

Diseño e implementación de un sistema embebido multimodal para la detección de fatiga visual en estudiantes basado en ESP32-CAM y Edge Impulse

Design and implementation of a multimodal embedded system for visual fatigue detection in students based on ESP32-CAM and Edge Impulse

Franz J. JIMÉNEZ-VILLALVA¹

Alessandro T. ROSAS-MIRABAL²

Cristian CASTRO-VARGAS³

Maritza R. CABANA-CACERES⁴

¹ Universidad Nacional Federico Villarreal, Lima, Perú. Email: 2024020454@unfv.edu.pe.

² Universidad Nacional Federico Villarreal, Lima, Perú. Email: 2024020686@unfv.edu.pe.

³ Universidad Nacional Federico Villarreal, Lima, Perú. Email: ccastrov@unfv.edu.pe. ORCID: <https://orcid.org/0000-0002-9696-8635>

⁴ Universidad Nacional Federico Villarreal, Lima, Perú. Email: mcabana@unfv.edu.pe. ORCID: <https://orcid.org/0000-0002-3442-5950>

RESUMEN

El Síndrome Visual Informático afecta a una alta proporción de estudiantes universitarios peruanos. Se diseñó, implementó y demostró funcionalmente un sistema embebido multimodal de bajo costo (S/ 125) para la detección de fatiga visual basado en ESP32-CAM con inferencia TinyML mediante Edge Impulse y sensores BH1750, MPU6050 e INMP441. El clasificador ocular alcanzó 90.0% de accuracy (F1: 0.900) sobre el conjunto de validación. Se incluye el firmware ejecutable y una demostración funcional del sistema completo con tres usuarios, verificando la operación de extremo a extremo.

Palabras clave: fatiga visual digital, ESP32-CAM, Edge Impulse, TinyML, PERCLOS

ABSTRACT

Computer Vision Syndrome affects a high proportion of Peruvian university students. A low-cost (S/ 125) multimodal embedded system for visual fatigue detection was designed, implemented and functionally demonstrated, based on ESP32-CAM with TinyML inference via Edge Impulse and BH1750, MPU6050 and INMP441 sensors. The eye classifier reached 90.0% accuracy (F1: 0.900) on the validation set. The executable firmware and a functional demonstration of the complete system with three users are included, verifying end-to-end operation.

Keywords: digital eye fatigue, ESP32-CAM, Edge Impulse, TinyML, PERCLOS

Recibido: 05/06/2026

Aprobado: 15/07/2026

Publicado: 30/07/2026

1. INTRODUCCIÓN

La detección automática y temprana de la fatiga visual mediante sistemas embebidos de bajo costo constituye un problema de ingeniería todavía abierto: las soluciones disponibles dependen mayoritariamente de visión por computadora ejecutada en computadoras de escritorio o servicios en la nube, lo que implica alta dependencia de infraestructura costosa y conectividad permanente (Safarov et al., 2023; Dewi et al., 2022). El paradigma de computación en el borde con microcontroladores como el ESP32-CAM permite trasladar la inferencia al punto de uso sin servidores externos (Gupta et al., 2023; Adi, 2022), aunque la gestión térmica del módulo es crítica en sesiones prolongadas (Nowroz et al., 2025). Este trabajo aborda el diseño, la implementación y la demostración funcional de un sistema embebido que integra estas capacidades en un dispositivo autónomo y económico.

El Síndrome Visual Informático (SVI), conocido como Computer Vision Syndrome, hace referencia a un conjunto de síntomas tanto oculares como no oculares derivados del uso excesivo de las pantallas. La visión borrosa, la sequedad ocular o las cefaleas son los síntomas más frecuentes (Kaur et al., 2022), además de que su prevalencia supera el 66% a nivel mundial entre los usuarios regulares de computadoras (Anbesu y Lema, 2023), además de que se observa un aumento en la misma a raíz de la pandemia de COVID-19, en función del incremento del teletrabajo o la educación en línea (León-Figueroa et al., 2024), incluso con evidencias que sugieren que los síntomas empeoran conforme lo hace el tiempo de exposición (Beeson et al., 2024). En el caso de estudiantes universitarios peruanos se han reportado prevalencias altas asociadas a sesiones de más de cuatro horas diarias frente a pantallas (Huyhua-Gutiérrez et al., 2023; Estrada Araoz et al., 2022; Lindo-Cano et al., 2022), todo lo cual invita al desarrollo de herramientas de control a modo de prevención accesibles a la realidad educativa local.

La frecuencia de parpadeo es el biomarcador fundamental del SVI, ya que esta disminuye con el uso de pantallas durante un tiempo considerable y se relaciona inversamente a la intensidad de los síntomas (Lapa et al., 2023). Las condiciones de inferioridad de luz entre 150 y 300 lux incrementan el riesgo de presentar síndrome de fatiga visual (McKee & Hedge, 2022) lo cual condice con la integración de sensorado lumínico junto con el esquema multimodal. Los sistemas de seguimiento ocular en tiempo real y de detección de parpadeos por señales de bajo consumo constituyen líneas activas para la detección automatizada del SVI (Witte & Lins, 2023; Sharmila et al., 2024; Liu et al., 2024).

Las contribuciones aportadas por el presente artículo son: (1) una arquitectura embebida de cuatro etapas sobre ESP32-CAM con Edge Impulse que integra análisis ocular mediante PERCLOS, lumínico (BH1750), postural (MPU6050) y auditivo (INMP441); (2) el firmware ejecutable completo del sistema, incluyendo el cálculo de PERCLOS y la lógica de alerta; y (3) el prototipo funcional desarrollado como una demostración del sistema funcional extremo a extremo con usuarios reales, respaldada por la evidencia de la aplicación móvil. Considerando que la validación cuantitativa con una muestra representativa se plantea explícitamente como trabajo futuro.

El análisis causa-efecto mediante el diagrama de Ishikawa (Figura 1) identificó seis categorías de factores contribuyentes al SVI. El sistema desarrollado aborda directamente las categorías Machine (pantallas), Environment (iluminación) y Manpower (conciencia preventiva mediante alertas).



Figura 1. Diagrama de Ishikawa: factores contribuyentes al Síndrome Visual Informático en estudiantes

2. METODOLOGÍA

El desarrollo siguió un enfoque de ingeniería estructurado en cinco fases secuenciales (Figura 2): identificación de variables, selección de componentes, diseño del circuito, desarrollo del firmware y construcción del prototipo con su demostración funcional. El énfasis del trabajo recae en la implementación del firmware ejecutable y la integración física del sistema, descritos en las secciones siguientes.

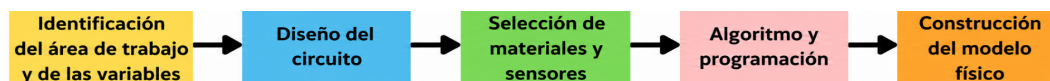


Figura 2. Fases del desarrollo del sistema embebido

2.1. Contexto de aplicación

El prototipo se orientó a entornos educativos universitarios de Lima, donde la alta densidad de salas de cómputo y el uso intensivo de laptops y tablets sostienen una exposición continua a pantallas (Lindo-Cano et al., 2022). Las Figuras 3 y 4 caracterizan la distribución de universidades por departamento y la matrícula nacional por campo de estudio, contextualizando el ámbito de aplicación.

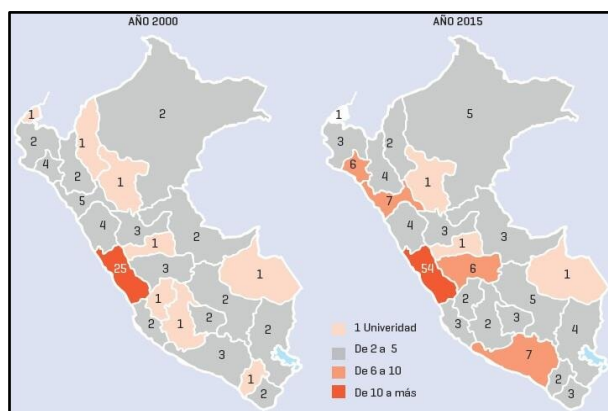
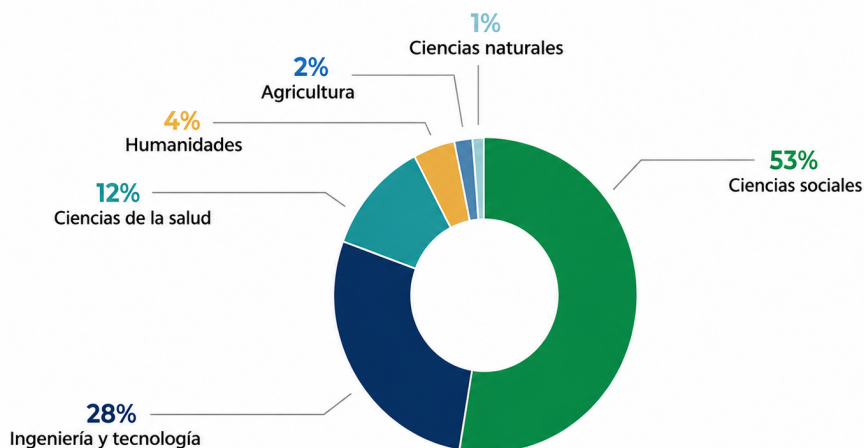


Figura 3 Distribución geográfica de universidades por departamento en el Perú (2000 y 2015)

Matrícula en educación superior por campo de estudio (2017)



Fuente: UNESCO

© 2022 Servicios Mundiales de Educación
www.wes.org



Figura 4 Distribución porcentual de la matrícula universitaria por campo de estudio

2.2. Selección de componentes

La selección se rigió por bajo costo, bajo consumo, compatibilidad con Edge Impulse y suficiencia técnica para las métricas del SVI. La Tabla 1 resume las especificaciones técnicas y el costo de cada componente a precios del mercado local (Paruro, Lima).

Tabla 1. Especificaciones técnicas, consumo y costo de los componentes del sistema

Componente	Función principal	V (V)	I prom. (mA)	I pico (mA)	Fórmula	Costo
ESP32-CAM	Procesamiento, captura OV2640 e inferencia TinyML	5	250	600	$P = V \cdot I$	S/ 50
BH1750	Illuminances 1-65,535 lux; alerta < 300 lux	3-5	0.12	0.18	$E = I/A$	S/ 20
MPU6050	IMU 6 ejes; inclinación de cabeza > 30°	3.3	3.9	5.5	$\theta = \text{atan2}$	Incl.
INMP441	Micrófono MEMS I ² S; detección de bostezos	3.3	1.4	3.6	SPL dBFS	S/ 30
LED indicador	Alerta visual local	3-5	10	20	N/A	S/ 5
Batería 18650 (x2)	Alimentación portátil; 7-14 h	3.7	N/A	N/A	$t = C/I$	S/ 20
TOTAL	Consumo ~305 mA prom. / 830 mA pico @ 5V	5	~305	~830		S/ 125

2.3. Arquitectura del sistema

La arquitectura (Figura 5) se organiza en cuatro etapas: adquisición (ESP32-CAM/OV2640, BH1750, MPU6050, INMP441); control (inferencia TinyML, cálculo de PERCLOS y evaluación ambiental en el ESP32-CAM); comunicación (envío a ThingSpeak por Wi-Fi); y visualización (aplicación móvil del estudiante). El diagrama lógico de la Figura 6 detalla el flujo desde la alimentación hasta la notificación.

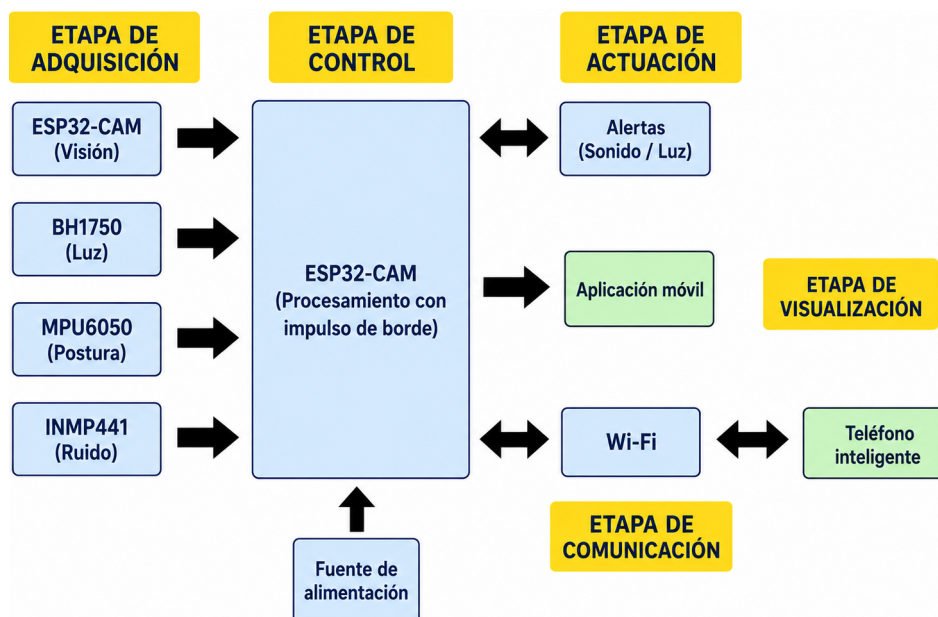


Figura 5. Diagrama de bloques del sistema multimodal

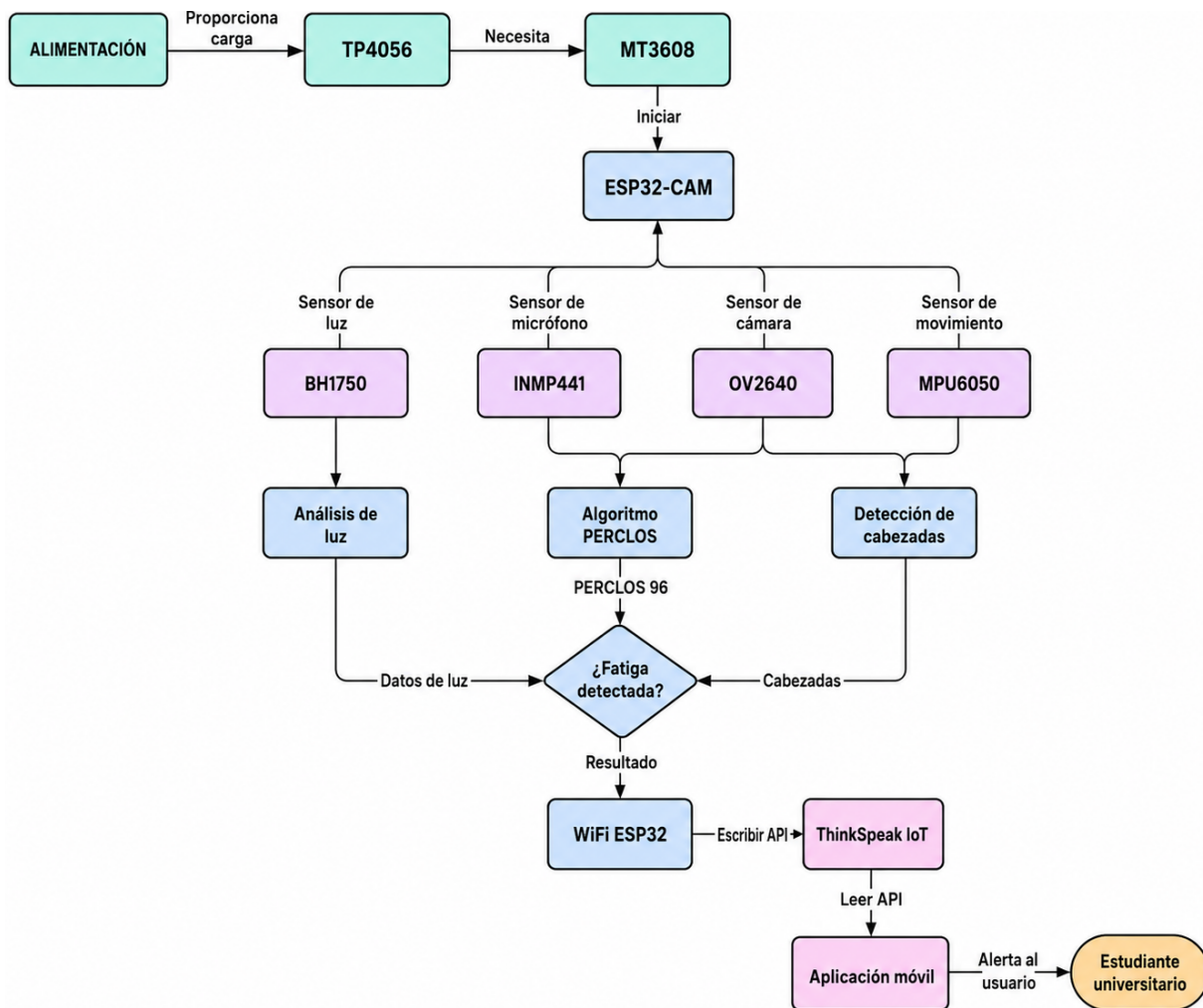


Figura 6. Diagrama lógico del sistema: flujo de los sensores hasta notificación

2.4. Diseño del circuito y modelo 3D

El ESP32-CAM se conecta al BH1750 y al MPU6050 por bus I²C, al INMP441 por bus I²S y al LED indicador por GPIO. El módulo TP4056 gestiona la carga de las baterías 18650 y el convertidor MT3608 eleva la tensión a 5 V. Las Figuras 7 y 8 muestran la disposición física y el esquema del circuito; las Figuras 9 a 11, las vistas del modelo 3D de la carcasa impresa en PLA.

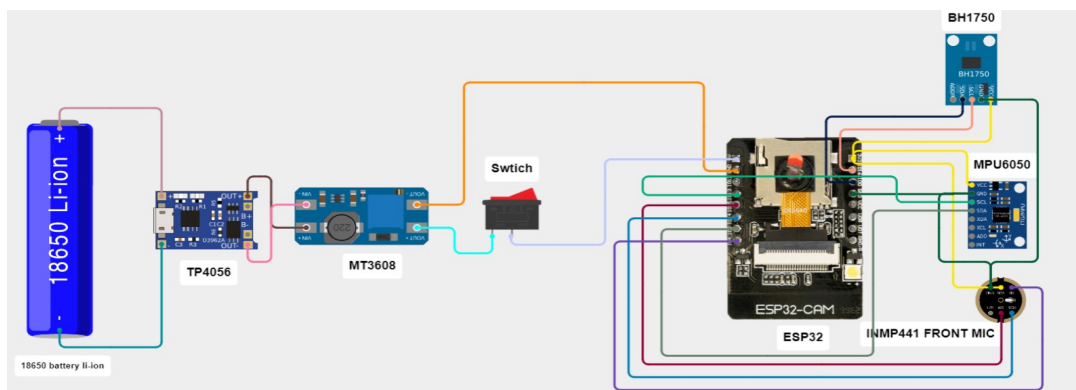


Figura 7. Disposición física de los componentes

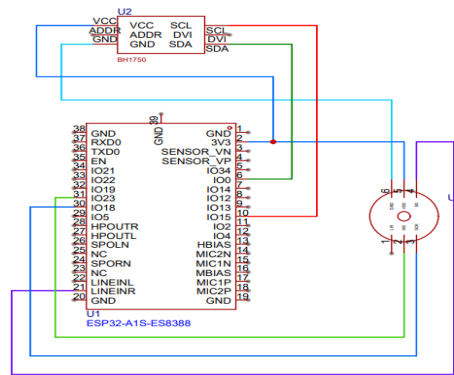


Figura 8. Esquema del circuito (ESP32-CAM y sensores)

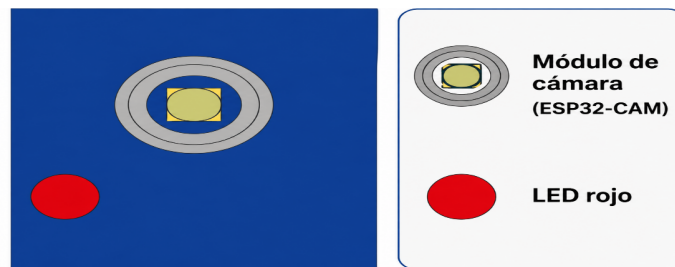


Figura 9. Modelo 3D de la carcasa: vista superior

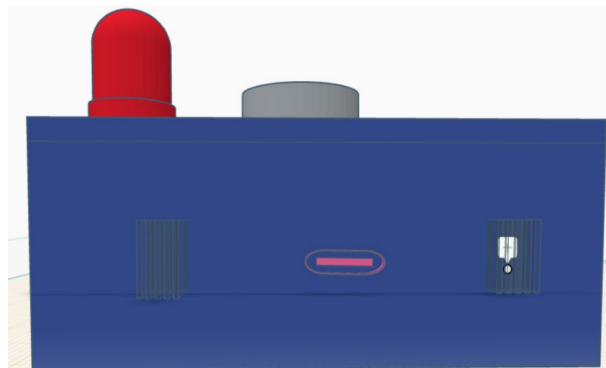


Figura 10. Modelo 3D: vista lateral



Figura 11. Modelo 3D: vista frontal

2.5. Entrenamiento del modelo y firmware ejecutable

El clasificador del estado ocular (abierto/cerrado) se entrenó en Edge Impulse con la arquitectura MobileNetV1 cuantizada a int8. El dataset comprendió 2,400 imágenes por clase, capturadas por los

autores bajo cuatro niveles de iluminación, divididas en 80% entrenamiento, 10% validación y 10% prueba; el preprocesamiento aplicó escala de grises, reescalado a 96x96 px y normalización. El modelo cuantizado ocupa 0.6 MB y registra una latencia de inferencia de 87 ms por fotograma.

El firmware se desarrolló en Arduino/C++ e integra la captura de la cámara, la ejecución del clasificador exportado por Edge Impulse, el cálculo del indicador PERCLOS en ventanas de 30 s y el envío a ThingSpeak. Los fragmentos siguientes presentan la lógica central del sistema (Véase Fragmento 1, Fragmento 2 y Fragmento 3).

Fragmento 1. Parámetros de configuración de PERCLOS y variables de estado.

```
#define PERCLOS_WINDOW_SECONDS 30 // ventana de observacion
#define PERCLOS_THRESHOLD 20.0 // umbral de fatiga (%)
#define FRAME_INTERVAL_MS 100 // periodo del bucle

static uint32_t frame_count = 0; // cuadros totales
static uint32_t closed_eye_frames = 0; // cuadros 'cerrado'
static uint32_t start_time = 0; // inicio de ventana
static float perclos_percentage = 0.0;
static bool fatigue_detected = false;
```

Fragmento 2. Bucle principal: inferencia y conteo de cierre ocular.

```
void loop() {
    ei_sleep(FRAME_INTERVAL_MS);
    frame_count++;
    // captura + clasificacion con el modelo Edge Impulse
    ei_impulse_result_t result = { 0 };
    run_classifier(&signal, &result, debug_nn);

    bool open_detected = false;
    for (uint16_t i = 0; i < EI_CLASSIFIER_LABEL_COUNT; i++) {
        if (strcmp(ei_classifier_inferencing_categories[i],
            "abierto") == 0 &&
            result.classification[i].value > 0.5) {
            open_detected = true; break;
        }
    }
    int estado = open_detected ? 0 : 1; // 1 = cerrado
    if (!open_detected) closed_eye_frames++;
    sendToThingSpeak(0, estado, 0);
    check_perclos_window();
}
```

Fragmento 3. Cálculo de PERCLOS y disparo de la alerta de fatiga.

```
void check_perclos_window() {
    uint32_t now = millis();
    if (now - start_time >= PERCLOS_WINDOW_SECONDS * 1000) {
        perclos_percentage =
            (float)closed_eye_frames / frame_count * 100.0;
        if (perclos_percentage > PERCLOS_THRESHOLD) {
            fatigue_detected = true; // activa alerta
            sendToThingSpeak(perclos_percentage, 1, 1);
        }
        frame_count = 0; closed_eye_frames = 0;
        start_time = now; // reinicia ventana
    }
}
```

El firmware completo, el modelo entrenado y los recursos del proyecto están disponibles bajo solicitud razonada al autor de correspondencia.

El diagrama de flujo del firmware (Figura 12) resume el ciclo: inicialización, captura e inferencia por fotograma, conteo de cierre ocular, evaluación de la ventana PERCLOS y, ante fatiga, activación de la alerta y reporte a ThingSpeak. La arquitectura de comunicación (Figura 13) describe el envío de datos hacia la aplicación móvil del usuario.

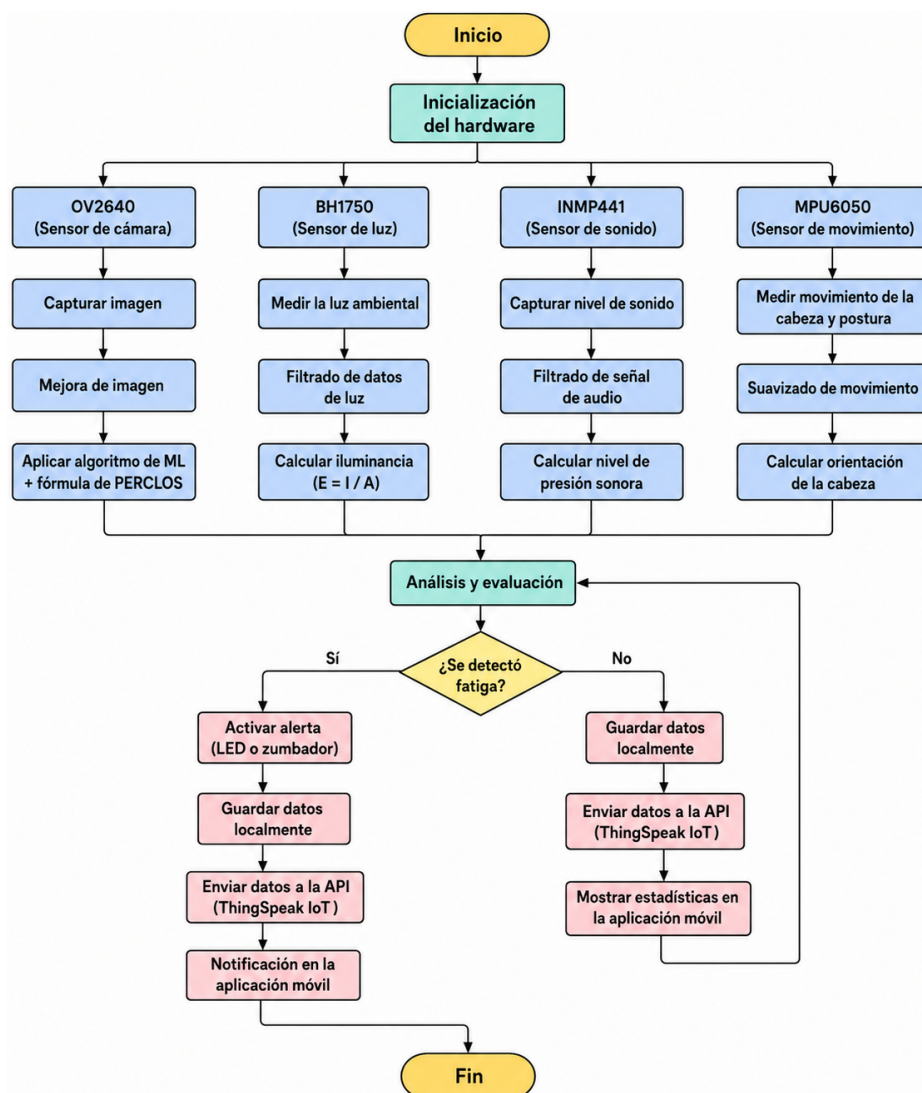


Figura 12. Diagrama de flujo del firmware del sistema

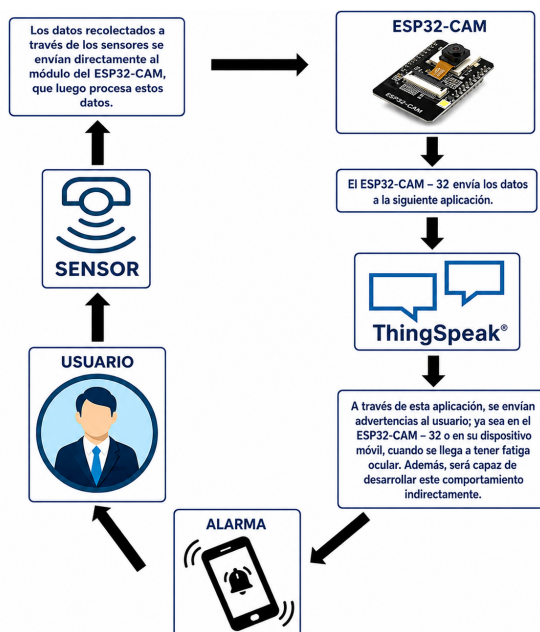


Figura 13. Arquitectura de comunicación: ESP32-CAM, ThingSpeak y aplicación móvil

3. RESULTADOS

3.1. Rendimiento del clasificador en el conjunto de validación

La Tabla 2 reporta el rendimiento del clasificador ocular sobre el conjunto de validación de Edge Impulse, no sobre usuarios. El modelo alcanzó 90.0% de accuracy promedio (F1: 0.900), con su mejor desempeño en iluminación óptima (94.2%) y el menor en iluminación deficiente (84.7%), lo que confirma la dependencia del clasificador respecto de las condiciones lumínicas y motiva la alerta ambiental basada en el BH1750.

Tabla 2. Rendimiento del clasificador ocular sobre el conjunto de validación de Edge Impulse

Condición del conjunto de validación	Accuracy (%)	Precisión (%)	Recall (%)	F1-score
Iluminación óptima (> 500 lux)	94.2	93.8	94.7	0.943
Iluminación moderada (300-500 lux)	91.5	90.9	92.1	0.915
Iluminación deficiente (< 300 lux)	84.7	83.1	86.3	0.847
Promedio del conjunto de validación	90.0	89.1	90.8	0.900

3.2. Demostración funcional del sistema completo

Se realizó una demostración funcional de extremo a extremo del prototipo con tres usuarios voluntarios del equipo, con el objetivo de verificar la operación integrada del sistema (captura, inferencia embebida, transmisión y visualización), no de obtener métricas estadísticas de precisión. La Tabla 3 resume las condiciones y la respuesta observada del sistema en cada caso.

Tabla 3. Demostración funcional del sistema completo con tres usuarios (verificación cualitativa)

Usuario	Condición de prueba	Evento inducido	Respuesta observada del sistema en la app
U1	Iluminación de aula (~350 lux)	Cierre ocular sostenido	Fatiga ocular reflejada en la app (50%); alerta visual activada
U2	Iluminación de aula (~350 lux)	Bostezo simulado	Nivel de bostezo reflejado en la app (65%); recomendación mostrada
U3	Iluminación reducida (< 300 lux)	Uso normal bajo baja luz	Luminosidad baja reflejada en la app (37%); recomendación de mejorar iluminación

La Figura 14 muestra la salida real de la aplicación móvil durante la demostración: a la izquierda, la pantalla de bienvenida; a la derecha, la pantalla de resultados, que reporta los indicadores medidos por el sistema (fatiga ocular 50%, nivel de bostezo 65% y luminosidad 37%) junto con recomendaciones preventivas concretas. Esta evidencia confirma que el flujo completo del sistema, desde la captura en el ESP32-CAM y la inferencia embebida hasta la visualización de indicadores cuantificados en el dispositivo del usuario, opera de manera integrada y produce salidas accionables.

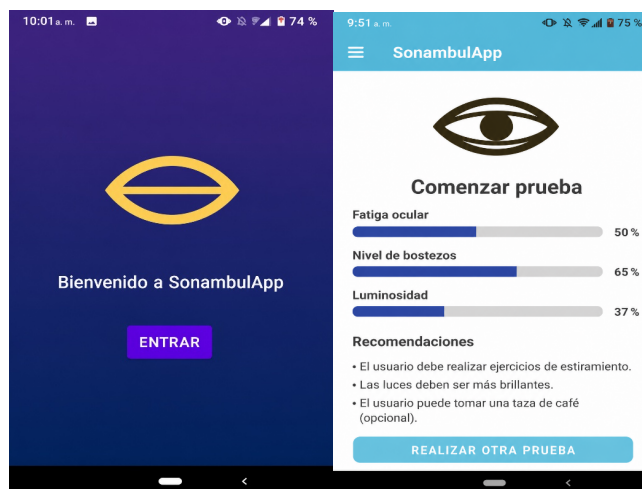


Figura 14. Salida real de la aplicación móvil durante la demostración funcional: pantalla de bienvenida (izquierda) y pantalla de resultados con indicadores de fatiga ocular, bostezo, luminosidad y recomendaciones (derecha)

3.3. Costo del prototipo

El costo para los componentes totaliza S/ 125 (el equivalente a unos 33 USD), incluyendo la electrónica propia, la carcasa impresa en 3D y baterías. Para soluciones comerciales basadas en computadoras, que elevan un precio en torno de los S/ 1,500, este prototipo representaría una alternativa más económica, por lo que podría considerarse una alternativa para los entornos educativos con recursos reducidos.

4. DISCUSIÓN

El rendimiento del clasificador (90.0% de precisión asociado al conjunto de validación) corresponde a un comportamiento conocido para clasificaciones desde ESP32-CAM y propiedades de aspecto ocular (Gupta et al., 2023; Dewi et al., 2022; Kraft et al., 2022). La caída a 84.7% en luminosidad baja es predecible, tanto como se indican las limitaciones del módulo de captura por el OV2640 (Adi, 2022; Nowroz et al., 2025) y apoya la decisión de diseño sobre el comportamiento implementado del módulo de captura del BH1750 como mecanismo de alerta adicional para llegar a la conclusión de que se han cruzado condiciones de iluminancia de por debajo de 300 lux (McKee & Hedge, 2022).

4.1. Análisis de falsos positivos y fuentes de error

Supondremos que el sistema tiene fuentes de errores que hay que contemplar en cualquier puesta en práctica real. Así, el canal acústico ya cuenta con el INMP441 que, por supuesto, es sensible al ruido ambiental que se recibe en el aula (conversaciones, ventiladores, tráfico), el cual puede confundirse con la firma espectral de un bostezo, provocando falsos positivos; evitarlo requeriría umbrales adaptativos y, por supuesto, filtrados espectrales todavía no implementados. Para el canal visual, también la oclusión secundaria (lentes con reflejos, propias manos, ángulos extremos) en el rostro o la contraluz intensa pueden inducir la clasificación errónea de los cierres de los ojos, inflando el valor de PERCLOS. En el caso del canal postural, el MPU6050 puede registrar inclinaciones de cabeza que no tengan relación con la fatiga (consultar el material en el pupitre), con falsos positivos posturales. La fusión multimodal atenúa parcialmente estos efectos al requerir que los canales se escaneen coincidan, pero la caracterización formal de las tasas de falsos positivos y negativos queda pendiente.

4.2. Limitaciones del estudio

Este escrito representa una prueba de concepto de tipo funcional. Las principales limitaciones son: (a) las métricas de precisión provienen del conjunto de validación del modelo y no de una muestra de sujetos; (b) la demostración funcional implicó únicamente a tres usuarios e implicó un carácter cualitativo, siendo sin recogida de datos demográficos ni medición contra criterio de referencia clínico; (c) la transmisión a ThingSpeak no se caracterizó según las condiciones de la red universitaria real. La verificación cuantitativa con muestra representativa, protocolo de medida formal, consentimiento informado y comparación contra ground-truth se plantea como la principal línea de trabajo futuro.

5. CONCLUSIONES

Se diseñó, implementó y demostró funcionalmente un sistema embebido multimodal de bajo coste (S/ 125) para la detección de fatiga visual basado en ESP32-CAM y Edge Impulse. El clasificador ocular alcanzó 90.0% de precisión (F1: 0.900) sobre el set de validación, y se presentó el firmware ejecutable que implementa el cálculo de PERCLOS y la lógica de alerta. La demostración funcional con tres usuarios validó la operación extremo a extremo del prototipo, la cual fue evidenciada por la salida de la aplicación móvil.

La contribución central es un dispositivo autónomo, de bajo coste y reproducible con su firmware, el cual traslada la inferencia al borde y reduce la dependencia de la nube. Las líneas de trabajo futuro prioritarias son: la validación cuantitativa con una muestra representativa de estudiantes; la caracterización formal de falsos positivos por canal; la incorporación de filtrado adaptativo en el

canal acústico; y el rediseño en PCB para la mejora del manejo térmico del ESP32-CAM en sesiones prolongadas.

Declaración de Ética, Transparencia y Uso de Inteligencia Artificial

Ética y transparencia

Originalidad y plagio: Los autores declaran que el manuscrito es original e inédito, no publicado ni en evaluación en otra revista.

Conflictos de interés: Los autores declaran no tener conflictos de interés.

Participación: Todos los autores contribuyeron al diseño, implementación, demostración funcional y redacción del trabajo.

Privacidad y protección de datos biométricos: El sistema procesa la inferencia ocular localmente en el dispositivo y no almacena ni transmite imágenes faciales identificables; solo se transmiten el valor numérico de PERCLOS y variables de estado. La demostración funcional se realizó con voluntarios del propio equipo, con su consentimiento, sin recopilación de datos demográficos identificables. Cualquier estudio futuro con participantes externos requerirá consentimiento informado y aprobación del Comité de Ética institucional.

Uso de Inteligencia Artificial (IA)

No se utilizaron herramientas de IA generativa en la redacción ni en el análisis o interpretación de los resultados. La responsabilidad intelectual del contenido es exclusiva de los autores.

REFERENCIAS

- Adi, P. D. P. (2022). Performance evaluation of ESP32 Camera face recognition for various projects. *Internet of Things and Artificial Intelligence Journal*, 2(1). <https://doi.org/10.31763/iota.v2i1.512>
- Anbesu, E. W., & Lema, A. K. (2023). Prevalence of computer vision syndrome: A systematic review and meta-analysis. *Scientific Reports*, 13, 1801. <https://doi.org/10.1038/s41598-023-28750-6>
- Beeson, D., Wolffsohn, J. S., Baigum, T., et al. (2024). Digital eye strain symptoms worsen during prolonged digital tasks. *Computers in Human Behavior Reports*, 16, 100489. <https://doi.org/10.1016/j.chbr.2024.100489>
- Dewi, C., Chen, R.-C., Chang, C.-W., Wu, S.-H., Jiang, X., & Yu, H. (2022). Eye aspect ratio for real-time drowsiness detection to improve driver safety. *Electronics*, 11(19), 3183. <https://doi.org/10.3390/electronics11193183>
- Estrada Araoz, E. G., Paricahua Peralta, J. N., Zuloaga Araoz, M. C., Gallegos Ramos, N. A., Paredes Valverde, Y., Quispe Herrera, R., & Velásquez Giersch, L. (2022). Prevalence of computer vision syndrome in Peruvian university students during the COVID-19 health emergency. *Archivos Venezolanos de Farmacología y Terapéutica*, 41(4), 264–270. <https://doi.org/10.5281/zenodo.6945062>
- Gupta, K. C., Kaur, G., & Chaudhary, A. (2023). Drowsiness detection using ESP32CAM. En 2023 3rd Asian Conference on Innovation in Technology (ASIANCON). IEEE. <https://doi.org/10.1109/ASIANCON58793.2023.10269977>
- Huyhua-Gutiérrez, S. C., Zeladita-Huaman, J. A., Díaz-Manchay, R. J., Dominguez-Palacios, A. B., Zegarra-Chapoñan, R., Rivas-Souza, M. A., & Tejada-Muñoz, S. (2023). Digital eye strain among Peruvian nursing students: Prevalence and associated factors. *International Journal of Environmental Research and Public Health*, 20(6), 5067. <https://doi.org/10.3390/ijerph20065067>
- Kaur, K., Gurnani, B., Nayak, S., et al. (2022). Digital eye strain: A comprehensive review. *Ophthalmology and Therapy*, 11(5), 1655–1680. <https://doi.org/10.1007/s40123-022-00540-9>
- Kraft, D., Hartmann, F., & Bieber, G. (2022). Camera-based blink detection using 3D-landmarks. En *Proceedings IWOAR* (pp. 1–7). <https://doi.org/10.1145/3558884.3558890>
- Lapa, I., Ferreira, S., Mateus, C., Rocha, N., & Rodrigues, M. A. (2023). Real-time blink detection as an indicator of computer vision syndrome in real-life settings: An exploratory study. *International Journal of Environmental Research and Public Health*, 20(5), 4569. <https://doi.org/10.3390/ijerph20054569>
- León-Figueroa, D. A., Barboza, J. J., et al. (2024). Prevalence of computer vision syndrome during the COVID-19 pandemic: A systematic review and meta-analysis. *BMC Public Health*, 24, 640. <https://doi.org/10.1186/s12889-024-17636-5>

- Lindo-Cano, E. F., García-Monge, V. A., Castillo-Cadillo, K. J., Sánchez-Tirado, E. A., Távara, I. M., & Morales, J. (2022). Computer-digital vision syndrome among university students of Lima city. *Open Public Health Journal*, 15. <https://doi.org/10.2174/18749445-v15-e2208104>
- Liu, M., Bian, S., Zhao, Z., Zhou, B., & Lukowicz, P. (2024). Energy-efficient, low-latency, and non-contact eye blink detection with capacitive sensing. *Frontiers in Computer Science*, 6. <https://doi.org/10.3389/fcomp.2024.1394397>
- McKee, C., & Hedge, A. (2022). Ergonomic lighting considerations for the home office workplace. *Work*, 71(1), 1–9. <https://doi.org/10.3233/WOR-210704>
- Nowroz, S. T., Saleh, N. M., Shakur, S., Banerjee, S., & Amsaad, F. (2025). A benchmark reference for ESP32-CAM module. *arXiv:2505.24081*. <https://doi.org/10.48550/arXiv.2505.24081>
- Safarov, F., Akhmedov, F., Abdusalomov, A. B., Nasimov, R., & Cho, Y. I. (2023). Real-time deep learning-based drowsiness detection. *Sensors*, 23(14), 6459. <https://doi.org/10.3390/s23146459>
- Sharmila, A. S., Jennifer, A., Vinodhini, R., & Sabarirajan, M. A. (2024). A review of eye blink detection methods using computer vision techniques for computer vision syndrome prevention. *Frontiers in Computer Science*, 6, Article 1394397. <https://doi.org/10.3389/fcomp.2024.1394397>
- Witte, A., & Lins, C. (2023). A real-time eye tracking system for the detection of eye blinks. *Studies in Health Technology and Informatics*, 302, 843–847. <https://doi.org/10.3233/SHTI230719>



Esta obra está bajo una Licencia Creative Commons
Atribución-NoComercial 4.0 Internacional