()
84     def change_duty_cycle(self, duty):
85         self.duty = max(0.0, min(1.0, duty))
86     def stop(self):
87         self.run = False
88         if self.thread and self.thread.is_alive(): self.thread.join(timeout=0.2)
89         self.gpio.output(self.pin, GPIO.LOW)
90
91 # ---- Funciones de Captura y Control ----
92 def flash_and_capture(camera):
93     """Captura ultrarrapida (buffer=1). Retorna (ret, frame)."""
94     return camera.read()
95 def soft_start_leds(pwm_list, target):
96     """Rampa suave de encendido para LEDs."""
97     print(f" -> [Power] Rampa LEDs -> {target*100:.0f} %...")
98     cur = 0.0
99     step = target / LED_RAMP_STEPS

```

```

100     for _ in range(LED_RAMP_STEPS):
101         cur = min(cur + step, target)
102         for p in pwm_list: p.change_duty_cycle(cur)
103         time.sleep(LED_RAMP_DELAY)
104     for p in pwm_list: p.change_duty_cycle(target)
105 def instant_leds_off(pwm_list):
106     """Apagado instantaneo de LEDs post—captura."""
107     print(" -> [Power] LEDs OFF (instantaneo)")
108     for p in pwm_list: p.change_duty_cycle(0.0)

```

C.4.3 Display OLED

```

1 #display.py
2 import time, datetime
3 try:
4     from luma.core.interface.serial import i2c
5     from luma.oled.device import sh1106
6     from luma.core.render import canvas
7     from PIL import ImageFont
8     LUMA_AVAILABLE = True
9 except ImportError:
10    LUMA_AVAILABLE = False
11
12 # ---- Constantes de Pantalla ----
13 I2C_BUS = 2 # Bus I2C en Orange Pi 5
14 I2C_ADDR = 0x3C # Direccion estandar OLED
15 # Variables globales del dispositivo
16 oled_device = None
17 font = None
18 serial_interface = None
19
20 def init_oled():
21     """
22     Inicializa la conexion I2C y el dispositivo OLED SH1106.
23     Retorna True si fue exitoso.
24     """
25     global oled_device, font, serial_interface
26     if not LUMA_AVAILABLE: return False
27     try:
28         serial_interface = i2c(port=I2C_BUS, address=I2C_ADDR)
29         oled_device = sh1106(serial_interface, width=128, height=64)
30         font = ImageFont.load_default()
31         oled_device.clear()
32         print("Pantalla OLED inicializada.")
33         return True
34     except Exception as e:
35         print(f"[OLED] Fallo inicializacion: {e}")
36         oled_device = None
37         return False
38
39 def update_oled_display(status, detail=""):
40     """
41     Muestra estado basico sin bateria (4 lineas).
42     Layout: Titulo(0), Linea(12), Estado(20), Detalle(40).
43     """
44     global oled_device
45     if not LUMA_AVAILABLE: return
46     def draw_screen():
47         with canvas(oled_device) as draw:
48             # Titulo fijo
49             draw.text((0, 0), "Analisis Hemoglobina", font=font, fill=255)
50             draw.line((0, 12, 128, 12), fill=255)
51             # Estado del sistema
52             draw.text((0, 20), f"Estado: {status}", font=font, fill=255)
53             # Detalle (truncado a 21 chars)
54             if detail:
55                 draw.text((0, 40), detail[:21], font=font, fill=255)
56     if oled_device:
57         try: draw_screen(); return
58     except: pass
59     # Intento de reinicializacion si falla
60     if init_oled():
61         try: draw_screen()
62     except: pass
63
64 def update_oled_with_battery(status, battery_mgr, detail=""):
65     """
66     Muestra estado con info de bateria al final (5 lineas).
67     Layout: Titulo(0), Linea(12), Estado(20), Detalle(32), Bateria(50).
68     """
69     global oled_device
70     if not LUMA_AVAILABLE: return

```

```

71 def draw_screen():
72     with canvas(oled_device) as draw:
73         draw.text((0, 0), "Análisis Hemoglobina", font=font, fill=255)
74         draw.line((0, 12, 128, 12), fill=255)
75         draw.text((0, 20), f"Estado: {status}", font=font, fill=255)
76         if detail:
77             draw.text((0, 32), detail[:21], font=font, fill=255)
78         # Info batería en la última línea
79         try:
80             bat_status = battery_mgr.get_battery_status()
81             draw.text((0, 50), f"Bat: {bat_status[:21]}", font=font, fill=255)
82         except:
83             draw.text((0, 50), "Bat: N/A", font=font, fill=255)
84     if oled_device:
85         try: draw_screen(); return
86     except Exception as e: print(f"[OLED] Error update bat: {e}")
87     if init_oled():
88         try: draw_screen()
89     except Exception as e: print(f"[OLED] Error reinit: {e}")
90
91 def show_shutdown_screen():
92     """Muestra mensaje de apagado sin cabeceras."""
93     global oled_device
94     if not oled_device: return
95     try:
96         with canvas(oled_device) as draw:
97             draw.text((10, 25), "APAGANDO", font=font, fill=255)
98             draw.text((10, 40), "DISPOSITIVO...", font=font, fill=255)
99     except Exception as e:
100         print(f"[OLED] Error shutdown screen: {e}")
101
102 def show_battery_warning_screen(battery_mgr):
103     """
104     Pantalla de advertencia batería baja.
105     Muestra tiempo restante y pide acción de usuario.
106     """
107     global oled_device
108     if not oled_device: return
109     try:
110         with canvas(oled_device) as draw:
111             draw.text((0, 0), "ATENCIÓN!", font=font, fill=255)
112             draw.line((0, 12, 128, 12), fill=255)
113             draw.text((0, 20), "Batería Baja", font=font, fill=255)
114             # Tiempo restante estimado
115             remaining = battery_mgr.get_remaining_time()
116             draw.text((0, 32), f"Quedan: {remaining}", font=font, fill=255)
117             draw.text((0, 50), "Presione botón", font=font, fill=255)
118     except Exception as e:
119         print(f"[OLED] Error warn screen: {e}")
120
121 def show_battery_critical_screen():
122     """Pantalla de error crítico por batería agotada."""
123     global oled_device
124     if not oled_device: return
125     try:
126         with canvas(oled_device) as draw:
127             draw.text((10, 15), "BATERÍA", font=font, fill=255)
128             draw.text((10, 30), "AGOTADA", font=font, fill=255)
129             draw.line((0, 42, 128, 42), fill=255)
130             draw.text((10, 48), "Recargue", font=font, fill=255)
131     except Exception as e:
132         print(f"[OLED] Error crit screen: {e}")
133
134 def update_oled_with_diagnosis(hgb_value, is_anemic):
135     """
136     Muestra resultado de hemoglobina y posible anemia.
137     Args:
138         hgb_value: float (g/dL)
139         is_anemic: bool
140     """
141     global oled_device
142     if not oled_device: return
143     try:
144         with canvas(oled_device) as draw:
145             draw.text((0, 0), "Resultado:", font=font, fill=255)
146             draw.line((0, 12, 128, 12), fill=255)
147             # Valor centrado aprox
148             draw.text((15, 22), f"{hgb_value:.2f} g/dL", font=font, fill=255)
149             if is_anemic:
150                 draw.text((5, 42), "POSIBLE ANEMIA", font=font, fill=255)
151     except Exception as e:
152         print(f"[OLED] Error diag screen: {e}")
153
154 def clear_oled():

```

```

155     """Limpia el contenido de la pantalla."""
156     global oled_device
157     if oled_device:
158         try: oled_device.clear()
159             except: pass

```

C.4.4 Procesamiento de inteligencia artificial

```

1 # PROCESAMIENTO DE INTELIGENCIA ARTIFICIAL
2 import cv2
3 import numpy as np
4 import time
5 from rknnlite.api import RKNNLite
6 from . import utils
7 # --- CONFIGURACION DE MODELOS ---
8 SEG_MODEL_PATH = 'ruta/hacia/modelo/unet_seg.rknn'
9 HGB_MODEL_PATH = 'ruta/hacia/modelo/hemo_convit.rknn'
10 SEG_INPUT_SIZE = (512, 512)
11 HGB_INPUT_SIZE = (224, 224)
12 #GFLOPS teoricos (Referencia)
13 SEG_MODEL_GFLOPS = 6.7
14 HGB_MODEL_GFLOPS = 2.8
15 # Umbral minimo de ROI para considerar valida la segmentacion
16 MIN_ROI_PERCENTAGE = 2.0
17
18 def _unified_preprocess(image_bgr, target_size):
19     """
20     PREPROCESAMIENTO UNIFICADO (NHWC — RGB — uint8)
21     Diseñado para modelos RKNN convertidos con:
22     — mean_values=[[0, 0, 0]]
23     — std_values=[[255, 255, 255]]
24     Esto delega la normalizacion matematica a la NPU.
25     Python solo entrega los bytes crudos de la imagen.
26     """
27     # 1. Redimensionar (Linear es el estandar para IA)
28     img_resized = cv2.resize(image_bgr, target_size, interpolation=cv2.INTER_LINEAR)
29     # 2. BGR a RGB (CRITICO: OpenCV lee BGR, los modelos se entrenaron en RGB)
30     img_rgb = cv2.cvtColor(img_resized, cv2.COLOR_BGR2RGB)
31     # 3. Expandir dimension de batch: (H, W, C) —> (1, H, W, C)
32     input_data = np.expand_dims(img_rgb, axis=0)
33     # 4. Asegurar tipo uint8 (0–255)
34     return input_data.astype(np.uint8)
35
36 def _keep_largest_island(mask_512):
37     """
38     Conserva solo el contorno mas grande (la conjuntiva) y elimina ruido.
39     """
40     contours, _ = cv2.findContours(mask_512, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
41     if not contours:
42         return mask_512
43     largest_contour = max(contours, key=cv2.contourArea)
44     clean_mask = np.zeros_like(mask_512)
45     cv2.drawContours(clean_mask, [largest_contour], -1, 255, thickness=cv2.FILLED)
46     return clean_mask
47
48 def _postprocess_unet_mask(output_data, original_shape):
49     """
50     Convierte la salida de la NPU en una mascara binaria limpia.
51     """
52     if output_data is None or len(output_data) == 0:
53         return None, None, 0.0
54     # 1. Aplanar salida. RKNN suele devolver (1, 1, 512, 512) o (1, 512, 512, 1)
55     mask_prob = output_data[0].squeeze()
56     # 2. Convertir probabilidad a 8 bits
57     if mask_prob.dtype == np.float32:
58         mask_8bit = (mask_prob * 255).astype(np.uint8)
59     else:
60         mask_8bit = mask_prob.astype(np.uint8)
61     # 3. Binarizacion
62     _, mask_binary_512 = cv2.threshold(mask_8bit, 127, 255, cv2.THRESH_BINARY)
63     # 4. Limpieza morfologica (quitar ruido/piel)
64     mask_cleaned_512 = _keep_largest_island(mask_binary_512)
65     # 5. Calcular porcentaje de ROI
66     total_pixels_512 = mask_cleaned_512.size
67     white_pixels_512 = np.sum(mask_cleaned_512 == 255)
68     roi_percentage = (white_pixels_512 / total_pixels_512) * 100.0
69     # 6. Redimensionar al tamaño original de la camara
70     mask_resized = cv2.resize(mask_cleaned_512, (original_shape[1], original_shape[0]),
71                               interpolation=cv2.INTER_NEAREST)
72
73     return mask_resized, mask_cleaned_512, roi_percentage
74

```

```

75 def run_analysis(frame, rknn_segformer, rknn_estimator):
76     """
77     Pipeline unificado: Segmentacion -> Limpieza -> Regresion HGB.
78     """
79     analysis_results = {
80         'success': False,
81         'metrics': {},
82         'images': {},
83         'error_message': 'Error desconocido en IA',
84         'error_code': None
85     }
86     try:
87         start_total = time.time()
88         # =====
89         # 1. ETAPA DE SEGMENTACION (UNET)
90         # =====
91         input_seg = _unified_preprocess(frame, SEG_INPUT_SIZE)
92         start_seg_inference = time.time()
93         outputs_seg = rknn_segformer.inference(inputs=[input_seg])
94         seg_inference_time = time.time() - start_seg_inference
95         if outputs_seg is None or outputs_seg[0] is None:
96             analysis_results['error_message'] = "Fallo inferencia segmentacion"
97             analysis_results['error_code'] = 'SEG_INFERENCE_FAIL'
98             return analysis_results
99         # Post-procesamiento
100         mask, mask_512, roi_percentage = _postprocess_unet_mask(outputs_seg, frame.shape)
101         # Validacion de ROI
102         if roi_percentage < MIN_ROI_PERCENTAGE:
103             print(f"ROI insuficiente: {roi_percentage:.2f} % < {MIN_ROI_PERCENTAGE} %")
104             analysis_results['error_message'] = f"ROI muy pequeno: {roi_percentage:.1f} %"
105             analysis_results['error_code'] = 'ROI_TOO_SMALL'
106             analysis_results['metrics']['roi_percentage_512'] = roi_percentage
107             analysis_results['images']['mask_512'] = mask_512
108             return analysis_results
109         if mask is None or np.sum(mask == 255) < 500:
110             analysis_results['error_message'] = "Conjuntiva no detectada"
111             analysis_results['error_code'] = 'NO_ROI_DETECTED'
112             return analysis_results
113         print(f"ROI valido: {roi_percentage:.2f} %")
114         analysis_results['images']['mask'] = mask
115         analysis_results['images']['mask_512'] = mask_512
116         analysis_results['metrics']['roi_percentage_512'] = roi_percentage
117         # Metricas de imagen
118         metrics, roi = utils.calculate_image_metrics(frame, mask)
119         metrics['seg_inference_time_ms'] = seg_inference_time * 1000
120         metrics['seg_effective_gflops'] = SEG_MODEL_GFLOPS / seg_inference_time if seg_inference_time > 0 else 0
121         analysis_results['images']['roi'] = roi
122
123         # =====
124         # 2. ETAPA DE ESTIMACION (HEMOCONVIT)
125         # =====
126         roi_recortado = utils.crop_black_borders(roi)
127         analysis_results['images']['roi_cropped'] = roi_recortado
128         if roi_recortado.size > 0:
129             # Usamos EL MISMO preprocesamiento unificado, solo cambiando el tamaño (224x224)
130             # Esto funciona porque el modelo RKNN ya tiene la configuración interna correcta
131             input_hgb = _unified_preprocess(roi_recortado, HGB_INPUT_SIZE)
132             # Guardamos imagen redimensionada para debug visual si se necesita
133             # (El input_hgb tiene dims extra y es RGB, aqui guardamos BGR para visualizacion)
134             analysis_results['images']['roi_resized'] = cv2.resize(roi_recortado, HGB_INPUT_SIZE)
135             start_hgb_inference = time.time()
136             outputs_hgb = rknn_estimator.inference(inputs=[input_hgb])
137             hgb_inference_time = time.time() - start_hgb_inference
138             if outputs_hgb is None or len(outputs_hgb) == 0:
139                 analysis_results['error_message'] = "Fallo estimacion HGB"
140                 analysis_results['error_code'] = 'HGB_INFERENCE_FAIL'
141                 return analysis_results
142             # Extraccion del valor (flotante)
143             hgb_value = float(outputs_hgb[0].reshape(-1)[0])
144             metrics['hemoglobin_g_dl'] = hgb_value
145             metrics['hgb_inference_time_ms'] = hgb_inference_time * 1000
146             metrics['hgb_effective_gflops'] = HGB_MODEL_GFLOPS / hgb_inference_time if hgb_inference_time > 0 else 0
147         else:
148             analysis_results['error_message'] = "ROI vacio tras limpieza"
149             analysis_results['error_code'] = 'ROI_EMPTY_AFTER_CROP'
150             return analysis_results
151         # Finalizacion
152         total_processing_time = time.time() - start_total
153         metrics['total_processing_time_ms'] = total_processing_time * 1000
154         analysis_results['metrics'] = metrics
155         analysis_results['success'] = True
156         analysis_results['error_message'] = ''
157         analysis_results['error_code'] = None
158         return analysis_results

```

```

159     except Exception as e:
160         print(f"ERROR CRÍTICO en run_analysis: {e}")
161         import traceback
162         traceback.print_exc()
163         analysis_results['error_message'] = "Error Interno IA"
164         analysis_results['error_code'] = 'INTERNAL_ERROR'
165         return analysis_results

```

C.4.5 Gestor de batería

```

1  import os, time, datetime, json
2  # ---- Constantes Gestion Batería ----
3  SESSION_LOG_FILE = os.path.expanduser("~/session_time.log")
4  WARNING_TIME_HOURS = 2.25 # 2:15 horas (10% restante)
5  SHUTDOWN_TIME_HOURS = 2.5 # 2:30 horas (Apagado forzoso)
6  UPDATE_INTERVAL_SECONDS = 600 # Guardado periodico 10 min
7
8  class BatteryManager:
9      """Gestor de tiempo de uso y estado de batería simulada."""
10     def __init__(self):
11         self.start_t = None
12         self.last_upd = None
13         self.acc_sec = 0
14         self.warn_shown = False
15         self.warn_ack = False
16         self.warn_t = None
17         self.load_previous_session()
18     def load_previous_session(self):
19         """Carga tiempo acumulado desde JSON si existe."""
20         try:
21             if os.path.exists(SESSION_LOG_FILE):
22                 with open(SESSION_LOG_FILE, 'r') as f:
23                     data = json.load(f)
24                     self.acc_sec = data.get('accumulated_seconds', 0)
25                     self.warn_shown = data.get('warning_shown', False)
26                     self.warn_ack = data.get('warning_acknowledged', False)
27                     print(f"[Bat] Sesión previa: {self._fmt(self.acc_sec)}")
28                     if self.warn_shown: print(f"[Bat] Warning previo detectado (ACK: {self.warn_ack})")
29             else:
30                 print("[Bat] Nueva sesión 0:00")
31                 self.acc_sec = 0
32     except Exception as e:
33         print(f"[Bat] Error carga sesión: {e}")
34         self.acc_sec = 0
35
36     def check_and_reset_if_needed(self):
37         """Resetea contadores si la sesión anterior terminó por batería baja (simula recarga)."""
38         if self.should_reset_on_next_boot():
39             print("[Bat] Detectado apagado por batería baja. Asumiendo recarga.")
40             self.reset_session_log()
41             return True
42         return False
43     def should_reset_on_next_boot(self):
44         """Determina si debe resetear basado en horas acumuladas."""
45         hrs = self.acc_sec / 3600.0
46         # Si supero límite de apagado o estaba en warning
47         return hrs >= SHUTDOWN_TIME_HOURS or hrs >= WARNING_TIME_HOURS
48     def start_session(self):
49         """Inicia cronómetro de sesión actual."""
50         self.start_t = time.time()
51         self.last_upd = self.start_t
52         print(f"[Bat] Inicio sesión. Acumulado: {self._fmt(self.acc_sec)}")
53     def update_time(self):
54         """Actualiza timestamp de última actividad."""
55         if self.start_t: self.last_upd = time.time()
56     def get_total_elapsed_hours(self):
57         """Retorna horas totales (acumulado + actual)."""
58         if self.start_t is None: return self.acc_sec / 3600.0
59         total = self.acc_sec + (time.time() - self.start_t)
60         return total / 3600.0
61     def should_show_warning(self):
62         """Verifica si cruzo umbral de advertencia (2:15)."""
63         return self.get_total_elapsed_hours() >= WARNING_TIME_HOURS and not self.warn_shown
64     def should_shutdown(self):
65         """Verifica si cruzo umbral de apagado (2:30)."""
66         return self.get_total_elapsed_hours() >= SHUTDOWN_TIME_HOURS
67     def mark_warning_shown(self):
68         """Registra que se mostro el warning."""
69         self.warn_shown = True
70         self.warn_t = time.time()
71         print("[Bat] Warning mostrado")
72     def acknowledge_warning(self):

```

```

73     """Usuario reconocio warning (boton presionado)."""
74     self.warn_ack = True
75     print("[Bat] Warning reconocido por usuario")
76     self.save_session_state()
77     def is_conscious_shutdown_after_warning(self):
78         """True si usuario apaga manualmente en zona critica (2:20-2:30)."""
79         if not self.warn_shown or not self.warn_ack: return False
80         h = self.get_total_elapsed_hours()
81         return WARNING_TIME_HOURS <= h < SHUTDOWN_TIME_HOURS
82     def needs_periodic_update(self):
83         """Verifica intervalo de guardado (10 min)."""
84         if self.last_upd is None: return False
85         return (time.time() - self.last_upd) >= UPDATE_INTERVAL_SECONDS
86     def save_session_state(self):
87         """Guarda estado actual en JSON."""
88         try:
89             total = self.acc_sec + (time.time() - self.start_t) if self.start_t else self.acc_sec
90             data = {
91                 'accumulated_seconds': total,
92                 'last_session_date': datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S'),
93                 'warning_shown': self.warn_shown,
94                 'warning_acknowledged': self.warn_ack
95             }
96             with open(SESSION_LOG_FILE, 'w') as f: json.dump(data, f, indent=2)
97             print(f"[Bat] Guardado: {self._fmt(total)}")
98         except Exception as e: print(f"[Bat] Error guardado: {e}")
99     def reset_session_log(self):
100         """Elimina log y reinicia contadores a 0."""
101         try:
102             if os.path.exists(SESSION_LOG_FILE): os.remove(SESSION_LOG_FILE)
103             self.acc_sec = 0
104             self.start_t = None
105             self.last_upd = None
106             self.warn_shown = False
107             self.warn_ack = False
108             print("[Bat] Log reiniciado (Bateria OK)")
109         except Exception as e: print(f"[Bat] Error reset: {e}")
110     def get_formatted_time(self):
111         """Retorna HH:MM transcurrido."""
112         h = self.get_total_elapsed_hours()
113         return f"{int(h):02d}:{int((h - int(h)) * 60):02d}"
114     def get_remaining_time(self):
115         """Retorna HH:MM restante hasta shutdown."""
116         rem = SHUTDOWN_TIME_HOURS - self.get_total_elapsed_hours()
117         if rem <= 0: return "00:00"
118         return f"{int(rem):02d}:{int((rem - int(rem)) * 60):02d}"
119     def get_battery_percentage(self):
120         """Estima % basado en tiempo (100 % -> 0h, 0 % -> 2.5h)."""
121         p = max(0, 100 - (self.get_total_elapsed_hours() / SHUTDOWN_TIME_HOURS) * 100)
122         return int(p)
123     def get_battery_status(self):
124         """String estado bateria (Ok/Media/Baja)."""
125         p = self.get_battery_percentage()
126         if p > 50: return f"OK ({p} %)"
127         if p > 20: return f"Media ({p} %)"
128         return f"Baja ({p} %)"
129     def _fmt(self, s):
130         """Helper formato HH:MM:SS."""
131         return f"{int(s//3600):02d}:{int((s%3600)//60):02d}:{int(s%60):02d}"

```

C.4.6 Utilidades

```

1 import cv2, os, psutil
2 import numpy as np
3 def get_thermal_metrics():
4     """Obtiene temperaturas NPU y SoC desde sysfs."""
5     temps = {'temp_npu_c': 0.0, 'temp_soc_c': 0.0}
6     base = "/sys/class/thermal/"
7     try:
8         for zone in os.listdir(base):
9             if not zone.startswith("thermal_zone"): continue
10            path = os.path.join(base, zone)
11            if os.path.exists(f"{path}/type"):
12                with open(f"{path}/type") as f: t_type = f.read().strip()
13                if t_type in ["npu-thermal", "soc-thermal"]:
14                    with open(f"{path}/temp") as f: val = int(f.read().strip()) / 1000.0
15                    if t_type == "npu-thermal": temps['temp_npu_c'] = val
16                    elif t_type == "soc-thermal": temps['temp_soc_c'] = val
17            except Exception as e: print(f"Error termico: {e}")
18        return temps
19 def get_power_metrics():
20     """Placeholder para metricas de energia."""

```

```

21     return {'voltage_v': 0.0, 'current_ma': 0.0, 'power_mw': 0.0}
22 def crop_black_borders(img):
23     """Recorta bordes negros usando contornos."""
24     if img is None or img.size == 0: return img
25     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
26     pts = cv2.findNonZero(gray)
27     if pts is None: return img
28     x, y, w, h = cv2.boundingRect(pts)
29     return img[y:y+h, x:x+w]
30 def get_sharpness_score(img):
31     """Calcula nitidez mediante varianza Laplaciana."""
32     if img is None or img.size == 0: return 0.0
33     return cv2.Laplacian(cv2.cvtColor(img, cv2.COLOR_BGR2GRAY), cv2.CV_64F).var()
34 def calculate_image_metrics(img, mask):
35     """Calcula metricas detalladas del sistema, imagen y ROI."""
36     m = {}
37     # --- Metricas de Sistema ---
38     m['memory_usage_mb'] = psutil.Process().memory_info().rss / 1048576
39     m.update(get_power_metrics())
40     m.update(get_thermal_metrics())
41     # --- Metricas Globales de Imagen ---
42     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
43     m['original_avg_brightness'] = float(np.mean(gray))
44     m['original_contrast_std'] = float(np.std(gray))
45     tot_px = img.shape[0] * img.shape[1]
46     m['sharpness_score'] = get_sharpness_score(img)
47     m['overexposed_percentage'] = float(np.sum(gray == 255) / tot_px) * 100
48     m['underexposed_percentage'] = float(np.sum(gray == 0) / tot_px) * 100
49     # --- Analisis ROI ---
50     roi = cv2.bitwise_and(img, img, mask=mask)
51     roi_px = img[mask > 0]
52     m['segmented_area_pixels'] = len(roi_px)
53     m['segmented_area_percentage'] = (len(roi_px) / tot_px) * 100
54     if len(roi_px) > 0:
55         gray_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)[mask > 0]
56         m['roi_avg_brightness'] = float(np.mean(gray_roi))
57         m['roi_contrast_std'] = float(np.std(gray_roi))
58         # Separar canales para vectorizacion
59         b, g, r = roi_px[:,0].astype(float), roi_px[:,1].astype(float), roi_px[:,2].astype(float)
60         m['average_r'], m['average_g'], m['average_b'] = float(np.mean(r)), float(np.mean(g)), float(np.mean(b))
61         # Indice de rojo (evita division por cero)
62         red_idx = (r - (g + b) / 2) / (r + 1e-6)
63         m['redness_index'] = float(np.mean(red_idx))
64     else:
65         # Valores por defecto si no hay ROI
66         keys_zero = ['roi_avg_brightness', 'roi_contrast_std', 'average_r', 'average_g', 'average_b', 'redness_index']
67         for k in keys_zero: m[k] = 0.0
68         for k in ['red_histogram', 'green_histogram', 'blue_histogram']: m[k] = [0]*256
69         m['redness_index_histogram'] = [0]*100
70         m['redness_index_bins'] = []
71
72     return m, roi

```

Anexo E: Especificaciones Técnicas


E.1. Orange Pi 5 Max



1. 3. Hardware features of Orange Pi 5

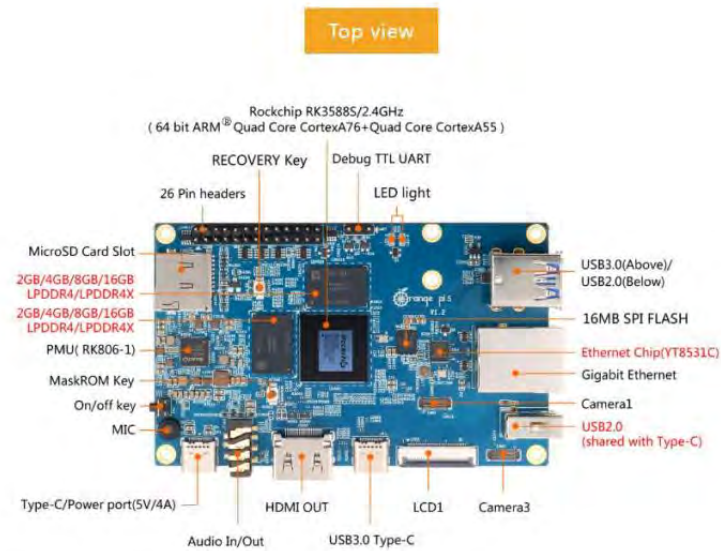
Introduction to hardware features	
CPU	<ul style="list-style-type: none">• Rockchip RK3588S (8nm LP processor)• 8-core 64-bit processor• 4-core Cortex-A76 and 4-core Cortex-A55 core architecture• The main frequency of the large core is up to 2.4GHz, and the main frequency of the small core is up to 1.8GHz
GPU	<ul style="list-style-type: none">• Integrated ARM Mali-G610• OpenGL ES1.1/2.0/3.2, OpenCL 2.2 and Vulkan 1.2
NPU	<ul style="list-style-type: none">• Built-in AI accelerator NPU with a computing power of up to 6 Tops• Support INT4/INT8/INT16 mixed operation
Video Output	<ul style="list-style-type: none">• HDMI 2.1, up to 8K @60Hz• DP1.4 (DisplayPort)• 2*MIPI D-PHY TX 4Lane
Memory	4GB/8GB/16GB (LPDDR4/4x)
Camera	<ul style="list-style-type: none">• 1 * MIPI CSI 4Lane• 2 * MIPI D-PHY RX 4Lane
PMU	RK806-1
Onboard Storage	<ul style="list-style-type: none">• 16MB QSPI Nor FLASH• MicroSD (TF) Card slot• PCIe2.0x1 M.2 M-KEY (SSD) slot
Ethernet	10/100/1000Mbps ethernet (YT8531C)
Audio	<ul style="list-style-type: none">• 3.5mm headphone jack audio in/out• Onboard MIC input• HDMI output
PCIe M.2 M-KEY	<ul style="list-style-type: none">• Support PCIe WIFI6+BT5.0+BLE• Support SSD
USB Interface	<ul style="list-style-type: none">1 * USB3.0 Interface2 * USB2.0 Interface (One of them is shared with the



	Type-C interface) 1 * USB3.0 Type-C Interface
26pin Extension Header	Used to expand UART, PWM, I2C, SPI, CAN and GPIO interfaces
Debug Serial Port	3pin debug serial port
LED Light	Power light and status light
Button	1 * Mask ROM key, 1 * RECOVERY, 1 * switch key
Power Supply	5V/4A Type-C power supply
Supported OS	Orange Pi OS (Droid)、Orange Pi OS (Arch)、Android12.1、Debian11、Ubuntu20.04 and Ubuntu22.04 operating systems
Introduction of Appearance Specifications	
Product Size	100mm*62mm
Weight	46g
 Orange Pi™ is a registered trademark of Shenzhen Xunlong Software Co., Ltd	



1.5. Interface details of Orange Pi 5



E.2. Cámara (Sensor CMOS)

C. Descripción de la cámara

- **Cámara de alta resolución:**Capture cada detalle con la cámara de alta resolución de la cámara web Mairuige 4K, que ofrece una resolución máxima de 3840x2160 para una calidad de video increíblemente clara.
- **Función de enfoque automático:**Nunca te preocupes por los videos borrosos de nuevo, ya que la función de enfoque automático de la cámara web garantiza que todo tu contenido esté enfocado, mejorando la experiencia visual general.
- **Micrófono integrado:**La cámara web viene con un micrófono integrado, proporcionando audio de alta calidad para complementar sus videos, por lo que es ideal para videoconferencias o clases en línea.
- **Interfaz USB:**Fácil conectividad con la Interfaz USB de la cámara web, lo que garantiza una integración perfecta con su PC o portátil para un uso sin complicaciones.
- **Compacto y portátil:**Su diseño compacto y portátil hace que sea fácil de transportar, perfecto para aquellos que siempre están en el camino.
- **Producto certificado:**Garantía de calidad con la webcam certificada por la FCC, garantizando su seguridad y fiabilidad de uso.

Velocidad de fotogramas de vídeo: 1080P:30fps,2K 4K:30fps
 Formato de salida: MJPG/YUY2
 Micrófono: Micrófono digital
 Interfaz: USB2.0
 Longitud del cable: 140cm
 Sistema de soporte: Windows XP, Win7,Win8,Win10M,Mac OS ,Vista, Linux



1080P

2K

4K

Sensor pixel 2 Mega
Resolution 1920x1080/30fps
Autofocus No
Microphone Yes
Format MJPEG, YUY2
Interface USB 2.0

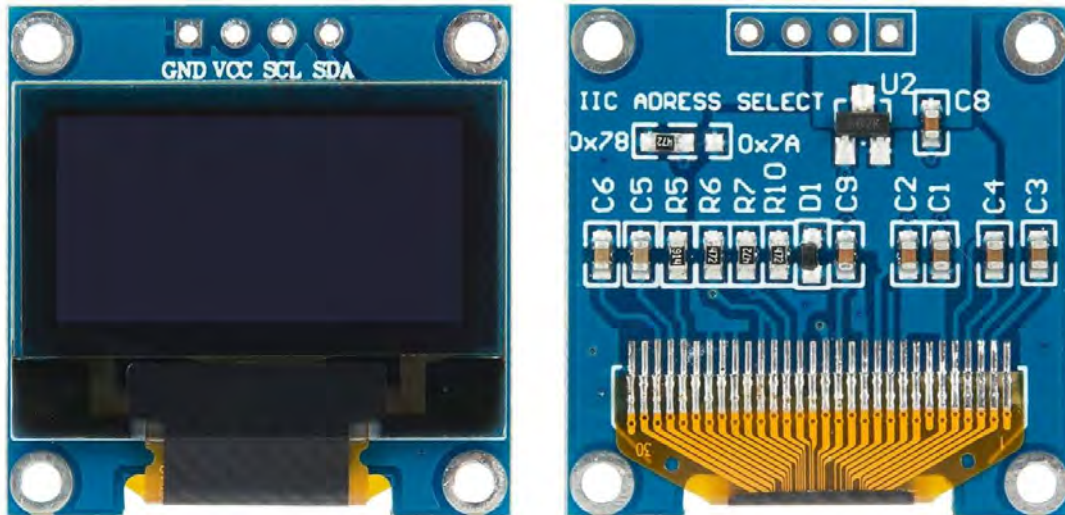
Sensor pixel 3 Mega
Resolution 2048x1536/30fps
Autofocus Yes
Microphone Yes
Format MJPEG, YUY2
Interface USB 2.0

Sensor pixel 8 Mega
Resolution 3840x2160/30fps
Autofocus Yes
Microphone Yes
Format MJPEG, YUY2
Interface USB 2.0

E.3. Pantalla OLED I2C

RG Rajguru Electronics (I) Pvt. Ltd.

OLED 4 Pin 128*64 Display Module 0.96" Blue Color



In contrast to LCD technology, Organic Light-Emitting Diode (OLED) displays do not require a backlight and are regarded as the ultimate technology for the next generation of flat-panel displays.

OLED displays are composed of a thin, multi-layered organic film placed between an anode and cathode, which are made up of electric conductive transparent Indium Tin Oxide.

The multi-layered organic film includes a Hole Transporting Layer, Emission Layer and Electron Transporting Layer.

By applying an appropriate electrical voltage, the holes and electrons are injected into the Emission Layer from the anode and cathode respectively and combine to form excitons, after which electroluminescence occurs.

This 0.96" 128*64 Blue OLED Module offers 128*64-pixel resolution. They are featuring much less thickness than LCD Displays with good brightness and produce better and true colors.

This OLED Display Module is very compact and will add a great ever user interface experience to your Arduino project. The connection of this display with Arduino is made through the I2C (also called as IIC) serial interface.

The 0.96" 4 pin 128*64 Blue OLED Display Module produces blue text on black background with very good contrast when supplied with 3.3V-5V Supply. The OLED Display Modules also offers a very wide viewing angle.

FEATURES:

- Supply voltage: 3.3V-5V
- Pixel: 128*64
- Display size- 0.96 inch

- Operating temperature range: -40°C - +80°C
- Use I2C Interface
- Chip No: SSD1306
- Color: Blue
- Drive Duty: 1/64 Duty
- Only need 2 I/O port to control
- Supported platforms: For Arduino,51 series, MSP430 series, STIM32/2, SCR chips
- Super high contrast and brightness(adjustable)
- Low power consumption
- High contrast, thus supporting clear display with no need of backlight
- For OLED SSD1306, a more elaborate and beautiful screen than LCD with more functions

PIN DESCRIPTION:

Pin No.	Pin Name	Description
1.	Supply Voltage (Vcc, 5V)	Can be powered by either 3.3V or 5V
2.	Ground (GND)	Pin Ground
3.	Serial Clock(SCL)	Pin SCL of I2C interface
4.	Serial Data(SDA)	Pin SDA of I2C interface

MECHANICAL SPECIFICATIONS:

ITEM	NORMAL DIMENSION
Module Dimension	27.30*27.30*2.37
Active Area	21.74*10.86
Pixel Size	0.148*0.148
Pixel Pitch	0.17*0.17

ABSOLUTE MAXIMUM RATING:

PARAMETER	SYMBOL	MIN	MAX	UNIT
Supply voltage for logic	VCC	1.65	5.5	V
Operating temperature	TOP	-40	+80	°C
Storage temperature	TSTG	-40	+80	°C

E.3. Sistema de Iluminación (LEDs Difusos)

TECHNICAL DATA SHEET

PRO-SIGNAL

**RoHS
Compliant**

5mm Diffused LEDs



Part No.	Pack Size	Colour	Wave Length at IF = 20mA			Forward Voltage (VF) at IF =20mA		Luminous Intensity at IF = 20mA	
			Min.	Typ.	Max.	Min.	Max.	Min.	Max.
PSG91860	10	Red	620nm		630nm	1.9V	2.1V	200mcd	300mcd
PSG91861		Green	565nm		575nm	1.9V	2.1V	50mcd	60mcd
PSG91862		Pure Green	520nm		525nm	3V	3.2V	1800mcd	2000mcd
PSG91863		Yellow	587nm		590nm	1.9V	2.1V	100mcd	150mcd
PSG91864		Blue	460nm	465nm	470nm	1.9V	2.1V	300mcd	400mcd
PSG91865		Warm White	3000k		4000k	3V	3.4V	3000mcd	4000mcd
PSG91866		Cool White	6000k		6500k	3V	3.4V	3000mcd	4000mcd
PSG91867		Red	620nm		630nm	1.9V	2.1V	200mcd	300mcd
PSG91868		Green	565nm		575nm	1.9V	2.1V	50mcd	60mcd
PSG91869		Pure Green	520nm		525nm	3V	3.2V	1800mcd	2000mcd
PSG91870	100	Yellow	587nm		590nm	1.9V	2.1V	100mcd	150mcd
PSG91871		Blue	460nm	465nm	470nm	1.9V	2.1V	300mcd	400mcd
PSG91872		Warm White	3000k		4000k	3V	3.4V	3000mcd	4000mcd
PSG91873		Cool White	6000k		6500k	3V	3.4V	3000mcd	4000mcd

Important Notice: This data sheet and its contents (the "Information") belong to pro-SIGNAL. No licence is granted for the use of it other than for information purposes in connection with the products to which it relates. No licence of any intellectual property rights is granted. The Information is subject to change without notice and replaces all data sheets previously supplied. The Information supplied is believed to be accurate but pro-SIGNAL assumes no responsibility for its accuracy or completeness, any error in or omission from it or for any use made of it. Users of this data sheet should check for themselves the Information and the suitability of the products for their purpose and not make any assumptions based on information included or omitted. Liability for loss or damage resulting from any reliance on the Information or use of it (including liability resulting from negligence or where pro-SIGNAL was aware of the possibility of such loss or damage arising) is excluded. This will not operate to limit or restrict pro-SIGNAL's liability for death or personal injury resulting from its negligence.

PRO-SIGNAL

TECHNICAL DATA SHEET

PRO-SIGNAL

**RoHS
Compliant**

Cont.

Part No.	Part Size	Colour	Viewing Angle	Max. Power Dissipation	Recommended Operating Forward Current	Max. Forward Current (IFM)	Reverse Voltage
PSG91860	10	Red	60°	120mW	15mA to 18mA	30mA	5V
PSG91861		Green	60°				
PSG91862		Pure Green	90°	80mW			
PSG91863		Yellow	60°				
PSG91864		Blue	60°	120mW			
PSG91865		Warm White	40°				
PSG91866		Cool White	40°	80mW			
PSG91867		Red	60°	120mW			
PSG91868		Green	60°				
PSG91869		Pure Green	90°	80mW			
PSG91870	100	Yellow	60°		15mA to 18mA	30mA	5V
PSG91871		Blue	60°	120mW			
PSG91872		Warm White	40°				
PSG91873		Cool White	40°	80mW			

Important Notice: This data sheet and its contents (the "Information") belong to pro-SIGNAL. No licence is granted for the use of it other than for information purposes in connection with the products to which it relates. No licence of any intellectual property rights is granted. The Information is subject to change without notice and replaces all data sheets previously supplied. The Information supplied is believed to be accurate but pro-SIGNAL assumes no responsibility for its accuracy or completeness, any error in or omission from it or for any use made of it. Users of this data sheet should check for themselves the Information and the suitability of the products for their purpose and not make any assumptions based on information included or omitted. Liability for loss or damage resulting from any reliance on the Information or use of it (including liability resulting from negligence or where pro-SIGNAL was aware of the possibility of such loss or damage arising) is excluded. This will not operate to limit or restrict pro-SIGNAL's liability for death or personal injury resulting from its negligence.

PRO-SIGNAL

E.4. Banco de poder

SPECIFICATIONS

RU: Характеристики | ES: Especificaciones | DE: Spezifikationen | FR: Caractéristiques | IT: Specifiche | PL: Specyfikacje | UA: Технічні характеристики | VI: Các thông số kỹ thuật | PT: Especificações

Model	J134
Capacity	5000 mAh / 18.5 Wh
Rated capacity	2600 mAh (5V, TYP 2A)
Input	USB-C: 5V / 2A
Output	USB-A: 5V / 2A USB-C cable / iP cable: 5V / 2A
Total output	5V / 2A (MAX)
Battery type	Lithium-ion polymer battery

WARNING

RU: Предупреждение | ES: Advertencia | DE: Warnung | FR: Avertissement | IT: Avvertimento | PL: Ostrzeżenie | UA: Попередження | VI: Cảnh báo | PT: Aviso

EN:

- If the power bank is not used for a long period, charge it at least once every three months.
- Keep the product dry and do not clean it with chemical agents.
- Do not allow children or unauthorized persons to use this device.
- Do not disassemble, crush, puncture, short-circuit, immerse in water or fire, or expose the product to temperatures above 60°C (140°F).

E.4. Instrumentación para Validación Energética

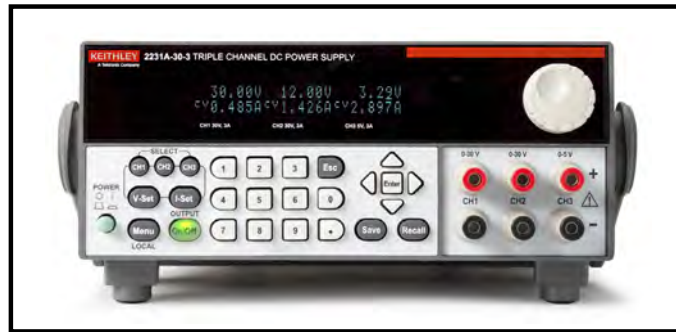
2231A-30-3

- Get three isolated, independent, and adjustable outputs in one instrument
- Output a total of 195W of power with two 30V@3A outputs and one 5V@3A output
- Set voltage outputs with 0.06% basic accuracy and current outputs with 0.2% basic accuracy
- Output DC power with less than 5mVp-p noise
- Display voltage and current measurements continuously from all three outputs
- Double output levels by connecting the two 30V channels in series or parallel
- Store frequently used configurations in any of 30 setup memory locations
- Turn off any output after a predetermined test time with each channel's output timer
- Control the supply from a PC with an optional USB adapter

APPLICATIONS

- Engineering and science student labs
- Service work
- Electronic design

195W Triple Channel DC Power Supply



The Model 2231A-30-3 Triple Channel DC Power Supply can output a total of 195W of power, providing the power levels needed to energize a wide range of circuits and devices for benchtop work. Two channels can supply up to 30V at 3A each; the third channel can provide up to 5V at 3A. The Model 2231A-30-1 does not compromise on performance or convenience features, offering the versatility and ease of use you need, so it can be the only DC power supply on your bench.

Quality DC Power

The Model 2231A-30-3 is a linear-based design with less than 5mVp-p noise per output. Voltage and current settings have basic accuracies of 0.06% and 0.2% respectively. The load voltage and load current readback results provide the same levels of accuracy, so you can be confident you are sourcing accurate, low noise voltages and currents to your device-under-test (DUT).

Three Fully Independent Outputs

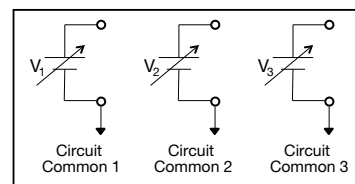
Each output of the Model 2231A-30-3 is fully programmable and can be turned on and off independently from the other channels; it essentially provides three power supplies in one instrument. You can power up both the analog circuitry and the digital circuitry of a printed circuit board or a complete device, all with the same instrument.

All Outputs Are Isolated from Each Other and from Ground

Each output can power a circuit on a different ground reference. The low connection on each output is not connected to any of the other outputs. That means one supply can test three separate circuits on three different ground references or can power circuits that are transformer or optically isolated from each other. Furthermore, all outputs are not referenced to ground, so the Model 2231A-30-3 can power both floating circuits and grounded circuits.

Power Bipolar Circuits

Because the Model 2231A-30-3's outputs are isolated, one output can be turned effectively into a negative source by connecting its high terminal to the common reference point of a bipolar circuit. A second output can connect its low terminal to the same common reference point. The result is a positive output and a negative output. If the two 30V channels are used in this configuration, both outputs can be changed at the same time by using the Model 2231A-30-3's tracking function. That allows varying both channels at the same time while keeping their outputs at identical magnitudes.



The Model 2231A-30-3 has three fully independent, programmable outputs that are electrically isolated from each other. This allows them to power circuits that are on different grounds, such as optically isolated circuits.

KEITHLEY
A Tektronix Company

**3-Year
Warranty**

195W Triple Channel DC Power Supply

DC POWER SUPPLIES

2231A-30-3

195W Triple Channel DC Power Supply

Specifications

DC OUTPUT RATING

	Channel 1	Channel 2	Channel 3
Voltage	0–30 V	0–30 V	0–5 V
Current	0–3 A	0–3 A	0–3 A

MAXIMUM POWER: 195W

LOAD REGULATION:

Voltage: $\leq 0.02\% + 4\text{mV}$
Current: $\leq 0.2\% + 3\text{mA}$

LINE REGULATION:

Voltage: $\leq 0.02\% + 4\text{mV}$
Current: $\leq 0.2\% + 3\text{mA}$

RIPPLE AND NOISE (20Hz–20MHz):

Voltage: $\leq 1\text{mVrms} \leq 5\text{mVp-p}$
Current: $\leq 6\text{Arms}$

SETTING RESOLUTION:

Voltage: 10mV
Current: 1mA

SETTING ACCURACY:

Voltage: $\leq 0.06\% + 20\text{mV}$
Current: $\leq 0.2\% + 10\text{mA}$

READBACK RESOLUTION:

Voltage: 10mV
Current: 1mA

READBACK ACCURACY:

Voltage: $\leq 0.06\% + 20\text{mV}$
Current: $\leq 0.2\% + 10\text{mA}$

ISOLATION VOLTAGE, OUTPUT TO CHASSIS: Any output can be isolated up to 240V (DC + peak AC with AC limited to a maximum of 3Vpk-pk and a maximum of 60Hz) relative to the earth ground terminal.

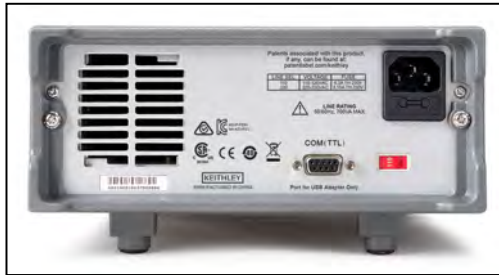
ISOLATION VOLTAGE, OUTPUT TO OUTPUT: Any output can be isolated up to 240V (DC + peak AC with AC limited to a maximum of 3Vpk-pk and a maximum of 60Hz) relative to any other output terminal.

TRACKING AND COMBINATION MODES:

Tracking Mode: Maintains the ratio on the two 30V output channels that is present when the control is activated.

Combination $V_1 + V_2$ Series Mode: Deliver up to 60V when CH1 and CH2 are wired in series. Meter reads back combined voltage.

Combination $I_1 + I_2$ Parallel Mode: Deliver up to 6A when CH1 and CH2 are wired in parallel. Meter reads back combined current.



Model 2231A-30-3 rear panel showing the communication port, line power setting switch, and the power input connector with fuse holder.

GENERAL

MEMORY: 30 setup memory locations.

OUTPUT TIMER RANGE: 0.1s to 999999s.

DISPLAY: Vacuum fluorescent display.

CONNECTIONS:

Front: Power output jacks: 3 sets, safety-shrouded banana jacks.

Rear: DB9 connector for remote control.

OVERTEMPERATURE PROTECTION: If the internal temperature of the supply exceeds 85°C, the supply will automatically turn off.

EMC COMPLIANCE: Conforms to European Union EMC Directive.

SAFETY COMPLIANCE: Conforms to European Union Low Voltage Directive.

POWER LINE RATINGS: 110VAC/230VAC $\pm 10\%$.

POWER LINE FREQUENCY: 47Hz–63Hz.

MAXIMUM POWER CONSUMPTION: 750VA.

OPERATING ENVIRONMENT: 0° to 40°C, 5% to 80% relative humidity up to 40°C.

STORAGE ENVIRONMENT: –20° to 70°C, 5% to 80% relative humidity up to 40°C, and 5% to 60% relative humidity from 40° to 70°C.

PHYSICAL CHARACTERISTICS

Height: 88.2 mm (3.5 in)

Width: 214.5 mm (8.5 in)

Depth: 354.6 mm (14 in)

Net Weight: 7.10kg (15.7 lbs.).

Shipping Weight: 9.40kg (20.7 lbs.).

WARRANTY: 3 years.

Model 2231A-30-3 specifications

DC POWER SUPPLIES

KEITHLEY
A Tektronix Company

Anexo F: Manual de Usuario del Prototipo

MANUAL DE USUARIO

Dispositivo No Invasivo de Estimación de Hemoglobina y Anemia

Gracias por su adquisición. Este dispositivo registra y analiza la conjuntiva inferior del paciente para determinar nivel de hemoglobina. Lea esta guía para garantizar una operación segura y precisa.

1. Conozca su Dispositivo

Identifique los componentes numerados:

- 1 Interruptor Principal (LED Rojo)**
Corte total de energía. Si el LED brilla, hay corriente.
- 2 Botón de Captura (Pulsador)**
Uso dual: **Click corto** para capturar foto / confirmar. **Click largo (4s)** para apagar el sistema (Apagado suave).
- 3 Pantalla OLED**
Interfaz visual de resultados y alertas.
- 4 Módulo de Adquisición**
Contiene la cámara y LEDs difusos. Diseñado para aislar la luz externa.
- 5 Puerto de Carga USB-C**
Úselo solo cuando el equipo esté apagado.

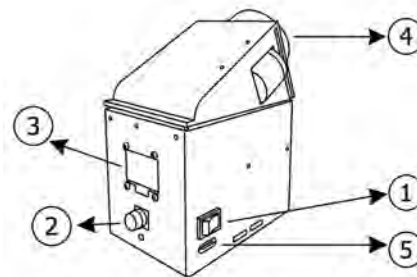


Figura 1: Vista del equipo.

2. Guía Rápida de Uso

Paso A: Encendido

Presione el **Interruptor Principal** **1**. El LED rojo se encenderá. La pantalla mostrará "Iniciando..". Espere a ver "Listo".

Paso B: Posicionamiento del Paciente



Figura 2: Posición correcta.

El sistema requiere una imagen limpia de la conjuntiva.

1. Apoye el borde superior del cilindro del **Módulo de Adquisición** **4** sobre la ceja (hueso orbital).
2. Pida al paciente que **mantenga el ojo abierto y mire hacia arriba**.
3. Solicite o ayude al paciente a bajar un poco la piel alrededor de la parte inferior del ojo con el dedo. Se recomienda bajar la piel alrededor del pómulo.
4. La cámara debe ver la parte rosácea interna del párpado inferior (conjuntiva).

Paso C: Captura

Pulse brevemente el **Botón de Captura** **2**. **Retire el dispositivo** apenas se apaguen los LEDs de iluminación interna del Módulo de Adquisición y/o la pantalla **3** muestre un aviso sobre la captura.

3. Lectura de Pantalla y Alertas

Mensaje	Acción Requerida
XX.XX g/dL	Éxito. Valor estimado. Si indica "POSIBLE ANEMIA", consultar médico.
Fallo en Segmentación	Error de Toma. El paciente parpadeó o no se vio la conjuntiva. Repita el paso B.
Batería Baja	Queda menos de 10% de carga. Pulse el botón 2 para confirmar. Finalice la sesión pronto.
Batería Agotada	Apagado Automático. El sistema ha alcanzado el límite de uso seguro y se apagará solo. Recargue inmediatamente.
Recargue antes de usar	Recordatorio al Apagar. Aparece si apaga el equipo tras recibir la alerta de batería baja.
ERROR FATAL / Reinicie	Fallo de Hardware. La cámara no responde. Apague y encienda desde el interruptor principal 1 .

4. Procedimientos de apagado

4. PROCEDIMIENTO DE APAGADO CORRECTO

Para proteger el sistema operativo y la tarjeta SD, siga este orden estricto:

- 1. Apagado de Software:** Mantenga presionado el ****Botón de Captura**** **2** durante **4 segundos** o más.
- Espera a que la pantalla diga "Apagando..." y se ponga negra completamente.
- 3. Corte de Energía:** Una vez apagada la pantalla, presione el ****Interruptor Principal**** **1** para cortar la corriente. El LED rojo interno de ese botón debe apagarse.

5. Precauciones de Batería

- **NO usar mientras carga:** Desconecte el cargador antes de examinar a un paciente.
- **Modo de Carga:** Cargue solo cuando el interruptor principal **1** esté APAGADO (LED apagado).
- **Tiempo:** Se recomienda cargar por al menos 1 hora cuando reciba la alerta de batería baja antes de una siguiente sesión de uso.