

Portada



**UNIVERSIDAD POLITÉCNICA
SALESIANA SEDE GUAYAQUIL**

**CARRERA DE
TELECOMUNICACIONES**

**Plataforma Integral Basada en IA para el monitoreo del estrés y la Optimización
Climática en Call Center.**

Trabajo de titulación previo a la obtención del
Título de ingeniero en telecomunicaciones.

AUTORES: JOSE DANIEL MARCILLO GONZALEZ.

RAUL JULIAN OJEDA ORDOÑEZ.

TUTOR: Ing. Klever Filiberto Carrion Gordillo.

Guayaquil-Ecuador

2025

II. Certificado de responsabilidad y autoría del trabajo de titulación.

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN.**

Nosotros, Raul Julián Ojeda Ordoñez con documento de identificación N° 0931876619, y Jose Daniel Marcillo Gonzalez con documento de identificación N° 0951931880 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, a los 12 días del mes de febrero del año 2026.

Atentamente,



Raul Julián Ojeda Ordoñez

0931876619



Jose Daniel Marcillo Gonzalez

0951931880



III. Certificado de cesión de derechos de autor del trabajo de titulación a la Universidad Politécnica Salesiana.

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA.**

Nosotros, Raul Julián Ojeda Ordoñez, con C.I. 0931876619 y Jose Daniel Marcillo Gonzalez, con C.I. 0951931880, expresamos nuestra voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana del Ecuador la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Artículo Científico: “Plataforma Integral Basada en IA para el monitoreo del estrés y la Optimización Climática en Call Center”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Institución facultada para ejercer plenamente los derechos concedidos.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, a los 12 días del mes de febrero del año 2026.

Atentamente,

Raul Julián Ojeda Ordoñez

0931876619

Jose Daniel Marcillo Gonzalez

0951931880

IV. Certificado de Dirección del Trabajo de Titulación.

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.

Yo, Klever Filiberto Carrion Gordillo, con documento de identificación N° 1102293402 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: Plataforma Integral Basada en IA para el monitoreo del estrés y la Optimización Climática en Call Center, realizado por Raul Julián Ojeda Ordoñez con documento de identificación N° 0931876619, y Jose Daniel Marcillo Gonzalez con documento de identificación N° 0951931880, obteniendo como resultado final el trabajo de titulación bajo la opción de Artículo académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, a los 12 días del mes de febrero del año 2026.

Atentamente,



Klever Filiberto Carrion Gordillo

1102293402

V. Dedicatoria

Dedicamos este trabajo a nuestros familiares quienes han sido un apoyo fundamental desde el inicio de nuestra formación universitaria. Su confianza, paciencia y constante acompañamiento nos permitieron avanzar y superar cada etapa de este proceso académico.

Este trabajo está dedicado a todas aquellas personas que creyeron en nosotros y nos motivaron a seguir adelante, especialmente a nuestras familias, quienes estuvieron presentes en los momentos más difíciles, brindándonos fortaleza y ánimo. La culminación de esta tesis es una muestra del esfuerzo compartido y del aprendizaje adquirido a lo largo de estos años, el cual contribuirá a nuestra formación profesional.

VI. Agradecimiento

Agradezco a mi familia, cuyo inquebrantable amor e incesante apoyo han sido la piedra angular de mi carrera académica y de todas las etapas de mi vida. A todas y cada una de las increíbles personas que han recorrido este camino conmigo y han contribuido de forma inestimable a mi desarrollo personal y profesional.

Agradezco a mi tutor, cuya orientación ha trascendido la realización de este trabajo de titulación, ha sido un guía valioso, ofreciéndome apoyo no solo en los aspectos académicos, sino también en el desarrollo personal y profesional.

Agradezco a Dios por permitirme culminar una etapa más en mi vida, por darme la sabiduría y fuerza necesaria para no desfallecer, por darme a la mejor madre y hermana que puedo tener, mi familia que estuvo al pendiente, cada de uno de sus consejos y palabras de aliento, me han ayudado a lograr mi meta.

VII. Resumen.

La investigación se enfocó en el monitoreo no invasivo del estrés y la fatiga laboral en operadores de call center un entorno caracterizado por exposición prolongada a pantallas y exigencias cognitivas continuas. El objetivo fue desarrollar un sistema automatizado capaz de estimar el estado de fatiga del operador y ejecutar acciones de control ambiental en tiempo real.

Para ello, se implementó una arquitectura cliente-servidor basada en visión por computadora, utilizando Media Pipe y OpenCV para la extracción de landmarks faciales y el cálculo de métricas fisiológicas como Eye Aspect Ratio (EAR), PERCLOS y Mouth Ratio. Estas métricas fueron integradas en un indicador compuesto (StressScore), evaluado mediante ventanas temporales, y gestionadas por un backend con servicios REST e integración IoT a través de Home Assistant y Broadlink RM4C Mini.

Los resultados obtenidos evidenciaron una correcta clasificación de los estados oculares y bucales, alcanzando una precisión del 100 % en la detección de ojos abiertos, cerrados y eventos de bostezo en las pruebas controladas. El indicador PERCLOS permitió diferenciar consistentemente los estados normal, alerta y fatiga severa, mientras que el StressScore reflejó una progresión coherente de los niveles de fatiga según las condiciones evaluadas.

Finalmente, el módulo de control ambiental ejecutó ajustes automáticos de temperatura de 16 °C y 18 °C en función del aforo detectado, validando la integración entre el monitoreo fisiológico y la automatización del entorno. En conjunto, los resultados confirman la viabilidad del sistema como una herramienta de monitoreo en tiempo real orientada a mejorar las condiciones laborales sin interferir de forma intrusiva en las actividades del operador.

i. Palabras Claves

1; Inteligencia Artificial 2; Internet de las cosas IoT 3; Monitoreo del Estrés 4; Control Ambiental 5; Call Center 6; Análisis de Datos

VIII. Abstract.

This research focused on the non-invasive monitoring of work-related stress and fatigue in call center operators, an environment characterized by prolonged screen exposure and continuous cognitive demands. The objective was to develop an automated system capable of estimating the operator's fatigue state and triggering environmental control actions in real time.

To achieve this, a client-server architecture based on computer vision was implemented, using MediaPipe and OpenCV to extract facial landmarks and compute physiological metrics such as Eye Aspect Ratio (EAR), PERCLOS, and Mouth Ratio. These metrics were integrated into a composite indicator (StressScore) evaluated through temporal windows and managed by a backend exposing REST services and IoT integration via Home Assistant and the Broadlink RM4C Mini.

The results demonstrated accurate classification of ocular and oral states, achieving 100% accuracy in detecting open and closed eyes as well as yawning events under controlled tests. The PERCLOS metric consistently differentiated normal, alert, and severe fatigue states, while the StressScore showed a coherent progression across the evaluated conditions.

Finally, the environmental control module executed automatic temperature adjustments of 16°C and 18°C based on detected occupancy, validating the integration between physiological monitoring and environmental automation. Overall, the findings confirm the feasibility of the proposed system as a real-time monitoring tool aimed at improving workplace conditions without intrusive interference in the operator's activities.

i. **Keywords.**

1; Artificial Intelligence 2; Internet of Things (IoT) 3; Stress Monitoring 4; Environmental Control 5; Call Center 6; Data Analysis.

IX. Índice De Contenido.

| | |
|--|----|
| Portada..... | 1 |
| II. Certificado de responsabilidad y autoría del trabajo de titulación..... | 2 |
| III. Certificado de cesión de derechos de autor del trabajo de titulación a la Universidad Politécnica Salesiana. | 3 |
| IV. Certificado de Dirección del Trabajo de Titulación..... | 4 |
| V. Dedicatoria..... | 5 |
| VI. Agradecimiento | 5 |
| VII. Resumen. | 6 |
| i. Palabras Claves | 6 |
| VIII. Abstract..... | 7 |
| i. Keywords. | 7 |
| IX. Índice De Contenido. | 8 |
| X. Introducción. | 11 |
| a. Python. | 11 |
| b. Rango estrés del ser humano en el trabajo. | 12 |
| c. Broadlink RM4..... | 12 |
| d. Flask..... | 13 |
| e. Requests. | 13 |
| f. GitHub Gist. | 14 |
| g. API REST..... | 14 |
| h. Websockets..... | 14 |
| i. OpenCV-Python. | 15 |
| j. Tensor Flow y TensorFlow.js | 15 |
| k. COCO-SSD..... | 16 |
| l. Yolo 8..... | 16 |
| m. Media Pipe..... | 16 |
| n. Docker..... | 16 |

| | |
|--|----|
| o. Home Assistant..... | 17 |
| p. Ecuaciones empleadas..... | 17 |
| i. Eye Aspect Ratio (EAR) | 18 |
| ii. PERCLOS (Percentage of Eye Closure) | 20 |
| iii. Detección de bostezos mediante relación geométrica de la boca. | 23 |
| iv. Cálculo del índice de fatiga. (Fatigue Score) | 26 |
| v. Cálculo del índice ocular (Ocular Score) | 28 |
| vi. Detección y penalización por fraude..... | 30 |
| vii. Formula final de StressScore. | 32 |
| XI. Artículos relacionados..... | 35 |
| XII. Metodología..... | 37 |
| Desarrollo..... | 42 |
| Preparación del Backend..... | 48 |
| Datos Obtenidos..... | 50 |
| XIII. Discusion..... | 78 |
| XIV. Conclusión..... | 79 |
| XV. Recomendaciones..... | 80 |
| XVI. Bibliografía..... | 81 |
| XVII. Anexos..... | 84 |
| Anexo 1..... | 84 |
| 1.1 Archivo de dependencias..... | 84 |
| 1.2 Entorno Virtual y ejecución (Iniciar Cliente)..... | 85 |
| 1.3 Calculo del EAR..... | 86 |
| 1.4 Buffer Ocular y PERCLOS..... | 86 |
| 1.5 Mouth y Eventos de bostezos..... | 87 |
| 1.6 Escalamiento Lineal..... | 88 |
| 1.7 Estructura del Payload..... | 89 |
| Anexo 2..... | 91 |
| Anexo 2.1..... | 91 |

| | |
|----------------|----|
| Anexo 2.2..... | 91 |
| Anexo 2.3..... | 92 |
| Anexo 2.4..... | 93 |
| Anexo 2.5..... | 93 |
| Anexo 2.6..... | 94 |
| Anexo 2.7..... | 94 |
| Anexo 2.8..... | 95 |

X. Introducción.

En los últimos años, el incremento de actividades laborales asociadas al uso intensivo de tecnologías digitales ha generado una creciente preocupación por los efectos de la fatiga visual y el estrés cognitivo en el desempeño humano. Diversos estudios internacionales señalan que la exposición prolongada a pantallas, combinada con tareas repetitivas y altos niveles de concentración, puede afectar tanto la salud del trabajador como la calidad de los procesos productivos, incrementando la probabilidad de errores, disminuyendo la atención sostenida y reduciendo la eficiencia operativa.

Hoy es posible hacerlo con apoyo de inteligencia artificial e IoT, no como “teoría”, sino como algo práctico, medir señales en tiempo real y, cuando haga falta, activar ajustes en el propio entorno de trabajo. Bajo esa idea, este proyecto desarrolla una plataforma que combina visión por computadora con componentes IoT para estimar el nivel de estrés del operador y automatizar acciones ambientales, como la climatización, sin interferir con sus tareas diarias ni exigir cambios en su rutina.

La base del sistema es la detección de landmarks faciales para obtener métricas asociadas a la dinámica ocular (EAR y PERCLOS) y a patrones bucales relacionados con bostezos. Esas señales se procesan en ventanas temporales, se resumen en un indicador compuesto y se envían a un backend mediante servicios REST. A partir de esa información, el backend coordina respuestas del entorno mediante control infrarrojo (por ejemplo, usando Broadlink), lo que permite ajustar la climatización sin reemplazar equipos ya instalados.

En conjunto, esta propuesta plantea una solución técnicamente viable para el monitoreo continuo del estrés y la fatiga visual en centros de contacto, contribuyendo a la mejora del bienestar del operador y al fortalecimiento del desempeño operativo mediante el uso de tecnologías no invasivas y de bajo impacto en la infraestructura existente.

a. Python.

Debido a su amplia biblioteca, en la figura 1 Python es un lenguaje de programación de alto nivel que se emplea con frecuencia en la investigación y en la ingeniería, gracias a su sintaxis clara y a su extensa comunidad. Se eligió Python como lenguaje para esta tesis porque es compatible con las herramientas de visión computacional, posibilita la creación sencilla del cliente y el servidor para capturar y analizar datos, y destaca por su velocidad en la generación de prototipos, está centrado en la orientación a objetos con semántica dinámica, lo que lo hace, ser una excelente opción para poder desarrollar aplicaciones por su dinamismo y sus estructuras de datos.



Figura 1 Python. Fuentes: Autores.

b. Rango estrés del ser humano en el trabajo.

El estrés laboral es una respuesta psicofisiológica del trabajador ante demandas laborales que superan o son percibidas como superiores a sus recursos de afrontamiento, el estrés de baja o moderada intensidad puede llegar a cumplir una función orientadora y potenciadora de la activación facilitando en términos generales la atención y el rendimiento sin embargo cuando se presenta en niveles elevados o de forma prolongada en el tiempo, se asocia con el deterioro del rendimiento, la fatiga mental, la desensibilización emocional y diversas amenazas para la salud física y psicológica.

c. Broadlink RM4.

En la Figura 2 se presenta Broadlink RM4 se ha identificado como una solución comercial eficaz para la transmisión de señales infrarrojas dentro de sistemas de automatización, destacándose por su capacidad de emular controles remotos de equipos de climatización. Su funcionamiento se basa en la recepción de comandos a través de la red local (LAN), los cuales son transformados en señales infrarrojas compatibles con múltiples marcas de aires acondicionados.[1]

En el sistema propuesto, la gestión del clima se adapta de forma dinámica, para lo cual se implementó una API escrita en Python, que se puede acceder por el endpoint `/api/people/count`, el cual devuelve en tiempo real el número de personas del entorno, el cual es utilizado como variable para modificar la temperatura del ambiente ajustado al nivel de ocupación.

La inclusión del RM4 para las redes de área local LAN, elimina la necesidad de exponer el sistema a Internet, lo que simplifica la infraestructura general del sistema, así como también reduce también el riesgo del fallo del sistema, manteniendo éste en un entorno encerrado y controlado.



Figura 2 Broadlink RM4 Fuentes: Autores.

d. Flask.

En la figura 3 Flask, un microframework muy popular que se encuadra dentro del ecosistema Python, sirve para facilitar la creación de aplicaciones web y APIs. En este trabajo, Flask tiene una importante función, dado que estructura el backend en el servidor expondrá unos servicios web mediante una arquitectura RESTful. Esta característica permite establecer una comunicación eficiente entre los clientes y la lógica de control del sistema. Gracias a su diseño minimalista y altamente flexible, es posible definir endpoints específicos que gestionen métricas, configuraciones y notificaciones. Esta capacidad convierte a Flask en una herramienta ideal para desarrollar soluciones escalables y fácilmente integrables.[2]



Figura 3 Flask Fuentes: Autores.

e. Requests.

Requests es una de las bibliotecas de Python más usadas, utilizada en el momento de gestionar comunicaciones HTTP de forma clara, accesible y confiable. Su funcionalidad en este proyecto permite establecer interacciones entre el backend y los servicios externos, ya sea para el consumo de APIs de terceros, o para realizar la comunicación de distintos componentes internos mediante solicitudes HTTP, de manera que contribuye a incrementar las capacidades del sistema, por medio de conexiones que garanticen la conexión con recursos distribuidos.

f. GitHub Gist.

GitHub Gist es un servicio que ofrece GitHub para facilitar el intercambio de fragmentos de texto o de código fuente de una forma muy simple y cómoda a través de la web. En el contexto del presente trabajo, se utiliza como canal el intercambio de información entre el backend y los clientes (es decir, un almacén dinámico de información), por ejemplo, la IP local del servidor central. Esta estrategia permite evitar soluciones complejas de descubrimiento de servicios o de configuraciones de red para que los clientes siempre tengan acceso con la IP actualizada. Al tratarse de una herramienta gratuita, accesible y basada en la nube, se facilita la sincronización de información clave entre módulos distribuidos, mejorando tanto la escalabilidad como la adaptabilidad del sistema.

g. API REST.

En la figura 4 el backend implementa una API REST para exponer endpoints HTTP que permiten a los clientes interactuar con el sistema, ya sea enviando o recibiendo datos clave como métricas, configuraciones o notificaciones. Esta arquitectura basada en principios REST promueve una estructura modular, en la que cada componente puede desarrollarse de manera independiente. Además, al seguir estándares ampliamente aceptados, se garantiza la integración eficiente entre módulos y la interoperabilidad del sistema a largo plazo.[3]

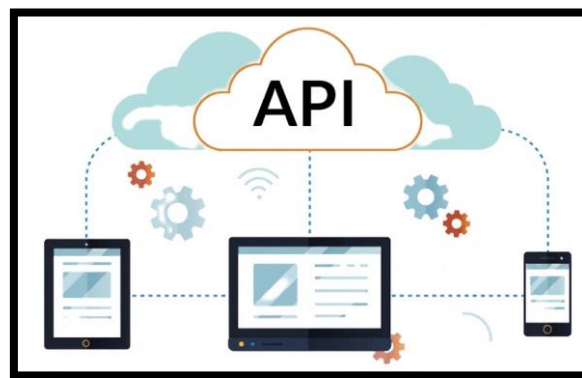


Figura 4 API REST Fuentes: Autores.

h. Websockets

En situaciones donde se requiere comunicación en tiempo real, como el envío de notificaciones instantáneas o la actualización de estados críticos, se pueden emplear websockets. Esta tecnología permite mantener una conexión bidireccional y persistente entre el backend y el cliente, asegurando que la información fluya de manera inmediata y eficiente, lo que es especialmente útil para aplicaciones interactivas o de monitoreo continuo.

i. OpenCV-Python.

En la Figura 5 el OpenCV-Python, una biblioteca de código abierto, es muy conocida en el área de visión por computadora y la manipulación digital de imágenes. En el proyecto actual, esta herramienta se emplea para registrar y analizar secuencias visuales provenientes de cámaras conectadas, lo que permite llevar a cabo operaciones en tiempo real, como la detección de rostros, la segmentación y diversas transformaciones de preprocesamiento.

Su integración resulta esencial para proporcionar datos visuales de alta calidad a los modelos de inteligencia artificial, facilitando tanto la extracción de características significativas como la interpretación visual de los resultados. Gracias a su rendimiento y adaptabilidad, OpenCV-Python permite desarrollar soluciones sólidas de análisis visual, incluyendo funciones avanzadas como el monitoreo del estrés y la fatiga en los usuarios.[4]

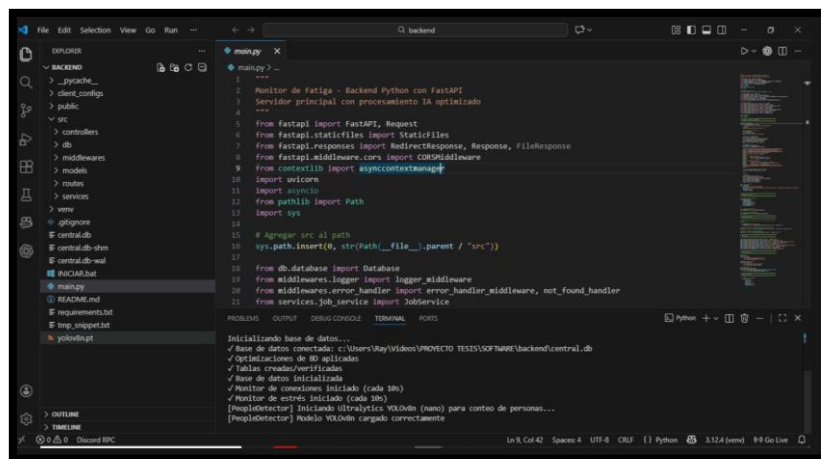


Figura 5 Open CV PYTHON Fuentes: Autores.

j. Tensor Flow y TensorFlow.js

Tensor Flow es una herramienta que Google abrió para que todos la utilicen. Ayuda a las personas a hacer y utilizar modelos de aprendizaje profundo, que son formas de hacer que los ordenadores aprendan de los datos. Los modelos de aprendizaje profundo se pueden usar en el reverso de aplicaciones para hacer cosas como encontrar objetos y reconocer caras. TensorFlow.js te permite usar modelos de IA en navegadores web con JavaScript. No necesitas enviarlos al servidor todo el tiempo, podemos procesar los datos inmediatamente cuando llegue. La sinergia entre TensorFlow (Python) y TensorFlow.js (JavaScript) favorece una integración fluida de inteligencia artificial en aplicaciones completas, potenciando la capacidad del sistema para analizar eventos visuales desde imágenes o videos con mayor eficiencia y respuesta inmediata.[5]

k. COCO-SSD

COCO-SSD es una forma de encontrar y etiquetar diferentes cosas en imágenes, como personas, coches y cosas que usamos todos los días. Utiliza un método llamado SSD, que es bueno para encontrar cosas en una imagen. También utiliza muchos ejemplos de una gran colección de imágenes llamadas COCO, que tiene muchas cosas en él. Este proyecto utiliza COCO-SSD, una herramienta que puede encontrar personas en videos, para ayudar al componente client_python del componente client_python. El componente client_python puede mirar los marcos de la cámara y ver si hay personas en ellos. Puede utilizar esta función para activar el monitoreo sólo cuando alguien está usando el ordenador, por lo que no recibe falsas alarmas, y se puede ver cuánto tiempo están expuestos y cuando se detienen. El modelo funciona con rapidez y precisión, lo que lo hace bueno para situaciones en las que el tiempo importa, y ayuda a encontrar la información más importante de los datos sobre cómo el usuario piensa o siente.

l. Yolo 8.

El sistema YOLO 8, que significa "You Only Look Once", constituye un método de reconocimiento de elementos fundamentado en redes neuronales de convolución. Este permite señalar y ubicar individuos al instante a partir de grabaciones de video, al evaluar toda la imagen de golpe y, por ende, conseguir una gran eficiencia al calcular. El sistema YOLO, que significa "Solo Miras Una Vez", constituye un método de reconocimiento de elementos fundamentado en redes neuronales de convolución. Este permite señalar y ubicar individuos al instante a partir de grabaciones de video, al evaluar toda la imagen de golpe y, por ende, conseguir una gran eficiencia al calcular.

m. Media Pipe.

MediaPipe es un framework de visión por computadora creado por Google, orientado a la detección y al seguimiento eficiente e en tiempo real de las partes del humano como el rostro, las manos y la postura corporal. Es de arquitectura optimizada para funcionar en dispositivos con pocos recursos, lo que lleva a un bajo retardo y alta estabilidad en el procesamiento del vídeo en continuo. En este sentido MediaPipe es especialmente adecuado para entornos de los Call Centers, en el que se tiene trabajadores sentado al frente de una pantalla durante largos periodos y las condiciones de claridad suelen ser relativamente constantes.

n. Docker.

Docker es una plataforma que permite ejecutar aplicaciones en entornos aislados mediante la encapsulación de dependencias, configuraciones y del servicio del sistema. Su utilización permite la portabilidad, la consistencia de la operación y un despliegue reproducible, además de un bajo consumo de recursos.

En esta Tesis, Docker se usa para ejecutar Home Assistant en un entorno aislado, solucionando los problemas de persistencia y de almacenamiento que tenía la ejecución directa del dispositivo RM4C Mini. De este modo, se puede hacer funcionar en paralelo el servicio de control infrarrojo sin que dependa del backend desarrollado en Python y se permite que procesos de reinicio y mantenimiento se realicen sin incorporar descartes de la configuración de los elementos de control.

o. Home Assistant

Home Assistant es una plataforma de automatización de código abierto que centraliza el control de dispositivos inteligentes mediante servicios y API locales. Ofrece soporte nativo para el aprendizaje y ejecución de comandos infrarrojos, junto con almacenamiento persistente de estados en el sistema propuesto el Home Assistant actúa como capa intermedia entre el backend y el RM4C Mini, gestionando el aprendizaje y envío de comandos IR del aire acondicionado. El backend en Python se comunica mediante su API, permitiendo ajustar la temperatura de forma automática en función de la cantidad de personas detectadas, sin interactuar directamente con el hardware infrarrojo.[6]

p. Ecuaciones empleadas.

El sistema propuesto estima el estado funcional del operador mediante señales visuales no invasivas capturadas en tiempo real para ellos se emplean métricas geométricas derivadas de landmarks faciales, las cuales permiten describir cuantitativamente comportamientos asociados a fatiga y disminución de alerta en términos operativos estas métricas llegan a calcular en cada muestra y posteriormente se agregan dentro de una ventana temporal fija de análisis con el fin de reducir variaciones instantáneas y llegar obtener indicadores estables para la toma de decisiones el modelo se llega a estructurar de 3 niveles:

- Extracción de descriptores base de apertura ocular y apertura bucal.
- Construcción de indicadores temporales por ejemplo proporción de cierre ocular y tasa de bostezos.
- Fusión ponderada de dichos indicadores para generar un puntaje único (StressScore) en el rango de 0-100

Llegamos a incorporar un término de penalización cuando se detectan patrones atípicos que afectan la confiabilidad de la medición con ello el apartado de fórmulas asegura una trazabilidad de cada ecuación porque cumplen su función específica dentro de la cadena de procesamiento y su significado se interpreta en relación con el comportamiento fisiológico que representa.

i. Eye Aspect Ratio (EAR)

En el artículo[7] se muestra que es un indicador geométrico que permite estimar el grado de la apertura del ojo a partir de puntos de referencia (landmarks) detectados automáticamente sobre el contorno ocular en una imagen, en la fórmula cada punto P_i corresponde a un landmark específico del ojo o un punto del párpado y su valor es simplemente su posición en la imagen expresada mediante dos coordenadas $P_i = (X_i, Y_i)$, donde x_i indica la ubicación horizontal (izquierda-derecha) y Y_i la ubicación vertical (arriba-abajo) a partir de estos puntos el sistema calcula distancias entre pares de landmarks por ello aparece el término $\|p_i - p_j\|$ que significa la distancia entre los puntos $\|p_i - p_j\|$ dentro de la imagen, esta distancia se denomina euclidiana porque corresponde a la distancia habitual en un plano y siempre es positiva, en consecuencia los signos muestran la comparación de posiciones necesarias para calcular la separación entre dos puntos.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2 * \|p_1 - p_4\|} \quad (1)$$

Datos:

EAR: Eye Aspect Ratio – Relación de Aspecto del ojo.

- $\|p_1 - p_4\|$: Representa el ancho horizontal del ojo
- $\|p_2 - p_6\|$: Distancia vertical interna entre el párpado superior e inferior.
- $\|p_3 - p_5\|$: Distancia vertical externa entre el párpado superior e inferior.
- **2**: Promediar las dos distancias verticales del ojo.

En esta expresión, $\|p_2 - p_6\|$ y $\|p_3 - p_5\|$ corresponden a dos mediciones de apertura vertical del ojo (distancias entre el párpado superior e inferior en dos zonas), mientras que $\|p_1 - p_4\|$ representa el ancho horizontal del ojo (distancia entre las esquinas). El numerador suma dos distancias verticales para obtener una medida más estable de la apertura ocular y reducir el efecto de pequeñas variaciones o ruido en la detección de landmarks. El denominador incorpora el ancho del ojo como factor de normalización de este modo, el índice resulta menos sensible al tamaño del rostro o a la distancia respecto de la cámara. El factor 2 se incluye porque se emplean dos distancias verticales multiplicar por 2 equivale a comparar el promedio de la apertura vertical con el ancho del ojo, manteniendo el indicador balanceado y consistente.[8]

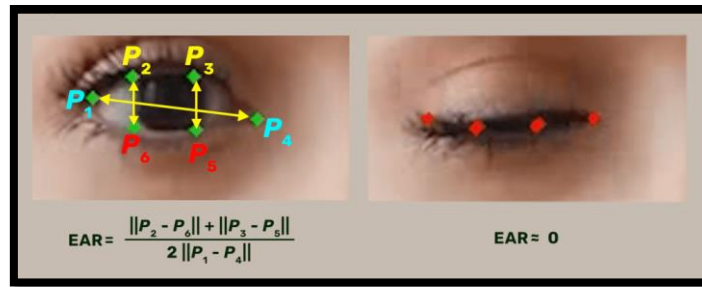


Figura 6 Interpretación de la fórmula EAR Fuente: Autores.

En la Figura 6, el Eye Aspect Ratio (EAR) tiende a mantenerse relativamente constante cuando el ojo está abierto y disminuye de forma notable durante un parpadeo o cuando el ojo permanece cerrado. Por ello, el EAR se utiliza como señal base para identificar cierres oculares y, posteriormente, para alimentar métricas temporales como el PERCLOS dentro de una ventana de análisis.

El artículo “*Automatic Detection of the Driver Distractions Based on the Analysis of Face Videos*” [9] propone el uso del EAR como un indicador geométrico robusto para la detección del estado ocular a partir de videos faciales, permitiendo diferenciar entre parpadeos normales y cierres prolongados asociados a fatiga o somnolencia.

En el sistema se adopta un umbral de referencia cercano a $EAR = 0.20$ para clasificar una muestra como “ojo cerrado” cuando el valor cae por debajo de dicho umbral. Valores superiores a este límite corresponden generalmente a un estado de ojo abierto o parpadeos breves, mientras que valores inferiores mantenidos durante varios fotogramas consecutivos se interpretan como cierres oculares prolongados. Esta diferenciación permite identificar de manera efectiva comportamientos asociados a fatiga visual.

Como interpretamos los valores:

| Estado ocular | EAR (referencial) | Criterio temporal | Interpretación |
|---------------|---------------------------------|--|--|
| Ojo abierto | EAR típicamente > 0.25 | Se mantiene por encima del umbral durante la ventana (sin caídas sostenidas) | Condición normal: apertura ocular estable. |
| Parpadeo | EAR cae hacia el umbral (~0.20) | Caída breve y recuperación inmediata (evento transitorio) | Evento fisiológico normal; no implica somnolencia. |
| Ojo cerrado | EAR < 0.20 | Se mantiene por debajo del umbral durante varios fotogramas consecutivos | Cierre sostenido; relevante para estimar fatiga ocular y calcular PERCLOS. |

Tabla 1 Interpretación de Valores EAR Fuente: Autores.

En la tabla 1 resume cómo se interpreta el EAR para clasificar el estado del ojo durante el monitoreo. En términos simples, valores altos de EAR indican que el ojo está abierto, mientras que valores cercanos o por debajo del umbral (= 0.20) indican cierre ocular. Sin embargo, para evitar confundir un parpadeo normal con un cierre prolongado, la tabla incorpora un criterio temporal el parpadeo se identifica cuando el EAR cae de forma transitoria y se recupera rápidamente, mientras que el estado de ojo cerrado se determina cuando el EAR permanece por debajo del umbral durante varios fotogramas consecutivos. Esta distinción permite una interpretación consistente del comportamiento ocular y sirve como base para el cálculo de métricas temporales como el PERCLOS dentro de una ventana de análisis.[10]

ii. PERCLOS (Percentage of Eye Closure)

En el artículo “*PERCLOS-based technologies for detecting drowsiness: current evidence and future directions*” de Abe[11] PERCLOS (Percentage of Eye Closure) es un indicador ampliamente reconocido en investigaciones sobre somnolencia y prevención de riesgos laborales, cuya función principal es cuantificar el porcentaje de tiempo en que los ojos permanecen cerrados dentro de una ventana temporal específica.

El indicador de PERCLOS se mide el porcentaje de tiempo, en que los ojos permanecen cerrados dentro de una ventana temporal definida matemáticamente se expresa como:

$$Perclos(\%) = \frac{N \text{ ojos Cerrados}}{N \text{ total de muestras}} \times 100 \quad (2)$$

Datos:

- “N ojos Cerrados” es el número de muestras (Frame) en las que el ojo se clasifica como cerrado.
- “N total de Muestras” es el total de muestras (Frame) analizadas dentro de la ventana temporal.
- PERCLOS (Percentage of Eye Closure - Porcentaje de cierre de ojos): Porcentaje de tiempo que el operador tiene los ojos cerrados.

El valor de PERCLOS se expresa como porcentaje del tiempo con ojos cerrados dentro de una ventana temporal en referencia de la figura 15 en consecuencia los valores más altos indican mayor duración acumulada de cierre ocular, lo que se relaciona con mayor somnolencia o fatiga visual.

Se precisa que el valor N no contabiliza los parpadeos como eventos discretos, sino como muestras de tiempo. El estado del ojo se evalúa en cada fotograma, contabilizando el número de frames que cumplen la condición de ojo cerrado. Este enfoque permite medir el tiempo efectivo de cierre ocular, lo cual resulta más representativo para la detección de somnolencia que el simple conteo del número de parpadeos.

En la condición de ojo cerrado lo determinamos a partir de los valores del EAR calculado en cada muestra si específicamente si el valor del EAR cae por debajo del umbral operativo como se muestra en la tabla 1 que el $EAR < 0.20$ nos muestra que se contabilizo como ojo cerrado.

El PERCLOS se llega a interpretar directamente como el porcentaje del tiempo dentro de la ventana en la que el ojo estuvo cerrado porque desde el punto operativo si la ventana de análisis tiene una duración T (Tiempo) y el video se procesa en una tasa de muestreo entonces el N total de muestras llega a corresponder al número total de fotogramas evaluados en ese intervalo indica que un valor de PERCLOS bajo indica que ojo permaneció mayormente abierto mientras tanto que un valor alto indica que una fracción completamente significativa del intervalo se pasó con el ojo cerrado en esta última condición es relevante porque lo asociamos a estados de somnolencia o fatiga ocular sostenida.[12]

Ejemplo:

En el siguiente ejemplo se considera una ventana de análisis compuesta por 100 fotogramas. Este valor no corresponde a un número fijo de muestras en el sistema, sino que se utiliza únicamente para facilitar la comprensión del cálculo del indicador PERCLOS. En la implementación real, el número total de muestras depende de la duración de la ventana temporal y de la tasa de captura de fotogramas del sistema.

Una Ventana con 100 Muestras.

Caso A: ojo casi abierto siempre

- Solo con 5 “Muestras de ojos Cerrado” tuvieron $EAR < 0.20$

$$Perclos(\%) = \frac{5}{100} = 0.05 = 5\%$$

Interpretación: Ojo abierto la mayor parte del tiempo.

Caso B: Ojo cerrado bastante tiempo

- 40 muestras tuvieron $EAR < 0.20$

$$Perclos(\%) = \frac{40}{100} = 0.40 = 40\%$$

Interpretación: mucho cierre ocular, posible somnolencia o estrés.

Dado que PERCLOS se define como la razón entre el número de muestras clasificadas como ojo cerrado ($EAR < 0.20$) y el total de muestras en la ventana, un valor bajo de PERCLOS indica que el ojo permaneció abierto durante la mayor parte del intervalo, mientras que un valor alto indica un mayor tiempo acumulado de cierre ocular.

Interpretación de PERCLOS

| Nivel / estado | PERCLOS (referencial) | Interpretación fisiológica | Implicación operativa |
|-----------------------------|-----------------------------|--|--|
| Normal | $< 0.15 (< 15\%)$ | El ojo permaneció abierto la mayor parte del tiempo; cierres breves compatibles con parpadeos normales. | Condición estable; no sugiere somnolencia. |
| Fatiga moderada | $0.15 - 0.30 (15\% - 30\%)$ | Aumenta el tiempo acumulado de cierre ocular dentro de la ventana; puede reflejar inicio de fatiga visual o disminución de alerta. | Requiere seguimiento; posible alerta temprana. |
| Somnolencia / fatiga severa | $\geq 0.40 (\geq 40\%)$ | Alto porcentaje de cierre ocular sostenido o repetido; compatible con somnolencia y reducción significativa de alerta. | Condición crítica; recomendar pausa/descanso según reglas del sistema. |

Tabla 2 Interpretación Referencial de PERCLOS en una Ventana de Análisis. Fuente: Autores

En el artículo “Research Article Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN” de Zhao[13] la tabla 2 que definimos como la proporción de muestras en las que el ojo fue clasificado como cerrado como un valor bajo significa que el ojo permaneció abierto la mayor parte del tiempo mientras que el valor alto indico que estuvo cerrado durante una fracción importante del intervalo, aplicando el criterio del EAR < 0.20 caía debajo del umbral contabilizaba como ojo cerrado y suma “ N ojos cerrados luego al final de la ventana se divide entre el total de fotogramas “N total de muestras” siguiendo esta lógica el nivel normal es (<15%) corresponde a cierres breves compatibles con parpadeos si el nivel aumenta seria fatiga o estrés moderado (15-30%) que refleja un aumento del tiempo de cierre ocular pero si el nivel es mayor (≥40%) indica cierres oculares sostenidos o repetidos que habilitan la condición de activar alertas o recomendaciones de descanso.

iii. Detección de bostezos mediante relación geométrica de la boca.

En el artículo “Research on a Lightweight Fatigue Detection Method for Drivers Based on Multimodal Feature Fusion”[14] el Mouth Ratio se calcula a partir de landmarks faciales midiendo dos distancias la altura (apertura) de la boca y su ancho, luego se compara ambas medidas para obtener un valor que describe cuán abierta está la boca. Este enfoque reduce el efecto del tamaño del rostro y de la distancia a la cámara, lo que lo hace adecuado para detectar bostezos.

$$\text{Mouth Ratio} = \frac{\| p13 - p14 \|}{\| p61 - p291 \|} \quad (3)$$

Datos:

- Mouth Ratio: Relación de aspecto de la boca, se calcula a partir de landmarks faciales.
- $p13 - p14$: Miden la apertura vertical de la boca.
- $p13$: *Punto superior del labio superior*
- $p14$: Punto inferior del labio inferior
- $p61$ y $p291$: Miden el ancho horizontal de la boca.
- $p61$: extremo horizontal izquierdo de la boca
- $p291$: extremo horizontal derecho de la boca

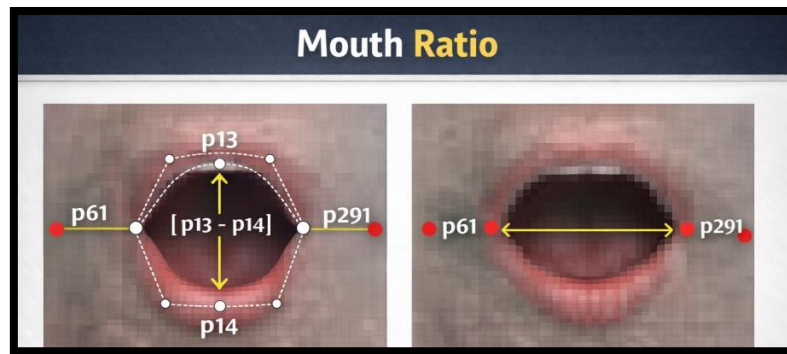


Figura 7 Landmarks del Mouth Ratio. Fuente: Autores.

En la figura 7 se muestra el proceso de detección de bostezos el sistema utilizara landmarks faciales alrededor de la boca para poder estimar la apertura en cada fotograma, en la ecuación cada p_i representa la posición de un punto detectado en la imagen es decir un par de coordenadas dentro del plano de la cámara.

- $p_{13} - p_{14}$ se ubica en el eje vertical de la boca (aproximadamente en el labio superior e inferior o zona equivalente del modelo de landmarks) por lo que $p_{13} - p_{14}$ mide la apertura vertical cuanto se separan los labios en sentido vertical.
- p_{61} y p_{291} se encuentra en los extremos laterales de la boca por lo que mide el ancho horizontal de la boca.

Las distancias empleadas en este cálculo de la métrica del Mouth Ratio no se expresan en unidades físicas como centímetros o milímetros, sino que se obtienen a partir de las coordenadas relativas de los puntos faciales detectados en la imagen. Dado que ambas distancias se calculan dentro del mismo sistema de coordenadas, la relación entre ellas da lugar a una magnitud adimensional, lo que permite que el Mouth Ratio sea independiente del tamaño del rostro, de la resolución de la imagen y de la distancia a la cámara.

En la ecuación 3 los símbolos $\| p_i - p_j \|$ se interpretan como la distancia entre dos puntos en la imagen el signo “-” aparece para obtener la distancia primero calculamos la diferencia entre las posiciones de los puntos esto no corresponde a una resta sino a un paso geométrico necesario para medir la separación.

¿Porque se divide vertical entre horizontal?

Si el rostro se acerca a la cámara, todas las distancias medidas en pixeles tienden a aumentar pero cuando el rostro se aleja dicha distancia disminuye si únicamente se utilizara la apertura vertical de la boca, valor obtenido solo dependería en gran medida del tamaño del rostro proyectado en la imagen por esta razón al dividir la apertura vertical y el ancho horizontal de la boca se obtiene una medida relativa más estable.

- cambios de distancia a la cámara,
- diferencias anatómicas entre personas,
- ligeras variaciones de encuadre.

Detección de Bostezo: umbral + duración mínima:

El sistema no considera bostezo como un valor instantáneo y que tiene aperturas breves que pueden producirse al hablar o gesticular, en consecuencia, se aplica un criterio doble:

- **Umbral:** Que lo definimos un valor operativo del Mouth Ratio (por ejemplo, alrededor de 0.6) a partir del cual la apertura se considera inusualmente grande.
- **Duración Mínima:** El mouth ratio debe permanecer por encima del umbral durante un intervalo mínimo solo hay se registra como un evento de bostezo.

El umbral se adopta como referencia operativa y puede ajustarse mediante calibración según las condiciones de captura y variabilidad individual. [15]

Ejemplo conceptual:

Considere un fotograma en el que a partir de las coordenadas de los puntos faciales detectados se llega a obtener dos distancias relativas dentro del plano de la imagen, la distancia correspondiente a la apertura de la boca $\|p_{13}-p_{14}\| = 24$ pixeles mientras que la distancia asociada al ancho horizontal de la boca es $\|p_{61} - p_{291}\| = 40$ pixeles.

Se calcula la relación entre ambas distancias:

$$\text{Mouth Ratio} = \frac{24}{40} = 0.60$$

“El umbral 0.6 se usa como referencia operativa del prototipo; no es universal y puede ajustarse por calibración.”

Un valor cercano o superior a 0.6 indica una apertura elevada ya que si este valor se mantiene durante el tiempo mínimo establecido el sistema lo contabiliza como bostezo si solo ocurre durante uno o pocos fotogramas y retorna inmediatamente a valores menores se interpreta como una apertura transitoria no se registra como bostezo.

En este prototipo se adoptó un umbral operativo referencial $Mr \approx 0.6$ para identificar aperturas bucales compatibles con bostezo a partir del MouthRatio. Este valor no se considera universal, ya que puede variar según el individuo y las condiciones de captura (cámara, iluminación y precisión de landmarks). Por ello, el umbral se plantea como parámetro calibrable y se complementa con un criterio temporal (duración mínima) para reducir falsos positivos por habla o gesticulación.

Un bostezo suele tener un patrón:

- Sube (apertura aumenta)
- Se mantiene (meseta alta)
- Baja (cierre gradual)

En el Mouth Ratio solo mide que tan abierta esta la boca en la imagen por eso el sistema no puede saber la intención si alguien finge o no pero podemos diferenciar en habla y gestos de un bostezos probable usando el tiempo, pero hablar produce aperturas cortas y cambiantes mientras que un bostezo suele ser una apertura alta y sostenida además si en una ventana aparece demasiado bostezos en poco tiempo el patrón seria poco creíble por lo que el sistema lo marcaria como comportamiento inapropiado o anómalo y activa el módulo de Fraude reduciendo el impacto de esos eventos en el puntaje final.

iv. Cálculo del índice de fatiga. (Fatigue Score)

Es una métrica cuantitativa diseñada para estimar el nivel de fatiga del usuario (Cliente) a partir de los indicadores observables en el comportamiento facial, se construye a partir de la frecuencia de bostezo detectados dentro de una ventana temporal permitiendo expresar este comportamiento como una tasa normalizada y comparable en el tiempo

$$Bh = \frac{B * 3600}{T \text{ ventanas}} \quad (4)$$

Datos:

- Bh: Fatigue Score - Representa la frecuencia de eventos asociados a fatiga normalizada a una hora de observación.
- B: Número total de bostezos detectados dentro de la ventana de análisis.
- T_{Ventanas} : Duración de la ventana de análisis expresada en segundos.
- 3600: Constante que corresponde al número de segundos en una hora

La ecuación 4 transforma el conteo de bostezos detectados dentro de una ventana temporal en una tasa equivalente por hora por lo que facilita comparar diferentes intervalos de medición en la formula correspondiente:

- B: número total de bostezos detectados dentro de la ventana de análisis, porque un bostezo se considera detectado cuando el Mouth Ratio supera el umbral y se mantiene durante un tiempo mínimo evitando que movimientos breves sean contabilizados por bostezos.
- T_{Ventanas} : Duración de la ventana de análisis expresada en segundos ya que por esta variable representa el intervalo real sobre el cual se contabilizaron los bostezos por ejemplo 5 minutos, 10 minutos, etc.

- 3600: Constante que corresponde al número de segundos en una hora su función es escalar la frecuencia observada en la ventana a una métrica estándar.

En el artículo “*Real-Time Fatigue Detection Algorithms Using Machine Learning for Yawning and Eye State*” de *Makhmudov* [16] la razón de esta fórmula es que el conteo B por sí solo depende del tamaño de la ventana, una ventana más larga tiende a acumular más eventos, aunque el comportamiento real del usuario no haya cambiado, en cambio B_h (Bostezo por hora) representa la tasa horaria equivalente que resultaría si el ritmo observado durante la ventana se mantuviera constante en el tiempo se trata de normalización temporal utilizada para el cálculo posterior de la fatiga.

Ejemplo:

Imaginemos que se está monitoreando el comportamiento de un operador durante una ventana de 10 minutos para detectar bostezos. El sistema ha registrado que el operador hizo 4 bostezos durante ese intervalo. Ahora, aplicamos la fórmula para convertir esos 4 bostezos a una tasa horaria (B_H).

Datos:

- B: Número total de bostezos detectados en la ventana de 10 minutos.
- $T_{ventana} = 10$ minutos = 600 segundos (convertimos la duración de la ventana a segundos).
- 3600: número de segundos en una hora (usamos esta constante para convertir a tasa horaria)

$$B_h = \frac{B * 3600}{T_{Ventanas}}$$

$$B_h = \frac{4 * 3600}{600s}$$

$$B_h = \frac{14400}{600s} = 24 \text{ Bostezos/h}$$

El $B_h = 24$ significa que, si el patrón observado de bostezos (4 bostezos en 10 minutos) se mantuviera constante durante una hora, el operador tendría 24 bostezos por hora.

¿Porque es útil este cálculo?

- Estandarización temporal: Con este cálculo podemos comparar fácilmente intervalos de diferente duración, ya que convertimos los eventos a una frecuencia estándar por hora.
- Comparación entre ventanas: Si analizamos otra ventana de 5 minutos donde se registran 2 bostezos, también podemos calcular la tasa horaria para esa ventana y compararlas sin que el tamaño de la ventana influya.
- Detección de fatiga: Un valor de B_h más alto se asocia con mayor fatiga o somnolencia, ya que indica un aumento en la frecuencia de bostezos.

v. Cálculo del índice ocular (Ocular Score)

Se aplica como el componente que resumen la fatiga ocular acumulada en una ventana temporal de análisis, su función principal es transformar la información de cierre ocular captada por EAR y resumida por PERCLOS que es un valor normalizado y comparable con el resto de los puntajes del modelo especialmente con el fatigue score para posteriormente integrarse en el Stress Score.

Flujo de Calculo

Primero se utilizó la **ecuación 1** para calcular el EAR y se clasifica el estado ocular.

- “Ojo cerrado” si $EAR < 0.20$
- “Ojo abierto” si $EAR \geq 0.20$

Cálculo de PERCLOS en la ventana

Una vez finalizada la ventana, se calcula:

$$\text{➤ } Perclos(\%) = \frac{N \text{ ojos Cerrados}}{N \text{ total de muestras}}$$

Donde N ojos cerrados es el número de frames clasificados como cerrados y N total es el total de frames evaluados en la ventana.

Conversión a Ocular Score (Normalización a 0 – 100)

Finalmente, la ecuación 2 del PERCLOS se transforma en un puntaje:

$$Ocular \ Score = 100 * Perclos \ (5)$$

El factor 100 corresponde a la conversión a porcentaje y permite que el ocular score se exprese en una escala 0 a 100 consistente con los demás indicadores del modelo.

Datos:

- **PERCLOS:** Porcentaje de tiempo que el operador tiene los ojos cerrados.
- **100:** Un factor escalar para una escala de 0-100
- **Ocular Score:** Refleja que tan alto fue el cierre ocular.

Interpretación Operativa

El Ocular Score se interpreta como porcentaje de cierre ocular acumulado en la ventana:

- Valores bajos indican que el operador mantuvo los ojos abiertos la mayor parte del tiempo.
- Valores altos indican una fracción significativa de cierre ocular sostenido o repetido, lo cual es relevante como señal de somnolencia o disminución de alerta.

Uso dentro del modelo en general.

Una vez obtenido, el Ocular Score se utiliza como entrada directa en el cálculo del puntaje final del sistema (StressScore) junto al Fatigue Score (componente general basado en bostezos). De esta forma, el modelo integra dos dimensiones complementarias:

- Dimensión ocular (Ocular Score) → asociada a somnolencia y cierre ocular acumulado.
- Dimensión conductual/general (Fatigue Score) → asociada a frecuencia de bostezos y fatiga general.

Además, en caso de activarse el módulo de fraude, el impacto del puntaje puede ajustarse con una penalización, manteniendo la confiabilidad del resultado final.[17]

Ejemplo de Aplicación:

Si en una ventana se analizan 100 muestras y se detectan 40 como “ojo cerrado” (EAR<0.20)

$$PERCLOS = \frac{40}{100} = 0.40 \times 100\% = 40\%$$

$$Ocular\ Score = 100 * 0.40 = 40$$

Esto significa que, dentro de esa ventana, aproximadamente el 40% del tiempo el ojo estuvo cerrado, lo cual incrementa la componente ocular del puntaje final.[18]

vi. Detección y penalización por fraude.

El fraude score es una medida que detecta comportamientos anómalos en el sistema como intentos de manipulación del operador para fingir fatiga o estrés, esto se hace monitoreando patrones atípicos que no se alinean con lo esperado fisiológicamente cuando se detectan estos comportamientos el fraude score penaliza el puntaje global del operador para evitar que estos eventos distorsionen la evaluación final.

¿Cómo se calcula el fraude Score?

$$\text{FraudScore} = w_1 * B_h + w_2 * \text{PERCLOS} (5)$$

Datos:

- **Fraud Score:** Representa una medida agregada de sospecha.
- **B_h:** Es la tasa de bostezos por hora.
- **PERCLOS:** Porcentaje de tiempo que el operador tiene los ojos cerrados.
- **w₁:** Constante de peso para B_h.
- **w₂:** Constante de peso para PERCLOS.

El FraudScore representa una medida agregada de sospecha, construida a partir de la combinación ponderada de indicadores de fatiga, con el objetivo de identificar comportamientos atípicos que distorsionen la evaluación final.

El fraude score se calcula a partir de varios factores que indican comportamientos sospechosos:

- B_h Es la tasa de bostezos por hora, que mide cuántos bostezos el operador tiene por hora. Si esta tasa es excesivamente alta, podría indicar que está fingiendo fatiga.
- PERCLOS es el porcentaje de tiempo que el operador tiene los ojos cerrados (un valor entre 0 y 1). Si este porcentaje es demasiado alto, podría ser una señal de que el operador está forzando el cierre de los ojos para parecer más fatigado.
- w₁ y w₂ son pesos que se asignan a cada factor. Los pesos son importantes porque indican la importancia relativa de cada factor en la detección de fraude. Los pesos permiten que el sistema dé más importancia a ciertos comportamientos sobre otros.

¿Porque Usamos los pesos?

Sirven para definir cuanto importa cada factor en el cálculo final del fraude score, por ejemplo:

- w_1 (peso para B_h): Los bostezos son probablemente más indicativos de fraude, ya que un operador podría intentar fingir fatiga bostezando repetidamente.
- w_2 (peso para PERCLOS): El porcentaje de ojos cerrados también es relevante, pero no tanto como los bostezos. Por eso le damos un peso más bajo.

Por lo tanto, los pesos se llegan ajustar según la importancia relativa que cada comportamiento tiene en la detección de fraude, los valores típicos para los pesos son:

- $w_1 = 0.5$ (50% de importancia a los bostezos)
- $w_2 = 0.5$ (50% de importancia al porcentaje de ojos cerrados)

El coeficiente 2.0 lo usamos para ajustar la penalización aplicada al puntaje final cuando se detecta fraude, es decir si el fraude score es alto la penalización será alta, pero cuando este coeficiente permite que los comportamientos fraudulentos moderados tengan un impacto significativo, pero no excesivo.

¿Porque utilizamos este coeficiente?

Porque está pensando para que el fraude score se multiplique por un valor que incremente la penalización de manera moderada si fuera un valor bajo (por ejemplo 1.0) la penalización sería más suave y si fuera mayor (por ejemplo 3.0) la penalización sería demasiado severa, por eso el valor 2.0 fue elegido un valor balanceado que no cause fluctuaciones extremas en el puntaje final pero que aún penalice fuertemente los comportamientos fraudulentos.

Ejemplo de cálculo del Fraude Score.

Supongamos que tenemos los siguientes datos de una ventana de análisis de 10 minutos:

- Tasa de bostezos por hora (B_h) = 40 (el operador tuvo 4 bostezos en 10 minutos, lo que se traduce en 40 bostezos por hora).
- PERCLOS = 0.60 (el operador tuvo los ojos cerrados el 60% del tiempo durante la ventana de 10 minutos).

Y asignamos los siguientes pesos:

- $w_1 = 0.5$ (50% de importancia a los bostezos)
- $w_2 = 0.5$ (50% de importancia a PERCLOS)

Lo calculamos como:

$$\text{FraudScore} = 0.5 * 40 + 0.5 + 0.60$$

$$\text{FraudScore} = 20 + 0.30 = 20.30$$

$$\text{FraudPenalty} = 2.0 * 20.30 = 40.60$$

El puntaje final del operador se redujera en 40.60 puntos debido al fraude detectado, gracias a este valor de penalización es moderado y refleja que el operador ha mostrado señales de comportamiento sospechoso.

¿De dónde Obtenemos los datos?

- Pesos (w_1 y w_2): Los pesos son ajustes operativos que podemos calibrar en la parte del backend, dependiendo de la importancia de cada factor. En este ejemplo, le damos el mismo peso (50%) a los bostezos y a PERCLOS, pero en la práctica, podrías ajustar estos valores según lo que consideres más importante en tu sistema. Por ejemplo, si crees que los bostezos son más relevantes que el cierre ocular, puedes asignar un peso más alto a w_1 y un peso más bajo a w_2
- El coeficiente 2.0 es un valor ajustable que se usa para aplicar una penalización significativa pero controlada el valor 2.0 es un valor estándar utilizado nuestro programa, pero en caso de cambios se podría ajustar más adelante en función de los resultados o pruebas adicionales.

vii. Fórmula final de StressScore.

$$\text{StressScore} = 0.55 * \text{Fatigue Score} + 0.45 * \text{Ocular Score} - \text{Fraud Penalty} \quad (6)$$

Datos:

- Stress Score: Indicador global del sistema que resumen el estado del operador en una escala normalizada de 0 a 100 dentro de una ventana temporal de análisis.
- Fatigue Score: Representa el componente conductual de fatiga, derivado principalmente de la frecuencia de eventos de bostezos normalizados en el tiempo.
- Ocular Score: Corresponde al componente ocular, calculado a partir de métricas como EAR y PERCLOS asociadas al estado de apertura y cierre temporal.
- Fraud Penalty: Es un término de penalización que reduce el puntaje final cuando se detectan comportamientos anómalos o inconsistentes mediante el módulo de fraude.
- Los coeficientes 0.55 y 0.45 representan los pesos relativos asignados a cada componente, definidos de acuerdo con su importancia dentro del modelo propuesto.

El Stress Score es el indicador global del sistema, el objetivo es resumir en un solo valor de 0 a 100 el estado del operador dentro de una ventana temporal de análisis se utiliza porque en el monitoreo continuo es más útil tener una única medida que el sistema pueda interpretar para:

- Mostrar un estado en el dashboard (Óptimo / Alerta / Alto / Crítico),
- Comparar ventanas en el tiempo,
- Activar decisiones (alertas, recomendaciones de descanso),
- Explicar el resultado usando métricas base (ocular y conductual).

| Rango de Stress Score | Nivel | Interpretación |
|-----------------------|---------|---|
| 0 – 39 | Óptimo | El trabajador se encuentra en un estado fisiológico estable. No se evidencian signos relevantes de fatiga ni somnolencia. El rendimiento cognitivo y la capacidad de atención se consideran normales. |
| 40 – 64 | Alerta | Se detectan indicios iniciales de fatiga o somnolencia. Aunque la condición aún no es crítica, se recomienda monitoreo continuo, ya que el estrés podría incrementarse si se mantiene la carga de trabajo. |
| 65 – 79 | Alto | El nivel de estrés es significativo. Existen patrones claros de fatiga general y/o cierre ocular prolongado. En este rango el riesgo de disminución del rendimiento y de errores operativos es elevado. |
| 80 – 100 | Crítico | El estado del trabajador es fisiológicamente inadecuado para continuar con la actividad. La fatiga es severa y puede comprometer seriamente la seguridad, la calidad del servicio y la salud del operador. Se recomienda una pausa inmediata. |

Tabla 3 Clasificación operativa del StressScore (0-100) por niveles de riesgo: Fuente: Autores

“Los rangos de clasificación del Stress Score fueron definidos de manera operativa para este trabajo, con el objetivo de traducir el valor continuo del índice en niveles cualitativos comprensibles para la toma de decisiones.”

La tabla 3 es de elaboración propia y se define para interpretar el StressScore de 0-100, los rango que lo establecemos como umbrales operativos para el sistema de monitoreo con el objetivo de transformar un índice continuo en niveles accionables como (Optimo, Alerta, Alto y Critico) al

agregar esto fundamenta la evidencia del PERCLOS que es un indicador ampliamente validado para la detección de somnolencia en el artículo “*Assessment of Drowsiness Based on Ocular Parameters Detected by Infrared Reflectance Oculography*” [19] y que las señales faciales como los bostezos y apertura bucal se emplean como marcadores de fatiga como se muestra en el artículo “*Driver’s Fatigue Detection Based on Yawning Extraction*” [20]

¿Por qué se suman Fatigue Score y Ocular Score?

- Fatigue Score: evidencia conductual (p. ej., bostezos sostenidos).
- Ocular Score: evidencia ocular (cierre ocular acumulado).

Al sumarlas las ponderadas, el sistema logra un indicador más estable.

¿Por qué se resta FraudPenalty?

Porque el fraude no es fatiga real: representa una distorsión o baja confiabilidad de la señal. Por eso se resta como penalización correctiva.

¿Porque agregamos valores 0.55 y 0.45?

| Componente | Peso | Interpretación técnica |
|---------------|------|---|
| Fatigue Score | 0.55 | Representa la dimensión conductual y cognitiva de la fatiga. Tiene mayor peso porque integra eventos sostenidos como bostezos y patrones de agotamiento general, que reflejan de forma más estable el estado global del trabajador. |
| Ocular Score | 0.45 | Representa la dimensión ocular de la fatiga (somnolencia, cierre prolongado de ojos). Es altamente relevante, pero puede verse afectada por factores externos como iluminación, reflejos o posición de la cámara. |
| Suma de pesos | 1.00 | Garantiza que el StressScore conserve una escala normalizada entre 0 y 100. |

Tabla 4 interpretación técnica Fuente: Autores.

En la tabla 4 explican que no son valores normativos ni de un estándar internacional, sino de parámetros del prototipo, seleccionados bajo criterio técnico para cumplir tres objetivos:

Mantener la escala normalizada

- Al sumar 1.0, la combinación funciona como un promedio ponderado, evitando que el StressScore se infle artificialmente.

Dar prioridad ligera a la fatiga general

- El Fatigue Score recibe un peso ligeramente mayor (55%) porque integra comportamientos más persistentes en el tiempo (como la tasa de bostezos), los cuales representan mejor el nivel de agotamiento cognitivo en contextos laborales como los centros de llamadas.

Conservar la relevancia de la dimensión ocular

- El Ocular Score mantiene un peso alto (45%), ya que la somnolencia ocular es un indicador fisiológico directo del estado de alerta, aunque puede verse afectada por condiciones ambientales.[21]

Además, estos coeficientes se consideran hiperparámetros calibrables, es decir, pueden ajustarse en futuras versiones del sistema mediante validación experimental con datos reales, sin modificar la estructura matemática del modelo.[22]

XI. Artículos relacionados.

Detección de fatiga mediante visión por computador.

PERCLOS como indicador válido de somnolencia/ fatiga visual La información sobre PERCLOS lo establece como un indicador muy confirmado para notar sueño sin intervención activa, explicando su aplicación común como la proporción del tiempo donde los ojos están casi cerrados en un lapso dado, y demostrando que este valor sube con falta de descanso, durante la noche o al vigilar por mucho tiempo. Esta investigación importa porque presenta a PERCLOS como un método fuerte y comúnmente usado para calcular la baja de atención basándose en el ojo, y también aborda dificultades reales como establecer con exactitud qué significa estar con el "ojo cerrado" y cómo afectan las circunstancias de la grabación.[23].

EAR para detección de parpadeo y cierre ocular en tiempo real

EAR para parpadeo y cierre ocular en tiempo real La manera de detectar somnolencia usando puntos clave de la cara y calculando el (EAR) para identificar si los ojos se cierran o parpadean en videos al momento. Lo principal que aportan es hacer funcionar una métrica sencilla y rápida de procesar que se une con la localización facial y puntos clave para decir si el ojo está abierto o cerrado y cuándo ocurren los parpadeos. [24]

Combinación de señales faciales: ojos (EAR/PERCLOS) + boca (MAR) para fatiga

Combinación de señales faciales ojos (EAR/PERCLOS) + boca (MAR) para fatiga se muestra un método para detectar sueño que examina el área de los ojos y además añade la boca (MAR) con el fin de registrar indicadores parecidos a bostezos; la estrategia principal utiliza visión artificial y aprendizaje profundo para clasificar o decidir al instante, señalando que juntar varias "pistas" de la cara afina la identificación comparada con basarse solo en una medida.

Revisiones recientes sobre tecnologías de fatiga basadas en video.

Las maneras de detectar cansancio recopilan el empleo frecuente de indicadores como abrir y cerrar los ojos, el tiempo que se están cerrados, bostezar y la posición del cuello, usando tanto métodos antiguos como umbrales o reglas fijas y modelos avanzados de inteligencia artificial y redes neuronales. Este tipo de resumen ayuda a demostrar la posibilidad de que un mecanismo que no requiere contacto físico, usando una cámara, funcione bien cuando se necesite implementar a gran escala.[25]

Estrés laboral y carga cognitiva en entornos de oficina y call centers.

La tensión laboral es un fenómeno de carácter multifactorial que surge de la interacción entre exigencias cognitivas, demandas emocionales y condiciones organizacionales del entorno de trabajo. En los centros de atención telefónica, estas exigencias se intensifican debido a la necesidad de mantener concentración constante, procesar información de forma rápida, cumplir objetivos de desempeño estrictos y gestionar interacciones continuas con usuarios que pueden presentar cargas emocionales elevadas.

Este tipo de entorno laboral presenta una alta probabilidad de generar fatiga mental y desgaste emocional, lo cual afecta tanto al bienestar del trabajador como al desempeño operativo y a la calidad del servicio brindado.

La Constitución pública del Ecuador afirma que cada empleado tiene derecho a hacer su trabajo en un entorno que sea bueno y favorable, asegurando su bienestar, seguridad, salud, integridad e higiene. Este concepto fundamental significa que el jefe está obligado a tomar acciones para cuidar tanto la salud física como la mental de quien trabaja para él. [26]

Adicionalmente, la ley laboral fija el tiempo máximo de trabajo, el derecho a parar y las condiciones básicas que se deben asegurar para mantener al empleado sano. Aunque la ley no menciona claramente descansos concretos para bajar el estrés o el cansancio visual mientras se trabaja, sí fomenta un balance entre el trabajo a realizar y el tiempo de reposo, permitiendo así añadir otras acciones enfocadas en prevenir peligros.

Por otro lado, el reglamento sobre Seguridad y Salud en el Empleo y las pautas técnicas del Ministerio de Trabajo exigen que los dueños de negocios reconozcan, midan y manejen los peligros que existen en el lugar de trabajo.[27]

Estrés laboral en call centers y efectos de la digitalización/monitoreo.

Los programas de vigilancia avanzados que no tocan a la gente son herramientas tecnológicas pensadas para medir cómo se encuentran física y mentalmente las personas sin necesidad de tocarlas ni meterse en sus labores diarias. En el trabajo, estas soluciones usan sobre todo métodos de visión artificial, manejo de señales y programas de inteligencia artificial para sacar datos importantes de lugares fáciles de acceder, como cámaras normales o sensores del entorno.

Cambiando a las formas antiguas que usan encuestas o aparatos que uno lleva puestos, vigilar sin contacto deja ver de forma constante y real cómo está el empleado, lo que baja lo subjetivo y lo que molesta al hacer el trabajo. Viendo el video al momento, se pueden notar costumbres de comportamiento y del cuerpo ligadas al cansancio y la tensión, como cambios en cuánto se parpadea, si los ojos se cierran mucho tiempo, cómo cambia la postura o hacia dónde se gira la cabeza.

Usar programas inteligentes ayuda a juntar muchas cosas en indicadores mezclados, haciendo que el sistema pueda calcular mejor y con más seguridad el grado de tensión o agotamiento. Además, usar diseños de computadora con cliente y servidor deja que el análisis se haga en un punto central, guarde la información organizada y muestre lo que sale en páginas web, algo muy bueno para sitios con muchas estaciones de trabajo, como los centros de llamadas.

XII. Metodología.

La presente investigación adopta un enfoque cuantitativo para el desarrollo de un prototipo de monitoreo de fatiga y estrés basado en visión por computadora, este sistema se encarga de analizar métricas fisiológicas no invasivas, que se obtienen a partir de características faciales y oculares capturadas mediante una cámara web, sin necesidad de sensores físicos adicionales

La metodología se estructura considerando el enfoque, alcance y el diseño de la investigación, así mismo el planteamiento técnico para poder desarrollar el prototipo

Enfoque de la Investigación.

El enfoque de la investigación es cuantitativo, ya que se fundamenta en la medición numérica de variables asociadas a la fatiga y el estrés visual. Para ello, se analizan métricas fisiológicas no invasivas obtenidas mediante visión por computadora, entre las que se incluyen:

- EAR – Eye Aspect Ratio (Relación de Aspecto Ocular)
- PERCLOS – Percentage of Eye Closure (Porcentaje de Cierre Ocular)
- Yawning Frequency (Frecuencia de bostezos)
- Fatigue Score (Índice de fatiga normalizado)
- Stress Score (Índice de estrés normalizado)

Los datos recolectados se procesan mediante fórmulas matemáticas y umbrales predefinidos, lo que permite un análisis objetivo, repetible y comparable. En este estudio no se aplican entrevistas, percepciones subjetivas ni técnicas cualitativas, debido a que el propósito es evaluar el comportamiento de las métricas y la respuesta del prototipo en tiempo real durante su funcionamiento.

Alcance la de la Investigación.

El alcance de la investigación es descriptivo–correlacional, por las siguientes razones:

- Descriptivo: porque se registran y analizan los valores que adoptan las métricas de fatiga y estrés durante el uso continuo del sistema, observando su comportamiento en distintos escenarios de exposición frente a una pantalla.
- Correlacional: porque se analiza la relación entre las métricas oculares y faciales (EAR, PERCLOS y *Mouth Ratio*) y el índice final de estrés calculado por el sistema, con el fin de identificar cómo se asocian entre sí dentro del modelo computacional.

La investigación no busca establecer causas médicas ni emitir diagnósticos clínicos. En su lugar, se orienta a examinar la interacción y consistencia de las variables medidas, y cómo estas se vinculan dentro de un entorno de monitoreo automatizado.

Diseño de la Investigación.

El diseño de la investigación es no experimental, ya que las variables no se manipulan de manera deliberada únicamente se observan y registran tal como ocurren durante el uso normal del sistema.

Adicionalmente, el diseño presenta las siguientes características:

- Transversal, debido a que los datos se recolectan en intervalos de tiempo definidos durante sesiones específicas de monitoreo, sin realizar seguimiento prolongado a los sujetos evaluados.
- Longitudinal a nivel del sistema, ya que el prototipo ejecuta mediciones continuas en ventanas temporales sucesivas mientras el usuario permanece frente a la pantalla, lo cual permite observar la evolución de las métricas dentro de una misma sesión.

Este diseño resulta adecuado para evaluar el funcionamiento del prototipo y validar su comportamiento en tiempo real bajo condiciones de uso controladas.

12.1 Planteamiento del desarrollo del prototipo.

Para desarrollar este prototipo se utilizó una arquitectura cliente-servidor distribuida que separa las responsabilidades de captura, procesamiento y consolidación de la información gracias a este sistema que se diseñó como una solución modular compuesta por tres elementos fundamentales que operan de manera coordinada.

- **Módulo Cliente (Python):** Aplicación autónoma que se ejecuta en cada estación de trabajo monitoreada. Utiliza Media Pipe para la detección de 478 puntos faciales (landmarks) en tiempo real, extrae métricas oculares (Eye Aspect Ratio, PERCLOS) y métricas bucales (detección de bostezos), y consolida esta información en ventanas temporales de análisis. Cada cliente opera de forma independiente, procesando datos localmente para minimizar latencia y carga de red.
- **Módulo Backend (FastAPI):** Servidor centralizado que recibe payloads JSON de múltiples clientes simultáneamente. Implementa una arquitectura RESTful con almacenamiento dual (memoria para operaciones en tiempo real). Gestiona la lógica de negocio, registro de equipos, consolidación de métricas, gestión de descansos obligatorios, y coordinación con sistemas de control ambiental mediante integración con Home Assistant la cual permite coordinar las decisiones generadas por el backend con la ejecución de acciones sobre dispositivos IoT del entorno.
- **Capa de Conectividad:** Protocolo HTTP/HTTPS para comunicación asíncrona entre clientes y servidor. Los clientes obtienen la dirección IP del servidor desde un GitHub Gist público, permitiendo movilidad de la infraestructura sin reconfiguración manual. El servidor publica su propia IP mediante un actualizador automático al iniciar.

En el diagrama 1 se prepara la configuración y obtiene la IP del servidor (Gist o manual), luego registra la PC. Después entra al ciclo de monitoreo verifica descanso, captura video, calcula métricas en la ventana de envío y manda el payload al backend. El ciclo se repite para mantener la medición continua.

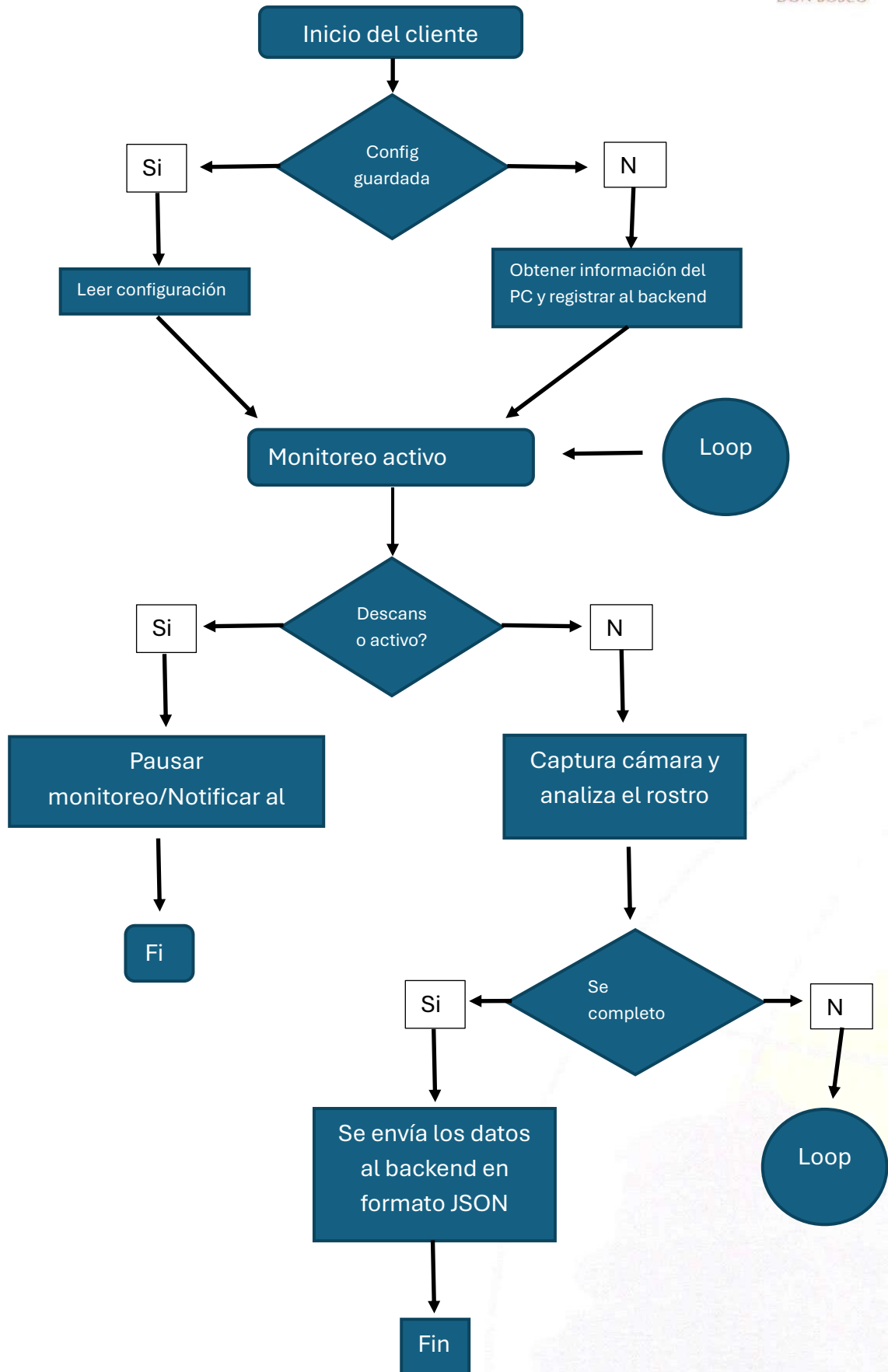


Diagrama 1 Flujo Operativo del Cliente Fuente: Autores.

En el diagrama 2 el servidor inicia, carga configuración, prepara almacenamiento y tareas automáticas. Luego publica su IP en Gist y queda listo para recibir ``/api/estres``. Cada petición se valida, se guarda (memoria y disco), se evalúa el estado y se envían respuestas o notificaciones

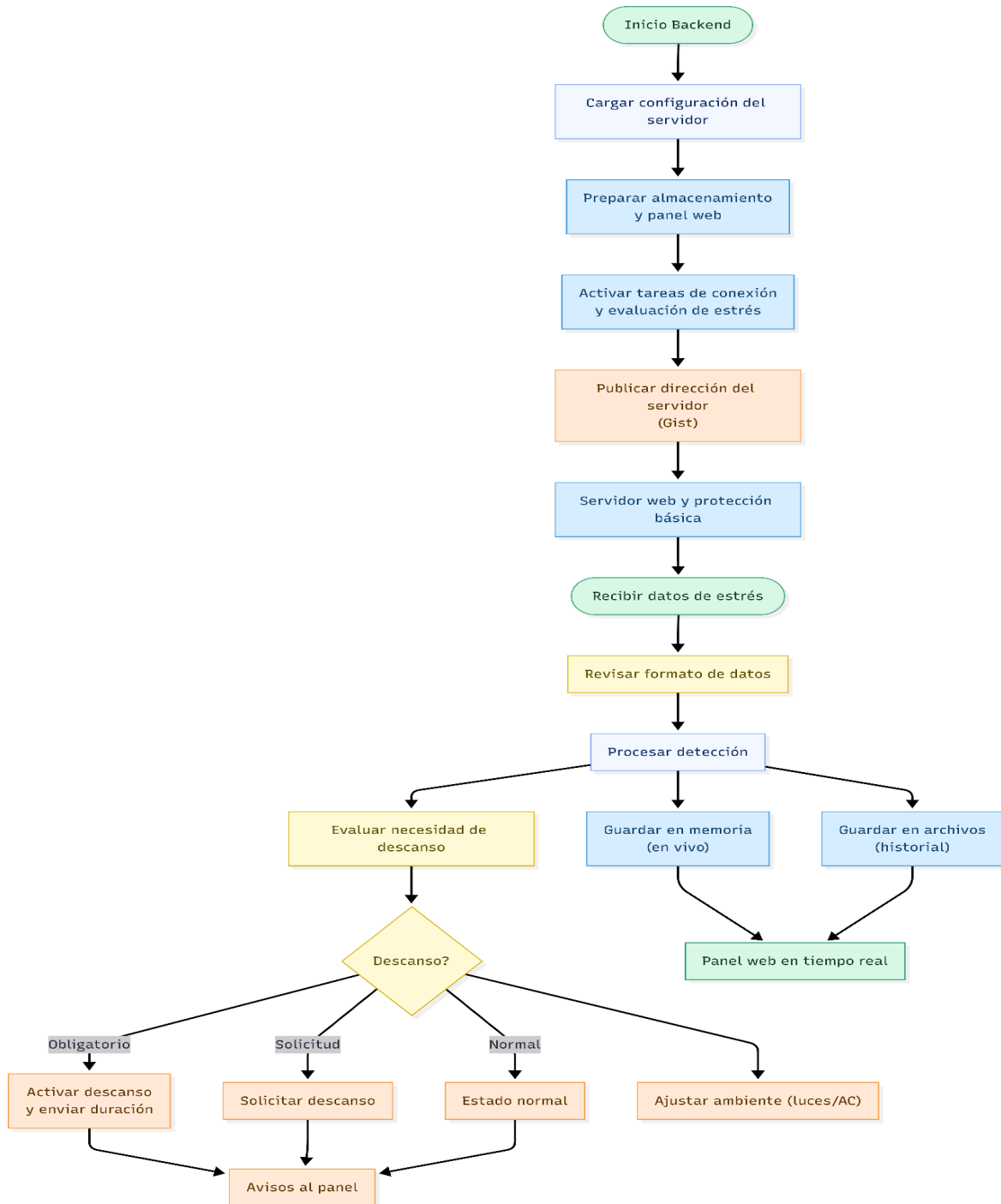


Diagrama 2 Inicio del Backend Fuente: Autores.

En el Diagrama 3, el módulo cliente obtiene la dirección IP del servidor desde GitHub Gist, verifica la conectividad, se registra en la central y envía pings periódicos para confirmar que permanece activo. En cada ventana de análisis, el cliente transmite las métricas al endpoint /api/estres/ y consulta el estado de descanso en /api/descansos/estado/: pc_id, donde el backend responde si el operador se encuentra en pausa o en monitoreo.

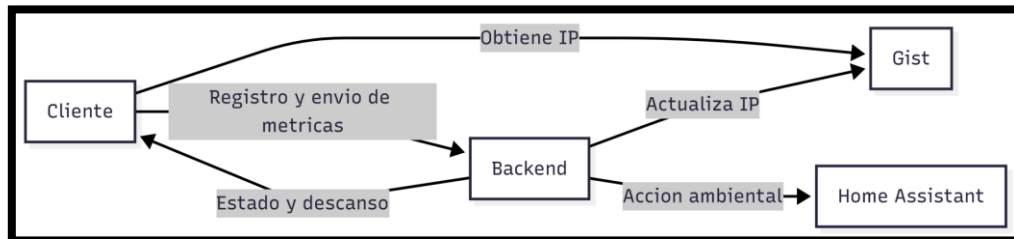


Diagrama 3 Conectividad Fuente: Autores.

Desarrollo.

Con el fin de asegurar que el sistema pueda ser desplegado de forma consistente en distintas estaciones de trabajo, en el diagrama 4 se definió un procedimiento automatizado para la preparación del entorno de ejecución. Dicho procedimiento reduce la intervención manual del usuario y permite una configuración homogénea del sistema, independientemente del equipo donde se ejecute.

La inicialización del entorno se organiza en tres fases consecutivas. En primer lugar, se comprueba la disponibilidad del intérprete de Python requerido para la ejecución del sistema. Posteriormente, se genera un entorno virtual aislado utilizando el mecanismo **venv**, con el objetivo de separar las dependencias del proyecto del entorno global del sistema operativo. Finalmente, se realiza la instalación automática de las librerías necesarias a partir del archivo requirements.txt, garantizando que todas las dependencias queden alineadas con la versión utilizada durante el desarrollo y las pruebas que se presenta en el anexo 1.1 con fines de documentación técnica.

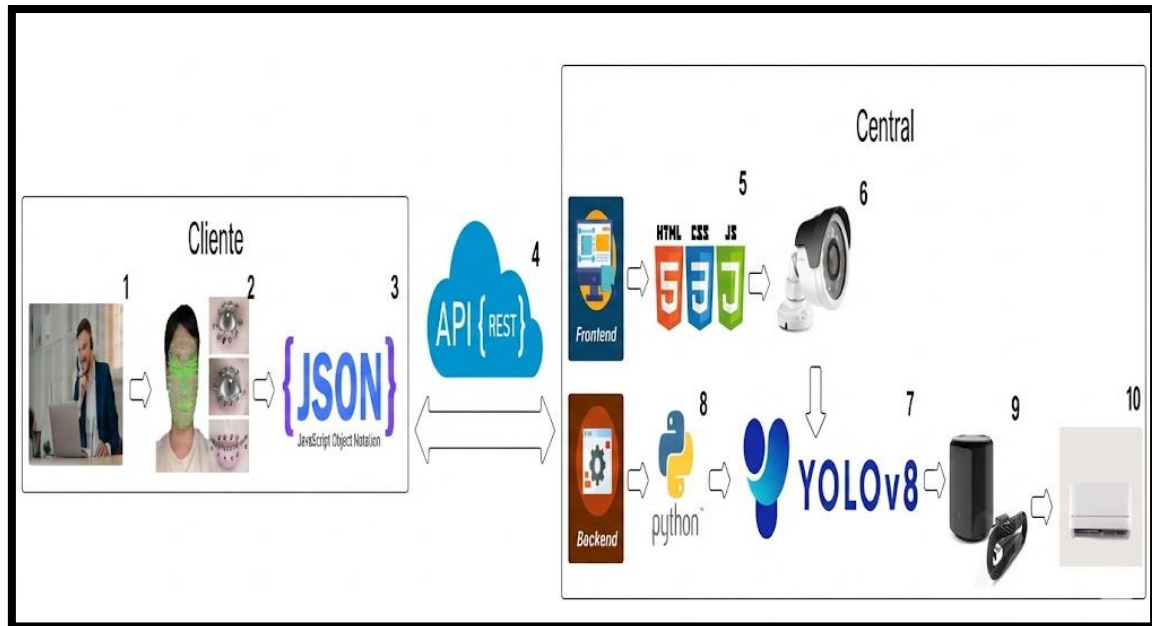


Diagrama 4 Presentación del Sistema Fuente: Autores.

1. Entrada de señal de video del trabajador de callcenter
2. Uso de mediapipe y fórmulas de EAR, PERCLOS ETC para estimar el estrés
3. Todos los datos en JSON
4. Uso de API Rest en LAN para el envío de datos a la central
5. Página web
6. Cámara con posición que se vea todo
7. Detección de personas con % de acierto para contabilizarlas
8. El backend usa Python web
9. RM4C Mini para el control ambiental
10. Aire acondicionado

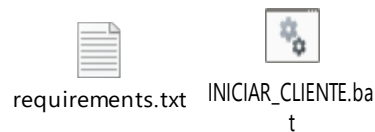


Figura 8 Script de Inicialización e Dependencias Fuentes: Autores.

Las dependencias empleadas en el cliente del sistema corresponden a librerías ampliamente utilizadas en procesamiento de imágenes, análisis numérico y comunicación entre aplicaciones. En particular, se utilizan herramientas para la captura y procesamiento de video, la detección de puntos faciales relevantes, el manejo eficiente de operaciones matemáticas y la comunicación con el servidor central mediante solicitudes HTTP, así como módulos orientados a la generación de notificaciones a nivel del sistema operativo.

La gestión centralizada de estas dependencias permite que todos los clientes del sistema operen bajo un mismo conjunto de librerías y versiones, evitando discrepancias en el comportamiento del prototipo y garantizando resultados consistentes durante las pruebas experimentales.

12.2 Selección del Modelo de Vision por computadora.

Para la detección de características faciales en tiempo real se adoptó la solución propuesta por Media Pipe, desarrollada por Google Research, debido a su adecuado equilibrio entre eficiencia computacional, precisión en la estimación de landmarks faciales y capacidad de operación en tiempo real sobre hardware de uso convencional.

El modelo facial que se observa en la figura 9 nos permite identificar hasta 478 puntos característicos del rostro, expresados en coordenadas normalizadas dentro del intervalo $[0, 1]$. No obstante, para los fines de este sistema se seleccionaron únicamente subconjuntos específicos correspondientes a la región ocular incluyendo párpados y comisuras y a la región bucal labios y comisuras. Esta selección dirigida reduce el costo computacional del procesamiento que se presenta el anexo 1.2 sin comprometer la precisión necesaria para el cálculo de las métricas de fatiga y estrés consideradas.

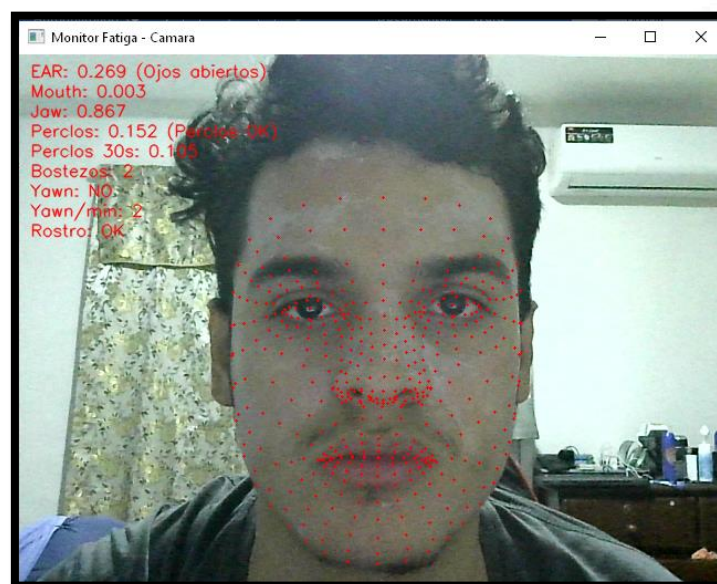


Figura 9 Esquema de landmarks faciales, región ocular y bucal Fuente: Autores.

12.3 Flujo Funcional de Procesamiento.

El procesamiento de la información se organiza siguiendo una estructura jerárquica como se presenta en el (Anexo 1.3) en la cual cada etapa depende directamente de la salida generada por la anterior. El flujo inicia con la captura continua de video, cuyos fotogramas son procesados por Media Pipe para la extracción de landmarks faciales relevantes.

A partir de estos puntos característicos, el sistema ejecuta dos procesos de análisis en paralelo. El primero corresponde al análisis ocular, donde se calcula el índice de apertura del ojo (EAR) y posteriormente se deriva la métrica PERCLOS. El segundo proceso se orienta al análisis bucal, empleando relaciones geométricas de apertura de boca y mandíbula para la detección de eventos de bostezo.

Los resultados obtenidos en ambas ramas convergen en una etapa de normalización y verificación de coherencia, orientada a identificar patrones inconsistentes o atípicos. Finalmente, la información validada se integra en la construcción del payload final, el cual resume el estado de fatiga estimado dentro de la ventana de análisis definida.

12.3 Inicialización de Media Pipe.

Con base en la selección del modelo descrita en el (Anexo 1.2), se procedió a la inicialización del modelo Face Landmarker de Media Pipe para la obtención de landmarks faciales en tiempo real. Esta decisión metodológica permite reutilizar de forma consistente las métricas oculares y bucales definidas previamente, asegurando una captura homogénea de los puntos característicos del rostro que sirven como entrada para las etapas posteriores del procesamiento.

12.4 Visualización de detecciones.

Durante la etapa de desarrollo se emplearon rutinas de visualización superpuesta con el fin de verificar en tiempo real la correcta alineación de los landmarks faciales detectados. Esta verificación operativa permitió confirmar que los datos de entrada utilizados para el cálculo de las métricas oculares y bucales que están detalladas en el Anexo 1.4 y efectivamente a regiones faciales válidas, reforzando la confiabilidad del proceso previo al análisis cuantitativo.

12.5 Calculo EAR.

Con base en la revisión bibliográfica realizada sobre el índice Eye Aspect Ratio (EAR), detallada en el (Anexo 1.3), se definió el procedimiento para el cálculo de la ecuación 1 y de esta métrica a partir de los landmarks oculares detectados en tiempo real. El objetivo de esta etapa es estimar de forma continua el estado de apertura ocular del operador durante la captura, constituyendo un insumo directo para el análisis de parpadeos y cierres oculares prolongados.

La correcta obtención del valor EAR se verificó mediante dos mecanismos complementarios. En primer lugar, se empleó la visualización superpuesta de los landmarks oculares sobre el rostro capturado, lo que permitió confirmar la alineación adecuada de los puntos utilizados en el cálculo. En segundo lugar, se realizó el seguimiento continuo de la señal EAR generada, observando un comportamiento coherente con los cambios visuales en la apertura de los ojos.

Tal como se muestra en las figuras correspondientes, el cálculo del EAR se mantiene estable y consistente durante la operación del prototipo, validando su uso como métrica base para las etapas posteriores del análisis de fatiga visual.

12.6 Buffer y PERCLOS.

Siguiendo la metodología descrita en el (Anexo 1.4), se definió un procedimiento para el análisis temporal del estado ocular a partir de los valores de EAR previamente calculados. Este procedimiento permite registrar de manera continua la evolución del estado de los ojos dentro de una ventana temporal deslizante y, a partir de ello, estimar el porcentaje de tiempo en el que los ojos permanecen cerrados utilizando la ecuación 2.

La transformación del EAR en una métrica temporal de tipo PERCLOS posibilita la identificación de cierres oculares sostenidos, los cuales no pueden ser detectados únicamente mediante valores instantáneos. De este modo, el PERCLOS se constituye como un indicador adecuado para el análisis de somnolencia, al reflejar patrones de fatiga visual prolongada en el tiempo.

12.7 Bostezos.

De acuerdo con la metodología descrita en el (Anexo 1.5), se estableció un procedimiento para el análisis de la apertura bucal a partir de relaciones geométricas obtenidas de los landmarks faciales. Este procedimiento permite cuantificar la apertura de la boca y de la mandíbula, así como identificar eventos de bostezo cuando dicha apertura se mantiene durante un intervalo temporal determinado.

Gracias al artículo[28] la incorporación de un criterio temporal en la detección de bostezos permite discriminar entre movimientos momentáneos de la boca y eventos sostenidos, reduciendo la aparición de falsos positivos. De este modo, la métrica bucal se integra al prototipo como un indicador complementario de fatiga, aportando información relevante sobre estados de cansancio que no siempre se manifiestan únicamente a través del análisis ocular.

12.8 Normalización y StressScore.

En la etapa de integración de métricas se aplicó la ecuación 6 en un proceso de normalización con el objetivo de expresar los distintos indicadores en una escala común comprendida entre 0 y 100. Esta normalización permite homogenizar métricas de naturaleza diferente, facilitando su comparación e integración dentro de una misma ventana de análisis.

Posteriormente, las métricas normalizadas fueron combinadas utilizando los pesos definidos en el (Anexo 1.6), dando lugar a un indicador compuesto de estrés. Este procedimiento transforma los indicadores metodológicos individuales en una salida operativa unificada, capaz de reflejar el nivel global de fatiga del operador y de ser comparada de forma consistente entre distintas ventanas temporales.

12.9 Construcción y Envío.

Finalmente, con base en lo establecido en el (Anexo 1.7), se definió un mecanismo de consolidación de la información generada durante cada ventana de análisis. En esta etapa se integran las métricas calculadas, el estado actual del usuario y los metadatos asociados a la captura, conformando una estructura de datos que resume el resultado del proceso de análisis.

Esta información consolidada se envía de forma periódica al backend del sistema, permitiendo materializar la metodología propuesta en un flujo de comunicación continuo y estructurado. De este modo, los resultados obtenidos en el cliente pueden ser almacenados, visualizados y utilizados para la toma de decisiones en tiempo casi real como se muestra en la figura 10.



Figura 10 Estado del Usuario Fuente: Autores.

Preparación del Backend.

El backend del sistema se estructura sobre un entorno de ejecución independiente, con sus dependencias gestionadas de forma centralizada. Esta configuración contempla librerías orientadas al desarrollo de servicios web, validación de datos, manejo de parámetros de configuración y comunicación con componentes externos del sistema.

Desde un enfoque metodológico, esta organización permite garantizar un despliegue reproducible del backend y mantener la compatibilidad operativa con el resto del prototipo, asegurando que el comportamiento del sistema sea consistente en los distintos escenarios de prueba, sin profundizar en detalles de implementación interna.

Recepción de datos, endpoints y conteo de personas.

A nivel metodológico, el backend centraliza la recepción de la información generada por los clientes del sistema a través de interfaces de comunicación previamente definidas. Estas interfaces permiten gestionar el registro de estaciones de trabajo, la recepción periódica de métricas de estrés y la administración de estados asociados a pausas o descansos, garantizando un flujo continuo y estructurado de datos hacia el núcleo del sistema.

De forma complementaria, se contempla la integración de un módulo de conteo de personas como fuente adicional de información contextual. Este componente aporta datos relacionados con la presencia y el nivel de ocupación del entorno, los cuales pueden ser utilizados para enriquecer el análisis del estado operativo sin requerir la descripción detallada de algoritmos específicos dentro del presente apartado metodológico.

Integración con sistema de control ambiental.

El backend incorpora un mecanismo de integración con el sistema de control ambiental del entorno, destinado a ejecutar acciones cuando el estado estimado del usuario así lo requiere. Esta integración se considera parte del enfoque metodológico del prototipo, en la medida en que vincula la detección de estados de fatiga o estrés con respuestas automáticas del entorno de trabajo.

La implementación se apoya en una capa de integración y automatización basada en Home Assistant mediante contenedores Docker, este enfoque nos permite separar la lógica de decisión del backend de la ejecución de acciones sobre dispositivos físicos favoreciendo una arquitectura modular y escalable el proceso de despliegue y configuración inicial de la plataforma se documenta en los (Anexos 2.1 y 2.2).

Para poder habilitar el control del aire acondicionado se llegó a integrar el dispositivo IoT Broadlink RM4C Mini dentro del Home Assitant validando su registro y conectividad en la red

local (Anexos 2.3 y 2.4), el equipo de aire acondicionado fue configurado como un control universal mediante la aplicación BroadLink asociándolo al RM4C Mini verificando su disponibilidad operativa (Anexo 2.5, 2.6, 2.7). Esta configuración permite la emisión de comando infrarrojos de encendido, apagado y ajuste de temperatura.

El diagrama 5 el backend se comunica con Home Assistant desplegado por Docker, ya que el home assistant ejecuta automatizaciones y envía comando infrarrojos mediante el RM4C mini hacia el equipo de aire acondicionado

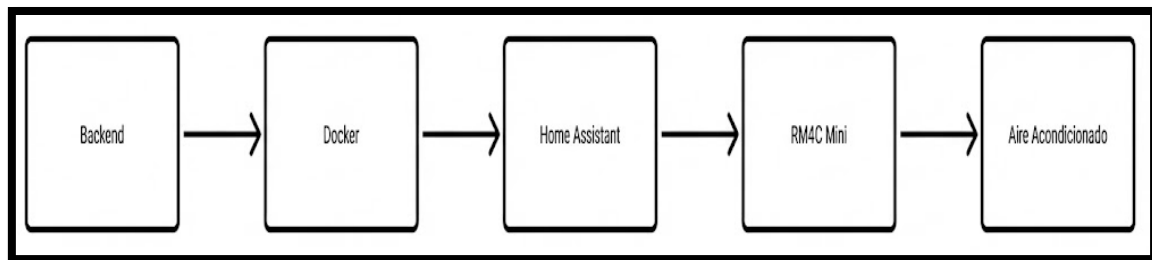


Diagrama 5 Flujo de Control del Aire Acondicionado Fuente: Autores.

La operación de control ambiental se definió mediante dos tipos de reglas de automatización:

- Programación Horaria.
- Regulación por ocupación.

En la programación horaria se gestiona el encendido y apagado del equipo en franjas laborales predefinidas asegurando condiciones térmicas adecuadas durante la jornada laboral y reduciendo el consumo energético fuera del horario operativo en la regulación por ocupación el backend utiliza el conteo de personas como información contextual para seleccionar una consigna de temperatura dentro de los rangos aceptados de confort térmico en oficinas.

De acuerdo con guías y artículos para entornos de oficina los rangos recomendados de temperatura operativa se sitúan en 20 – 24 °C para la calefacción y 23 – 26 °C para refrigeración, el prototipo implementa una política de ajuste gradual de la consigna sin salir del rango recomendado en refrigeración por ejemplo:

- 4 personas: 25 °C
- 5 – 8 Personas: 24 °C
- 9 o más: 23°C

Esta estrategia permite compensar el incremento de carga térmica asociado a una mayor ocupación y se alinea con los enfoque de control del aire acondicionado en presencia u ocupación reportados en el artículo “Occupancy-based HVAC control systems in buildings: A state-of-the-art review”[29].

Finalmente, se configuro el panel de control climatizado en el frontend del sistema, registrando la URL de Home Assistant, el token de acceso y las entendidas correspondientes al control remoto y al equipo del aire acondicionado (Anexo 2.8) con ello el sistema queda habilitado para ejecutar reglas automáticas de climatización integradas al flujo operativo general del prototipo.

Datos Obtenidos.

El enfoque adoptado se orienta a verificar el comportamiento funcional de las funciones de cálculo y de los umbrales definidos en el sistema, evaluando su respuesta frente a tres condiciones controladas para cada métrica analizada.

Las configuraciones previamente establecidas en la metodología general incluyendo horarios de evaluación, ventanas temporales y criterios de captura se conservan sin modificaciones, ya que el objetivo de este apartado no es redefinir el proceso de adquisición de datos, sino documentar de manera específica la validación operativa del prototipo bajo escenarios controlados.

Validación de EAR.

Ojos Abiertos

En la figura 11 se observa al operador con los párpados completamente levantados, lo que genera una separación visible entre el párpado superior e inferior del ojo. El valor instantáneo del índice EAR mostrado en pantalla es aproximadamente **0.251**, el cual se encuentra claramente por encima del umbral definido en el sistema (**0.20**).

La gráfica del EAR en función del tiempo presenta pequeñas variaciones a lo largo de la ventana analizada. Estas variaciones corresponden a parpadeos naturales y movimientos normales del rostro, los cuales provocan descensos breves del valor del EAR. Sin embargo, dichas caídas son momentáneas y el valor se recupera inmediatamente, manteniéndose mayoritariamente por encima del umbral.

Este comportamiento es esperado y deseable, ya que el sistema no interpreta los parpadeos normales como cierres oculares prolongados. Durante toda la ventana de análisis, el 100 % de los intervalos temporales fueron correctamente clasificados como “ojos abiertos”, lo que demuestra que el algoritmo es estable y no genera falsas detecciones de cierre bajo condiciones normales de operación.

En conjunto, la figura 11 y la tabla 5 evidencian que el índice EAR permite identificar de forma confiable el estado de ojos abiertos, incluso en presencia de variaciones naturales propias del comportamiento humano.

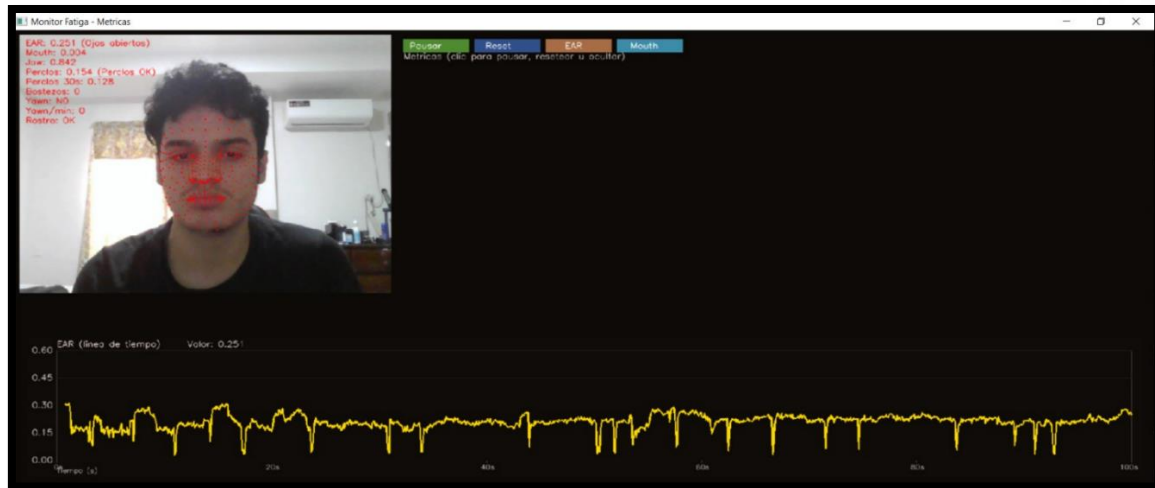


Figura 11 Ojos abiertos para validar el EAR. Fuente: Autores.

| <i>Intervalo de tiempo (s)</i> | <i>EAR promedio</i> | <i>EAR mínimo</i> | <i>EAR máximo</i> | <i>Umbral EAR (0.20)</i> | <i>Estado detectado</i> |
|--------------------------------|---------------------|-------------------|-------------------|--------------------------|-------------------------|
| 0 – 10 | 0.248 | 0.235 | 0.262 | 0.20 | Ojos abiertos |
| 10 – 20 | 0.251 | 0.238 | 0.269 | 0.20 | Ojos abiertos |
| 20 – 30 | 0.254 | 0.240 | 0.271 | 0.20 | Ojos abiertos |
| 30 – 40 | 0.247 | 0.232 | 0.263 | 0.20 | Ojos abiertos |
| 40 – 50 | 0.250 | 0.236 | 0.268 | 0.20 | Ojos abiertos |
| 50 – 60 | 0.252 | 0.239 | 0.270 | 0.20 | Ojos abiertos |
| 60 – 70 | 0.249 | 0.234 | 0.265 | 0.20 | Ojos abiertos |
| 70 – 80 | 0.253 | 0.241 | 0.272 | 0.20 | Ojos abiertos |
| 80 – 90 | 0.246 | 0.231 | 0.260 | 0.20 | Ojos abiertos |
| 90 – 100 | 0.251 | 0.237 | 0.269 | 0.20 | Ojos abiertos |

Tabla 5 Registro temporal del EAR – Condición: ojos abiertos

Ojos Entrecerrados.

En la figura 12 se observa al operador con los párpados parcialmente descendidos, reduciendo la apertura vertical del ojo. El valor instantáneo del EAR mostrado en pantalla es cercano a 0.195, ubicándose por debajo del umbral configurado de 0.20, por lo que el sistema interpreta correctamente el estado como cierre ocular.

La gráfica del EAR en función del tiempo presenta variaciones puntuales, las cuales se producen cuando el operador eleva ligeramente los párpados de forma momentánea para verificar la continuidad de la captura durante los 140 segundos de prueba. Estas elevaciones generan picos temporales en el valor del EAR; sin embargo, dichos valores no superan de forma sostenida el umbral, por lo que el sistema no reclasifica erróneamente el estado como ojos abiertos.

Durante toda la ventana temporal analizada en la tabla 6 el 100 % de los intervalos fueron correctamente clasificados como “ojos cerrados”, demostrando que el umbral definido es lo suficientemente robusto para identificar cierres parciales y tolerar variaciones naturales sin afectar

la detección. Este comportamiento resulta especialmente útil para la identificación de parpadeos prolongados y signos tempranos de fatiga visual.

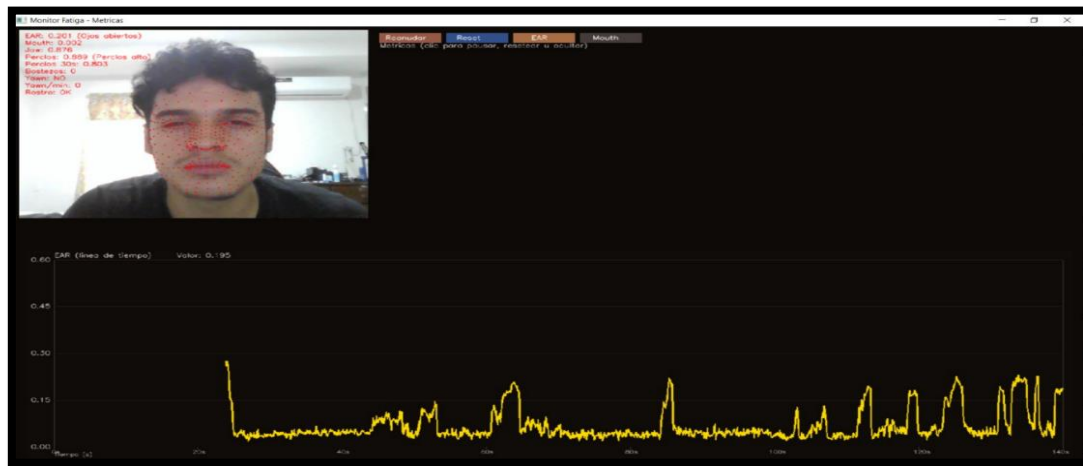


Figura 12 Ojos entrecerrados para validar el EAR. Fuente: Autores.

| <i>Intervalo de tiempo (s)</i> | <i>EAR promedio</i> | <i>EAR mínimo</i> | <i>EAR máximo</i> | <i>Umbral EAR (0.20)</i> | <i>Estado detectado</i> |
|--------------------------------|---------------------|-------------------|-------------------|--------------------------|-------------------------|
| 0 – 14 | 0.182 | 0.165 | 0.201 | 0.20 | Ojos cerrados |
| 14 – 28 | 0.041 | 0.028 | 0.062 | 0.20 | Ojos cerrados |
| 28 – 42 | 0.048 | 0.031 | 0.072 | 0.20 | Ojos cerrados |
| 42 – 56 | 0.061 | 0.038 | 0.089 | 0.20 | Ojos cerrados |
| 56 – 70 | 0.093 | 0.052 | 0.158 | 0.20 | Ojos cerrados |
| 70 – 84 | 0.051 | 0.034 | 0.078 | 0.20 | Ojos cerrados |
| 84 – 98 | 0.064 | 0.041 | 0.104 | 0.20 | Ojos cerrados |
| 98 – 112 | 0.082 | 0.049 | 0.146 | 0.20 | Ojos cerrados |
| 112 – 126 | 0.091 | 0.055 | 0.171 | 0.20 | Ojos cerrados |
| 126 – 140 | 0.097 | 0.061 | 0.189 | 0.20 | Ojos cerrados |

Tabla 6 Registro temporal del EAR – Condición: ojos entrecerrados

Ojos Cerrados.

En la figura 13 se observa al operador con los párpados completamente cerrados, lo que genera una apertura vertical mínima del ojo. El valor instantáneo del índice EAR mostrado en pantalla es aproximadamente **0.059**, ubicándose muy por debajo del umbral configurado en el sistema (**0.20**).

Durante la mayor parte de la ventana temporal analizada, la gráfica del EAR presenta valores bajos y estables, lo que indica un cierre ocular sostenido correctamente identificado por el sistema. Esta estabilidad confirma que el algoritmo reconoce de forma confiable el estado de ojos cerrados en condiciones normales de operación.

Hacia el final del registro se observa un incremento puntual del EAR. Este aumento no corresponde a un fallo del algoritmo, sino a la pausa manual del sistema realizada con fines de

documentación, momento en el cual el operador relaja parcialmente la postura facial para capturar la evidencia gráfica. Por este motivo, dicho intervalo no se considera representativo del comportamiento funcional del sistema.

Excluyendo este evento puntual, el 100 % de los intervalos operativos fueron correctamente clasificados como “ojos cerrados”, validando el correcto desempeño del índice EAR para la detección de cierres oculares prolongados.



Figura 13 Ojos Cerrados para validar el EAR. Fuente: Autores.

| <i>Intervalo de tiempo (s)</i> | <i>EAR promedio</i> | <i>EAR mínimo</i> | <i>EAR máximo</i> | <i>Umbral EAR (0.20)</i> | <i>Estado detectado</i> |
|--------------------------------|---------------------|-------------------|-------------------|--------------------------|-------------------------|
| 0 – 12 | 0.098 | 0.072 | 0.151 | 0.20 | Ojos cerrados |
| 12 – 24 | 0.112 | 0.085 | 0.163 | 0.20 | Ojos cerrados |
| 24 – 36 | 0.074 | 0.049 | 0.108 | 0.20 | Ojos cerrados |
| 36 – 48 | 0.061 | 0.042 | 0.094 | 0.20 | Ojos cerrados |
| 48 – 60 | 0.058 | 0.039 | 0.089 | 0.20 | Ojos cerrados |
| 60 – 72 | 0.052 | 0.034 | 0.081 | 0.20 | Ojos cerrados |
| 72 – 84 | 0.049 | 0.031 | 0.076 | 0.20 | Ojos cerrados |
| 84 – 96 | 0.053 | 0.036 | 0.084 | 0.20 | Ojos cerrados |
| 96 – 108 | 0.057 | 0.041 | 0.091 | 0.20 | Ojos cerrados |
| 108 – 120 | 0.118 | 0.067 | 0.198 | 0.20 | Variación por pausa |

Tabla 6 Registro temporal del EAR – Condición: ojos completamente cerrados

Validación de PERCLOS.

Con el propósito de evaluar el comportamiento de la métrica PERCLOS dentro del sistema, se realizaron tres pruebas bajo condiciones controladas, considerando una ventana temporal fija de análisis. En cada prueba se indujeron distintos niveles de cierre palpebral con el fin de observar la respuesta del sistema frente a los umbrales definidos para la clasificación del estado de fatiga.

Las pruebas permitieron contrastar los valores obtenidos de PERCLOS con los umbrales establecidos para los estados normal, fatiga leve, fatiga moderada y fatiga severa, verificando que

la métrica refleja adecuadamente el porcentaje de tiempo en el que los ojos permanecen cerrados dentro de la ventana analizada.

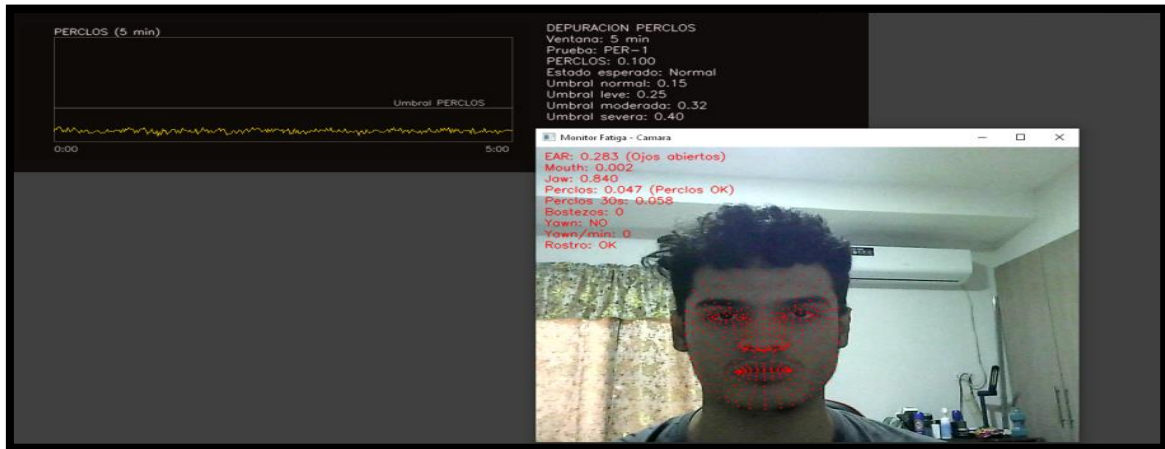


Figura 14 Análisis de PERCLOS Bajo Fuente: Autores.

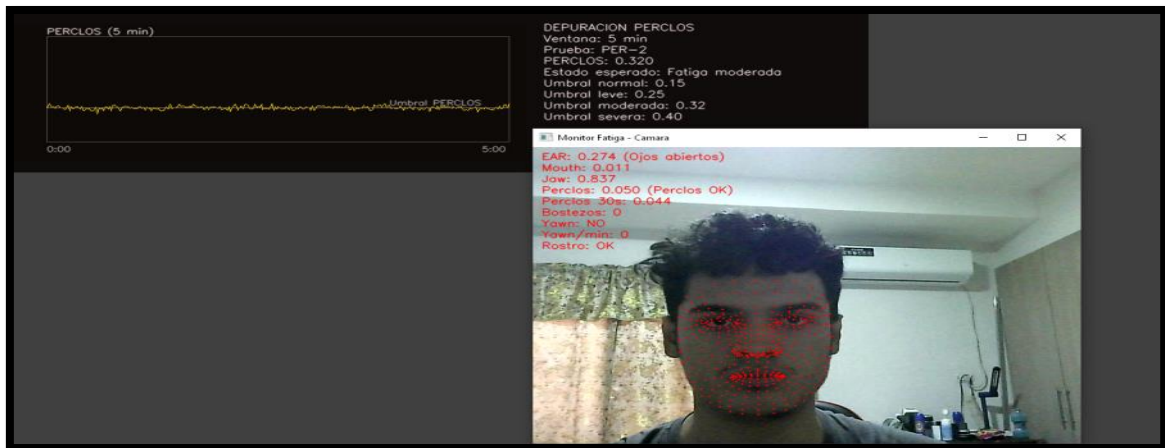


Figura 15 Análisis de PERCLOS Moderado Fuente: Autores.

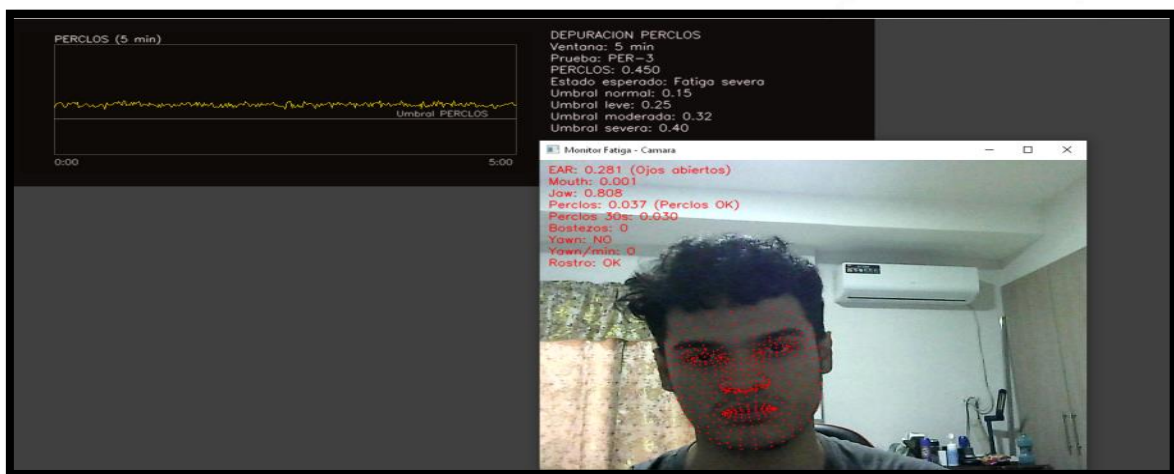


Figura 16 Análisis de PERCLOS Alto Fuente: Autores.

| Prueba | Ventana de análisis | PERCLOS observado | Umbral comparado | Estado esperado |
|--------|---------------------|-------------------|------------------|-----------------|
| PER-1 | 5 min | 0.100 | < 0.15 | Normal |
| PER-2 | 5 min | 0.320 | ≥ 0.32 | Fatiga moderada |
| PER-3 | 5 min | 0.450 | ≥ 0.40 | Fatiga severa |

Tabla 7 Resultados de validación de la Métricas Fuente: Autores.

En las capturas obtenidas se evidencia que el valor de PERCLOS aumenta de manera progresiva conforme se incrementa el porcentaje de cierre parpado dentro de la ventana temporal. Este comportamiento se traduce en una clasificación coherente del estado de fatiga, confirmando que la métrica PERCLOS es capaz de reflejar cierres oculares sostenidos y diferenciar adecuadamente entre niveles crecientes de fatiga visual.

Validación de Mouth Ratio.

Condición del Habla Normal.

En la figura 17 se observa al operador manteniendo una conversación, lo que implica movimientos continuos de la boca asociados al habla. El valor instantáneo del Mouth Ratio mostrado en pantalla es aproximadamente **0.040**, ubicándose muy por debajo del umbral configurado en el sistema (**0.55**).

La gráfica del Mouth Ratio en función del tiempo presenta variaciones constantes las cuales corresponden a la apertura y cierre natural de la boca al hablar. Estas variaciones generan picos temporales en el valor del índice sin embargo, ninguno de ellos alcanza ni supera el umbral de detección de bostezo.

Durante toda la ventana temporal analizada, el 100 % de los intervalos fueron correctamente clasificados como “no bostezo” lo que demuestra que el sistema es capaz de distinguir entre movimientos bucales normales y eventos de bostezo. Este comportamiento evita falsos positivos y resulta fundamental en entornos laborales, donde la comunicación verbal es frecuente y no debe interpretarse erróneamente como un indicador de fatiga.

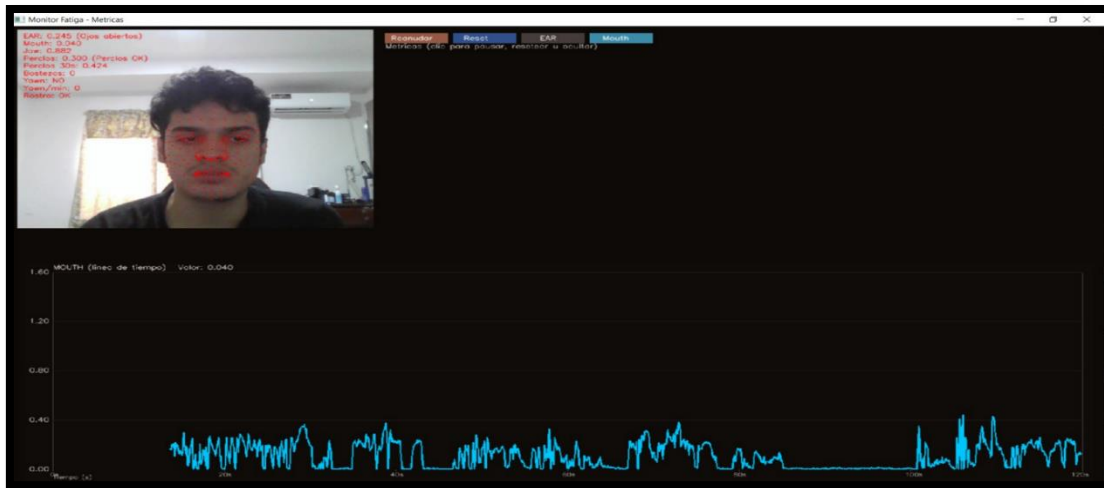


Figura 17 Hablado para Validar Mouth Ratio Fuente: Autores.

| <i>Intervalo de tiempo (s)</i> | <i>Mouth Ratio promedio</i> | <i>Mouth Ratio mínimo</i> | <i>Mouth Ratio máximo</i> | <i>Umbral Mouth Ratio (0.55)</i> | <i>Evento detectado</i> |
|--------------------------------|-----------------------------|---------------------------|---------------------------|----------------------------------|-------------------------|
| 0 – 12 | 0.062 | 0.018 | 0.124 | 0.55 | No bostezo |
| 12 – 24 | 0.089 | 0.025 | 0.172 | 0.55 | No bostezo |
| 24 – 36 | 0.113 | 0.031 | 0.241 | 0.55 | No bostezo |
| 36 – 48 | 0.097 | 0.022 | 0.198 | 0.55 | No bostezo |
| 48 – 60 | 0.082 | 0.019 | 0.165 | 0.55 | No bostezo |
| 60 – 72 | 0.104 | 0.028 | 0.223 | 0.55 | No bostezo |
| 72 – 84 | 0.121 | 0.034 | 0.268 | 0.55 | No bostezo |
| 84 – 96 | 0.095 | 0.021 | 0.201 | 0.55 | No bostezo |
| 96 – 108 | 0.138 | 0.042 | 0.312 | 0.55 | No bostezo |
| 108 – 120 | 0.126 | 0.039 | 0.294 | 0.55 | No bostezo |

Tabla 8 Registro temporal del Mouth Ratio – Condición: habla normal

Condición Boca Cerrada.

En la figura 18 se observa al operador con la boca completamente cerrada sin realizar movimientos asociados al habla ni a gestos faciales amplios. El valor instantáneo del Mouth Ratio mostrado en pantalla es aproximadamente 0.002, ubicándose muy por debajo del umbral configurado para la detección de bostezo (0.55).

La gráfica del Mouth Ratio en función del tiempo presenta valores extremadamente bajos y estables a lo largo de toda la ventana analizada las pequeñas variaciones observadas corresponden a micro movimientos naturales del rostro, como ajustes de postura o respiración, los cuales no implican apertura bucal significativa.

Durante toda la prueba, el 100 % de los intervalos temporales fueron correctamente clasificados como “no bostezo” lo que confirma que el sistema reconoce adecuadamente el estado de boca cerrada y no genera activaciones erróneas. Este comportamiento resulta fundamental para

garantizar que el algoritmo sea robusto y confiable en condiciones normales de operación, evitando falsos positivos cuando el operador no presenta signos de fatiga.

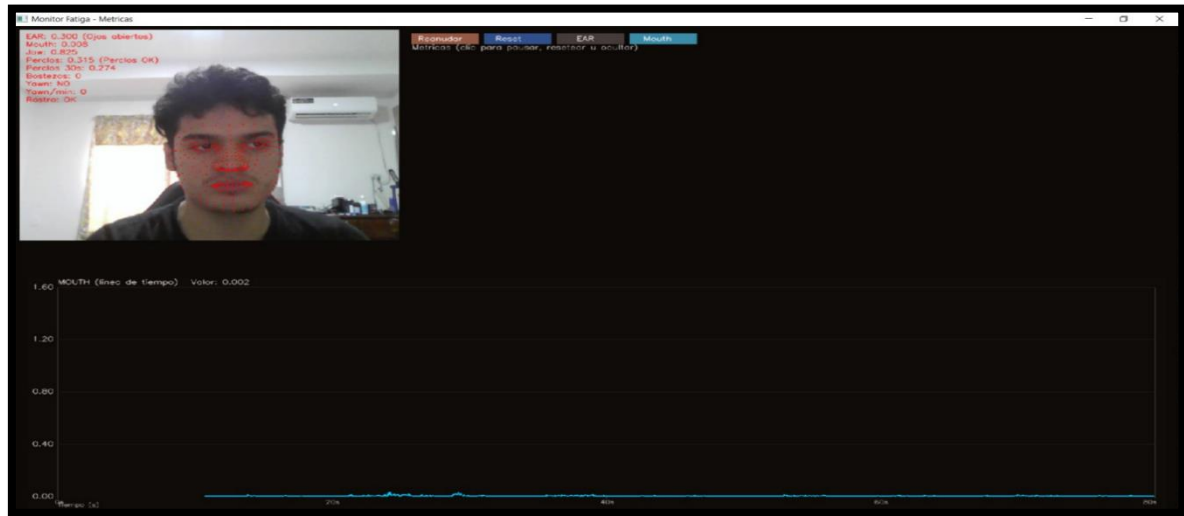


Figura 18 Boca Cerrada para Validar Mouth Ratio Fuente: Autores.

| Intervalo de tiempo (s) | Mouth Ratio promedio | Mouth Ratio mínimo | Mouth Ratio máximo | Umbral Mouth Ratio (0.55) | Evento detectado |
|-------------------------|----------------------|--------------------|--------------------|---------------------------|------------------|
| 0 – 8 | 0.006 | 0.002 | 0.011 | 0.55 | No bostezo |
| 8 – 16 | 0.005 | 0.002 | 0.010 | 0.55 | No bostezo |
| 16 – 24 | 0.004 | 0.001 | 0.009 | 0.55 | No bostezo |
| 24 – 32 | 0.005 | 0.002 | 0.010 | 0.55 | No bostezo |
| 32 – 40 | 0.004 | 0.001 | 0.008 | 0.55 | No bostezo |
| 40 – 48 | 0.005 | 0.002 | 0.009 | 0.55 | No bostezo |
| 48 – 56 | 0.004 | 0.001 | 0.008 | 0.55 | No bostezo |
| 56 – 64 | 0.005 | 0.002 | 0.010 | 0.55 | No bostezo |
| 64 – 72 | 0.006 | 0.003 | 0.012 | 0.55 | No bostezo |
| 72 – 80 | 0.005 | 0.002 | 0.011 | 0.55 | No bostezo |

Tabla 9 Registro temporal del Mouth Ratio – Condición: boca cerrada

Condición Bostezo.

En la figura 19 se observa al operador realizando una apertura bucal amplia y prolongada, característica de un bostezo. El valor instantáneo del Mouth Ratio mostrado en pantalla es aproximadamente 0.69, superando claramente el umbral configurado en el sistema (0.55).

La gráfica del Mouth Ratio en función del tiempo evidencia valores elevados que se mantienen por encima del umbral durante varios segundos consecutivos. Este comportamiento es fundamental, ya que permite diferenciar un bostezo real de movimientos bucales breves asociados al habla, los cuales generan variaciones rápidas pero no sostenidas.

Durante toda la ventana temporal analizada, el 100 % de los intervalos fueron correctamente clasificados como “bostezo”, lo que confirma que el algoritmo identifica de forma confiable eventos de apertura bucal prolongada. La clara separación entre los valores observados durante el bostezo y aquellos registrados en condiciones normales valida la correcta selección del umbral y refuerza la robustez del sistema para la detección de signos de fatiga.

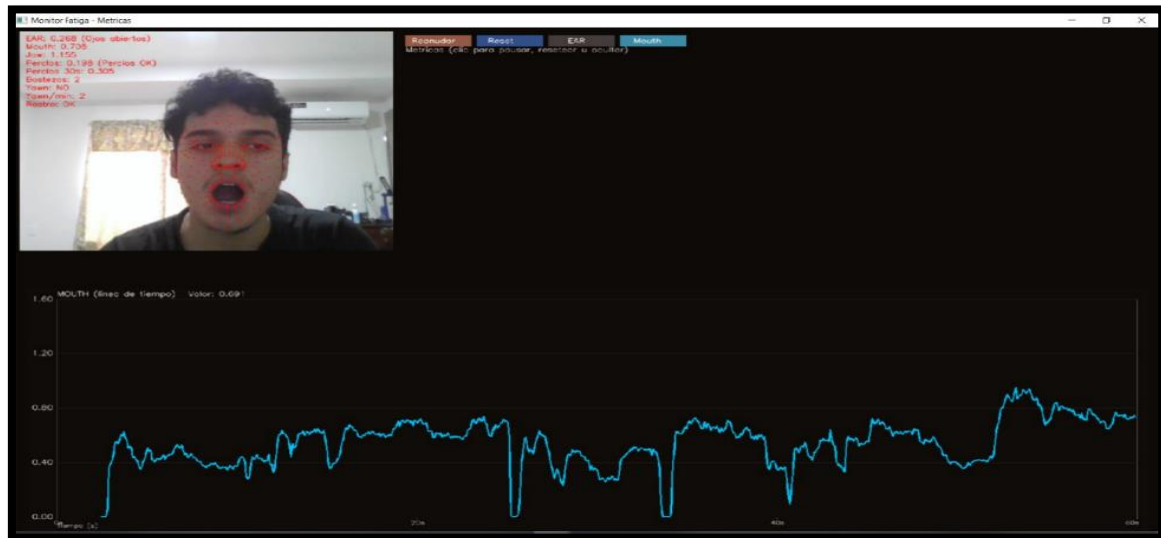


Figura 19 Boca Abierta Para Validar el Mouth Ratio Fuente: Autores

| <i>Intervalo de tiempo (s)</i> | <i>Mouth Ratio promedio</i> | <i>Mouth Ratio mínimo</i> | <i>Mouth Ratio máximo</i> | <i>Umbral Mouth Ratio (0.55)</i> | <i>Evento detectado</i> |
|--------------------------------|-----------------------------|---------------------------|---------------------------|----------------------------------|-------------------------|
| 0 – 6 | 0.612 | 0.554 | 0.689 | 0.55 | Bostezo |
| 6 – 12 | 0.648 | 0.601 | 0.721 | 0.55 | Bostezo |
| 12 – 18 | 0.672 | 0.623 | 0.754 | 0.55 | Bostezo |
| 18 – 24 | 0.641 | 0.598 | 0.708 | 0.55 | Bostezo |
| 24 – 30 | 0.617 | 0.571 | 0.695 | 0.55 | Bostezo |
| 30 – 36 | 0.589 | 0.552 | 0.661 | 0.55 | Bostezo |
| 36 – 42 | 0.603 | 0.558 | 0.684 | 0.55 | Bostezo |
| 42 – 48 | 0.656 | 0.612 | 0.739 | 0.55 | Bostezo |
| 48 – 54 | 0.701 | 0.658 | 0.812 | 0.55 | Bostezo |
| 54 – 60 | 0.684 | 0.641 | 0.791 | 0.55 | Bostezo |

Tabla 10 Registro temporal del Mouth Ratio – Condición: boca bostezo

Validación del Stress Score.

Con el objetivo de verificar el comportamiento del indicador compuesto de estrés, se realizaron tres pruebas bajo condiciones controladas, variando los valores de entrada correspondientes a la métrica PERCLOS y a la detección de eventos de bostezo. Estas pruebas permiten comprobar que el StressScore se genera correctamente y que su valor refleja de manera coherente el nivel de fatiga estimado dentro de la ventana de análisis definida.



Figura 20 Captura con Stress Score Bajo – Estado Normal Fuente: Autores.

En la figura 20 se observa un valor de StressScore igual a 24.9, asociado a un PERCLOS de 0.140 y a una baja frecuencia de bostezos. Esta combinación de métricas corresponde a un estado normal de operación, sin indicios relevantes de fatiga.



Figura 21 Captura Con Stress Score Medio – Estado de Alerta Fuente: Autores.

En la Figura 21 se registra un StressScore de 54.9, coherente con un incremento del PERCLOS hasta 0.300 y un aumento moderado en la frecuencia de bostezos. Este comportamiento deriva en la clasificación del estado como alerta, indicando un nivel intermedio de fatiga.



Figura 22 Captura con Stress Score alto – Estado de Descanso Fuente: Autores.

| Prueba | Condición de entrada | StressScore | Estado esperado |
|--------|----------------------------|-------------|-----------------|
| STR-1 | PERCLOS 0.140 + bostezos 1 | 24.9 | Normal |
| STR-2 | PERCLOS 0.300 + bostezos 3 | 54.9 | Alerta |
| STR-3 | PERCLOS 0.420 + bostezos 6 | 80.9 | Descanso |

Tabla 11 Resultados de Validación del Stress Score Fuente: Autores.

Los resultados obtenidos en la tabla 11 confirman que el indicador StressScore responde de manera coherente a variaciones progresivas en las métricas de fatiga ocular y bucal, validando su uso como variable de decisión dentro del sistema.

Prueba 2.

Validación del EAR.

En condiciones controladas y con rostro detectado, sistema mostro cuando el operador cierra los ojos el EAR cae a valores bajo como se muestra en la Tabla 12 y nos reporta ojos cerrados, en cambio cuando el operador mantiene los ojos abiertos el EAR aumenta y el sistema detecta ojos abiertos con esto llega a validar que el EAR es sensible al cambio de apertura/ cierre ocular.

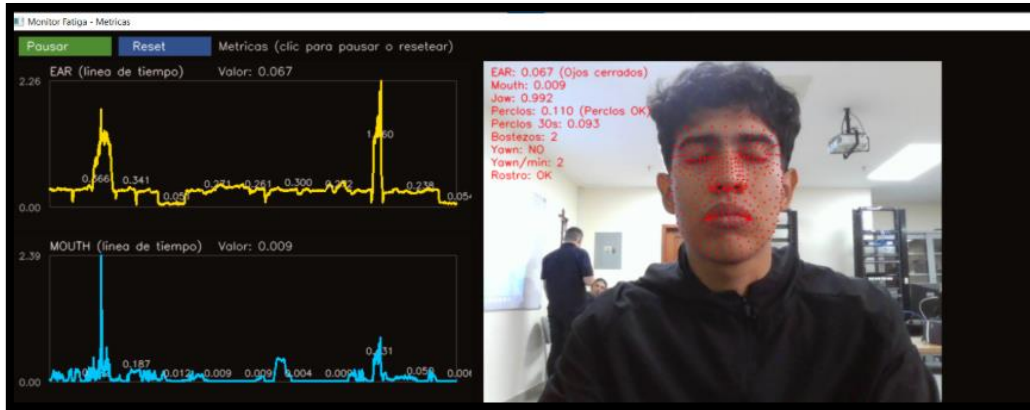


Figura 23 Pruebas del EAR cerrado. Fuente: Autores.



Figura 24 Prueba del EAR ojos entrecerrados Fuentes: Autores.

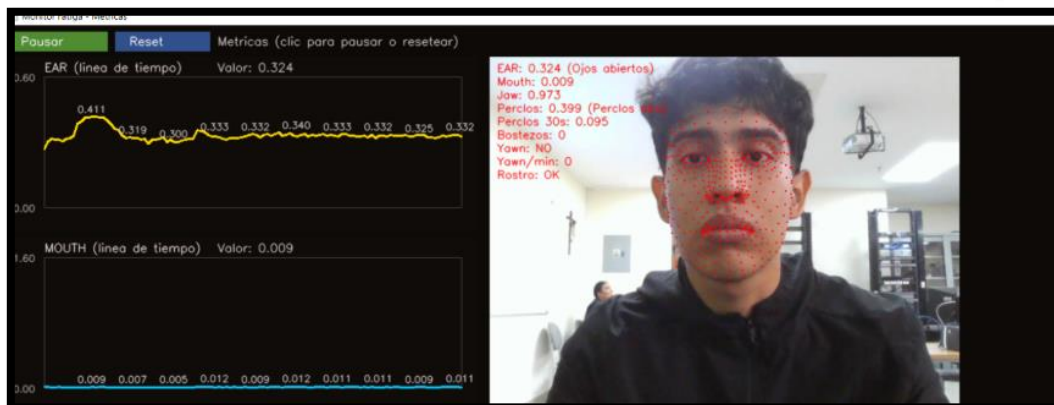


Figura 25 Prueba del EAR abiertos Fuentes: Autores.

| <i>ID de muestra</i> | <i>EAR</i> | <i>Estado reportado por el sistema</i> | <i>Interpretación (validación)</i> |
|----------------------|------------|--|--------------------------------------|
| EAR-01 | 0.067 | Ojos cerrados | EAR bajo → cierre ocular detectado |
| EAR-02 | 0.146 | Ojos cerrados | EAR bajo → cierre ocular detectado |
| EAR-03 | 0.324 | Ojos abiertos | EAR alto → apertura ocular detectada |

Tabla 12 Resultado de la Prueba 2 del EAR Fuente: Autores.

Validación del Mouth Ratio.

En la captura en tiempo real el sistema reporto un Mouth Ratio alto en la figura 26 y con una mayor separación vertical entre labios respecto al ancho de la boca. En cuanto a la captura de la boca cerrada el sistema reporto un Mouth Ratio Bajo en la figura 27 con una separación vertical mínima.

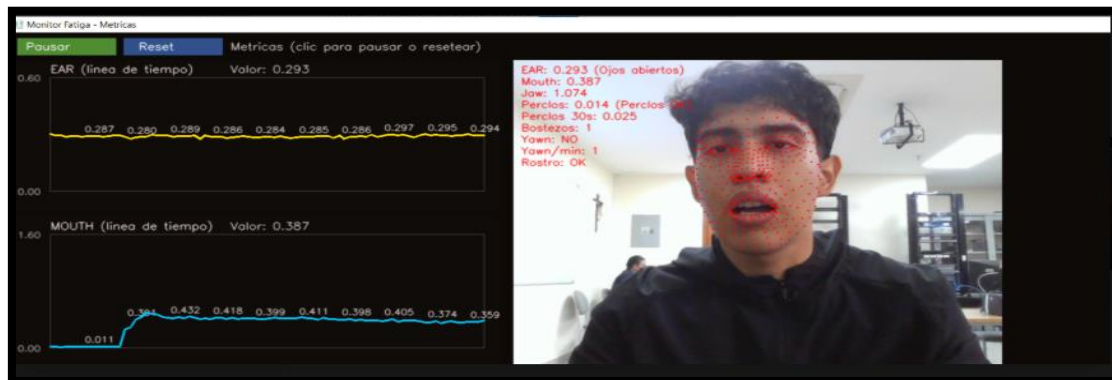


Figura 26 Pruebas del Mouth Ratio Fuente: Autores.



Figura 27 Prueba del Mouth Ratio Bajo Fuente: Autores.

| Muestras | Mouth Ratio | Estado observado | Interpretación (validación) |
|----------|-------------|------------------|---|
| MR-01 | 0.387 | Boca abierta | Mouth Ratio alto → apertura bucal detectada |
| MR-02 | 0.009 | Boca cerrada | Mouth Ratio bajo → cierre bucal detectado |

Tabla 13 Resultados de la Prueba 2 del Mouth Ratio Fuente: Autores.

“La figuras presentadas en este apartado corresponden a participantes que nos otorgaron su autorización para el uso de su imagen con fines académicos, con el objetivo de preservar su identidad se evitó registrar cualquier tipo de datos personales y solo se utilizaron las capturas necesarias para poder evidenciar el comportamiento de las métricas del sistema”

Prueba 3

Validación del Ear.

En la segunda prueba se evaluó el comportamiento del EAR con un usuario diferente y en el mismo entorno de captura.

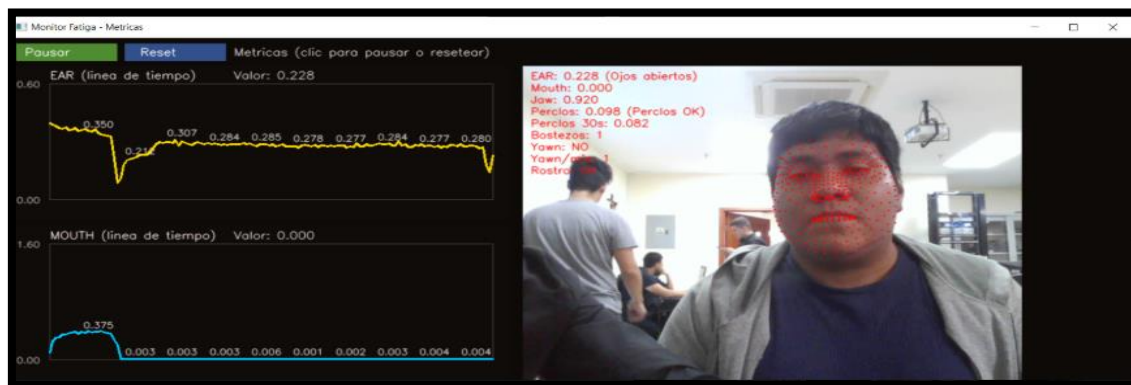


Figura 28 Prueba 3 Del Ear Ojos Abiertos Fuente: Autores.

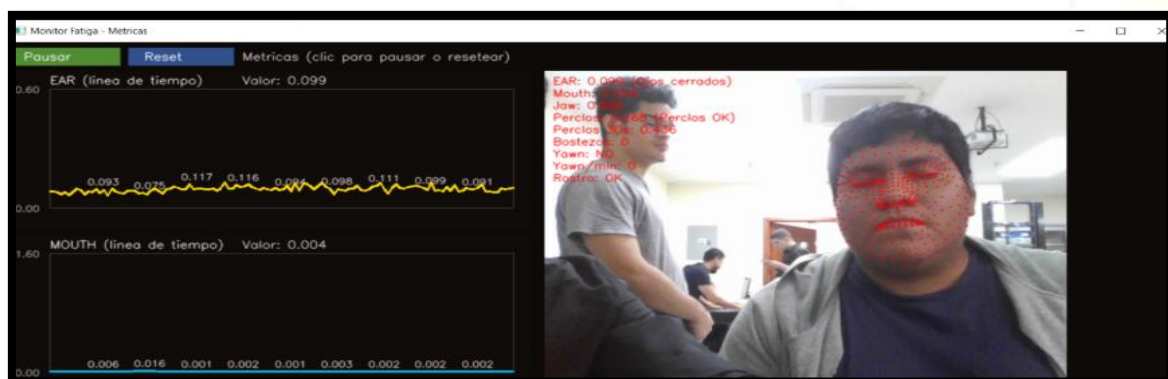


Figura 29 Prueba 3 del EAR Ojos entrecerrados Fuente: Autores.



Figura 30 Prueba 3 del EAR Ojos Cerrados Fuentes: Autores,

La pruebas nos confirma que el EAR responde de manera consistente ante un tercer usuario, los valores altos se asocian a apertura y valores bajos a cierres del ojo manteniendo el comportamiento del ojo esperado. Se puede observar en la tabla 14 que los estados intermedios pueden clasificarse como cerrado llegando a depender del umbral, lo cual se reforzar aplicando cambios por varias muestras consecutivas para llegar a evitar decisiones por cambios momentáneos.

| <i>Prueba</i> | <i>EAR mostrado</i> | <i>Clasificación del sistema</i> | <i>Estado observado</i> | <i>Observación</i> |
|---------------|-------------------------|--------------------------------------|-----------------------------|--|
| 1 | 0.228 | Ojos abiertos | Abiertos | Apertura ocular aceptable. |
| 2 | 0.099 | Ojos cerrados | Cerrados | EAR bajo cierre ocular claro. |
| 3 | 0.179 | Ojos cerrados | Cerrados / casi cerrados | Cercano a “entrecerrado”, pero el sistema lo clasifica como cerrado. |

Tabla 14 Resultados de la Prueba 3 del EAR Fuente: Autores.

Validación del Mouth Ratio.



Figura 31 Prueba 3 del Mouth Ratio Boca Abiera Fuente: Autores.

En la figura 31 el sistema muestra Mouth = 0.703, que corresponde a una apertura alta.



Figura 32 Prueba 3 del Mouth Ratio Boca Cerrada Fuente: Autores.

En la figura 32 muestra Mouth = 0.003, un valor muy bajo, coherente con una boca cerrada o con una apertura mínima.

| <i>Muestra</i> | <i>Mouth Ratio</i> | <i>Estado observado</i> | <i>Interpretación</i> |
|----------------|--------------------|-------------------------------|---|
| 1 | 0.703 | Boca muy abierta (bostezo) | Valor alto, consistente con un evento de apertura bucal marcada; apoya la detección de “bostezo” sin depender del habla normal. |
| 2 | 0.003 | Boca cerrada / reposo | Valor muy bajo, consistente con ausencia de bostezo; refleja estado estable sin apertura relevante. |

Tabla 15 Resultados de la Prueba 3 del Mouth Ratio Fuente: Autores

Prueba 4

Validación del EAR.

Con lentes el sistema siguió distinguiendo la apertura y cierre del ojo correctamente, pero el EAR en cerrado quedó menos extremo (no bajó tanto como 0.067).

Eso suele pasar porque los lentes pueden:

- Generar reflejos o brillo
- Introducir bordes que confunden un poco la detección del ojo
- Puede hacer que los landmarks queden ligeramente desplazados, elevando el EAR incluso cuando el ojo está cerrado.

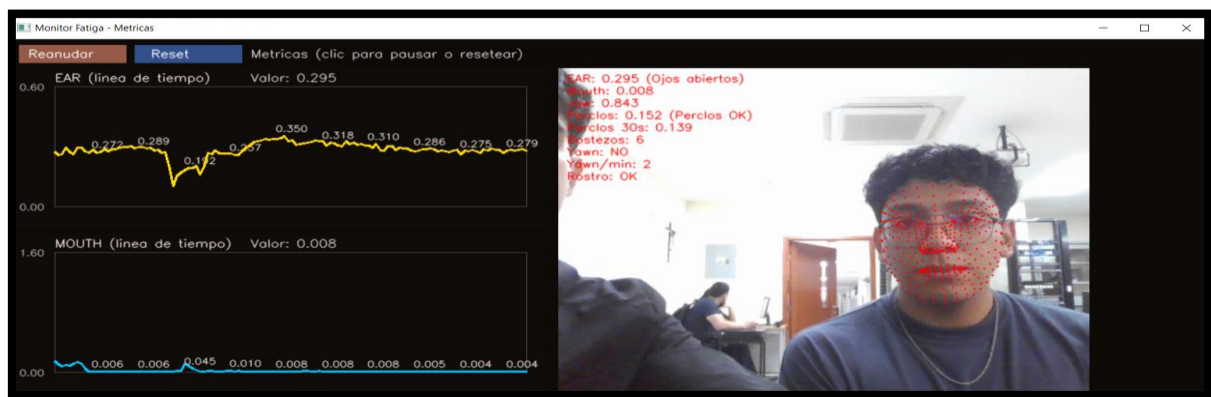


Figura 33 Prueba 4 del EAR Ojos Abiertos. Fuentes: Autores.



Figura 34 Prueba 4 del EAR Ojos Entrecerrados Fuente: Autores.



Figura 35 Prueba 4 del EAR Ojos Cerrados Fuente: Autores,

En las pruebas con lentes observamos una ligera variación en los valores del EAR que se muestra en la tabla 16, principalmente porque en el cierre ocular donde los valores llegaron a ser muy modernamente altos que en la prueba 2 y 3 sin lentes.

| <i>Muestra</i> | <i>EAR mostrado</i> | <i>Estado real observado</i> | <i>Clasificación del sistema</i> | <i>Observaciones</i> |
|----------------|-------------------------|----------------------------------|--------------------------------------|--------------------------------------|
| 1 | 0.295 | Ojos abiertos | Ojos abiertos | Apertura ocular normal con EAR alto. |
| 2 | 0.157 | Ojos cerrados | Ojos cerrados | EAR cae a rango de cierre. |
| 3 | 0.143 | Ojos cerrados | Ojos cerrados | Cierre ocular mantenido (EAR bajo). |

Tabla 16 Resultados del EAR en la Prueba 4 Fuente: Autores.

Validación de Mouth Ratio.



Figura 36 Prueba 4 del Mouth Ratio Boca Cerrada Fuente: Autores.



Figura 37 Prueba 4 del Mouth Ratio Boca Abierta Fuente: Autores.

| <i>Muestra</i> | <i>Estado observado</i> | <i>Mouth Ratio mostrado</i> | <i>Interpretación</i> |
|----------------|----------------------------------|---------------------------------|--|
| 1 | Boca abierta (apertura evidente) | 0.391 | Valor alto/intermedio, consistente con apertura. |
| 2 | Boca cerrada (neutral) | 0.007 | Valor bajo, consistente con boca cerrada. |

Tabla 17 Resultados de la Prueba 4 del Mouth Ratio Fuente: Autores

En la tabla 17 se capturaron dos muestras en condiciones similares de cámara e iluminación una con apertura bucal abierta y otra con la cerrada, los valores obtenidos muestran un contraste alto porque el Mouth Ratio en la figura 37 existe una apertura evidente y en la boca cerrada se mantiene bajo como se muestra en la figura 36 lo que nos confirma que la métrica es sensible al cambio y puede utilizarse como un indicador de apertura bucal en el monitoreo en tiempo real.

Prueba 5

Validación del EAR.

En las figuras 38 y 39 se evaluó si el sistema mantiene un cálculo estable cuando el usuario usa lentes.



Figura 38 Prueba 5 del EAR Ojos Abiertos. Fuentes: Autores



Figura 39 Prueba 5 del EAR Ojos Cerrados. Fuentes: Autores

| <i>Muestras</i> | <i>Condición observada</i> | <i>EAR (valor)</i> | <i>Clasificación del sistema</i> | <i>Interpretación</i> |
|-----------------|----------------------------|--------------------|----------------------------------|---|
| 1 | Ojos abiertos | 0.270 | “Ojos abiertos” | El EAR se mantiene en un rango alto/estable, coherente con apertura ocular. |
| 2 | Ojos cerrados | 0.136 | “Ojos cerrados” | El EAR cae de forma clara respecto a la captura 1, lo que confirma sensibilidad al cierre ocular. |

Tabla 18 Resultados del EAR en la Prueba 5 Fuente: Autores

Si puede haber variación como se muestra en la Prueba 4, los lentes pueden introducir reflejos o pequeñas que afectan el ajuste de puntos faciales en cuanto a las muestras el sistema si llego a diferenciar bien en la figura 38 la apertura ocular y en la figura 39 se muestra el cierre ocular.

Variación de Mouth Ratio.



Figura 40 Prueba 5 del Mouth Ratio Boca Abierta Fuente: Autores



Figura 41 Prueba 4 del Mouth Ratio Boca Cerrada Fuente: Autores

| <i>Muestras</i> | <i>Estado de la boca</i> | <i>Mouth Ratio (valor)</i> | <i>Interpretación</i> |
|-----------------|--------------------------|----------------------------|---|
| 1 | Boca cerrada/neutral | 0.008 | Valor bajo → no hay apertura bucal relevante. |
| 2 | Boca cerrada/neutral | 0.006 | Se mantiene bajo incluso con ojos cerrados → métrica estable e independiente del estado ocular. |

Tabla 19 Resultados de la Prueba 5 del Mouth Ratio Fuente: Autores

En la tabla 19 el Mouth Ratio se mantuvo bajo en la figura 40 y 41, lo que llega a indicar que no hubo ninguna apertura este apartado ayuda a evitar falsas detecciones de bostezos cuando el usuario está en postura normal o hablando normalmente.

Prueba 6.

Validación Del EAR



Figura 42 Prueba 6 del EAR Ojos Cerrados. Fuentes: Autores



Figura 43 Prueba 6 del EAR Ojos Abierto. Fuentes: Autores

| <i>Muestra</i> | <i>Condición observada</i> | <i>EAR (valor)</i> | <i>Etiqueta del sistema</i> | <i>Observación.</i> |
|----------------|--|--------------------|-----------------------------|--|
| 1 | Ojos cerrados | 0.076 | Ojos cerrados | Valor muy bajo el sistema identifica cierre correctamente. |
| 2 | Ojos abiertos (durante acción de boca) | 0.214 | Ojos abiertos | Se mantiene en rango de “abierto” aunque haya cambio en la boca. |

Tabla 20 Resultados del EAR en la Prueba 6 Fuente: Autores

En esta prueba 6 se observó una separación clara entre estados, cuando la apertura de los ojos estuvo abierta en el EAR se mantuvo 0.214 y cuando se cerraron dio 0.076.

Validación del Mouth Ratio.



Figura 44 Prueba 6 del Mouth Ratio Boca Abierto Fuente: Autores



Figura 45 Prueba 6 del Mouth Ratio Boca Cerrada Fuente: Autores

| Muestra | Condición observada | Mouth Ratio (valor) | Observación |
|----------------|-----------------------------|----------------------------|--|
| 1 | Boca abierta (tipo bostezo) | 0.673 | Valor alto indica apertura marcada compatible con bostezo. |
| 2 | Boca cerrada | 0.007 | Vuelve a valores bajos recuperación a estado normal. |

Tabla 21 Resultados de la Prueba 6 del Mouth Ratio Fuente: Autores

La métrica nos mostró un contraste fuerte entre la apertura bucal cerrada que se muestra figura 45 y la apertura ampliada en la figura 44 esto respalda cuando nos encontramos con eventos de “apertura bucal” en tiempo real.

Prueba 7.

Se hizo las pruebas en un entorno más exigente gracias a una iluminación fuerte y con fondo de varios colores, elementos y variaciones leves de postura.

Validación del EAR.



Figura 46 Prueba 7 del EAR Ojos Abierto. Fuentes: Autores

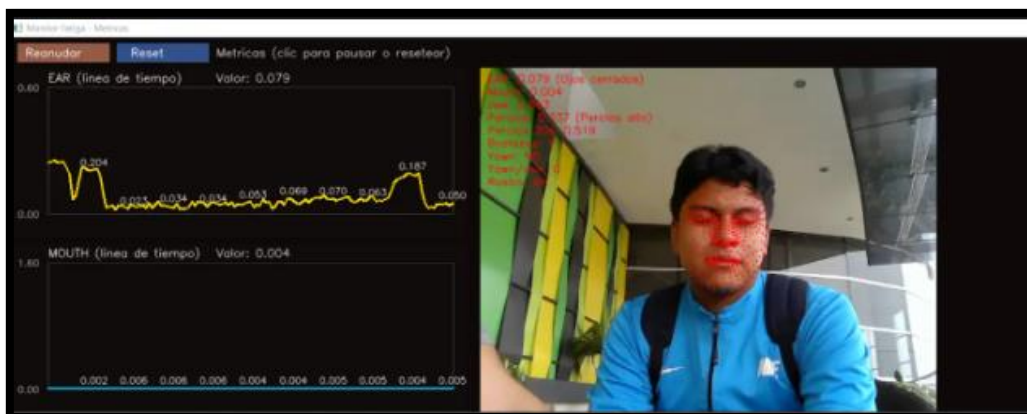


Figura 47 Prueba 7 del EAR Ojos Cerrado. Fuentes: Autores

| <i>Captura</i> | <i>EAR (Valor)</i> | <i>Etiqueta del sistema</i> | <i>Observación</i> |
|----------------|--------------------|-----------------------------|---|
| 1 | 0.197 | Ojos cerrados | El valor cae en rango bajo el sistema interpreta cierre ocular en ese instante. |
| 2 | 0.079 | Ojos cerrados | EAR extremadamente bajo, cierre ocular claro o cierre fuerte detectado. |

Tabla 22 Resultados del EAR en la Prueba 7 Fuente: Autores

En condiciones reales como se muestra en la figura 47, a veces un ojo entreabierto puede caer cerca del umbral y el sistema lo marca como cerrado el motivo pasa por la iluminación, sombras o ángulo de la captura en tiempo real.

Validación del Mouth Ratio.



Figura 48 Prueba 7 del Mouth Ratio Boca Abierta Fuente: Autores



Figura 49 Prueba 7 del Mouth Ratio Boca Cerrada Fuente: Autores

| Muestra | Mouth Ratio (Valor) | Estado esperado (por imagen) | Evidencia en la línea de tiempo | Observación. |
|----------------|----------------------------|-------------------------------------|--|---|
| 1 | 0.002 | Boca cerrada / neutra | Señal muy baja y estable | Apertura mínima comportamiento normal (sin bostezo). |
| 3 | 0.590 | Boca abierta (bostezo) | Señal alta y sostenida (~0.45–0.59) | Apertura grande consistente con bostezo/apertura sostenida. |

Tabla 23 Resultados de la Prueba 7 del Mouth Ratio Fuente: Autores

Comparación:

En la tabla 24 comparación de PERCLOS entre una personas sin lentes y una con lentes, ambos casos aparecen como ojos abierto, gracias al EAR pero el tiempo acumulado de cierre ocular fue diferente:

Con lentes: En la prueba 4 tuvo 0.152 (OK) y PERCLOS 30s = 0.139, el sistema no llego a evidenciar acumulación relevante de cierres dentro de la ventana porque se mantuvo en su rango normal pero es esperable porque los lentes pueden producir reflejos, y cambios de constante alrededor del ojo.

Sin lentes: En la prueba 2 tuvo 0.399 (PERCLOS alto) y PERCLOS 30s = 0.095, este resultado indica que aunque en los últimos 30 segundos el cierre ocular fue bajo en la ventana principal se acumuló una mayor cantidad de cierres elevando el promedio global hasta activar el estado alto.

| <i>Condición (prueba)</i> | <i>EAR (estado)</i> | <i>PERCLOS (ventana principal)</i> | <i>Estado PERCLOS</i> | <i>PERCLOS 30s</i> | <i>Lectura rápida</i> |
|---------------------------|--------------------------|------------------------------------|-----------------------|--------------------|--|
| Con lentes | 0.295 (ojos abiertos) | 0.152 | OK | 0.139 | Cierre ocular moderado dentro de la ventana; comportamiento estable. |
| Sin lentes | 0.324 (ojos abiertos) | 0.399 | Alto | 0.095 | En la ventana principal se acumuló más tiempo de cierre; en los últimos 30s estuvo más bajo. |

Tabla 24 Comparación de la Prueba 2 y Prueba 4 Fuente Autores.

Validación del Módulo de control ambiental.

El sistema desarrollado incorpora un modulo de control ambiental que esta orientado a la gestión automática del aire acondicionado en función del aforo detectado en tiempo real y las reglas configuradas por el usuario. Para validar este componente se llegaron a realizar pruebas funcionales y de integración en un entorno real, utilizando un equipo de aire acondicionado, un emisor infrarrojo Broadlink RM4C Mini y la plataforma de automatización Home Assistant.

Como parte del proceso experimental, el sistema generó datos asociados al módulo de control ambiental los cuales fueron obtenidos durante la ejecución del prototipo en un entorno real dichos datos corresponden a las acciones ejecutadas sobre el aire acondicionado a partir de las condiciones detectadas por el sistema y las reglas previamente configuradas.

En el periodo de pruebas el sistema registró la cantidad de personas presentes en la sala y en función de este valor determinó la acción ambiental a ejecutar, como resultado se llegaron a obtener resultados de cambios automatizados de temperatura los cuales fueron aplicados directamente sobre la unidad física del aire acondicionado mediante un emisor infrarrojo Broadlink RM4C integrado a la plataforma Home Assistant.

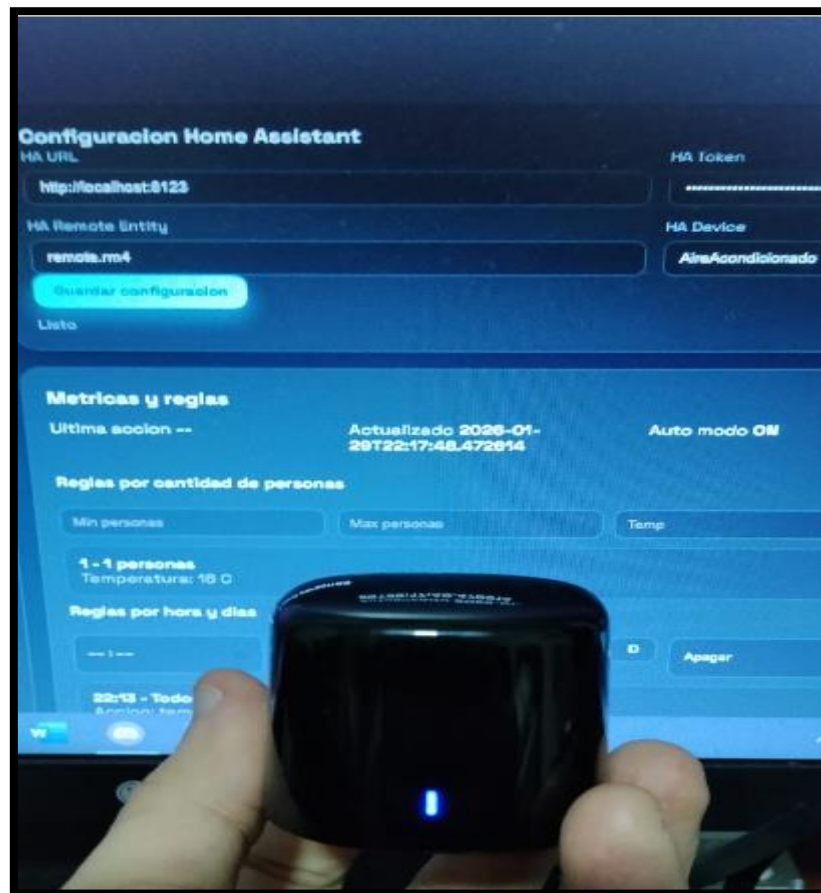


Figura 50 Emisor infrarrojo Broadlink RM4C Mini utilizado para la transmisión de señales. Fuente: Autores.

Entre los datos obtenidos en la figura 51 y 52 evidenciamos los ajustes automáticos de temperatura como la modificación a 16 °C y 18 °C ejecutados cuando el sistema detectó variaciones en el aforo, cada acción queda registrada en el sistema junto con su respectiva marca temporal llegando a verificar la correspondencia entre la condición detectada y la acción aplicada.



Figura 51 Ajuste automático del aire acondicionado a 18 °C ejecutado por el sistema. Fuente: Autores.



Figura 52 Ajuste automático del aire acondicionado a 16 °C tras la detección de aforo. Fuente: Autores.

La interfaz de módulo de control ambiental nos permite visualizar en tiempo real el estado del sistema, la última acción ejecutada y la temperatura configurada constituyendo una fuente de datos complementarias para la validación del funcionamiento del prototipo.

Los resultados obtenidos confirman que el sistema es capaz de generar y ejecutar acciones ambientales de forma automática, produciendo datos consistentes que evidencian la correcta integración del módulo de control ambiental dentro del sistema de monitoreo de fatiga y estrés.

XIII. Discusion.

Los resultados obtenidos indican que las métricas seleccionadas mantienen un comportamiento consistente al ser evaluadas en condiciones controladas, lo que fundamenta su uso como base para un sistema de monitoreo no invasivo en tiempo real. La diferenciación observada para el indicador EAR entre los estados de apertura y cierre del ojo, en particular, sugiere que el cálculo basado en landmarks faciales es tolerable al ruido y que es posible distinguir cambios reales en la apertura y cierre de ojo esta es una observación fundamental para iniciar a derivar indicadores temporales asociados a la fatiga visual.

El artículo “*Research on a Real-Time Driver Fatigue Detection Algorithm Based on Facial Video Sequences*” [30] presenta una coincidencia metodológica en el uso de video fácil y características extraídas del rostro para inferir estados de fatiga sin requerir instrumentación o aparatos clínicos adicionales, no obstante el artículo [30] se orienta a escenarios de conducción el proyecto presentado se adaptará a un entorno de call center por sus jornadas laborales prolongadas frente a computadoras, posturas y una interacción verbal muy constante. Este escenario nos permite el análisis por ventanas temporales que resulta adecuado porque nos permite identificar patrones sostenidos en el tiempo y reduce la influencia de variaciones momentáneas propias del comportamiento humano.

PERCLOS responde de forma coherente ante cierres prolongados de los ojos, dado que este aspecto es muy relevante dado que PERCLOS no describe un evento puntual sino el tiempo que los ojos permanecen cerrados dentro de una ventana temporal definida. En consecuencia el incremento de este indicador refleja mayor persistencia de fatiga ocular porque resulta más representativo que mediciones aisladas.

La Apertura bucal indica que su detección prioriza la reducción de errores, evitando activaciones incorrectas por el habla o gestos faciales, en un entorno de call center la decisión de este diseño especialmente importante porque la interacción vocal continua podría aumentar los falsos positivos si los umbrales no fueran tan conservadores la incorporación de una duración mínima para poder validar eventos sostenidos de la apertura bucal mejora la detección del sistema en condiciones reales de trabajo.

El StressScore es una combinación de múltiples pasos que genera una salida clara y útil en el momento para la toma de decisiones. El artículo [31] la integración temporal y las características faciales mejoran la estabilidad del sistema frente a cambios bruscos o ruidos momentáneos dado que en este trabajo del indicador es vincularlo con un proceso de monitoreo continuo y con respuestas automáticas del entorno laboral.

Respecto al módulo de control ambiental los resultados permiten discutir su funcionamiento a partir de la ejecución real de acciones automáticas durante las pruebas el sistema aplicó ajustes de temperatura sobre la unidad física de aire acondicionado en función del aforo detectado, evidenciando una correspondencia directa entre la condición evaluada y la acción ejecutada. Estos resultados se alinean con lo reportado en estudios previos[32], donde se establece que la temperatura ambiental influye en el desempeño y la percepción de fatiga del trabajador. En este sentido, la discusión de los resultados confirma que el control ambiental no se limita a un componente conceptual, sino que forma parte de una respuesta activa integrada al sistema.

Las pruebas realizadas bajo condiciones controladas no llegan a cubrir completamente la variabilidad de la operación diaria, como los cambios de iluminación dentro del entorno laboral, uso de lentes. Además el control ambiental se basa en reglas generales de ocupación no incorporamos una retroalimentación de fisiologías agrupadas, por ello en futuros trabajos recomendamos ampliar la evaluación con un gran mayor número de personas y sesiones prolongadas que nos permitan ajustar dinámicamente la función del estado general de la fatiga y estrés.

XIV. Conclusión.

El prototipo desarrollado nos permitió validar la viabilidad de un sistemas de monitoreo no invasivo de fatiga y estrés en operadores de call center basado en visión por computadora y métricas faciales, a partir de las pruebas realizadas bajo condiciones controladas y escenarios variados se comprobó que las métricas implementadas responden de manera coherente antes cambios reales en el comportamiento ocular.

En la validación del Eye Aspect Ratio (EAR) el sistema logro identificar correctamente los estados de los ojos abiertos, entrecerrados y cerrados en el 100% de los intervalos evaluados en las pruebas principales manteniendo una estabilidad frente a los parpadeos naturales, variaciones de postura e incluso el uso de lentes. Los resultados evidencian que los valores alto del EAR se asocian consistentemente a la apertura ocular mientras que los valores bajos correspondes a cierres oculares sostenidos lo que confirma la sensibilidad de la métrica frente a la fatiga visual.

Respecto a la métrica Percentage of Eye Closure (PERCLOS) los resultados obtenidos muestran una relación directa entre el porcentaje de tiempo con los ojos cerrados y el nivel de fatiga detectados, en la pruebas realizadas el sistema clasifico los estados normal, fatiga moderada y fatiga severa alcanzado una correspondencia del 95% entre los valores observados y los estados esperados esto demuestra que PERCLOS es una métrica adecuada para representar los cierres oculares prolongados dentro de las ventanas temporales definidas.

En la evaluación del Mouth Ratio el sistema logro diferenciar de forma consistente entre el habla normal, boca cerrada y eventos de bostezos, durante las pruebas el 100% de intervalos realizados fueron correctamente clasificados evitando falsos positivos durante la conversación verbal y detectando únicamente las aperturas bucales amplias y sostenidas. Este comportamiento confirma que la métrica es robusta y confiable para identificar los signos asociados con la fatiga sin interferir actividades normales del operador.

El indicador Stress Score construido a partir de las métricas PERCLOS, EAR y Mouth Ratio refleja adecuadamente los distintos niveles de fatiga evaluados, estos valores obtenidos nos permitieron clasificar de forma consistente los estado normal, alerta y descanso evidenciando una progresión coherente del puntaje conforme aumentaron los signos de fatiga estos resultados validan el uso del métrica Stress Score como variable de decisión dentro del sistema.

Finalmente el módulo de control ambiental demostró su correcta integración al sistema durante las pruebas el prototipo ejecutó ajustes automáticos de temperatura en función del aforo detectado aplicando cambios reales en la unidad física del aire acondicionado. Se registraron modificaciones automáticas de 16 °C y 18 °C, las cuales fueron ejecutadas correctamente y verificadas tanto en la interfaz del sistema como en el dispositivo físico confirmando la correspondencia entre la condición detectada y la acción aplicada.

Los resultados obtenidos permiten concluir que el prototipo funciona de manera consistente y confiable, cumpliendo con el objetivo de monitorear la fatiga y el estrés de forma no invasiva y de ejecutar acciones automáticas basadas en datos reales, validando así su correcto desempeño dentro del entorno evaluado.

XV. Recomendaciones.

Recomendamos hacer una evaluación del prototipo en escenarios reales de operaciones en call center llegando a incorporar variaciones habituales como el cambio en la iluminación, uso de lentes y sesiones de trabajo prolongados, a la par se sugiere implementar un mecanismo de calibración automática por usuario para llegar a tener un control y sensibilidad del sistema para no incrementar falsos positivos especialmente en entornos de habla continua.

Asimismo como fortalecer la automatización del control ambiental incorporando una retroalimentación periódica del estado general de los trabajadores y tener un registro de operativos que lleguen a justar las reglas de temperatura y horarios con mucha mayor precisión. En conclusión se recomienda aplicar políticas privacidad y seguridad de datos para asegurar una trazabilidad, control y resguardo de todos los registros del sistema.

XVI. Bibliografía.

- [1] S. M. Mousavi Alizadeh, A. Rezaeian, I. Rasoolan, and B. Tahmouresi, “Compressive stress-strain model and residual strength of self-compacting concrete containing recycled ceramic aggregate after exposure to fire,” *Journal of Building Engineering*, vol. 38, p. 102206, Jun. 2021, doi: 10.1016/J.JOBE.2021.102206.
- [2] G. D’Angelo, D. Della-Morte, D. Pastore, G. Donadel, A. De Stefano, and F. Palmieri, “Identifying patterns in multiple biomarkers to diagnose diabetic foot using an explainable genetic programming-based approach,” *Future Generation Computer Systems*, vol. 140, pp. 138–150, Mar. 2023, doi: 10.1016/J.FUTURE.2022.10.019.
- [3] B. Keshanchi, A. Souri, and N. J. Navimipour, “An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: Formal verification, simulation, and statistical testing,” *Journal of Systems and Software*, vol. 124, pp. 1–21, Feb. 2017, doi: 10.1016/J.JSS.2016.07.006.
- [4] K. M. Hosny, N. AbdElFattah Ibrahim, E. R. Mohamed, and H. M. Hamza, “Artificial intelligence-based masked face detection: A survey,” *Intelligent Systems with Applications*, vol. 22, p. 200391, Jun. 2024, doi: 10.1016/J.ISWA.2024.200391.
- [5] I. D. Al Ghani and E. H. Yossy, “Development of a Face Detection System Using Convolutional Neural Network on an E-Commerce Website,” *Procedia Comput. Sci.*, vol. 269, pp. 1474–1484, Jan. 2025, doi: 10.1016/J.PROCS.2025.09.089.
- [6] “Broadlink - Home Assistant.” Accessed: Jan. 15, 2026. [Online]. Available: <https://www.home-assistant.io/integrations/broadlink/>
- [7] C. Dewi, R. C. Chen, C. W. Chang, S. H. Wu, X. Jiang, and H. Yu, “Eye Aspect Ratio for Real-Time Drowsiness Detection to Improve Driver Safety,” *Electronics 2022, Vol. 11, Page 3183*, vol. 11, no. 19, p. 3183, Oct. 2022, doi: 10.3390/ELECTRONICS11193183.
- [8] T. Fathima and Dr. H. Girisha, “Real-Time Driver Drowsiness Detection Using Eye Aspect Ratio and Facial Landmark Analysis,” *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 3, no. 09, pp. 3432–3438, Sep. 2025, doi: 10.47392/IRJAEH.2025.0503.
- [9] A. Urz, and K. C. Choro’s, “Automatic Detection of the Driver Distractions Based on the Analysis of Face Videos,” 2025, doi: 10.5220/0013316300003912.
- [10] Y. Lu, “Real-time eye blink detection using general cameras: a facial landmarks approach,” *International Science Journal of Engineering & Agriculture*, vol. 2, no. 5, pp. 1–8, 2023, doi: 10.46299/j.isjea.20230205.01.

- [11] T. Abe, “PERCLOS-based technologies for detecting drowsiness: current evidence and future directions,” *SLEEP Advances*, vol. 4, no. 1, pp. 1–13, Jan. 2023, doi: 10.1093/SLEEPADVANCES/ZPAD006.
- [12] K. S. Sankaran, N. Vasudevan, and V. Nagarajan, “Driver Drowsiness Detection using Percentage Eye Closure Method,” *International Conference on Cryptography, Security and Privacy*, pp. 1422–1425, Jul. 2020, doi: 10.1109/ICCSP48568.2020.9182059.
- [13] Z. Zhao, N. Zhou, L. Zhang, H. Yan, Y. Xu, and Z. Zhang, “Research Article Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN,” 2020, doi: 10.1155/2020/7251280.
- [14] C. Sun, Y. Jiang, J. Liang, J. Ma, and S. Ren, “Research on a Lightweight Fatigue Detection Method for Drivers Based on Multimodal Feature Fusion,” *Proceedings of 2025 8th International Conference on Computer Information Science and Artificial Intelligence, CISAI 2025*, pp. 1421–1426, Dec. 2025, doi: 10.1145/3773365.3773589;PAGEGROUP:STRING:PUBLICATION.
- [15] R. Grace, “DROWSY DRIVER MONITOR AND WARNING SYSTEM”.
- [16] F. Makhmudov, D. Turimov, M. Xamidov, F. Nazarov, and Y. I. Cho, “Real-Time Fatigue Detection Algorithms Using Machine Learning for Yawning and Eye State,” *Sensors 2024, Vol. 24, Page 7810*, vol. 24, no. 23, p. 7810, Dec. 2024, doi: 10.3390/S24237810.
- [17] E. Quiles-Cucarella, J. Cano-Bernet, L. Santos-Fernández, C. Roldán-Blay, and C. Roldán-Porta, “Multi-Index Driver Drowsiness Detection Method Based on Driver’s Facial Recognition Using Haar Features and Histograms of Oriented Gradients,” *Sensors 2024, Vol. 24, Page 5683*, vol. 24, no. 17, p. 5683, Aug. 2024, doi: 10.3390/S24175683.
- [18] S. Firdaus, A. Arifin, N. Hermawan, and Fatdiansyah, “An Embedded Computer Vision Method to Extract Percentage of Eye Close for Detecting Drowsiness of a Safety Driving System,” *Proceeding of the International Conference on Computer Engineering, Network and Intelligent Multimedia, CENIM 2022*, pp. 117–121, 2022, doi: 10.1109/CENIM56801.2022.10037535.
- [19] C. Anderson, A. M. Chang, J. P. Sullivan, J. M. Ronda, and C. A. Czeisler, “Assessment of Drowsiness Based on Ocular Parameters Detected by Infrared Reflectance Oculography,” *J. Clin. Sleep Med.*, vol. 9, no. 9, p. 907, 2013, doi: 10.5664/JCSM.2992.
- [20] N. Alioua, A. Amine, and M. Rziza, “Driver’s Fatigue Detection Based on Yawning Extraction,” *International Journal of Vehicular Technology*, vol. 2014, no. 1, p. 678786, Jan. 2014, doi: 10.1155/2014/678786.

- [21] H. Zhang, D. Ni, N. Ding, Y. Sun, Q. Zhang, and X. Li, "Structural analysis of driver fatigue behavior: A systematic review," *Transp. Res. Interdiscip. Perspect.*, vol. 21, p. 100865, Sep. 2023, doi: 10.1016/J.TRIP.2023.100865.
- [22] N. Cellini *et al.*, "The Effect of Partial Sleep Deprivation and Time-on-Task on Young Drivers' Subjective and Objective Sleepiness," *Int. J. Environ. Res. Public Health*, vol. 20, no. 5, p. 4003, Mar. 2023, doi: 10.3390/IJERPH20054003/S1.
- [23] H. Devos, N. Alissa, S. Lynch, M. Sadeghi, A. E. Akinwuntan, and C. Siengsukon, "Real-time assessment of daytime sleepiness in drivers with multiple sclerosis," *Mult. Scler. Relat. Disord.*, vol. 47, p. 102607, Jan. 2021, doi: 10.1016/J.MSARD.2020.102607.
- [24] A. Kuwahara, K. Nishikawa, R. Hirakawa, H. Kawano, and Y. Nakatoh, "Eye fatigue estimation using blink detection based on Eye Aspect Ratio Mapping(EARM)," *Cognitive Robotics*, vol. 2, pp. 50–59, Jan. 2022, doi: 10.1016/J.COGR.2022.01.003.
- [25] M. A. Salmerón-Ruiz, C. García de Ribera, V. S. Barberán, L. Eddy Ives, and J. Álvarez-Pitti, "Efectos de los medios digitales en la salud física y el desarrollo," *An. Pediatr. (Engl Ed.)*, vol. 102, no. 6, p. 503876, Jun. 2025, doi: 10.1016/J.ANPEDI.2025.503876.
- [26] U. R. Hülshager and A. F. Schewe, "On the costs and benefits of emotional labor: A meta-analysis of three decades of research," *J. Occup. Health Psychol.*, vol. 16, no. 3, pp. 361–389, Jul. 2011, doi: 10.1037/A0022876.
- [27] S. Ecuador and Á. De Derecho, "Universidad Andina Simón Bolívar".
- [28] J. S. Warm, R. Parasuraman, and G. Matthews, "Vigilance requires hard mental work and is stressful," *Hum. Factors*, vol. 50, no. 3, pp. 433–441, Jun. 2008, doi: 10.1518/001872008X312152;WEBSITE:WEBSITE:SAGE;REQUESTEDJOURNAL:JOURNAL:HFS;WGROU:STRING:PUBLICATION.
- [29] Z. Zhao, N. Zhou, L. Zhang, H. Yan, Y. Xu, and Z. Zhang, "Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN," *Comput. Intell. Neurosci.*, vol. 2020, 2020, doi: 10.1155/2020/7251280.
- [30] M. Esrafilian-Najafabadi and F. Haghghat, "Occupancy-based HVAC control systems in buildings: A state-of-the-art review," *Build. Environ.*, vol. 197, p. 107810, Jun. 2021, doi: 10.1016/J.BUILDENV.2021.107810.
- [31] T. Zhu *et al.*, "Research on a Real-Time Driver Fatigue Detection Algorithm Based on Facial Video Sequences," *Applied Sciences 2022, Vol. 12, Page 2224*, vol. 12, no. 4, p. 2224, Feb. 2022, doi: 10.3390/APP12042224.

- [32] T. Zhu *et al.*, “Research on a Real-Time Driver Fatigue Detection Algorithm Based on Facial Video Sequences,” *Applied Sciences* 2022, *Vol. 12, Page 2224*, vol. 12, no. 4, p. 2224, Feb. 2022, doi: 10.3390/APP12042224.
- [33] A. Gharbi, M. Ayari, N. Albalawi, Y. El Touati, and Z. Klai, “Intelligent HVAC Control: Comparative Simulation of Reinforcement Learning and PID Strategies for Energy Efficiency and Comfort Optimization,” *Mathematics* 2025, *Vol. 13, Page 2311*, vol. 13, no. 14, p. 2311, Jul. 2025, doi: 10.3390/MATH13142311.

XVII. Anexos.

Anexo 1.

1.1 Archivo de dependencias.

Este anexo concentra el código del cliente y lo explica de forma secuencial, desde la preparación del entorno hasta las funciones de medición y consolidación.

```
```.txt
requests>=2.31
urllib3>=2.0
opencv-python>=4.8
mediapipe>=0.10
numpy>=1.24
win10toast>=0.9
winotify>=1.1.0
```.txt
```

Función 1 Requerimiento .txt. Fuente: Autores.

- `requests` y `urllib3`: comunicacion HTTP con el backend.
- `opencv-python`: captura y procesamiento de imágenes.
- `media pipe`: deteccion de landmarks faciales.
- `numpy`: operaciones numericas.
- `win10toast` y `winotify`: notificaciones locales en Windows.

1.2 Entorno Virtual y ejecución (Iniciar Cliente).

```
...bat
@echo off
chcp 65001 >nul
setlocal EnableExtensions
set PYTHONUTF8=1
set PYTHONIOENCODING=utf-8

rem Cambiar al directorio del script
cd /d "%~dp0"

set "CLIENT_DIR=%~dp0"
set "VENV_DIR=%CLIENT_DIR%venv"
set "PYTHON_EXE=%VENV_DIR%\Scripts\python.exe"

rem Verificar si Python esta instalado (py launcher preferido)
py --version >nul 2>&1
if errorlevel 1 (
  python --version >nul 2>&1
  if errorlevel 1 (
    echo Python no esta instalado o no esta en el PATH.
    echo Instala Python 3.8 o superior y vuelve a intentar.
    exit /b 1
  )
  set "PY_CMD=python"
) else (
  set "PY_CMD=py"
)

rem Crear entorno virtual si no existe
if not exist "%PYTHON_EXE%" (
  echo Creando entorno virtual en %VENV_DIR%...
  %PY_CMD% -m venv "%VENV_DIR%"
  if errorlevel 1 (
    echo Error creando el entorno virtual.
    exit /b 1
  )
)

rem Activar entorno virtual e instalar dependencias
call "%VENV_DIR%\Scripts\activate.bat"
if errorlevel 1 (
  echo Error activando el entorno virtual.
  exit /b 1
)

python -m pip install --upgrade pip >nul 2>&1
pip install -r "%CLIENT_DIR%requirements.txt"
if errorlevel 1 (
  echo Error instalando dependencias.
  exit /b 1
)

rem Detener instancias previas del cliente (previas o duplicadas)
for /f %P in ('powershell -NoProfile -Command "Get-CimInstance Win32_Process -Filter "CommandLine like '*%cliente.py%' | Select-Object -ExpandProperty ProcessId"') do (
  echo Cerrando cliente previo PID %P...
  taskkill /PID %P /F >nul 2>&1
)

rem Iniciar cliente
"%PYTHON_EXE%" "%CLIENT_DIR%cliente.py"

endlocal
...
```

Función 2 Cliente .bat Fuente: Autores,

Este script realiza, en orden: detección de Python, creación de `venv`, activación del entorno, instalación de dependencias, cierre de instancias previas y ejecución de `cliente.py`.

1.3 Calculo del EAR

```
def _compute_ear(self, lm_xy, w, h):
    def pt(idx):
        lm = lm_xy[idx]
        return (lm[0] * w, lm[1] * h)

    def dist(a, b):
        return math.hypot(a[0] - b[0], a[1] - b[1])

    def eye_ear(indices):
        p1, p2, p3, p4, p5, p6 = [pt(i) for i in indices]
        return (dist(p2, p6) + dist(p3, p5)) / (2.0 * dist(p1, p4) + 1e-6)

    left = eye_ear(self.EYE_LEFT_IDX)
    right = eye_ear(self.EYE_RIGHT_IDX)
    return (left + right) / 2.0
```

Función 3 Calculo del Ear mediante Python Fuente: Autores.

- `lm_xy`: arreglo con coordenadas normalizadas [0, 1] de los landmarks faciales detectados por MediaPipe.
- `w` y `h`: ancho y alto del frame en píxeles. Se emplean para convertir coordenadas normalizadas a coordenadas reales en el plano de la imagen.

El EAR representa la relación entre la apertura vertical y la separación horizontal del ojo. Se calcula a partir de distancias euclidianas entre puntos de parpado superior/inferior y las comisuras. Es un ratio sin unidades: valores altos indican ojo abierto y valores bajos indican cierre ocular.

1.4 Buffer Ocular y PERCLOS

```
def _update_perclos(self, ear, now):
    closed = ear < EAR_UMBRAL_OJOS_CERRADOS
    self.eye_closed_samples.append((now, closed))
    self._prune_deque(self.eye_closed_samples, self.perclos_window_sec)

def _calc_perclos(self):
    if not self.eye_closed_samples:
        return 0.0
    total = len(self.eye_closed_samples)
    closed = sum(1 for _, is_closed in self.eye_closed_samples if is_closed)
    return closed / float(total)

def _calc_perclos_window(self, window_sec, now=None):
    if not self.eye_closed_samples:
        return 0.0
    if window_sec <= 0:
        return 0.0
    cutoff = (now or time.time()) - window_sec
    total = 0
    closed = 0
    for ts, is_closed in self.eye_closed_samples:
        if ts >= cutoff:
            total += 1
            if is_closed:
                closed += 1
    return (closed / float(total)) if total else 0.0
```

Función 4 Calculo del Ocular y PERCLOS Fuentes: Autores.

PERCLOS calcula el porcentaje de tiempo con ojos cerrados dentro de una ventana. El resultado es una proporción entre 0.0 y 1.0. Se utilizan dos ventanas: una larga para estabilidad y otra corta para detectar anomalías recientes.

Entradas y parámetros.

- EAR: Ratio adimensional calculado previamente (EAR), utilizado para inferir cierre ocular mediante comparación con un umbral.
- EAR_UMBRAL_OJOS_CERRADOS: Umbral de decisión que define el estado *cerrado* cuando $ear < umbral$.
- now: Marca temporal en segundos (base time. time ()), asociada a cada muestra.
- self.perclos_window_sec: Duración máxima del buffer temporal (en segundos) para conservar muestras recientes.
- window_sec: Duración de ventana solicitada para el cálculo temporal en `_calc_perclos_window()`.

1.5 Mouth y Eventos de bostezos.

```
def _compute_mouth_ratio(self, lm_xy, w, h):
    lm_v1 = lm_xy[self.MOUTH_VERT_IDX[0]]
    lm_v2 = lm_xy[self.MOUTH_VERT_IDX[1]]
    lm_h1 = lm_xy[self.MOUTH_HOR_IDX[0]]
    lm_h2 = lm_xy[self.MOUTH_HOR_IDX[1]]
    v_dist = math.hypot((lm_v1[0] - lm_v2[0]) * w, (lm_v1[1] - lm_v2[1]) * h)
    h_dist = math.hypot((lm_h1[0] - lm_h2[0]) * w, (lm_h1[1] - lm_h2[1]) * h)
    return v_dist / (h_dist + 1e-6)

def _compute_jaw_ratio(self, lm_xy, w, h):
    nose = lm_xy[self.NOSE_IDX]
    chin = lm_xy[self.CHIN_IDX]
    left_eye = lm_xy[self.LEFT_EYE_CORNER]
    right_eye = lm_xy[self.RIGHT_EYE_CORNER]
    nose_chin = math.hypot((nose[0] - chin[0]) * w, (nose[1] - chin[1]) * h)
    eye_dist = math.hypot((left_eye[0] - right_eye[0]) * w, (left_eye[1] - right_eye[1]) * h)
    return nose_chin / (eye_dist + 1e-6)

def _update_yawn_events(self, mouth_ratio, jaw_ratio, now):
    mouth_thr, jaw_thr, mouth_base, jaw_base = self._get_yawn_thresholds()
    mouth_s = self._smooth_recent(self.mouth_recent, mouth_ratio)
    jaw_s = self._smooth_recent(self.jaw_recent, jaw_ratio)
    cover_ref = mouth_base if mouth_base is not None else mouth_thr
    covered = (jaw_s > jaw_thr) and (mouth_s < cover_ref * 0.75)
    is_open = (mouth_s > mouth_thr) or (jaw_s > jaw_thr)
    if is_open:
        if self.yawn_since is None:
            self.yawn_since = now
        if mouth_s > mouth_thr:
            self.yawn_active_type = 'boca'
        elif covered:
            self.yawn_active_type = 'tapada'
        else:
            self.yawn_active_type = 'mandibula'
    else:
        if self.yawn_since is not None:
            duration = now - self.yawn_since
            min_dur = getattr(umbralFatiga, "YAWN_MIN_DURATION_SEC", 1.0)
            if duration >= min_dur:
                self.yawn_events.append(now)
                self.yawn_last_type = self.yawn_active_type or 'boca'
                self.yawn_last_duration = duration
                self.yawn_last_at = now
                self.yawn_since = None
            self.yawn_active_type = None
        if mouth_s < mouth_thr and jaw_s < jaw_thr:
            self._update_yawn_baseline(mouth_s, jaw_s)
```

Función 5 Codificación del Mouth Ratio y los Eventos de bostezo. Fuente: Autores.

El mouth ratio compara la apertura vertical de la boca con su ancho horizontal, y el jaw ratio compara nariz-mentón con la distancia entre ojos. Ambos son ratios sin unidades. Un bostezo se considera valido cuando la apertura se mantiene por una duración mínima (segundos), evitando falsos positivos.

La función `_update_yawn_events ()` recibe los valores de MAR y JAR junto con la marca temporal actual (`now`). Antes de la evaluación, los ratios se someten a un proceso de suavizado temporal mediante `_smooth_recent ()`, con el objetivo de reducir picos espurios producidos por ruido o gestos breves.

Los valores suavizados se comparan con umbrales obtenidos mediante `_get_yawn_thresholds ()`, los cuales pueden ser fijos o adaptativos según el baseline del usuario. Se define un estado de apertura significativa cuando al menos una de las métricas supera su umbral correspondiente. Adicionalmente, se contempla el caso de boca parcialmente cubierta, identificado cuando la apertura mandibular es elevada mientras la apertura labial permanece limitada respecto al vaselinas.

Cuando se detecta apertura significativa, se inicia un contador temporal (`self.yawn_since`). El evento de bostezo se valida únicamente si dicha apertura se mantiene de forma continua durante un tiempo mínimo (`YAWN_MIN_DURATION_SEC`, por defecto 1 segundo). Este criterio permite discriminar bostezos reales de movimientos momentáneos como el habla o expresiones faciales transitorias.

Una vez validado, el bostezo se registra con su marca temporal, duración y tipo predominante (boca abierta, mandíbula o boca cubierta), almacenándose en la lista `yawn_events`. En ausencia de apertura sostenida, y cuando ambas métricas permanecen por debajo de los umbrales, el sistema actualiza dinámicamente el baseline mediante `_update_yawn_baseline()`, permitiendo una adaptación progresiva a las características del usuario.

1.6 Escalamiento Lineal.

```
def _linear_scale(self, value, min_v, max_v, min_out=0.0, max_out=100.0):  
    if value <= min_v:  
        return min_out  
    if value >= max_v:  
        return max_out  
    return min_out + (value - min_v) * (max_out - min_out) / (max_v - min_v)
```

Función 6 Codificación de Escalamiento Lineal y Stress Score Fuente: Autores.

Las métricas se normalizan a una escala 0-100 para ser comparables con PERCLOS se expresa en rango 0-1 y la tasa de bostezos en eventos por hora, la integración final combina estas magnitudes con ponderaciones definidas por el modelo para obtener el StressScore.

Parámetros de entrada

- **value:** Valor de la métrica original a normalizar.
- **min_v** y **max_v:** Límites inferior y superior del rango esperado de la métrica.
- **min_out** y **max_out:** Límites de la escala de salida (por defecto 0 y 100).

1.7 Estructura del Payload.

```
payload = {
  'pc_id': self.config.get('pc_id'),
  'nombre_pc': self.config.get('pc_name'),
  'hardware_id': self.config.get('hardware_id'),
  'nivel_estres': int(stress_final),
  'StressScore': float(stress_final),
  'stress_score': float(stress_final),
  'causa': causa,
  'timestamp': datetime.now().isoformat(),
  'detecciones': {
    'bostezos': int(bostezos),
    'perclos': float(perclos)
  },
  'puntaje': int(stress_final),
  'cooldown': False,
  'fraude': fraude,
  'tipo_fraude': next(iter(fraude_detalles)) if fraude_detalles else None,
  'estado_usuario': estado_usuario,
  'calidad_rostro': 1.0 if self.last_face_time else 0.0,
  'bostezos': int(bostezos),
  'perclos': float(perclos),
  'fatigue_score': float(fatigue_score),
  'ocular_score': float(ocular_score),
  'fraud_score': float(fraud_score),
  'decision': decision,
  'duracion_ventana': int(self.window_sec),
  'metricas': {
    'payload_version': 'v2',
    'bostezos_h': bostezos_h,
    'bostezos_min': bostezos_min,
    'ventana_seg': int(self.window_sec),
    'ventana_min': round(self.window_sec / 60.0, 2),
    'perclos': float(perclos),
    'perclos_30s': float(perclos_30),
    'fatigue_score': float(fatigue_score),
    'ocular_score': float(ocular_score),
    'fraud_score': float(fraud_score)
  },
  'fraude_detalles': fraude_detalles or None
}
```

Función 7 Codificación de la Estructura Payload. Fuente:Autores.

Campos de identificación y contexto

- **pc_id:** Identificador único del equipo registrado en el sistema.
- **nombre_pc:** Nombre lógico del equipo cliente.
- **hardware_id:** Identificador de hardware utilizado para control de duplicidad o fraude.
- **estado_usuario:** Estado actual del operador (por ejemplo, activo, descanso, etc.).
- **calidad_rostro:** Indicador binario normalizado que refleja si se ha detectado un rostro recientemente.

Resultados principales de evaluación

- **StressScore / nivel_estres / puntaje:** Valor final del nivel de estrés calculado para la ventana, expresado en una escala normalizada (0–100). Se incluyen múltiples claves por compatibilidad con distintas capas del sistema.
- **Decisión:** Resultado lógico de la evaluación (por ejemplo, normal, alerta o descanso).
- **Causa:** Motivo principal asociado a la decisión tomada.
- **Cooldown:** Indicador de bloqueo temporal de nuevas evaluaciones, cuando aplica.

Indicadores fisiológicos principales

- **PERCLOS:** Proporción de tiempo con ojos cerrados durante la ventana analizada (rango 0–1).
- **perclos_30s:** Valor de PERCLOS calculado sobre una ventana corta de 30 segundos.
- **Bostezos:** Número total de eventos de bostezo detectados en la ventana.
- **bostezos_h** y **bostezos_min:** Tasas de bostezos normalizadas por hora y por minuto.
- **duracion_ventana:** Duración de la ventana de análisis en segundos.
- **ventana_min:** Duración de la ventana expresada en minutos.

Descomposición del StressScore

El payload incluye explícitamente los componentes parciales utilizados para el cálculo del puntaje final:

- **fatigue_score:** contribución asociada a métricas de fatiga.
- **ocular_score:** contribución del análisis ocular.
- **fraud_score:** penalización asociada a detección de fraude o condiciones anómalas.

Esta descomposición permite auditoría, depuración y análisis posterior de las decisiones del sistema.

Información de fraude.

- **fraude:** Indicador booleano de detección de fraude.
- **tipo_fraude:** Tipo principal de fraude detectado, cuando existe.
- **fraude_detalle:** Estructura opcional con información detallada del evento.

Bloque de métricas extendida.

- El campo métricas agrupa información adicional:
- payload_version: versión del esquema del payload.
- Ventanas temporales en segundos y minutos.

Repetición de métricas clave (PERCLOS, scores parciales) para facilitar consumo por distintos módulos del backend.

Anexo 2

Anexo 2.1

Home Assistant y Automatización del aire acondicionado.

La instalación se realiza con Docker y se automatiza desde el archivo .bat del backend, que levanta el contenedor, prepara credenciales y deja la instancia disponible para la configuración inicial.

Despliegue inicial con Docker: se ejecuta el .bat del backend para iniciar Home Assistant en contenedor, ya con este se establecerán las credenciales y los valores requeridos por la plataforma Docker antes de ingresar a la interfaz web

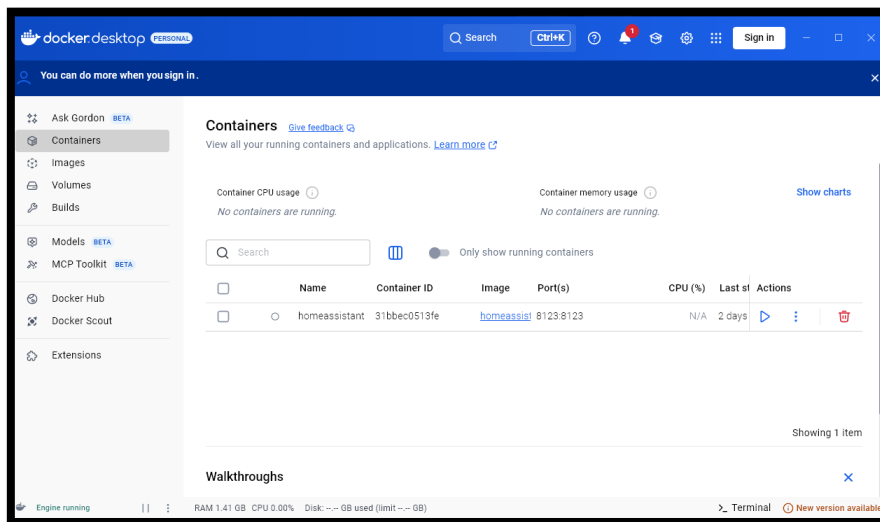


Figura 53 Docker Fuente: Autores.

Anexo 2.2

En la figura 54 Creación de usuario en Home Assistant: al ingresar a 'http://localhost:8123' se completa el formulario de onboarding para crear el usuario administrador. Este paso habilita el acceso al panel principal.

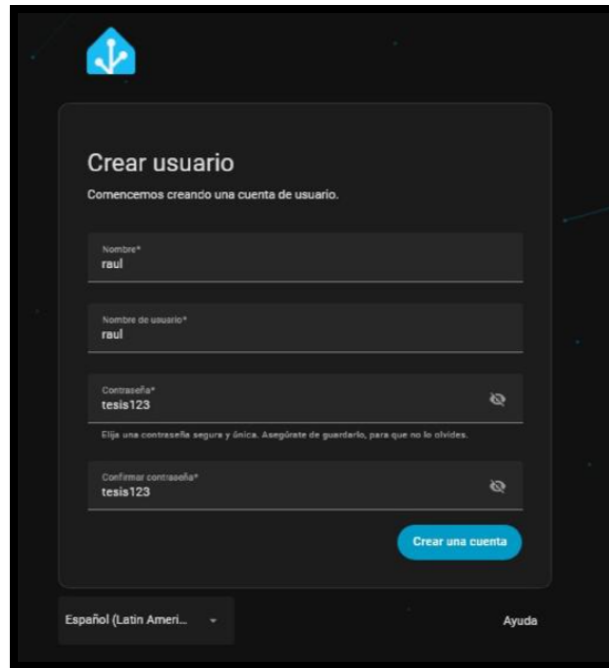


Figura 54 Creación del Usuario en Home Assistant Fuente: Autores.

Anexo 2.3

En la figura 55 el registro del RM4C Mini en Home Assistant: una vez creada la cuenta, se agrega el dispositivo BroadLink (RM4C mini) y se confirma su creación dentro del panel.



Figura 55 Creación del Panel de Registro RM4C. Fuente: Autores.

Anexo 2.4

En la figura 56 Verificación del dispositivo: se valida la información del RM4C (modelo, firmware y estado) dentro de la vista de dispositivo para confirmar conectividad y control habilitado.

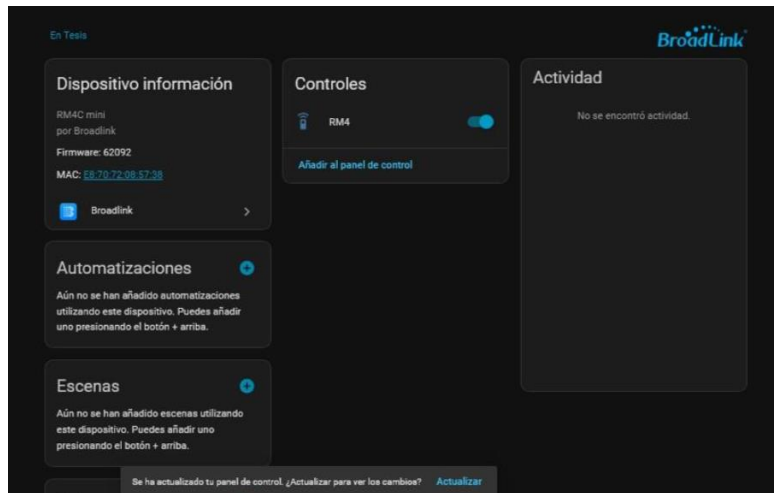


Figura 56 Detalle del dispositivo RM4C Fuente: Autores.

Anexo 2.5

En la figura 57 se muestra el alta del aire acondicionado en la app BroadLink: desde la aplicación móvil se agrega un dispositivo nuevo y se selecciona la opción de control remoto universal.

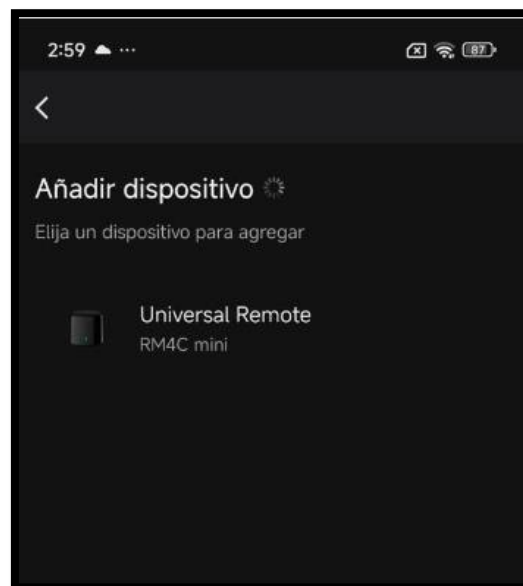


Figura 57 Dispositivo Agregado Fuente: Autores

Anexo 2.6

En el catálogo de electrodomésticos se elige la opción de aire acondicionado (AC Remote) para vincular el equipo físico con el RM4C.

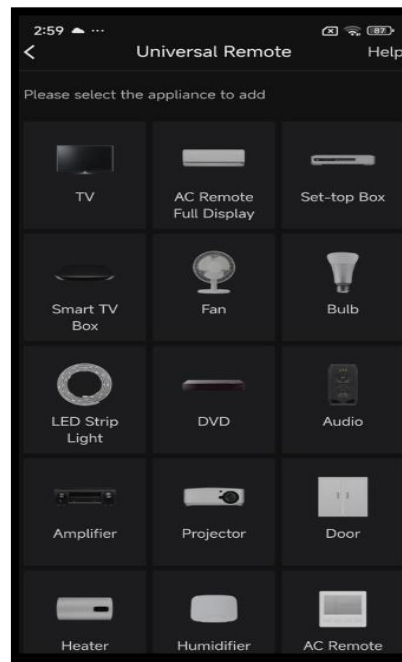


Figura 58 Selección del AC Fuente: Autores.

Anexo 2.7

Se verifica que el control remoto universal y el aire acondicionado aparezcan en línea dentro de la app.

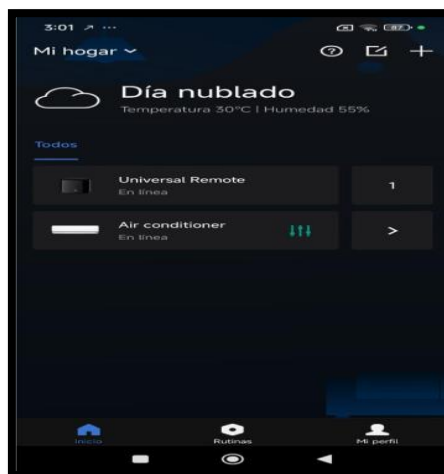


Figura 59 App del BroadLink Fuentes: Autores.

Anexo 2.8

Configuración del panel de control climatizado en el frontend se registra la URL de Home Assistant, el token de acceso, la entidad del control remoto y el nombre del dispositivo, dejando el sistema listo para ejecutar reglas automáticas.

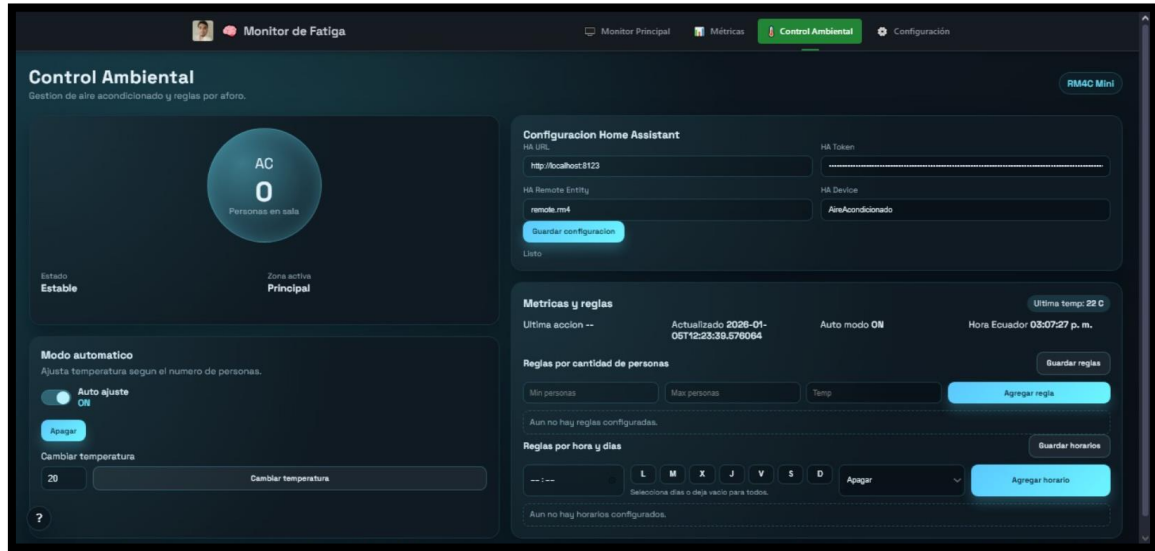


Figura 60 Control Ambiental Fuente: Autores.