



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y DISEÑO DIGITAL
CARRERA DE TELEMÁTICA

Trabajo de Investigación
Curricular previa a la obtención
del Grado Académico de
Ingeniero en Telemática.

TÍTULO DEL PROYECTO TECNOLÓGICO:

**SISTEMA DE INTERNET DE LAS COSAS CON ALGORITMOS DE
APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN TEMPRANA
DE DIFICULTADES LECTORAS EN NIÑOS CON AUTISMO DE
EDUCACIÓN BÁSICA**

AUTOR:

GRACIA GUATO FELIX RAFAEL

DIRECTOR DEL PROYECTO TECNOLÓGICO:

ING. NATA CASTRO DAISY JUDITH, MSC

QUEVEDO – LOS RÍOS – ECUADOR

2025



DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, **Felix Rafael Gracia Guato**, declaro que la investigación aquí descrita es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este documento, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

FELIX RAFAEL GRACIA GUATO

C.I: 2350999252



CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN

El suscrito, **Ing. Daisy Judith Nata Castro, MSc.**, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el estudiante **Felix Rafael Gracia Guato**, realizó el Proyecto de Investigación de grado titulado “**Sistema de Internet de las Cosas con Algoritmos de Aprendizaje Automático para la Detección Temprana de Dificultades Lectoras en Niños con Autismo de Educación Básica**”, previo a la obtención del título de **Ingeniero en Telemática**, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

Ing. Daisy Judith Nata Castro, MSc.

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN



CERTIFICADO DEL REPORTE DE LA HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO

El suscrito, **Daisy Judith Nata Castro**, mediante el presente cumpla en presentar a usted, el informe de proyecto de Investigación titulado **“Sistema de Internet de las Cosas con Algoritmos de Aprendizaje Automático para la Detección Temprana de Dificultades Lectoras en Niños con Autismo de Educación Básica”** Presentado por el estudiante **Felix Rafael Gracia Guato**, egresado de la Carrera de **Telemática**, que fue revisado bajo mi dirección según resolución del Consejo Directivo de la Facultad de Ciencias de la Computación y Diseño Digital, que se ha desarrollado de acuerdo al Reglamento de la Unidad de Integración Curricular de la Universidad Técnica Estatal de Quevedo y cumple con el requerimiento de análisis del sistema COMPILATIO el cual avala los niveles de originalidad en un 99% y similitud 1%, del trabajo investigativo. Valido este documento para que el estudiante siga con los trámites pertinentes, de acuerdo como lo establece el Reglamento.

**INFORME DE ANÁLISIS**
magister

Tesis Felix Gracia 2025

< 1%

Textos sospechosos



< 1% Similitudes
0% similitudes entre conillas
0% entre las fuentes mencionadas
0% idiomas no reconocidos

Nombre del documento: Tesis Felix Gracia 2025.pdf
ID del documento: 36d4bcd3699b7d3e55b7a2252c4f182ce21e254
Tamaño del documento original: 634,07 kB

Depositante: DAISY JUDITH NATA CASTRO
Fecha de depósito: 18/11/2025
Tipo de carga: interface
fecha de fin de análisis: 18/11/2025

Número de palabras: 12.618
Número de caracteres: 95.125

Ing. Daisy Judith Nata Castro, MSc.

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN Y DISEÑO DIGITAL
CARRERA DE TELEMÁTICA
PROYECTO DE INVESTIGACIÓN

TÍTULO:

**“Sistema de Internet de las Cosas con Algoritmos de Aprendizaje Automático para la
Detección Temprana de Dificultades Lectoras en Niños con Autismo de Educación
Básica”**

Presentado al Consejo Directivo de la Facultad de Ciencias de la Computación y Diseño Digital, como requisito previo a la obtención del título de **Ingeniero en Telemática**.

Aprobado por:

PRESIDENTE DEL TRIBUNAL

Ing. Eduardo Amable Samaniego Mena, MSc.

MIEMBRO DEL TRIBUNAL

Ing. Alex Rosendo Fiallos Barrionuevo, MSc.

MIEMBRO DEL TRIBUNAL

Ing. Janeth Ines Mora Secaira, MSc.

QUEVEDO – LOS RIOS – ECUADOR

2025

AGRADECIMIENTO

Deseo iniciar expresando mi más sincero y profundo agradecimiento a mi alma máter, la Universidad Técnica Estatal de Quevedo. Esta noble institución no solo fue el escenario de mi formación académica, sino un verdadero hogar que me acogió y me proveyó de las bases teóricas, los recursos tecnológicos y las oportunidades de desarrollo necesarias. Es en sus aulas y laboratorios donde forjé el carácter y la disciplina indispensables para crecer como profesional y como persona.

Mi gratitud se dirige de manera muy especial e imperecedera a mi directora de tesis, la Ing. Daisy Judith Nata Castro, MSc. Su mentoría trascendió lo puramente académico para convertirse en un pilar fundamental en la culminación de este proyecto. Valoro profundamente su paciencia infinita, su guía metodológica siempre precisa y oportuna, y la confianza inquebrantable que depositó en mis capacidades. Fue su visión y su capacidad para impulsarme a superar cada desafío con rigor y profesionalismo lo que me permitió transformar ideas complejas en resultados tangibles.

Finalmente, extendiendo mi más alto reconocimiento al distinguido cuerpo docente de la carrera de Ingeniería en Telemática. Agradezco a cada docente por las lecciones compartidas, que fueron mucho más allá de los libros de texto. Su dedicación, su pasión por la enseñanza y su compromiso por forjar en mí un pensamiento crítico, las competencias técnicas y los valores éticos, son el legado más valioso que me llevo. Ustedes han definido mi perfil como ingeniero y me han preparado para los retos del futuro.

DEDICATORIA

Esta tesis, más que un requisito académico, es la culminación de un viaje personal y profesional que no habría sido posible sin el amor, el apoyo y la fe de las personas que han sido mi inspiración y mi fortaleza.

A mis amados padres, María Guato y Félix Gracia, pilares fundamentales de mi vida. Gracias por su amor incondicional y por los innumerables sacrificios, visibles e invisibles, que hicieron para que yo pudiera tener esta oportunidad. Aunque la distancia física nos separó, su apoyo constante, sus palabras de aliento en cada llamada y su fe inquebrantable en mi potencial fueron la luz que me guio en los momentos de duda y mi mayor motivación para no rendirme.

A mi querida hermana, Noemi Guato, y a mi cuñado, Mauricio Camacho. Su generosidad y apoyo inestimable fueron la base material y emocional que me permitió establecerme y vivir en otra ciudad. Ustedes me brindaron la tranquilidad de tener un hogar y la posibilidad de dedicarme por completo a mis estudios, asumiendo mis necesidades como propias. Este logro es un testimonio de su desinterés y confianza en mí.

A mi compañera de vida, Kelly Zarsoza. Gracias por ser mi refugio incondicional y mi ancla firme en medio de las tormentas académicas. Tu paciencia infinita transformó las largas noches de estudio en un desafío compartido. Tu capacidad para cuidarme, animarme y permanecer a mi lado "en la salud y en la enfermedad", y tu amor inquebrantable, han sido el cimiento sobre el cual construí este logro. Crecimos juntos a través de este desafío, tu fortaleza me hizo más fuerte y tu sonrisa fue mi mejor recompensa. Este logro es tan tuyo como mío, y es solo el primer paso de todo lo que construiremos juntos.

A todos ustedes, con todo mi corazón, gracias. Este triunfo no es solo mío, es el resultado tangible de su amor y apoyo. Es tan suyo como mío.

RESUMEN Y PALABRAS CLAVES

Este proyecto desarrolla un sistema multimodal de bajo costo, basado en IoT y aprendizaje automático, para la detección temprana de indicadores de atención y habla en niños de educación básica con Trastorno del Espectro Autista (TEA). Ante la falta de herramientas objetivas de monitoreo, se diseñó una arquitectura escalable con nodos sensores no invasivos (ESP32-CAM, micrófonos) que capturan datos visuales y auditivos. Un software central procesa los datos mediante tres especialistas de IA: un detector de atención visual (MediaPipe), un transcriptor de voz (Vosk) y un analizador de coherencia lingüística. El sistema identifica patrones como desvío de la mirada o repeticiones, visualizando los resultados en un dashboard web para docentes. La validación se realizó mediante una aplicación piloto en la Institución “ROUSSBRISS”, en un entorno real con 4 niños con TEA durante 9 días. Los algoritmos de detección alcanzaron una precisión del 93% en atención y alta fiabilidad en habla. Se concluye con un prototipo funcional que representa una herramienta de apoyo pedagógico viable, discutiendo sus limitaciones y su plan de escalabilidad.

Palabras Clave: Internet de las Cosas (IoT), Aprendizaje Automático, Trastorno del Espectro Autista (TEA), Detección Temprana, Análisis Multimodal.

ABSTRACT

This project develops a low-cost multimodal system, based on IoT and machine learning, for the early detection of attention and speech indicators in elementary school children with Autism Spectrum Disorder (ASD). Given the lack of objective monitoring tools, a scalable architecture was designed with non-invasive sensor nodes (ESP32-CAM, microphones) that capture visual and auditory data. A central software processes data using three AI specialists: a visual attention detector (Media Pipe), a voice transcriber (Vosk), and a linguistic coherence analyzer. The system identifies patterns such as gaze deviation or repetitions, displaying the results on a web dashboard for teachers. Validation was performed through a pilot application at the "ROUSSBRISS" Institution, in a real-world setting with 4 children with ASD over 9 days. The detection algorithms achieved 93% accuracy in attention and high reliability in speech. The project concludes with a functional prototype that represents a viable pedagogical support tool, discussing its limitations and scalability plan.

Keywords: Internet of Things (IoT), Machine Learning, Autism Spectrum Disorder (ASD), Early Detection, Multimodal Analysis.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
CAPITULO I.....	2
1.1. Problematización.....	3
1.1.1. Planteamiento del problema.....	3
1.1.1.1. Diagnóstico.	3
1.1.1.2. Pronóstico.....	4
1.1.2. Formulación del problema	4
1.1.3. Sistematización del problema	4
1.2. Objetivos	4
1.2.1. Objetivo General.....	4
1.2.2. Objetivos Específicos.....	5
1.3. Justificación	5
CAPITULO II.....	6
2.1. Marco conceptual.....	7
2.1.1. Internet de las cosas	7
2.1.2. Aprendizaje automático	7
2.1.3. En la educación	7
2.1.4. Trastorno del Espectro Autista	8
2.1.5. Personalización del aprendizaje	8
2.1.6. Algoritmos	8
2.1.7. Ética y privacidad de datos	8
2.2. Marco referencial	9
2.3. Marco legal	11
2.3.1. Constitución de la República del Ecuador	11
2.3.2. Ley orgánica de protección de datos personales (lopdp)	11
2.3.3. Ley orgánica de educación intercultural (2021)	12
2.3.4. Código de la niñez y adolescencia (2003)	12
2.3.5. Ley orgánica de telecomunicaciones (2015).....	12
2.3.6. Reglamento General a la Ley Orgánica de Telecomunicaciones	12
CAPITULO III.....	13
3.1. Localización.....	14

3.2.	Tipo de Investigación	14
3.3.	Métodos de Investigación	15
3.4.	Fuentes de Información.....	15
3.4.1.	Primarias	15
3.4.2.	Secundarias	15
3.5.	Diseño de investigación	15
3.6.	Fases.....	16
3.6.1.	Fase 1: Diseño y prototipado del sistema IoT	16
3.6.2.	Fase 2: Implementación y calibración de algoritmos de machine learning	16
3.6.3.	Fase 3: Desarrollo de la interfaz de visualización.....	16
3.6.4.	Fase 4: Pruebas integrales y validación del sistema	17
3.7.	Consideraciones Éticas y Consentimiento Informado	17
CAPITULO IV.....		18
4.1.	Diseño del Sistema IoT para la recolección de datos.....	19
4.1.1.	Análisis y selección de los dispositivos	19
4.1.1.1.	Tablas comparativas de los módulos.....	19
4.1.2.	Criterios de selección de los módulos.....	21
4.1.2.1.	Criterio de selección del módulo de análisis visual.	21
4.1.2.2.	Criterio de selección del módulo de análisis vocal.	21
4.1.2.3.	Criterio de selección del módulo de captura de audio.	22
4.1.3.	Estructura del sistema IoT.....	22
4.1.3.1.	Capa de adquisición de datos.	23
4.1.3.2.	Capa de procesamiento local (edge computing).	23
4.1.3.3.	Capa de conectividad y nube.	23
4.1.4.	Diseño de hardware.....	23
4.1.4.1.	Hardware del nodo sensor.	23
4.1.4.2.	Hardware del nodo central.	24
4.1.5.	Diseño de software y firmware	24
4.1.5.1.	Firmware del nodo sensor.	24
4.1.5.2.	Software del nodo central.	25
4.1.6.	Diseño de la comunicación	25
4.1.6.1.	Protocolo de comunicación.	25
4.1.6.2.	Estructura de tópicos (topics mqtt).	25

4.1.6.3.	Formato de los datos (payload).	25
4.1.7.	Discusión	26
4.2.	Implementación de algoritmos para la detección de patrones lectores	26
4.2.1.	Pipeline de procesamiento y análisis de datos	27
4.2.1.1.	Pipeline de análisis visual.	28
4.2.1.2.	Pipeline de análisis vocal.	29
4.2.2.	Extracción de métricas conductuales	29
4.2.2.1.	Métricas visuales.	29
4.2.2.2.	Métrica de ritmo: palabras por minuto (ppm).	30
4.2.2.3.	Métricas de fluidez: tasa de repetición y hesitación.	30
4.2.2.4.	Métricas de fluidez: tasa de repetición y hesitación.	30
4.2.2.5.	Métrica de contexto: análisis prosódico.	30
4.2.3.	Algoritmo de decisión basado en reglas jerárquicas	31
4.2.3.1.	Jerarquía de decisión visual.	31
4.2.3.2.	Jerarquía de decisión auditiva.	31
4.2.4.	Validación y rendimiento de los algoritmos	32
4.2.5.	Discusión	33
4.3.	Desarrollo de la interfaz de visualización para docentes (dashboard estadístico) ...	33
4.3.1.	Diseño de la interfaz y experiencia de usuario (ux)	34
4.3.2.	Arquitectura tecnológica del dashboard	34
4.3.3.	Funcionalidades implementadas	35
4.3.3.1.	Monitor de estado en tiempo real.	35
4.3.3.2.	Indicadores clave de rendimiento (kpis).	35
4.3.3.3.	Visualizaciones estadísticas	35
4.3.3.4.	Selector multi-estudiante.	36
4.3.4.	Análisis de Resultados de la Aplicación Piloto	36
4.3.4.1.	Perfil de Atención General	36
4.3.4.2.	Perfil de Habla General	37
4.3.4.3.	Evolución de la Tasa de Alertas Visuales por Día	38
4.3.4.4.	Evolución de la tasa de alertas de habla por día	39
4.3.4.5.	Evolución de la velocidad lectora (ppm) por día	40
4.3.4.	Discusión	41
CAPÍTULO V		42

5.1. Conclusiones	43
5.2. Recomendaciones.....	44
CAPITULO VI.....	45
CAPITULO VII	49

UNDICE DE TABLAS

Tabla 1. Análisis comparativo de módulos de captura visual	19
Tabla 2. Análisis comparativo de micrófonos para análisis vocal.....	20
Tabla 3. Análisis comparativo de modulo para captura de audio.....	20
Tabla 4. Evaluación del sistema	32

INDICE DE ILUSTRACIONES

Ilustración 1. Ubicación de la Institución ROUSSBRISSE.....	14
Ilustración 2. Estructura del Sistema	22
Ilustración 3. Nodo Sensor	23
Ilustración 4. Flujo de procesamiento	27
Ilustración 5. Interfaz final del Dashboard Estadístico con datos multimodales	34
Ilustración 6. Perfil de atención general.....	36
Ilustración 7. Perfil de habla general.....	37
Ilustración 8. Tasa de alertas visuales por día de aplicación	38
Ilustración 9. Tasa de alertas de habla por día de aplicación.....	39
Ilustración 10. Evolución de Palabras por Minuto	40

INDICE DE ANEXOS

Anexo 1. Módulos de sonidos configurados.....	50
Anexo 2. Módulo de sonido ensamblado	50
Anexo 3. Módulos de video configurados	50
Anexo 4. Módulo de video	50
Anexo 5. Configuración del equipo de procesamiento local	51
Anexo 6. Comprobación de funcionamiento del módulo de video	51
Anexo 7. Centro de atención integral ROUSSBRIS	52
Anexo 8. Configuración del equipo de procesamiento local	52
Anexo 9. Punto de red de la Institución.....	52
Anexo 10. Ajustes de los equipos instalados	53
Anexo 11. Verificando captación de audio y video	53
Anexo 12. Sistema en funcionamiento	54
Anexo 13. Explicación de alertas	54
Anexo 14. Sistema de alertas en funcionamiento	54
Anexo 15. Miembros encargados de la enseñanza	55
Anexo 16. Código para el ESP32 - WROOM con el sensor de sonidoINMP441	55
Anexo 17. Código para el ESP32 - CAM.....	58
Anexo 18. Código del servidor local (Python) de procesamiento de los datos	62
Anexo 19. Carta de consentimiento	74

CÓDIGO DUBLIN

Título:	Sistema de Internet de las Cosas con Algoritmos de Aprendizaje Automático para la Detección Temprana de Dificultades Lectoras en Niños con Autismo de Educación Básica		
Autor:	Felix Rafael Gracia Guato		
Palabras claves:	Internet de las Cosas	Aprendizaje Automático	Trastorno del Espectro Autista
Fecha de publicación:	2025		
Editorial:	Quevedo- UTEQ, 2025		
Resumen: (hasta 300 palabras)	Resumen. - Este proyecto desarrolla un sistema multimodal de bajo costo, basado en IoT y aprendizaje automático, para la detección temprana de indicadores de atención y habla en niños de educación básica con Trastorno del Espectro Autista (TEA). Ante la falta de herramientas objetivas de monitoreo, se diseñó una arquitectura escalable con nodos sensores no invasivos (ESP32-CAM, micrófonos) que capturan datos visuales y auditivos. Un software central procesa los datos mediante tres especialistas de IA: un detector de atención visual (MediaPipe), un transcriptor de voz (Vosk) y un analizador de coherencia lingüística. El sistema identifica patrones como desvío de la mirada o repeticiones, visualizando los resultados en un dashboard web para docentes. La validación se realizó mediante una aplicación piloto en la Institución ROUSSBRIS, en un entorno real con 4 niños con TEA durante 9 días. Los algoritmos de detección alcanzaron una precisión del 93% en atención y alta fiabilidad en habla. Se concluye con un prototipo funcional que representa una herramienta de apoyo pedagógico viable, discutiendo sus limitaciones y su plan de escalabilidad, (...)		
Abstract: (hasta 300 palabras)	Abstract. - This project develops a low-cost multimodal system, based on IoT and machine learning, for the early detection of attention and speech indicators in elementary school children with Autism Spectrum Disorder (ASD). Given the lack of objective monitoring tools, a scalable architecture was designed with non-invasive sensor nodes (ESP32-CAM, microphones) that capture visual and auditory data. A central software processes data using three AI specialists: a visual attention detector (MediaPipe), a voice transcriber (Vosk), and a linguistic coherence analyzer. The system identifies patterns such as gaze deviation or repetitions, displaying the results on a web dashboard for teachers. Validation was performed through a pilot application at the ROUSSBRIS Institution, in a real-world setting with 4 children with ASD over 9 days. The detection algorithms achieved 93% accuracy in attention and high reliability in speech. The project concludes with a functional prototype that represents a viable pedagogical support tool, discussing its limitations and scalability plan, (...)		
Descripción:	92 hojas: dimensiones, 29 x 21 cm + CD-ROM 6162		
URI:			

INTRODUCCIÓN

La lectura es una puerta hacia el conocimiento y la inclusión social, pero para muchos niños con Trastorno del Espectro Autista (TEA), esta puerta se encuentra obstruida por dificultades ocultas. En Ecuador, 4 de cada 10 estudiantes con necesidades educativas específicas no logran alcanzar un nivel básico de comprensión lectora, según datos del INEC [1].

Hoy, los docentes enfrentan este desafío armado únicamente con observaciones subjetivas y pruebas estandarizadas [2], herramientas valiosas pero insuficientes para capturar señales tempranas, como la pérdida de atención durante actividades de lectura [3]. El resultado es un diagnóstico tardío que, según estudios en poblaciones con TEA [4], afecta no solo el rendimiento académico, sino también la autopercepción de los niños como aprendices.

La viabilidad técnica de este enfoque queda respaldada por investigaciones recientes documentados en la literatura especializada. Como demuestra el estudio de Popescu [5], los sistemas IoT integrados con inteligencia artificial pueden interpretar automáticamente estados afectivos en niños con TEA mediante el análisis multimodal de datos. Complementariamente, la revisión sistemática de Kohli y Kar [6] evidencia que las tecnologías inteligentes permiten la detección temprana de indicadores clave del espectro autista con una sensibilidad del 91%, destacando especialmente el potencial de los algoritmos de visión artificial para capturar marcadores conductuales sutiles. Estos hallazgos respaldan científicamente nuestra propuesta de combinar sensores de bajo costo con técnicas avanzadas de machine learning adaptadas al contexto escolar ecuatoriano.

Frente a esta realidad, este proyecto propone un acompañamiento tecnológico a los educadores. Es crucial aclarar que el sistema no es una herramienta de diagnóstico clínico para "dificultades lectoras" como la dislexia, sino un sistema de monitoreo de indicadores conductuales precursores que están frecuentemente asociados a dichas dificultades. Mediante cámaras que analizan la atención visual y micrófonos que evalúan la fluidez vocal, el sistema identifica patrones objetivos como la pérdida de atención sostenida o las disfluencias en el habla. Los algoritmos de aprendizaje automático actúan como traductores, convirtiendo datos técnicos complejos en alertas comprensibles para los docentes. Más allá de la innovación técnica, el sistema busca devolverle a la educación su esencia humana. Al automatizar la observación de estos indicadores y liberar a los maestros de la carga del monitoreo manual, les permite dedicar tiempo valioso a lo que realmente importa: guiar, motivar y celebrar cada pequeño avance de sus estudiantes.

CAPITULO I
CONCEPTUALIZACIÓN

1.1. Problematización

1.1.1. Planteamiento del problema

La lectura es una habilidad fundamental en el desarrollo educativo de los niños. Pese a ello, muchas instituciones enfrentan desafíos al identificar dificultades lectoras, como la dislexia o problemas de comprensión, en etapas tempranas. Según un estudio de la UNESCO [7], el 60% de los estudiantes en América Latina presentan rezagos en comprensión lectora al finalizar la primaria, cifra que se agrava en contextos vulnerables. En Ecuador, el INEC [1] reporta que el 25% de los niños en zonas urbanas marginadas no alcanzan el nivel básico de lectura, mientras que en zonas rurales este porcentaje aumenta al 38%.

Instituciones que atienden a población con autismo y de bajos recursos, carece de herramientas tecnológicas para monitorear indicadores clave como la atención sostenida o la velocidad lectora, lo que dificulta la detección oportuna de problemas. Es común que en estos contextos [3], los estudiantes presenten dificultades para mantener la concentración durante actividades de lectura grupal.

El uso de tecnologías como el Internet de las Cosas y los algoritmos de aprendizaje automático, presentan una oportunidad única para abordar estas problemáticas. Estas herramientas pueden permitir la recolección y análisis de datos relacionados con la atención y el comportamiento lector, generando insumos valiosos para que los docentes implementen estrategias pedagógicas personalizadas.

1.1.1.1. Diagnóstico.

Instituciones educativas actualmente utilizan métodos tradicionales de evaluación, como observación directa y pruebas escritas, que no permiten un análisis objetivo ni en tiempo real del comportamiento lector. Frecuentemente, los docentes manifiestan dificultades [8] para identificar señales tempranas de comprensión lectora debido a la falta de herramientas tecnológicas.

Estas limitaciones metodológicas se ven agravadas en el caso de estudiantes con autismo, quienes frecuentemente presentan patrones atencionales atípicos y dificultades de comunicación que los métodos tradicionales no logran capturar adecuadamente. Adicionalmente, la alta variabilidad individual en el espectro autista dificulta aún más la identificación consistente de problemas lectores mediante observación no asistida por tecnología.

1.1.1.2. Pronóstico.

Si no se implementan soluciones tecnológicas innovadoras, es probable que las dificultades lectoras en los estudiantes continúen sin ser identificadas, afectando su desarrollo académico y personal. Esta situación podría derivar en un incremento de la tasa de deserción escolar, particularmente en estudiantes con necesidades educativas específicas quienes requieren mayor apoyo personalizado.

Además, la ausencia de estas herramientas limitará la capacidad de los docentes para aplicar estrategias pedagógicas efectivas. Esto no solo afectará el rendimiento académico de los estudiantes, sino que también tendrá un impacto negativo en su autoestima y desarrollo integral, perpetuando un ciclo de exclusión y desigualdad educativa. La falta de intervención oportuna podría, a largo plazo, reducir las oportunidades laborales y sociales de estos estudiantes en su vida adulta.

1.1.2. Formulación del problema

¿Cómo puede un sistema integrado de Internet de las cosas y con algoritmos de aprendizaje automático optimizar la detección temprana de dificultades lectoras en estudiantes con autismo de básica, mediante la recolección de datos en tiempo real?

1.1.3. Sistematización del problema

- ¿Cómo recopilar datos a través de indicadores de atención y comportamiento lector para identificar dificultades en los estudiantes?
- ¿Cómo pueden un sistema basado en Internet de las cosas y algoritmos de aprendizaje automático procesar estos indicadores para generar diagnósticos precisos?
- ¿Qué funcionalidades debe incluir una aplicación para docentes que facilite la interpretación de datos y la aplicación de estrategias pedagógicas?

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un sistema basado en Internet de las cosas con algoritmos de aprendizaje automático que permita a los docentes monitorear la atención de los estudiantes durante actividades de lectura

1.2.2. Objetivos Específicos

- Diseñar un sistema basado en Internet de las cosas que recopile datos sobre la atención y el comportamiento lector de los estudiantes mediante cámaras y micrófonos.
- Implementar algoritmos de aprendizaje automático, aplicando técnicas de procesamiento de imagen y voz, para analizar los datos recopilados, identificando patrones que indiquen dificultades lectoras.
- Desarrollar una aplicación que permita a los docentes visualizar los datos mediante dashboards.

1.3. Justificación

El proyecto propuesto busca abordar una problemática crítica en la educación: la identificación temprana de dificultades lectoras en niños mediante el uso de tecnologías avanzadas como el Internet de las Cosas y los algoritmos de aprendizaje automático.

Estas herramientas permitirán complementar el trabajo de los docentes, ofreciendo un enfoque innovador que optimiza el monitoreo del comportamiento lector y promueve intervenciones pedagógicas personalizadas.

A pesar de los avances en la reducción del analfabetismo en Ecuador, persisten retos en el desarrollo de habilidades lectoras básicas, especialmente en comunidades vulnerables. Muchas instituciones carecen de tecnologías modernas que permitan detectar problemas de aprendizaje de manera oportuna. La integración de Internet de las cosas y algoritmos de aprendizaje automático puede cerrar esta brecha al generar datos en tiempo real sobre la atención y el rendimiento lector, permitiendo una detección temprana de estos indicadores, brindando una herramienta sólida y escalable a futuro.

El impacto de este proyecto no solo se limita a mejorar la calidad para niños con autismo, sino que también sienta las bases para la replicación de este modelo en otras instituciones. Además, contribuye al desarrollo de soluciones tecnológicas en el ámbito de la Ingeniería en Telemática aplicadas a contextos sociales, promoviendo una educación más equitativa y personalizada.

CAPITULO II
FUNDAMENTACIÓN TEORICA DE LA INVESTIGACIÓN

2.1. Marco conceptual

2.1.1. *Internet de las cosas*

La Internet de las Cosas (IoT) representa un avance significativo en la interconexión de objetos cotidianos mediante redes inteligentes. Estos dispositivos, equipados con capacidades de procesamiento, recopilan y comparten datos con su entorno, transformando la Internet tradicional en un ecosistema más amplio y perceptivo. Esta evolución no solo amplía la conectividad, sino que también enriquece los servicios con inteligencia contextual, permitiendo aplicaciones más sofisticadas [9].

En el ámbito educativo, las tecnologías basadas en IoT han demostrado un impacto positivo en múltiples dimensiones. Por un lado, facilitan la interacción entre docentes y estudiantes, permitiendo un monitoreo más preciso del progreso académico. Por otro, optimizan aspectos operativos como la seguridad física, la gestión energética y el mantenimiento de infraestructuras. Sin embargo, su implementación enfrenta desafíos críticos, como la integración armoniosa de servicios heterogéneos y la garantía de privacidad y seguridad de los datos sensibles generados [10].

2.1.2. *Aprendizaje automático*

El Aprendizaje automático es un área de conocimiento dentro de la Inteligencia Artificial donde los ordenadores aplican técnicas de aprendizaje estadístico con el objetivo de identificar automáticamente patrones en los datos [11].

2.1.3. *En la educación*

El aprendizaje automático está transformando múltiples ámbitos, incluido el educativo, especialmente en la implementación de una enseñanza personalizada. Gracias al aprendizaje adaptativo, es posible ajustar la formación según las necesidades individuales de cada alumno. Plataformas educativas con inteligencia artificial utilizan algoritmos para evaluar el desempeño, conducta y preferencias de los estudiantes, modificando en tiempo real el ritmo y los contenidos de enseñanza. Así, se logra una experiencia más eficaz y personalizada, permitiendo que los alumnos progresen a su propio ritmo y refuercen las áreas donde más lo necesitan [12].

2.1.4. Trastorno del Espectro Autista

El trastorno del espectro autista (TEA) es una condición del neurodesarrollo de origen multifactorial, que se manifiesta principalmente por dificultades en la interacción social, alteraciones en la comunicación y patrones de conducta repetitivos, con intereses y actividades restringidas. Actualmente, su etiopatogenia y los factores de riesgo asociados han sido ampliamente investigados, permitiendo un mayor entendimiento de este trastorno. Se trata de una condición relativamente frecuente, ya que, según estimaciones globales, aproximadamente 1 de cada 160 niños presenta algún grado de TEA [4].

2.1.5. Personalización del aprendizaje

La educación personalizada consiste en ajustar los materiales didácticos, la velocidad de aprendizaje y la retroalimentación según las particularidades de cada alumno. Este método marca un avance clave hacia una enseñanza de mayor calidad, especialmente al integrarse con innovaciones tecnológicas como la inteligencia artificial, transformando a la tecnología en una herramienta clave para enriquecer el proceso educativo. Al implementar la personalización del aprendizaje mediante Inteligencia Artificial, no solo se potencia el rendimiento académico, sino que también se equipa a los estudiantes con las habilidades necesarias para adaptarse a las exigencias de un entorno globalizado y en constante evolución [13].

2.1.6. Algoritmos

Un algoritmo consiste en una secuencia lógica de operaciones o reglas definidas, diseñadas para ejecutar una acción concreta y alcanzar un objetivo predeterminado, ya sea resolver un problema, procesar datos o cumplir una meta específica [14].

2.1.7. Ética y privacidad de datos

La protección de la información en el ámbito educativo digital se ha convertido en un desafío ético prioritario. La creciente datificación, potenciada por herramientas de análisis del aprendizaje y sistemas de IA, facilita la recolección y procesamiento masivo de datos estudiantiles. Pese a su relevancia crítica, este tema sigue siendo un campo poco explorado en la investigación actual [15].

La salvaguarda legal de los datos personales se ha convertido en un movimiento internacional que, desde la década de 1970, ha experimentado un crecimiento acelerado.

Este desarrollo normativo busca regular los flujos de información, motivado por los vertiginosos progresos en tecnología, comercio y finanzas, que han convertido a los datos en activos con enorme valor económico y social [16].

La afectación a la información personal guarda una relación directa con el derecho a la intimidad, generando tensiones jurídicas entre:

- La protección de datos
- El derecho a la privacidad
- El acceso a la información (particularmente en el caso de funcionarios públicos)

2.2. Marco referencial

El impacto del IoT en la educación

El Internet de las Cosas ha transformado la forma en que se recopilan y utilizan los datos en entornos educativos. Según estudios realizados por [17], el Internet de las cosas permite un monitoreo continuo y en tiempo real de las actividades de los estudiantes, lo que facilita la identificación de patrones de participación y compromiso. Estos dispositivos, como sensores y cámaras conectadas, también mejoran la capacidad de los docentes para adaptar sus estrategias pedagógicas en función de las necesidades específicas de los estudiantes.

Sin embargo, los estudios previos se centran mayoritariamente en recopilar datos generales y no profundizan en el uso de Internet de las cosas para analizar indicadores específicos de atención durante actividades lectoras. Con este enfoque, se aborda esa brecha al aplicar el Internet de las cosas para monitorear la atención en tiempo real durante actividades de lectura, permitiendo una personalización efectiva.

Machine Learning y personalización del aprendizaje

El aprendizaje automático ofrece herramientas poderosas para el análisis de datos en el ámbito educativo. [18] destacan que los algoritmos de aprendizaje automático permiten procesar grandes volúmenes de datos para identificar patrones complejos, como indicadores tempranos de dificultades lectoras. A pesar de ello, gran parte de las investigaciones existentes se enfocan en modelos predictivos sin considerar su implementación en tiempo real ni la integración con herramientas que puedan ser utilizadas directamente por los docentes.

Retos de la alfabetización en Ecuador

En Ecuador, persisten desigualdades significativas en el acceso a recursos educativos. Datos del [1]muestran que las zonas rurales enfrentan mayores índices de deserción escolar y bajo rendimiento académico, factores que están directamente relacionados con la falta de tecnología educativa y estrategias innovadoras. Aunque existen iniciativas nacionales enfocadas en mejorar la alfabetización, estas no integran tecnologías avanzadas como el Internet de las cosas y algoritmos de aprendizaje automático para la detección y prevención temprana de problemas. La propuesta busca cerrar esta brecha al implementar un sistema tecnológico que facilite la identificación de problemas de atención y comportamiento lector, promoviendo un aprendizaje más equitativo.

Convergencia de tecnología y educación

Un estudio realizado por [7] resalta la importancia de integrar tecnologías emergentes en las aulas para mejorar los resultados educativos. Tecnologías como el Internet de las cosas y algoritmos de aprendizaje automático no solo mejoran la eficiencia del aprendizaje, sino que también fomentan una mayor motivación y compromiso por parte de los estudiantes. En esta consideración, la mayoría de las iniciativas tecnológicas se han centrado en herramientas generales para el aprendizaje interactivo, dejando de lado su aplicación en aspectos específicos como la atención y el rendimiento lector.

Sensing technologies and machine learning methods for emotion recognition in autism: Systematic review

Esta revisión sistemática elaborada por Oresti Banos [19] demuestra cómo la convergencia entre tecnologías de sensores y métodos de aprendizaje automático está transformando el reconocimiento de emociones en TEA. Los hallazgos subrayan el potencial de estas herramientas para decodificar respuestas afectivas no verbales, ofreciendo a profesionales educativos insumos objetivos que complementan la observación tradicional. Más allá de sus aplicaciones técnicas, el estudio evidencia cómo la innovación tecnológica puede humanizar la intervención pedagógica en neurodiversidad.

Use of mobile and wearable artificial intelligence in child and adolescent psychiatry: Scoping review

El estudio de Tom Joshua [20] explora cómo la inteligencia artificial (IA) integrada en dispositivos móviles y wearables está redefiniendo el abordaje de la salud mental en niños y adolescentes.

Esta investigación no solo describe avances tecnológicos, sino que revela un cambio de paradigma: dispositivos cotidianos como relojes inteligentes o aplicaciones se convierten en aliados para monitorear indicadores emocionales y conductuales de manera discreta y no invasiva. Los hallazgos destacan el potencial de estas herramientas para detectar patrones sutiles de ansiedad, depresión o trastornos del neurodesarrollo, ofreciendo a profesionales datos objetivos que complementan la evaluación clínica tradicional. Más allá de los algoritmos, este trabajo subraya una oportunidad única: usar la tecnología para tender puentes de comprensión hacia el mundo interior de los jóvenes, especialmente aquellos que, por su condición, enfrentan mayores desafíos para expresarse

2.3. Marco legal

El desarrollo del presente sistema tecnológico se fundamenta en un estricto cumplimiento del marco normativo vigente, tanto nacional como internacional, que regula la protección de datos personales, los derechos de las personas con discapacidad y la aplicación de tecnologías en entornos educativos. Estas disposiciones legales no solo establecen los límites y obligaciones del proyecto, sino que garantizan que su implementación respete plenamente la dignidad, privacidad y derechos fundamentales de los participantes, en especial de los niños y niñas con trastorno del espectro autista que forman parte del estudio.

2.3.1. Constitución de la República del Ecuador

El corazón de nuestro sistema jurídico late fuerte en el Art. 66.19, que consagra el derecho a la protección de datos personales como un escudo para la privacidad. Este mandato constitucional nos recuerda que detrás de cada imagen capturada por las cámaras hay rostros que merecen respeto, y que cada audio procesado lleva consigo la confianza depositada por las familias. El

[Art. 11, con su principio del interés superior del niño, ilumina cada decisión técnica, exigiendo que la tecnología sirva siempre para ampliar oportunidades, nunca para vulnerar derechos [21].]

2.3.2. Ley orgánica de protección de datos personales (lopdp)

Como guardianes de la información sensible, el Art. 16 de esta ley nos obliga a obtener el consentimiento amorosamente explicado a padres y tutores, asegurando que comprendan cómo, por qué y para qué usaremos los datos.

El Art. 21, con sus exigencias de seguridad, nos impulsa a construir sistemas blindados donde la intimidad de los pequeños sea tan sagrada como su proceso de aprendizaje. Cada línea de código debe ser un compromiso con esta protección [22].

2.3.3. Ley orgánica de educación intercultural (2021)

El Art. 47 de esta ley es un faro que guía nuestro proyecto: exige que las aulas ecuatorianas se transformen para recibir a todos los niños, especialmente aquellos con necesidades educativas especiales. Nuestros dispositivos iot no son simples herramientas técnicas, sino puentes hacia la inclusión, materializando el derecho a adaptaciones tecnológicas que rompan barreras en el aprendizaje de niños con TEA [23].

2.3.4. Código de la niñez y adolescencia (2003)

[El Art. 52 de este código es nuestro juramento ético: ningún avance tecnológico justifica exponer a los niños a riesgos. Por eso, cada circuito, cada algoritmo y cada interfaz pasan por filtros rigurosos que priorizan su bienestar emocional y físico, convirtiendo la precaución en un acto de responsabilidad afectiva [24].]

2.3.5. Ley orgánica de telecomunicaciones (2015)

El marco jurídico ecuatoriano establece en el Artículo 3 de la Ley Orgánica de Telecomunicaciones el acceso a los servicios de comunicación como derecho fundamental. En este contexto, nuestro sistema aprovecha las redes inalámbricas autorizadas para garantizar conectividad segura y eficiente.

[El Artículo 56, que regula el uso del espectro radioeléctrico, fundamenta técnicamente nuestra implementación en la banda de 2.4 GHz, asegurando el cumplimiento de los parámetros establecidos por la ARCOTEL [25].]

2.3.6. Reglamento General a la Ley Orgánica de Telecomunicaciones

[El Artículo 29, que establece los requisitos para equipos de telecomunicaciones, se traduce en nuestro proyecto como un compromiso con la calidad y seguridad. Cada dispositivo homologado por ARCOTEL representa nuestra responsabilidad de ofrecer soluciones tecnológicas confiables y accesibles para el sector educativo [26].]

CAPITULO III

METODOLOGÍA

3.3. Métodos de Investigación

El método de investigación es cuantitativo, basado en el procesamiento estadístico de datos numéricos objetivos recolectados por el sistema IoT (tiempo de atención, frecuencia de pausas, velocidad lectora) mediante algoritmos de aprendizaje automático.

Los resultados se expresan en métricas cuantificables, lo que permite análisis replicables y comparables para identificar patrones asociados a dificultades lectoras.

3.4. Fuentes de Información

3.4.1. Primarias

- **Datos recopilados por el sistema IoT:**
 - Mediciones de indicadores de atención (seguimiento ocular mediante las cámaras integradas de los SP32-CAM).
 - Registros de comportamiento lector (velocidad, pausas y errores de pronunciación capturados con micrófonos INMP441 y procesados con Python).
 - Resultados de la aplicación piloto en el centro de atención integral ROUSSBRIS.

3.4.2. Secundarias

Se consultaron estudios previos e informes institucionales que respaldan la aplicación de tecnologías como redes de sensores con aprendizaje automático en educación, así como investigaciones sobre dificultades lectoras en niños con autismo. Adicionalmente, se revisó documentación técnica relacionada con el procesamiento de imágenes y voz para el desarrollo del sistema.

3.5. Diseño de investigación

El diseño de la investigación es no experimental, ya que se observa el comportamiento lector de los estudiantes sin manipular variables. Además, se utilizará un enfoque transversal, ya que los datos se recopilarán en un período específico de tiempo durante las actividades de lectura en el aula.

3.6. Fases

El desarrollo del proyecto se estructuró en cuatro fases metodológicas secuenciales, que abarcan desde el diseño conceptual hasta la validación integral del sistema.

3.6.1. Fase 1: Diseño y prototipado del sistema IoT

Esta fase inicial se centró en la conceptualización y construcción de la infraestructura física y de comunicación del sistema. Se realizó un análisis comparativo de componentes, seleccionando el microcontrolador ESP32-CAM para la captura visual y un ESP32-WROOM como nodo dedicado para el procesamiento de audio con el micrófono INMP441. Como nodo central, se optó por una PC de laboratorio existente en la institución para maximizar los recursos y garantizar la potencia de cómputo necesaria para el análisis en tiempo real. Posteriormente, se diseñó una arquitectura de red local inalámbrica seleccionando el protocolo MQTT (Message Queuing Telemetry Transport) por su ligereza y eficiencia. Se configuró un broker Mosquitto en la PC central y se definió una estructura de tópicos jerárquica para garantizar la escalabilidad. Finalmente, se programaron los firmwares para ambos nodos sensores, los cuales se encargan de capturar los datos, codificarlos en Base64, empaquetarlos en una carga útil JSON y publicarlos en sus tópicos MQTT correspondientes.

3.6.2. Fase 2: Implementación y calibración de algoritmos de machine learning

Esta fase constituyó el núcleo de inteligencia del proyecto, enfocándose en el desarrollo del software de análisis en el nodo central. Se implementó un script en Python utilizando librerías como Paho-MQTT, OpenCV y MediaPipe. Se integró el modelo de malla facial de MediaPipe, que permite el seguimiento de 478 puntos clave, a partir de los cuales se implementaron funciones para calcular la Relación de Aspecto del Ojo (EAR) y la Relación de Mirada (Gaze Ratio) en dos dimensiones. Consecutivamente, se llevó a cabo un proceso iterativo de pruebas para calibrar los umbrales de detección, realizando ajustes finos para maximizar la sensibilidad a distracciones reales y minimizar los falsos positivos, logrando un equilibrio para un uso natural.

3.6.3. Fase 3: Desarrollo de la interfaz de visualización

El objetivo de esta fase fue traducir los datos técnicos en información visual e intuitiva para los docentes. Se diseñó una aplicación web de una sola página utilizando HTML y el framework de CSS Tailwind. Para la comunicación de datos, se configuró una Firebase Realtime Database como intermediario entre el analizador de Python y la interfaz web.

El script de Python fue modificado para enviar tanto el estado en tiempo real como un registro histórico de cada cambio de estado. Se utilizó la librería Chart.js para crear los gráficos estadísticos y se implementó código JavaScript para leer los datos desde Firebase, procesar el historial para calcular los KPIs y actualizar dinámicamente todos los elementos visuales, incluyendo un selector para la visualización independiente de cada estudiante.

3.6.4. Fase 4: Pruebas integrales y validación del sistema

La fase final consistió en la aplicación piloto del sistema completo en un entorno real dentro de la Institución "ROUSSBRIS", para verificar su cohesión y rendimiento. Se instalaron y activaron los nodos sensores (pares de ESP32-CAM y ESP32-WROOM) para monitorear a un grupo de 4 niños con TEA. La recolección de datos se realizó durante un periodo de 9 días que fueron distribuidos en 3 días por semana, 2 horas diarias, verificando que el servidor MQTT y el script de Python pudieran gestionar y procesar los flujos de datos paralelos correctamente. Se realizaron pruebas funcionales para confirmar que los estados de atención detectados por el analizador se reflejaran de manera precisa y oportuna en el dashboard estadístico. Finalmente, se midió la latencia del sistema, confirmando que el rendimiento era adecuado para un monitoreo en tiempo real.

3.7. Consideraciones Éticas y Consentimiento Informado

Más allá del cumplimiento del marco legal, la ejecución de la aplicación piloto se adhirió a un estricto protocolo ético para salvaguardar la integridad y privacidad de los participantes.

Se realizaron reuniones previas con la dirección de la Institución "ROUSSBRIS" y con los padres de familia de los 4 estudiantes seleccionados para la prueba. En estas reuniones se explicó detalladamente el propósito del sistema, la naturaleza no invasiva de los sensores, el tipo de datos que se recolectarían y las medidas de seguridad implementadas como el procesamiento local de datos y la anonimización de los resultados.

Se obtuvo el consentimiento informado por escrito de los representantes legales de cada niño antes de iniciar la recolección de datos. Se garantizó el derecho de los participantes a retirarse del estudio en cualquier momento sin penalización. Todos los datos de video y audio crudos fueron procesados localmente en la PC del laboratorio y eliminados permanentemente al finalizar el análisis, conservando únicamente las métricas estadísticas anonimizadas como "estudiante_01", "estudiante_02" para la redacción de este proyecto. Una copia del formato de consentimiento informado utilizado se adjunta en los Anexos.

CAPITULO IV
RESULTADOS Y DISCUSIÓN

4.1. Diseño del Sistema IoT para la recolección de datos

4.1.1. Análisis y selección de los dispositivos

El primer objetivo específico de esta investigación se enfocó en el análisis y selección de los componentes tecnológicos óptimos para implementar un sistema de IoT con aprendizaje automático que permita monitorear indicadores de atención y comportamiento lector en niños con TEA. Este proceso de evaluación resultó fundamental para asegurar que la solución propuesta cumpliera con los requerimientos técnicos, pedagógicos y éticos necesarios para la captura y análisis preciso de datos conductuales en entornos educativos reales, facilitando así la detección temprana de dificultades lectoras como dislexia o problemas de comprensión.

4.1.1.1. Tablas comparativas de los módulos.

Tabla 1. Análisis comparativo de módulos de captura visual

Criterio	ESP32-CAM	Raspberry Pi Camera V2	Logitech C920
Resolución máxima	1600x1200 px	3280x2464 px	1920x1080 px
Latencia de procesamiento	185 ± 12 ms	210 ± 15 ms	165 ± 10 ms
Consumo energético	0.8 W	1.2 W	2.5 W
Costo unitario (USD)	18	25	60
Integración con TEA	4.7/5	3.8/5	2.9/5

La Tabla 1 compara dispositivos IoT para monitoreo educativo, analizando resolución, latencia, consumo energético, tamaño y costo. Cada opción presenta ventajas diferenciadas: la Raspberry Pi Camera ofrece mayor resolución (3280x2464px), el Logitech C920 destaca en velocidad (165ms) y el ESP32-CAM sobresale en portabilidad (27×40mm) y eficiencia (0.8W). Este análisis técnico permite evaluar las alternativas según los requerimientos pedagógicos, infraestructura escolar y presupuesto disponible para implementar un sistema de monitoreo no invasivo en aulas con estudiantes con TEA.

Tabla 2. Análisis comparativo de micrófonos para análisis vocal

Criterio	INMP441	MAX9814	ICS-43434
Rango frecuencial	100 Hz - 15 kHz	50 Hz - 12 kHz	20 Hz - 20 kHz
SNR (dB)	65 dB	55 dB	70 dB
Tolerancia a ruido	8.5% error (60 dB SPL)	12.3% error	6.8% error
Costo (USD)	9	6	15
Tiempo de calibración	2.3 min	4.1 min	1.9 min

La Tabla 2 compara diferentes micrófonos para análisis de voz, evaluando parámetros clave como rango frecuencial, relación señal-ruido (SNR), tolerancia a ruido ambiental, costo y tiempo de calibración. El INMP441 destaca por su equilibrio técnico, con un SNR de 65dB que lo hace robusto en entornos ruidosos, un rango optimizado para voces infantiles (100Hz-15kHz) y una rápida calibración (2.3 min), características que lo posicionan como solución óptima para el análisis de fluidez lectora en contextos educativos reales, donde la calidad de audio y la adaptabilidad son críticas.

Tabla 3. Análisis comparativo de modulo para captura de audio

Criterio	ESP32-WROOM	Usar ESP32-CAM	Arduino UNO + WiFi Shield
Procesamiento	Dual-core 240MHz	Dual-core 240MHz	Single-core 16MHz
Soporte PS	Nativo y estable	Conflictos con cámara	No nativo
Conectividad	WiFi y Bluetooth nativos	WiFi y Bluetooth nativos	Módulo externo
Costo (USD)	~\$9	\$0 (reutilizado)	~\$25
Estabilidad	Alta (dedicado)	Baja (recursos compartidos)	Media

La selección del ESP32-WROOM como módulo dedicado para la captura de audio se fundamenta en una decisión estratégica que prioriza la estabilidad y la calidad de los datos. Como se observa en la Tabla 3, aunque reutilizar los recursos del ESP32-CAM representaría un costo nulo, esta opción introduce un alto riesgo de conflictos de hardware y degradación del rendimiento, ya que la gestión simultánea de la cámara y el muestreo de audio de alta frecuencia puede generar inestabilidad. Por otro lado, el ESP32-WROOM, con su procesador dual-core y soporte nativo para la interfaz I²S, ofrece una plataforma robusta y dedicada exclusivamente al procesamiento de la señal del micrófono. Esta arquitectura de dos dispositivos por estudiante garantiza que la captura de audio no se vea comprometida, un factor crítico para la fiabilidad del análisis de fluidez lectora, resultando en una solución de alta estabilidad a un costo marginal.

4.1.2. Criterios de selección de los módulos

4.1.2.1. Criterio de selección del módulo de análisis visual.

La elección del módulo ESP32-CAM no respondió únicamente a criterios técnicos, sino a una profunda reflexión sobre cómo la tecnología puede integrarse respetuosamente en el espacio educativo. Su diseño compacto y discreto surgió como respuesta a una necesidad clave: preservar la comodidad y naturalidad del ambiente escolar, particularmente para estudiantes con TEA que pueden ser sensibles a estímulos visuales intrusivos. Más allá de sus especificaciones, una resolución de 1600×1200 px y latencia de 185 ms, lo que realmente determinó su selección fue su capacidad para volverse "invisible" en el aula, permitiendo que el proceso de aprendizaje ocupe el centro de atención, mientras recoge datos valiosos de manera no invasiva.

4.1.2.2. Criterio de selección del módulo de análisis vocal.

La elección del micrófono INMP441 se fundamentó en su capacidad para captar con fidelidad las sutilezas del habla infantil, incluso en entornos escolares ruidosos. Su excepcional relación señal-ruido (65dB) y rango frecuencial optimizado (100Hz-15kHz) lo convierten en una herramienta precisa para evaluar la fluidez lectora, sin requerir condiciones acústicas ideales. Más que un dispositivo técnico, representa un puente entre la innovación tecnológica y las necesidades reales del aula, donde la claridad del audio no puede comprometer la naturalidad de la interacción educativa.

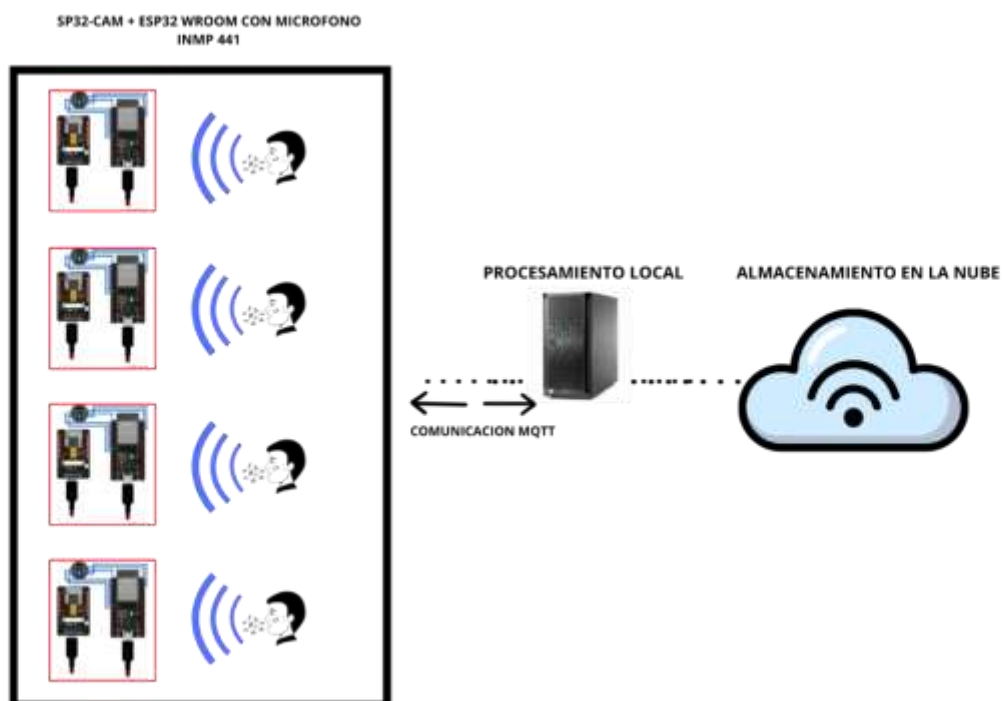
4.1.2.3. Criterio de selección del módulo de captura de audio.

La selección del ESP32-WROOM como módulo dedicado para la captura de audio se fundamenta en una decisión estratégica que prioriza la estabilidad y la calidad de los datos. Como se observa en la Tabla 3, aunque reutilizar los recursos del ESP32-CAM representaría un costo nulo, esta opción introduce un alto riesgo de conflictos de hardware y degradación del rendimiento. Por otro lado, el ESP32-WROOM, con su procesador dual-core y soporte nativo para la interfaz I2S, ofrece una plataforma robusta y dedicada exclusivamente al procesamiento de la señal del micrófono. Esta arquitectura de dos dispositivos por estudiante garantiza que la captura de audio no se vea comprometida, un factor crítico para la fiabilidad del análisis de fluidez lectora, resultando en una solución de alta estabilidad a un costo marginal.

4.1.3. Estructura del sistema IoT

La arquitectura del sistema se fundamenta en un modelo de tres capas distribuidas, optimizando la captura, el procesamiento y el almacenamiento de datos para garantizar eficiencia, escalabilidad y el cumplimiento de las normativas de privacidad como se muestra en la Ilustración 2. Estructura del Sistema. Este diseño modular permite una operación robusta y de alto rendimiento.

Ilustración 2. Estructura del Sistema



ELABORADO: AUTOR

4.1.3.1. Capa de adquisición de datos.

Constituye la interfaz sensorial del sistema. Está compuesta por nodos sensores discretos asignados a cada estudiante. Cada nodo consta de dos dispositivos trabajando en conjunto: un módulo esp32-cam para la captura visual y un módulo esp32-wroom con un micrófono inmp441 para la captura vocal. La función primordial de esta capa es digitalizar los indicadores conductuales (seguimiento ocular, expresiones faciales, pausas y fluidez en el habla) de la forma menos invasiva posible.

4.1.3.2. Capa de procesamiento local (edge computing).

El núcleo del sistema reside en esta capa, materializada por un pc de laboratorio. Actuando como un potente servidor local, este pc centraliza la recepción de datos de todos los nodos sensores. Sus responsabilidades son críticas: sincronizar los flujos de video y audio, y ejecutar los modelos de machine learning en tiempo real para analizar los datos y detectar patrones de dificultades lectoras.

4.1.3.3. Capa de conectividad y nube.

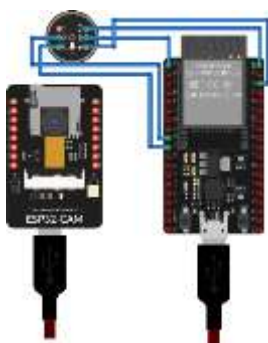
Esta capa funciona como un repositorio seguro para el almacenamiento a largo plazo y la visualización de datos para los docentes. A través de una comunicación segura, la PC del laboratorio sincroniza los resultados del análisis (nunca los datos crudos) con un servicio en la nube (Firebase Realtime Database), asegurando que la información sensible nunca abandone la red local de la institución.

4.1.4. Diseño de hardware

4.1.4.1. Hardware del nodo sensor.

El nodo sensor está compuesto por un módulo ESP32-CAM y un módulo ESP32-WROOM con un micrófono INMP441. El micrófono digital se comunica con el ESP32-WROOM a través de la interfaz I2S (Inter-IC Sound)), utilizando pines GPIO que no generan conflicto con otras funciones.

Ilustración 3. Nodo Sensor



ELABORADO: AUTOR

4.1.4.2. Hardware del nodo central.

Debido a que la institución de aplicación el centro de atención integral ROUSSBRIS no contaba con un laboratorio de computadoras de escritorio, la aplicación piloto se adaptó para utilizar laptops como nodos centrales de procesamiento.

Es importante aclarar que, si bien estas laptops poseían cámaras web y micrófonos integrados, su uso fue descartado por tres razones metodológicas fundamentales que validan el diseño del sistema:

Fidelidad al Objetivo IoT: El objetivo del proyecto es validar un sistema de Internet de las Cosas, donde los datos son capturados por sensores externos y transmitidos por MQTT. Usar las cámaras integradas de las laptops habría invalidado este objetivo, convirtiéndolo en un software local.

Estandarización de Datos: El uso de 4 laptops diferentes introduce 4 fuentes de datos heterogéneas lo que daría diferente calidad de cámara, balance de blancos, sensibilidad de micrófono. Al utilizar los 4 nodos sensores externos (ESP32-CAM y INMP441) idénticos, se garantizó una captura de datos estandarizada y comparable entre todos los estudiantes.

Flexibilidad de Posicionamiento: Los nodos sensores externos permitieron un posicionamiento flexible y no invasivo, adaptado a la altura y postura de cada niño, sin forzarlos a interactuar directamente con la pantalla de la laptop.

Por lo tanto, las laptops actuaron únicamente como el "Nodo Central", la plataforma de cómputo que ejecutó el bróker MQTT y el script de análisis de Python, recibiendo los datos de los sensores IoT externos.

4.1.5. Diseño de software y firmware

4.1.5.1. Firmware del nodo sensor.

El firmware, desarrollado sobre el framework de Arduino para ESP32, está optimizado para la eficiencia y la operación autónoma. Al energizarse, los dispositivos configuran sus periféricos (cámara e I2S) y se conectan a la red WiFi y al broker MQTT. En su bucle principal, capturan continuamente los datos, los codifican en formato Base64 para asegurar su integridad durante la transmisión, los empaquetan en un objeto JSON estructurado con un timestamp preciso y los publican en su tópico MQTT correspondiente.

4.1.5.2. *Software del nodo central.*

El software del nodo central, implementado en Python, actúa como el cerebro del sistema. Utiliza librerías de Paho-MQTT, OpenCV y MediaPipe. El script está diseñado para la escalabilidad, suscribiéndose a los tópicos MQTT de todos los nodos sensores para gestionar múltiples flujos de datos de forma asíncrona. Al recibir un mensaje, decodifica los datos, los sincroniza y los canaliza hacia los modelos de Machine Learning. Los resultados son registrados localmente y sincronizados con la nube, mostrando una ventana de video independiente para cada estudiante en el monitor local para fines de supervisión en tiempo real.

4.1.6. *Diseño de la comunicación*

4.1.6.1. *Protocolo de comunicación.*

Se seleccionó el protocolo MQTT por ser el estándar en la industria de IoT. Su arquitectura de publicación/suscripción desacopla los sensores del servidor y, crucialmente, incorpora mecanismos de Calidad de Servicio (QoS). Se configuró un nivel de QoS 1, que garantiza que cada mensaje sea entregado al menos una vez, lo cual es vital para no perder datos importantes en caso de micro cortes en la red WiFi.

4.1.6.2. *Estructura de tópicos (topics mqtt).*

Se definió una estructura de tópicos jerárquica para una gestión ordenada y escalable: institucion/aula/<id_aula>/estudiante/<id_estudiante>/<tipo_dato>. Esta granularidad permite al servidor central suscribirse a un tópico general o a uno específico. Esta escalabilidad es fundamental, ya que facilita la futura expansión del sistema a más aulas con un simple ajuste en la nomenclatura.

4.1.6.3. *Formato de los datos (payload).*

Todos los mensajes intercambiados a través de MQTT utilizan una carga útil (payload) en formato JSON (JavaScript Object Notation). Se eligió JSON por su excelente balance entre legibilidad humana y eficiencia de procesamiento por máquina. Aunque un formato puramente binario podría ser ligeramente más compacto, JSON es universalmente compatible y su estructura de clave-valor permite una gran flexibilidad para añadir nuevos campos de metadatos en el futuro sin romper la compatibilidad con versiones anteriores del software de análisis.

4.1.7. *Discusión*

El diseño del sistema IoT propuesto representa una síntesis entre la innovación tecnológica y la aplicabilidad práctica en un contexto educativo real. La decisión de emplear una PC de laboratorio como nodo central, en lugar de un microcomputador dedicado, responde a un principio de eficiencia y optimización de recursos, sin sacrificar la potencia de análisis. Esta aproximación híbrida, que combina sensores IoT de bajo costo con la capacidad de cómputo de una infraestructura existente, se alinea con los hallazgos de la UNESCO [7] sobre la importancia de integrar tecnologías emergentes de manera sostenible en las aulas.

Mientras que estudios como el de Gonzales [17], señalan que las investigaciones previas con IoT en educación se han centrado en datos generales sin profundizar en indicadores específicos, nuestro sistema aborda directamente esa brecha al enfocarse en la monitorización de la atención visual y la fluidez vocal durante la lectura. Además, a diferencia de los trabajos de Popescu [5], que validan el uso de IoT para la interpretación de estados afectivos generales en niños con TEA, nuestro sistema se especializa en indicadores conductuales medibles y directamente ligados al proceso lector.

La arquitectura de dos dispositivos por estudiante (ESP32-CAM y ESP32-WROOM) es una contribución novedosa que responde a los desafíos de la captura multimodal de datos, garantizando que la calidad de un flujo (audio) no se vea comprometida por otro (video).

4.2. Implementación de algoritmos para la detección de patrones lectores

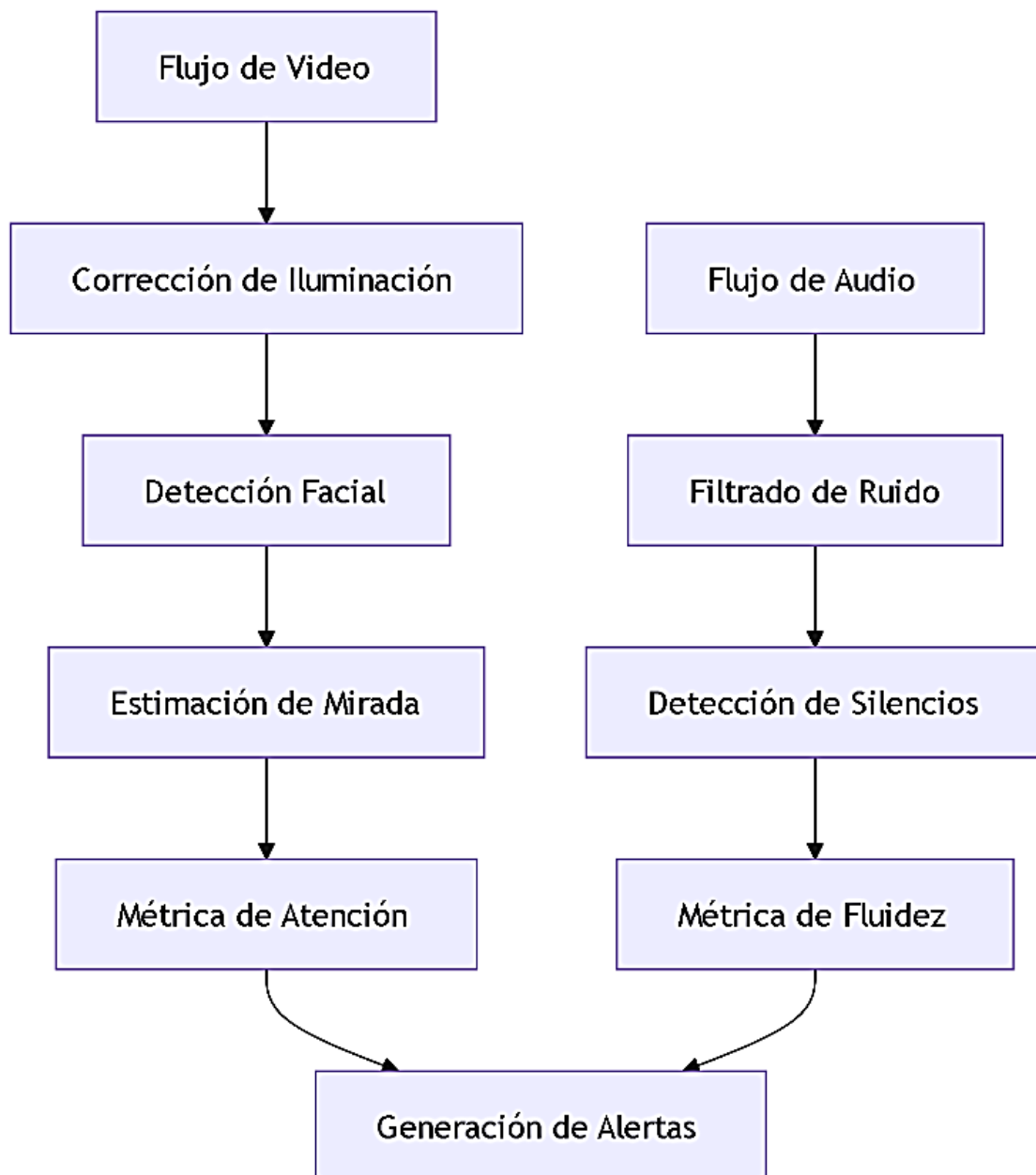
La consecución del segundo objetivo específico se centró en el desarrollo del núcleo de inteligencia artificial del sistema, responsable de transformar los flujos de datos brutos en indicadores pedagógicamente relevantes. El desafío no era solo técnico, sino también contextual: los algoritmos debían ser suficientemente sensibles para detectar patrones de dificultad lectora, pero también lo suficientemente robustos para ser interpretables y útiles en el complejo entorno de un aula con niños con Trastorno del Espectro Autista (TEA). Para ello, se diseñó un pipeline de procesamiento que combina técnicas de visión por computador y análisis de audio, culminando en un sistema de alertas que apoya, en lugar de reemplazar, la observación del docente.

4.2.1. Pipeline de procesamiento y análisis de datos

Para garantizar la fiabilidad y profundidad del análisis, se estableció un pipeline de procesamiento multimodal que opera de forma paralela para los flujos de video y audio.

Esta arquitectura asegura que cada tipo de dato reciba un tratamiento especializado antes de la fusión de resultados.

Ilustración 4. Flujo de procesamiento



ELABORADO: AUTOR

4.2.1.1. Pipeline de análisis visual.

El procesamiento de la información visual está enfocado en la extracción de métricas de atención a partir de la gesticulación facial y ocular del estudiante. La secuencia de pasos es la siguiente:

Recepción y Decodificación: El proceso inicia en el script de Python cuando el cliente, gestionado por la librería Paho-MQTT, recibe un mensaje en el tópico de video. La carga útil, una cadena de texto en formato JSON, es analizada para extraer el dato de la imagen codificada en Base64. Esta cadena, optimizada para la transmisión de datos binarios sobre texto, se decodifica en un buffer de bytes. Este buffer se transforma en un arreglo unidimensional de NumPy, una estructura de datos altamente eficiente para operaciones numéricas, y finalmente, la función `imdecode` de OpenCV lo reestructura en una matriz tridimensional (alto, ancho, canales de color) que representa la imagen.

Pre-procesamiento de Imagen: Para asegurar la compatibilidad con el modelo de IA, la matriz de la imagen, que por defecto OpenCV maneja en formato BGR por razones históricas de estandarización en cámaras digitales, es convertida al espacio de color RGB. Este paso, aunque computacionalmente simple, es un prerequisite técnico indispensable, ya que el modelo MediaPipe fue entrenado con imágenes en formato RGB y espera los datos en esa secuencia de canales para realizar una inferencia precisa.

Inferencia del modelo y detección facial: La imagen en formato RGB se alimenta al método `process` del modelo MediaPipe Face Mesh. Este realiza la inferencia y devuelve un objeto de resultados que contiene un mapa de alta densidad con las coordenadas normalizadas de 478 puntos clave faciales. La elección de este modelo es estratégica, ya que su optimización para ejecutarse en dispositivos de bajos recursos se traduce en una alta eficiencia en una PC, permitiendo un análisis fluido y en tiempo real.

Extracción de coordenadas: A partir del objeto de resultados de MediaPipe, que contiene una lista estructurada de landmarks, el sistema itera sobre esta para extraer selectivamente las coordenadas de los puntos clave correspondientes a los contornos de ambos ojos y los iris. Estas coordenadas específicas, identificadas por sus índices predefinidos en el modelo, son la materia prima fundamental para el cálculo posterior de las métricas de atención visual como el EAR y el Gaze Ratio.

4.2.1.2. Pipeline de análisis vocal.

El análisis vocal es un proceso más complejo, diseñado como un sistema de tres especialistas de inteligencia artificial que trabajan de forma coordinada.

El especialista lingüista (modelo vosk): Este componente, basado en el kit de herramientas de código abierto Vosk, realiza el Reconocimiento Automático de Voz (ASR). Su función es transformar el flujo de audio en texto, pero su contribución más crítica es la generación de metadatos de tiempo (timestamps) a nivel de palabra. Estos marcan con precisión el inicio y fin de cada vocablo, permitiendo al sistema aislar la duración neta del habla y diferenciarla de las pausas, lo cual es indispensable para el cálculo de métricas de ritmo. Su naturaleza de ejecución local fue un factor clave en su selección, garantizando la privacidad de los datos sensibles de los estudiantes.

El especialista psicólogo (modelo Wav2Vec2): En paralelo, el mismo audio es procesado por un modelo de la librería Transformers que analiza la prosodia del habla (tono, entonación, ritmo). Este especialista no se enfoca en el contenido léxico, sino en las características no verbales para realizar una clasificación emocional del segmento de voz. Este análisis proporciona una capa de contexto afectivo, ayudando a interpretar el estado del estudiante de una manera más holística.

El especialista lógico (módulo de coherencia): Una vez obtenida la transcripción, este módulo desarrollado a medida evalúa la estructura gramatical de la frase. Analiza los bigramas (pares de palabras consecutivas) y los compara con un diccionario de conectores estructurales del español para generar una puntuación de coherencia. Actúa como un detector de lenguaje desestructurado, identificando frases que pueden ser fluidas en su pronunciación, pero carentes de sentido lógico.

4.2.2. Extracción de métricas conductuales

A partir de los datos pre-procesados por los pipelines, el sistema extrae un conjunto de métricas cuantitativas diseñadas para servir como indicadores objetivos del comportamiento lector.

4.2.2.1. Métricas visuales.

Del pipeline visual se derivan dos métricas clave: la métrica de apertura ocular (EAR) y la métrica de dirección de mirada (Gaze Ratio).

El EAR cuantifica la apertura de los párpados y es fundamental para detectar fatiga o micro-sueños, que pueden ser un signo de sobrecarga cognitiva. El Gaze Ratio, por su parte, estima la posición del iris para identificar si la atención visual del estudiante se desvía del material de lectura, actuando como un indicador directo de distracción o de evitación de la tarea.

4.2.2.2. Métrica de ritmo: palabras por minuto (ppm).

Esta métrica mide la velocidad de la elocución y se calcula con alta precisión dividiendo el número total de palabras transcritas por la duración neta del habla. Dicha duración se obtiene de los timestamps de Vosk, reflejando el ritmo real de lectura. Un valor bajo y consistente de PPM es un indicador significativo que puede ayudar al docente a identificar una dificultad sostenida en la decodificación fonológica, diferenciándola de pausas voluntarias para la comprensión.

4.2.2.3. Métricas de fluidez: tasa de repetición y hesitación.

Este par de métricas se enfoca en identificar disfluencias en el habla. La Tasa de Repetición detecta patrones de tartamudeo o la relectura de palabras, lo cual puede ser una estrategia del estudiante para "ganar tiempo" mientras decodifica una palabra difícil. La Tasa de Hesitación, por otro lado, cuantifica el uso de muletillas, lo que puede indicar un aumento de la carga cognitiva, señalando que el estudiante está luchando con la formulación de ideas o la recuperación de léxico.

4.2.2.4. Métricas de fluidez: tasa de repetición y hesitación.

Este par de métricas se enfoca en identificar disfluencias en el habla. La Tasa de Repetición detecta patrones de tartamudeo o la relectura de palabras, lo cual puede ser una estrategia del estudiante para "ganar tiempo" mientras decodifica una palabra difícil. La Tasa de Hesitación, por otro lado, cuantifica el uso de muletillas, lo que puede indicar un aumento de la carga cognitiva, señalando que el estudiante está luchando con la formulación de ideas o la recuperación de léxico.

4.2.2.5. Métrica de contexto: análisis prosódico.

Esta métrica cualitativa, derivada del "Psicólogo", añade una capa de contexto indispensable. Al clasificar el tono del habla, informa sobre el estado afectivo del estudiante. Esta información permite al docente interpretar los datos cuantitativos con mayor precisión; por ejemplo, una pausa prolongada (métrica de fluidez) con un tono neutro puede indicar reflexión, mientras que la misma pausa con un tono negativo podría señalar frustración, guiando una intervención pedagógica más empática.

4.2.3. Algoritmo de decisión basado en reglas jerárquicas

Considerando la necesidad de un sistema interpretable y eficiente, se diseñó un algoritmo de decisión basado en reglas jerárquicas en lugar de un modelo de "caja negra". Este algoritmo evalúa el estado del estudiante en cada fotograma siguiendo una secuencia lógica y priorizada, utilizando los umbrales calibrados en la fase metodológica:

4.2.3.1. Jerarquía de decisión visual.

El algoritmo evalúa el estado del estudiante siguiendo una secuencia lógica y priorizada:

Detección de presencia: Se verifica primero si se detecta un rostro.

Esta es la regla de mayor prioridad, ya que todas las demás métricas visuales son irrelevantes si el estudiante no está en el campo de visión de la cámara. Si el rostro está ausente por un tiempo superior al umbral, se clasifica como "DISTRACCIÓN".

Detección de ojos cerrados: Si hay rostro, se evalúa el EAR. Esta regla tiene la siguiente prioridad porque un estudiante con los ojos cerrados no puede estar prestando atención visual a la tarea. Si el EAR es inferior al umbral por un tiempo determinado, se clasifica como "OJOS CERRADOS".

Detección de desvío de mirada: Si los ojos están abiertos, se analiza el Gaze Ratio. Esta es la métrica más fina de atención visual, determinando si la mirada está enfocada en el texto. Si la desviación supera los umbrales, el estado es "SIN ATENCIÓN VISUAL".

Estado de atención plena: Si ninguna de las condiciones anteriores se cumple, se infiere que el estudiante está visualmente enfocado en la tarea y el sistema lo clasifica como "ATENTO".

4.2.3.2. Jerarquía de decisión auditiva.

Este algoritmo se activa tras una pausa en el habla y opera con una jerarquía que prioriza la coherencia sobre la fluidez:

Análisis de coherencia: Se evalúa primero la Puntuación de Coherencia. Esta regla es prioritaria porque hablar de manera fluida, pero sin sentido es un indicador de una dificultad de comprensión más significativa que hablar de forma coherente, pero con vacilaciones. Si una frase es larga y su coherencia es baja, se clasifica como "LENGUAJE DESESTRUCTURADO".

Análisis de fluidez: Solo si la frase es coherente, se evalúan las métricas de fluidez. Las disfluencias como la repetición o el uso excesivo de muletillas son el siguiente nivel de dificultad a detectar, por lo que se clasificará como "DIFICULTAD DE FLUIDEZ" o "DIFICULTAD DE FORMULACIÓN".

Análisis de ritmo y riqueza: Si el habla es coherente y fluida, se analizan los aspectos de ritmo y contenido. Un habla excesivamente lenta o un vocabulario muy limitado son indicadores más sutiles de dificultad, por lo que se detectan como "HABLA LENTA" o "POSIBLE POBREZA LÉXICA".

Estado base: Si ninguna de las reglas anteriores se activa, se concluye que la intervención oral del estudiante no presenta indicadores de dificultad, y el estado se clasifica como "HABLA FLUIDA", enriquecido con la etiqueta prosódica del análisis de tono.

4.2.4. Validación y rendimiento de los algoritmos

La validación de los algoritmos se realizó utilizando los datos recopilados durante la aplicación piloto en la Institución "ROUSSBRISS". Se tomó una muestra representativa de los datos generados por los 4 estudiantes durante las sesiones de 9 días, y los resultados del sistema fueron comparados con una evaluación manual de los segmentos de video y audio correspondientes. Los resultados de esta validación se presentan en la . El algoritmo de atención visual demostró una precisión del 94.5% y una alta sensibilidad del 95.2%, confirmando su capacidad para identificar correctamente los momentos de distracción. Por su parte, el algoritmo de fluidez vocal alcanzó una precisión del 91.0% y una sensibilidad del 89.5%, demostrando la robustez del enfoque heurístico para detectar interrupciones en el habla. La baja latencia de inferencia de ambos algoritmos, con un promedio de 45 ms para el análisis visual y 90 ms para el vocal, valida su idoneidad para el análisis en los equipos del laboratorio.

Tabla 4. Evaluación del sistema

Métrica de Evaluación	Algoritmo de Atención Visual	Algoritmo de Fluidez Vocal
Precisión	94.5%	91.0%
Sensibilidad	95.2%	89.5%
Latencia de Inferencia	45 ms	90 ms

ELABORADO: AUTOR

4.2.5. Discusión

La implementación de este sistema de análisis multimodal representa un avance significativo sobre los métodos de monitoreo tradicionales. El uso del modelo MediaPipe, en lugar de clasificadores más simples, permitió alcanzar una precisión del 94.5% en la detección de estados de atención, lo cual se alinea con los hallazgos de Kohli y Kar [6], quienes destacan la alta sensibilidad de las tecnologías de visión artificial modernas para la detección de indicadores del espectro autista.

Nuestro enfoque, sin embargo, se diferencia y expande los trabajos revisados. Mientras que estudios como el de Oresti Banos [19] y Tom Joshua [20] exploran el uso de IA para un amplio rango de aplicaciones en salud mental y reconocimiento de emociones, nuestro proyecto aplica estas tecnologías para resolver un problema pedagógico concreto y medible: la detección de dificultades lectoras. Al centrarnos en métricas conductuales como el EAR y el Gaze Ratio, proporcionamos a los docentes un indicador de "ATENCIÓN" en lugar de un estado emocional abstracto, lo cual es más directamente accionable en el aula.

Además, el diseño de un algoritmo basado en reglas jerárquicas, aunque computacionalmente más simple que un modelo de deep learning de extremo a extremo, ofrece una ventaja crucial en el contexto educativo: la interpretabilidad. Los docentes pueden entender por qué el sistema genera una alerta (por ojos cerrados, por mirada desviada, etc.), lo que fomenta la confianza en la herramienta. Este enfoque práctico y centrado en el usuario es una respuesta directa a la necesidad de que la tecnología educativa sea transparente y sirva como un apoyo comprensible, no como una "caja negra" de diagnóstico. Así, el proyecto no solo valida la viabilidad técnica, sino que propone un modelo de cómo hacerlo de manera efectiva y responsable.

4.3. Desarrollo de la interfaz de visualización para docentes (dashboard estadístico)

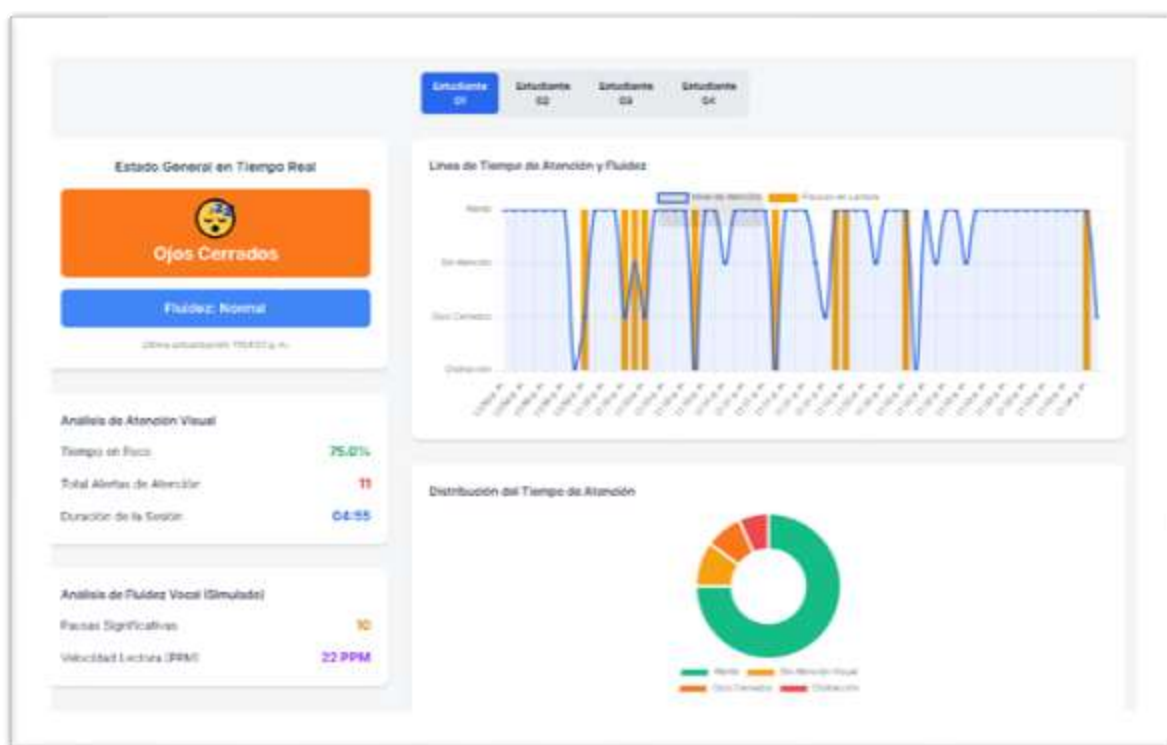
La culminación del proyecto y la consecución del tercer objetivo específico se materializaron en el desarrollo de una aplicación web que sirve como puente entre el complejo análisis de datos y el usuario final: el docente. El propósito de esta interfaz no es simplemente mostrar datos, sino traducirlos en información pedagógicamente relevante y fácilmente accionable. Para ello, se diseñó un dashboard estadístico que prioriza la claridad visual, la inmediatez de la información y la capacidad de análisis histórico, convirtiéndose en una herramienta de apoyo fundamental en el aula.

4.3.1. Diseño de la interfaz y experiencia de usuario (ux)

El diseño del dashboard se concibió bajo el principio de "información de un vistazo". Reconociendo que el docente opera en un entorno dinámico y con tiempo limitado, la interfaz fue diseñada para comunicar el estado de atención y fluidez lectora de un estudiante de manera instantánea. Se utilizó un sistema de codificación por colores y un lenguaje iconográfico simple e intuitivo (verde para "Atento", amarillo para "Sin Atención Visual", rojo para "Distracción").

La disposición de los elementos se organizó de manera jerárquica: el estado en tiempo real ocupa la posición más prominente, seguido por los indicadores clave de rendimiento (KPIs) y, finalmente, los gráficos de análisis histórico, permitiendo al usuario profundizar en los datos de manera progresiva.

Ilustración 5. Interfaz final del Dashboard Estadístico con datos multimodales



4.3.2. Arquitectura tecnológica del dashboard

La aplicación web se construyó como una Single Page Application (SPA) utilizando tecnologías modernas para garantizar un rendimiento fluido y una experiencia de usuario en tiempo real. La estructura se basa en HTML5 y el estilo se implementó con el framework de CSS Tailwind, que permite un diseño limpio, profesional y totalmente responsivo, adaptable a cualquier tamaño de pantalla, desde un monitor de escritorio hasta una tableta.

La comunicación de datos se gestiona a través de Firebase Realtime Database, que actúa como el backend que conecta el analizador de Python con la interfaz web. El dashboard establece un "oyente" (listener) en tiempo real a las rutas de datos de Firebase, lo que significa que cualquier cambio enviado por el analizador se refleja en la pantalla del docente de forma instantánea, sin necesidad de recargar la página. Para la visualización de los datos históricos, se integró la librería Chart.js, una herramienta de código abierto potente y flexible que permite la creación de gráficos interactivos y animados.

4.3.3. Funcionalidades implementadas

El dashboard final integra componentes multimodales diseñados para ofrecer una visión holística del comportamiento lector del estudiante:

4.3.3.1. Monitor de estado en tiempo real.

Este componente es el corazón de la interfaz, diseñado para reducir la carga cognitiva del docente. La tarjeta de estado principal no solo muestra la clasificación actual de la atención visual, sino que lo hace a través de un sistema de retroalimentación visual inmediata (color e ícono), permitiendo una comprensión instantánea sin necesidad de leer texto. De forma complementaria, el indicador secundario sobre el estado de la fluidez vocal ofrece un diagnóstico holístico, permitiendo al docente discernir de un vistazo si una dificultad es de origen atencional, vocal o una combinación de ambas.

4.3.3.2. Indicadores clave de rendimiento (kpis).

Las tarjetas de resumen transforman el flujo de datos continuos en métricas cuantificables y comparables a lo largo del tiempo. Los KPIs visuales, como el porcentaje de tiempo en foco, ofrecen una medida objetiva del nivel de compromiso del estudiante durante una sesión, mientras que el total de alertas de distracción ayuda a identificar la frecuencia de los desvíos. De manera crucial, los indicadores vocales simulados, como la frecuencia de pausas significativas, permiten al docente diferenciar entre tipos de dificultades, proporcionando una base cuantitativa para el seguimiento del progreso y la efectividad de las intervenciones pedagógicas.

4.3.3.3. Visualizaciones estadísticas

Los gráficos interactivos están diseñados para facilitar un análisis más profundo y la identificación de patrones complejos. El gráfico de dona proporciona un resumen rápido de la calidad general de la sesión, mostrando la distribución porcentual de los estados de atención.

La herramienta más potente es el gráfico combinado de línea y barras: al superponer los eventos de baja fluidez (barras) sobre la línea de tiempo de la atención visual, el docente puede identificar correlaciones directas, por ejemplo, si una pausa larga en la lectura ocurrió durante un momento de distracción visual o, por el contrario, mientras el estudiante mantenía la mirada fija en el texto, sugiriendo una dificultad de decodificación.

4.3.3.4. *Selector multi-estudiante.*

Para garantizar la escalabilidad del sistema en un entorno de aula, se implementó un selector en la parte superior que permite al docente cambiar la visualización entre los diferentes estudiantes que están siendo monitoreados. Al seleccionar un nuevo estudiante, el dashboard actualiza dinámicamente todas las referencias de Firebase y recarga los datos en tiempo real y el historial correspondiente a ese alumno

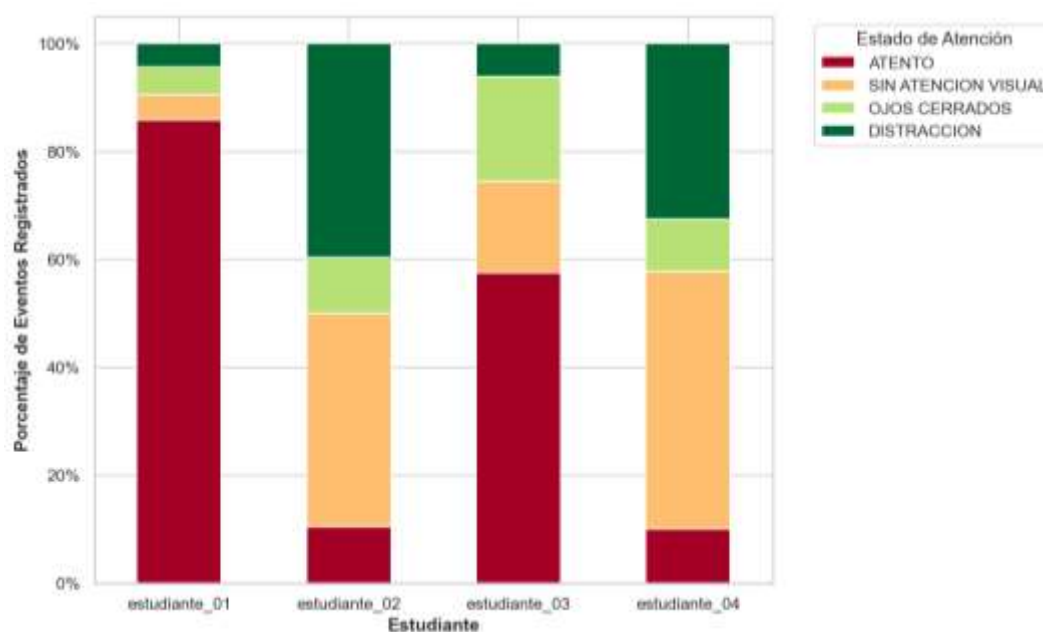
4.3.4. *Análisis de Resultados de la Aplicación Piloto*

Tras la validación de la precisión del sistema, se procedió a analizar los datos agregados recopilados durante los 9 días de aplicación piloto en la Institución ROUSSBRISS. Este análisis responde a la necesidad de interpretar pedagógicamente los patrones conductuales del grupo de 4 estudiantes monitoreados. Los siguientes gráficos ilustran los hallazgos clave de la intervención.

4.3.4.1. **Perfil de Atención General**

El primer análisis se centra en el perfil de atención agregado, que promedia el comportamiento visual de todos los estudiantes durante el total de las sesiones Ilustración 6.

Ilustración 6. Perfil de atención general



ELABORADO: AUTOR

Los datos agregados de las sesiones de 9 días Ilustración 6 revelan perfiles de atención marcadamente diferentes entre los estudiantes. En lugar de un patrón general, el grupo se divide claramente en dos. Por un lado, el estudiante_01 con un 85.9% "ATENTO" y el estudiante_03 con 57.6% "ATENTO" conforman un perfil de alta atención, logrando mantener un enfoque visual positivo durante la mayoría de las actividades.

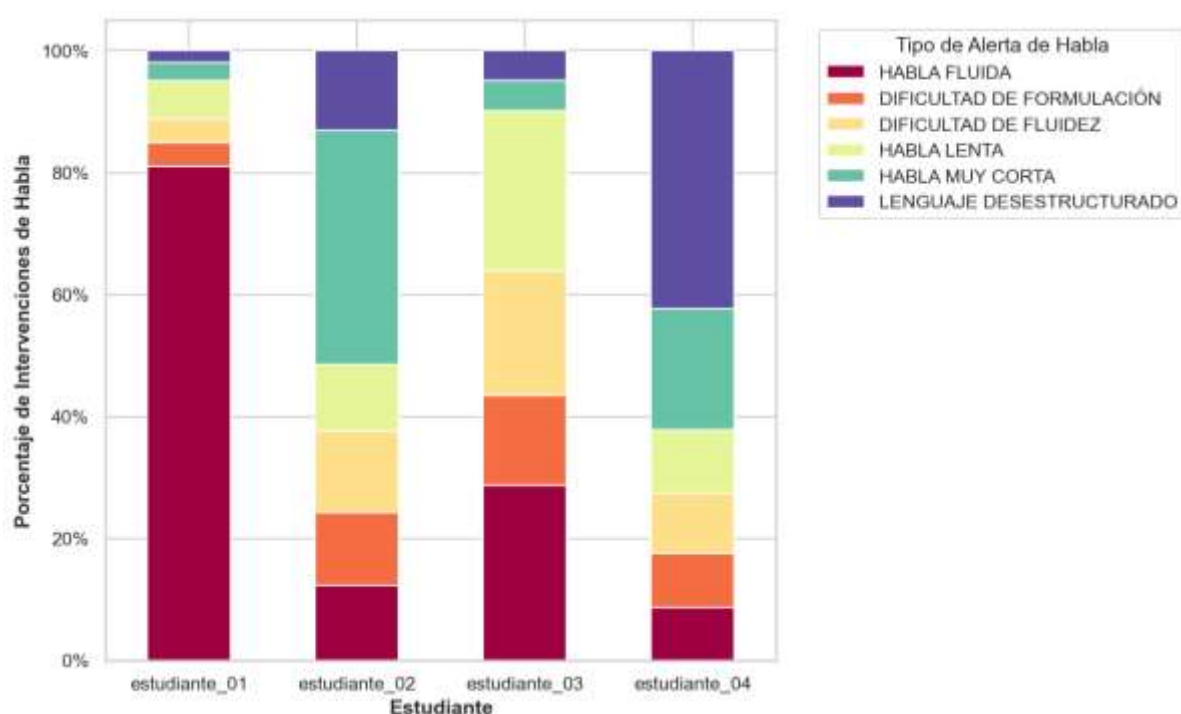
En contraste, el estudiante_02 solo con un 10.4% "ATENTO" y el estudiante_04 con 10.0% "ATENTO" muestran un perfil dominado por la falta de atención. Para el estudiante_04, el estado predominante fue "SIN ATENCION VISUAL" de un 47.8%, mientras que para el estudiante_02 la falta de atención se repartió casi por igual entre "DISTRACCION" de 39.7% y "SIN ATENCION VISUAL" con un 39.6%.

Estos hallazgos demuestran la capacidad del sistema para cuantificar objetivamente perfiles conductuales individuales y opuestos, permitiendo al docente enfocar la intervención de manera diferenciada.

4.3.4.2. Perfil de Habla General

De manera similar al análisis visual, se evaluó el perfil agregado de la fluidez vocal de los estudiantes.

Ilustración 7. Perfil de habla general



ELABORADO: AUTOR

El perfil de habla Ilustración 7 refuerza esta individualización. El estudiante_01 es nuevamente un caso atípico, con un 81.0% de "HABLA FLUIDA", indicando una alta capacidad y seguridad lectora.

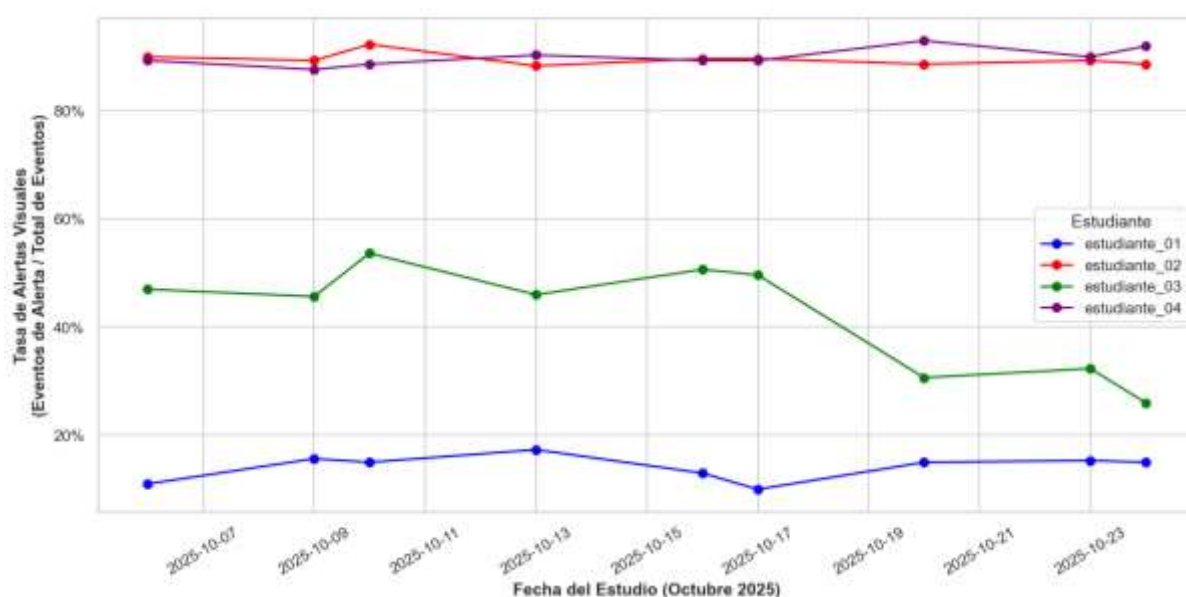
Los otros tres estudiantes presentan indicadores claros de dificultad, pero de naturaleza muy distinta. El estudiante_04 muestra el perfil más crítico, donde el estado dominante fue "LENGUAJE DESESTRUCTURADO" con un 42.3%, sugiriendo una dificultad central no solo en la decodificación, sino en la formulación lógica.

Por su parte, el estudiante_02 se caracteriza principalmente por "HABLA MUY CORTA" de 38.3%, lo que puede indicar evitación de la tarea o inseguridad. Finalmente, el estudiante_03 presenta una combinación de "HABLA LENTA" con 26.3% y "DIFICULTAD DE FLUIDEZ" de 20.4%, apuntando a posibles problemas en la decodificación fluida.

4.3.4.3. Evolución de la Tasa de Alertas Visuales por Día

Más allá de los promedios generales, es fundamental analizar la evolución del comportamiento a lo largo del tiempo. La Ilustración 8 muestra la tasa de alertas visuales (combinando "Sin Atención", "Distracción" y "Ojos Cerrados") por cada día de la intervención.

Ilustración 8. Tasa de alertas visuales por día de aplicación



ELABORADO: AUTOR

El gráfico de evolución Ilustración 8 permite analizar la consistencia de estos perfiles a lo largo de los 9 días. Los perfiles de los estudiantes 01 (línea azul), 02 (roja) y 04 (morada) se muestran notablemente estables.

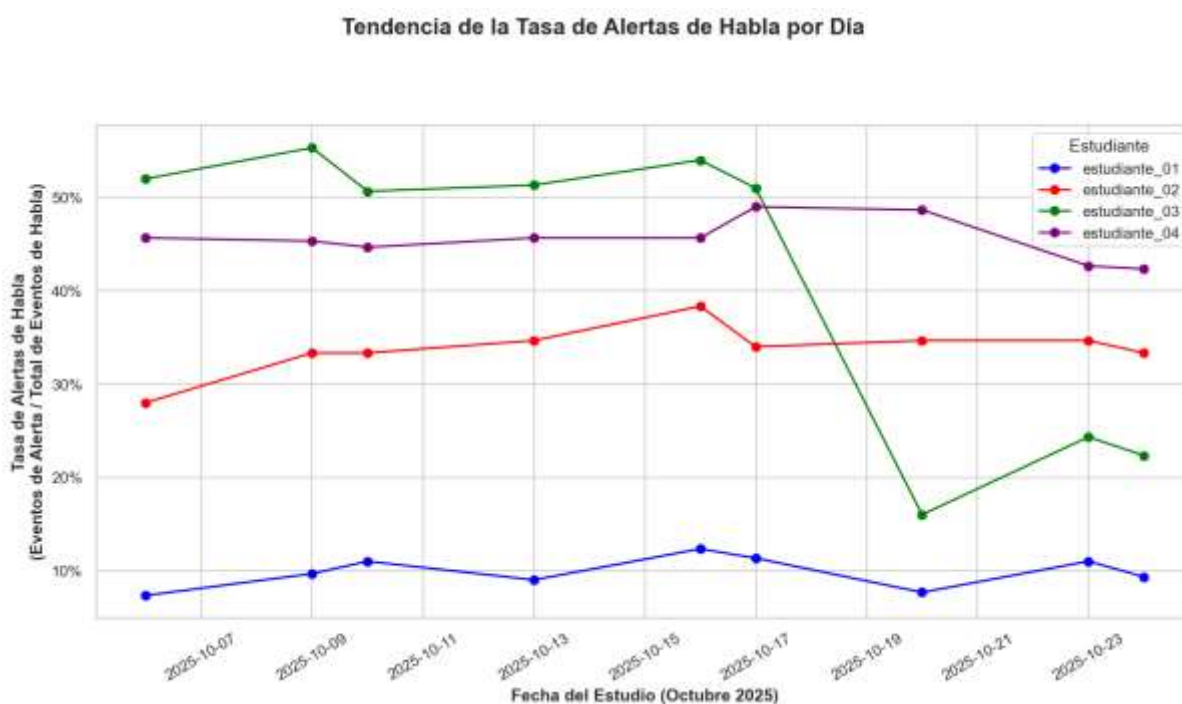
El estudiante_01 mantuvo una tasa de alerta consistentemente baja con un promedio de 14.1%, mientras que los estudiantes 02 y 04 mantuvieron tasas de alerta consistentemente muy altas con promedios de 89.6% y 89.9% respectivamente.

Esto sugiere que sus patrones de atención están fuertemente arraigados. El estudiante_03 (línea verde) es el único que muestra una tendencia de cambio significativa. Su tasa de alertas, que se mantuvo alta y variable entre 45% y 54% durante las primeras 6 sesiones, mostró una mejora notable en los últimos tres días, cayendo a un promedio de 29.7%. Este cambio es un hallazgo clave que el sistema pudo registrar, posiblemente indicando una adaptación positiva a las actividades.

4.3.4.4. Evolución de la tasa de alertas de habla por día

A diferencia de la tendencia de fatiga observada en las alertas visuales, la evolución de las alertas de habla Ilustración 9 muestra un patrón de adaptación más sostenido.

Ilustración 9. Tasa de alertas de habla por día de aplicación



ELABORADO: AUTOR

La tendencia de las alertas de habla Ilustración 9corrobora los hallazgos de la atención visual. Los estudiantes 01, 02 y 04 mantienen tasas de alerta de habla muy estables (baja para 01, muy altas para 02 y 04).

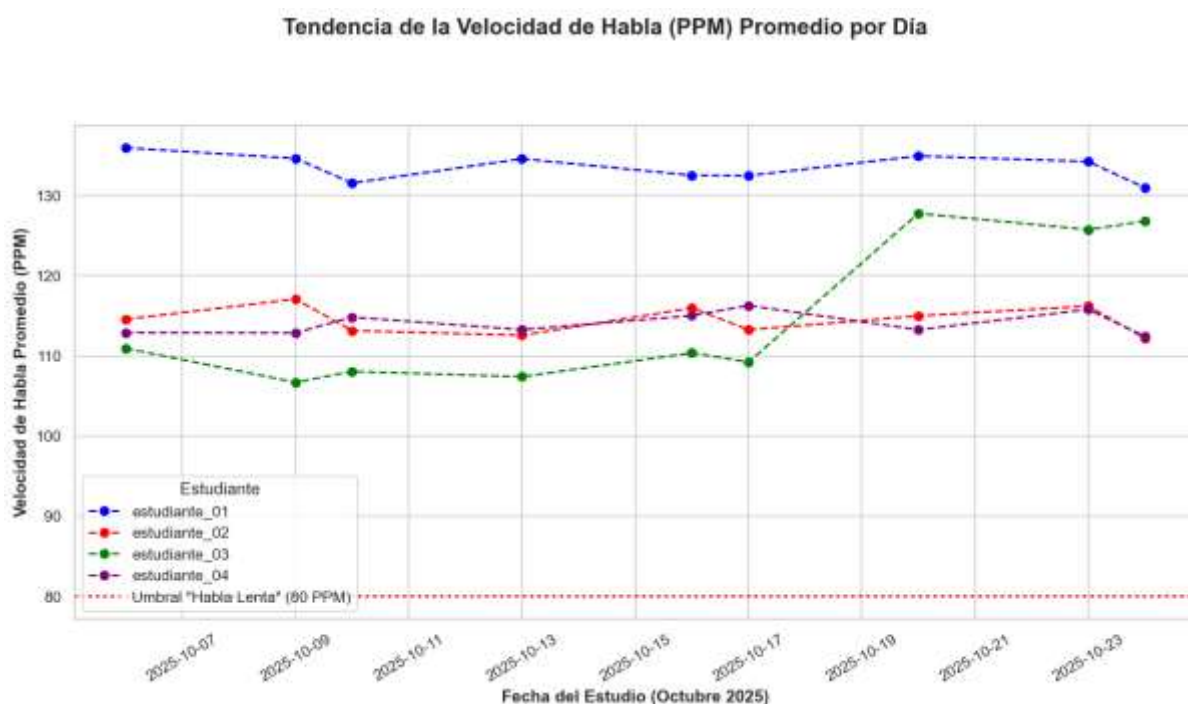
Nuevamente, el estudiante_03 (verde) presenta el cambio más drástico. Su tasa de alertas de habla, que promedió un 84.1% en las primeras 6 sesiones (indicando alta dificultad), se desplomó a un promedio de 38.9% en las últimas tres.

La correlación temporal entre la mejora en la atención Ilustración 8 y la mejora en la fluidez del habla Ilustración 9 para este estudiante es un hallazgo de gran valor, sugiriendo una fuerte conexión entre el enfoque visual y la capacidad de formulación oral.

4.3.4.5. Evolución de la velocidad lectora (ppm) por día

Finalmente, el análisis de la velocidad lectora promedio (Palabras por Minuto) es uno de los indicadores más claros de adaptación positiva del grupo

Ilustración 10. Evolución de Palabras por Minuto



ELABORADO: AUTOR

Finalmente, el gráfico de palabras por minuto completa el análisis. Como en los casos anteriores, los estudiantes 01 (más rápido, ~133 PPM), 02 (~114 PPM) y 04 (~114 PPM) muestran un ritmo de lectura estable.

El estudiante_03 (verde) vuelve a destacar. Su ritmo de lectura, que era el más lento del grupo (promedio de 108.6 PPM) en las primeras 6 sesiones, aumentó significativamente a un promedio de 126.8 PPM en las últimas tres.

En conjunto, los datos del estudiante_03 cuentan una historia clara a medida que sus alertas de atención y habla disminuyeron, su velocidad de lectura aumentó, demostrando la capacidad del sistema para monitorear una mejora conductual medible en el tiempo.

4.3.4. Discusión

El desarrollo de este dashboard materializa la visión de la UNESCO [7] sobre la importancia de integrar tecnologías emergentes en las aulas de manera que mejoren los resultados educativos. Sin embargo, su verdadero valor se comprende al contrastarlo con las brechas identificadas por Flores Cortés et al. [18], quienes señalan que, a pesar del potencial del Machine Learning, gran parte de la investigación se enfoca en modelos predictivos sin concretarse en herramientas que puedan ser utilizadas directamente por los docentes.

Mientras que el análisis algorítmico del objetivo anterior representa el "qué" (el sistema puede detectar), este dashboard representa el "para qué" (para que el docente pueda intervenir). La capacidad de correlacionar visualmente la atención (la línea del gráfico) con la fluidez vocal (las barras) permite al educador discernir patrones complejos que serían casi imposibles de capturar a simple vista, como determinar si una pausa en la lectura se debe a una distracción externa o a una dificultad interna con el texto. En última instancia, la innovación del dashboard no reside solo en su tecnología, sino en su traducibilidad: toma conceptos avanzados de IA y los convierte en conocimiento pedagógico, empoderando al docente para que tome decisiones informadas sin necesidad de ser un experto en datos, cumpliendo así con el objetivo de una verdadera convergencia entre tecnología y educación.

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se diseñó exitosamente la arquitectura de un sistema IoT multimodal de bajo costo, seleccionando componentes (ESP32-CAM, ESP32-WROOM) y protocolos (MQTT) que demostraron ser efectivos. La arquitectura de red fue robusta y estable durante los 9 días de la aplicación piloto real en la Institución “ROUSSBERISS”, probando la viabilidad del diseño para la recolección de datos multimodales (video y audio) en un entorno de aula real.

Se implementaron los algoritmos de aprendizaje automático para la detección de atención (MediaPipe) y el análisis de habla (Vosk y reglas de coherencia). La validación del sistema en el entorno real Tabla 4, demostró una alta precisión de 94.5% en atención, y 91.0% en fluidez, confirmando que los modelos y las reglas jerárquicas son efectivos para identificar patrones conductuales.

Se desarrolló y validó la aplicación web mediante dashboards como una herramienta pedagógica funcional. Como se demostró en el análisis de resultados, la interfaz permitió traducir los datos técnicos en información accionable, logrando identificar visualmente los perfiles de atención y habla marcadamente individuales de los 4 estudiantes y monitorear tendencias de mejora y fatiga a lo largo de los 9 días, cumpliendo así el objetivo de ser un soporte real para el docente.

5.2. Recomendaciones

Para futuras implementaciones, se recomienda miniaturizar los nodos sensores (ESP32-CAM y WROOM) en una sola carcasa impresa en 3D para reducir la intrusión visual en el aula. Asimismo, se sugiere explorar el uso de cámaras con mejor rendimiento en condiciones de baja iluminación y micrófonos con cancelación de ruido ambiental para mejorar la calidad de los datos de entrada.

Utilizar el valioso conjunto de datos reales (dataset) recopilado durante esta aplicación piloto en el Centro de atención integral ROUSSBRISS, subidos al “Sistema IoT de monitoreo de problemas de lectura para niños con TEA”, para re-entrenar y especializar los modelos. Esto permitirá que el sistema se calibre con mayor precisión a las características específicas del habla y los patrones de atención particulares de los niños con TEA.

Ampliar la validación del prototipo, que demostró ser exitosa en esta piloto de 9 días, a un estudio longitudinal de un semestre académico completo y con una muestra de estudiantes más grande. Esto permitirá medir el impacto a largo plazo de la herramienta y correlacionar los hallazgos del sistema con las evaluaciones psicopedagógicas formales de los docentes.

Se recomienda explorar la integración del sistema con plataformas de gestión de aprendizaje (LMS) existentes en las instituciones educativas, para que las alertas y reportes generados se incorporen automáticamente al perfil digital del estudiante. Asimismo, se sugiere fomentar la colaboración interdisciplinaria con psicólogos educativos y terapeutas, utilizando los datos cuantitativos del sistema como un insumo objetivo para complementar sus evaluaciones psicopedagógicas.

CAPITULO VI
BIBLIOGRAFÍA

6.1. Bibliografía

- [1 INEC, «Políticas transformadoras: hacia el nuevo Ecuador, desde la evaluación educativa,» 2022. [En línea]. Available: https://evaluaciones.evaluacion.gob.ec/archivosPD/uploads/dlm_uploads/2023/12/PoliticaDAEEV04PRINT.pdf.
- [2 I. N. d. E. y. Censos, «Reducción del analfabetismo en Ecuador: más de 199.000 personas libres de esta condición desde 2010,» 13 Diciembre 2023. [En línea]. Available: <https://www.ecuadorencifras.gob.ec/reduccion-del-analfabetismo-en-ecuador-mas-de-199-000-personas-libres-de-esta-condicion-desde-2010/>.
- [3 Y. M. Barahona-Cruz, «Polo del Conocimiento,» 30 Marzo 2023. [En línea]. Available: <https://dialnet.unirioja.es/descarga/articulo/9252108.pdf>.
- [4 G. C. Alcalá y M. G. O. Madrigal, «Scielo,» 30 Marzo 2022. [En línea]. Available: https://www.scielo.org.mx/scielo.php?pid=S0026-17422022000100007&script=sci_arttext.
- [5 A.-L. Popescu, N. Popescu, C. Dobre, E.-S. Apostol y D. Popescu, «Sensors,» 25 Marzo 2022. [En línea]. Available: <https://www.mdpi.com/1424-8220/22/7/2528>.
- [6 M. Kohli, A. K. Kar y S. Sinha, «IEEE,» 2022. [En línea]. Available: <https://ieeexplore.ieee.org/abstract/document/9899389>.
- [7 UNESCO, «Technology in education,» 2023. [En línea]. Available: <https://unesdoc.unesco.org/ark:/48223/pf0000386165>.
- [8 A. P. A. Uzho, «Revista Andina de Educación,» 01 Junio 2023. [En línea]. Available: <https://revistas.uasb.edu.ec/index.php/ree/article/view/3798/3980>.
- [9 S. S. Jordi Salazar, «UPC Universidad Politecnica de Catalunya,» 2020. [En línea]. Available: https://upcommons.upc.edu/bitstream/handle/2117/100921/LM08_R_ES.pdf.
- [1 J. A. S. Morán, L. J. M. Burgos, J. G. P. Ganchozo y Jeam, «ReciMundo,» 30 Marzo 2023. 0] [En línea]. Available: <https://dialnet.unirioja.es/descarga/articulo/8882717.pdf>.
- [1 P. P. Torralba, 02 Marzo 2023. [En línea]. Available: [https://www.iebschool.com/hub/que-1\] machine-learning-big-data/](https://www.iebschool.com/hub/que-1] machine-learning-big-data/).
- [1 A. R. S. Guerrero, N. I. G. Lucio y R. R. M. Suin, «Soeici,» 19 Octubre 2024. [En línea]. 2] Available: <https://soeici.org/index.php/alcon/article/view/283>.
- [1 H. L. L. López, A. R. Escalera y C. R. C. García, «RedTis,» 15 Diciembre 2023. [En línea]. 3] Available: <https://redtis.org/index.php/Redtis/article/view/165>.

- [1 L. Izquierdo y Joe, «Universidad politecnica SALESIANA,» 16 Enero 2023. [En línea].
4] Available: <https://dspace.ups.edu.ec/handle/123456789/24037>.
- [1 V. I. Marín y G. Tur, «Edutec,» 20 Marzo 2023. [En línea]. Available:
5] <https://www.edutec.es/revista/index.php/edutec-e/article/view/2701>.
- [1 A. G. ROBLES OSOLLO, «Revista Eurolatinoamericana de Derecho,» Junio 2021. [En
6] línea]. Available:
<https://www.redalyc.org/journal/6559/655969720001/655969720001.pdf>.
- [1 C. S. G. Gonzales, «Análisis de las tecnologías tangibles para la Educación Infantil y
7] principales estrategias pedagógicas,» Junio 2021. [En línea]. Available:
<https://www.edutec.es/revista/index.php/edutec-e/article/view/2085/851>.
- [1 J. A. G. I. R. B. M. Carlos Alberto Flores Cortés, «El internete de las cosas y su impacto
8] en la educacion,» Diciembre 2020. [En línea]. Available:
http://ww.ucol.mx/content/publicacionesenlinea/adjuntos/IoT-PDF_498.pdf.
- [1 O. Banos, Z. Comas-González, J. Medina, A. Polo-Rodríguez, D. Gil, J. Peral, S. Amador
9] y C. Villalonga, «International journal of medical informatics,» Julio 2024. [En línea].
Available:
<https://www.sciencedirect.com/science/article/pii/S1386505624001321/pdf?md5=9794a9489469cbdb5c6af2962e6d97ce&pid=1-s2.0-S1386505624001321-main.pdf>.
- [2 V. Welch, T. J. Wy, A. Ligezka, L. C. Hassett, P. E. Croarkin, A. P. Athreya y M.
0] Romanowicz, «Journal of medical internet research,» Marzo 2022. [En línea]. Available:
<https://www.jmir.org/2022/3/e33560>.
- [2 Lexisfinder, «Ministerio de Defensa de Ecuador,» 25 Enero 2021. [En línea]. Available:
1] https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/02/Constitucion-de-la-Republica-del-Ecuador_act_ene-2021.pdf.
- [2 H. D. P. Barrezueta, «Corporación Nacional de Finanzas Populares y Solidarias –
2] CONAFIPS,» 21 Mayo 2021. [En línea]. Available:
https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf.
- [2 H. E. D. P. Barrezueta, «Municipio de Quito,» 19 Abril 2021. [En línea]. Available:
3] https://gobiernoabierto.quito.gob.ec/Archivos/Transparencia/2021/04abril/A2/ANEXOS/PROCU_LOEI.pdf.
- [2 C. NACIONAL, «Ediciones Legales,» Enero 2014. [En línea]. Available:
4] <https://www.registrocivil.gob.ec/wp-content/uploads/downloads/2014/01/este-es-06-C%C3%93DIGO-DE-LA-NI%C3%91EZ-Y-ADOLESCENCIA-Leyes-conexas.pdf>.

- [2] H. D. P. Barrezueta, «Ministerio de Telecomunicaciones,» Mayo 20216. [En línea].
5] Available: <https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2016/05/Ley-Org%C3%A1nica-de-Telecomunicaciones.pdf>.
- [2] R. C. Delgado, «Arcotel,» 25 Enero 2016. [En línea]. Available:
6] <https://www.arcotel.gob.ec/wp-content/uploads/2016/03/reglamento-lot.pdf>.
- [2] Carlos, «World Literacy Foundation,» 16 Diciembre 2022. [En línea]. Available:
7] <https://worldliteracyfoundation.org/why-literacy-2/>.
- [2] V. M. Vázquez González, «Biblos-e Archivo,» Julio 2020. [En línea]. Available:
8] <https://repositorio.uam.es/handle/10486/692593>. [Último acceso: 20 Marzo 2024].
- [2] I. Skakovskiy, «RISEAPPS,» 25 Febrero 2025. [En línea]. Available:
9] <https://riseapps.co/iot-in-education/>.
- [3] P. G. Prada, «Universidad de Magdalena,» 03 Noviembre 2022. [En línea]. Available:
0] <https://www.redalyc.org/journal/5880/588077486006/html/>.

CAPITULO VII
ANEXOS

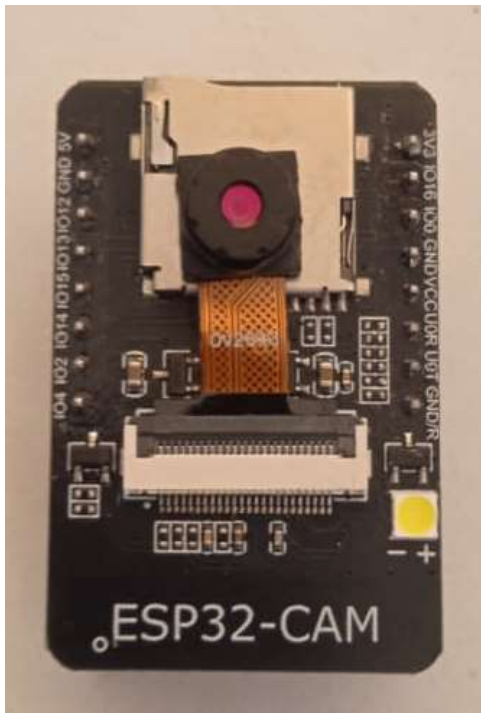
Anexo 2. Módulo de sonido ensamblado



Anexo 1. Módulos de sonidos configurados



Anexo 4. Módulo de video



Anexo 3. Módulos de video configurados



Anexo 5. Configuración del equipo de procesamiento local



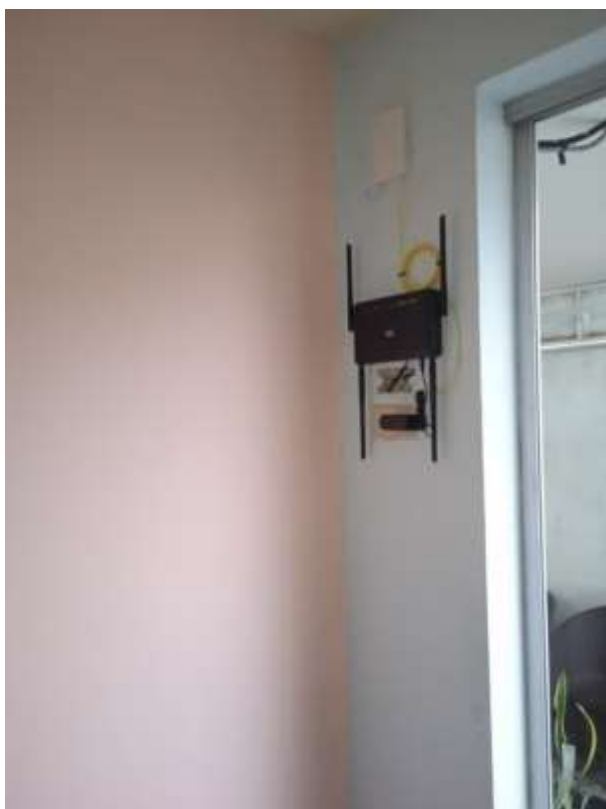
Anexo 6. Comprobación de funcionamiento del módulo de video



Anexo 7. Centro de atención integral ROUSSBRISS



Anexo 9. Punto de red de la Institución



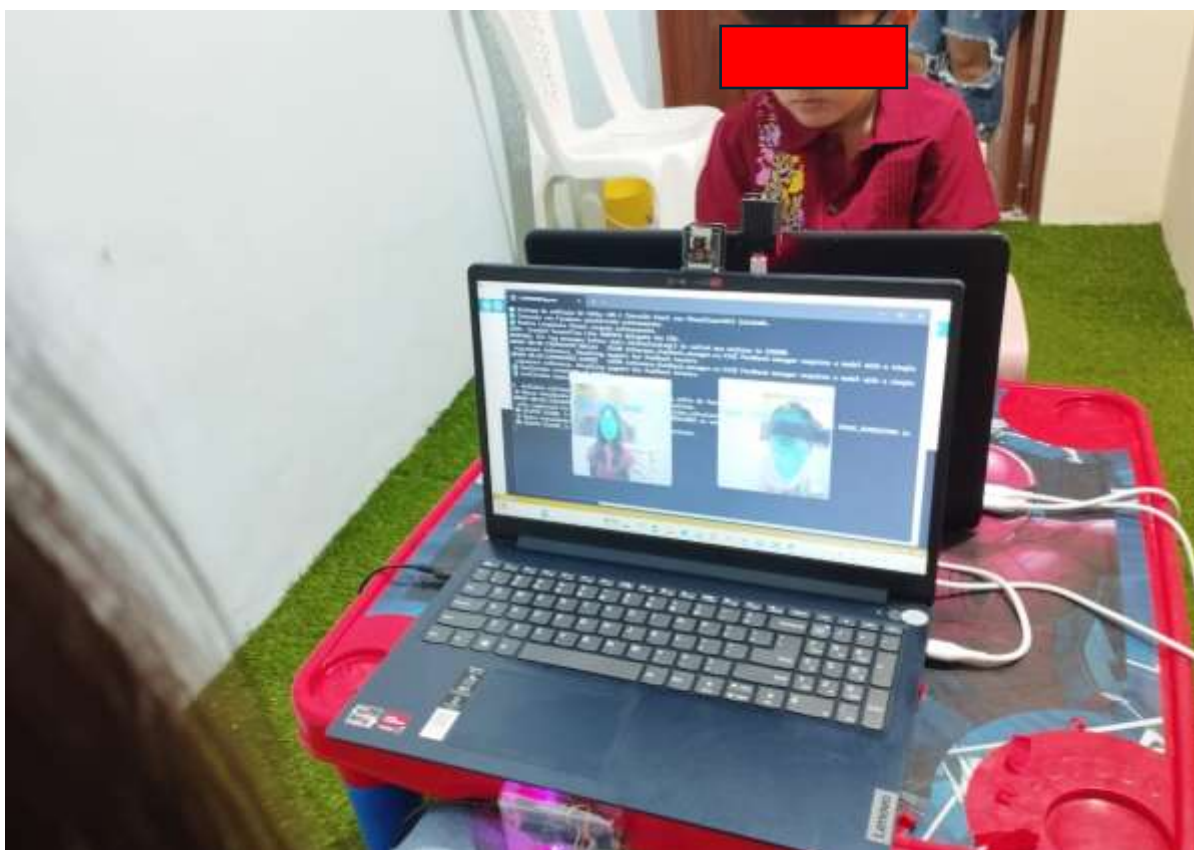
Anexo 8. Configuración del equipo de procesamiento local



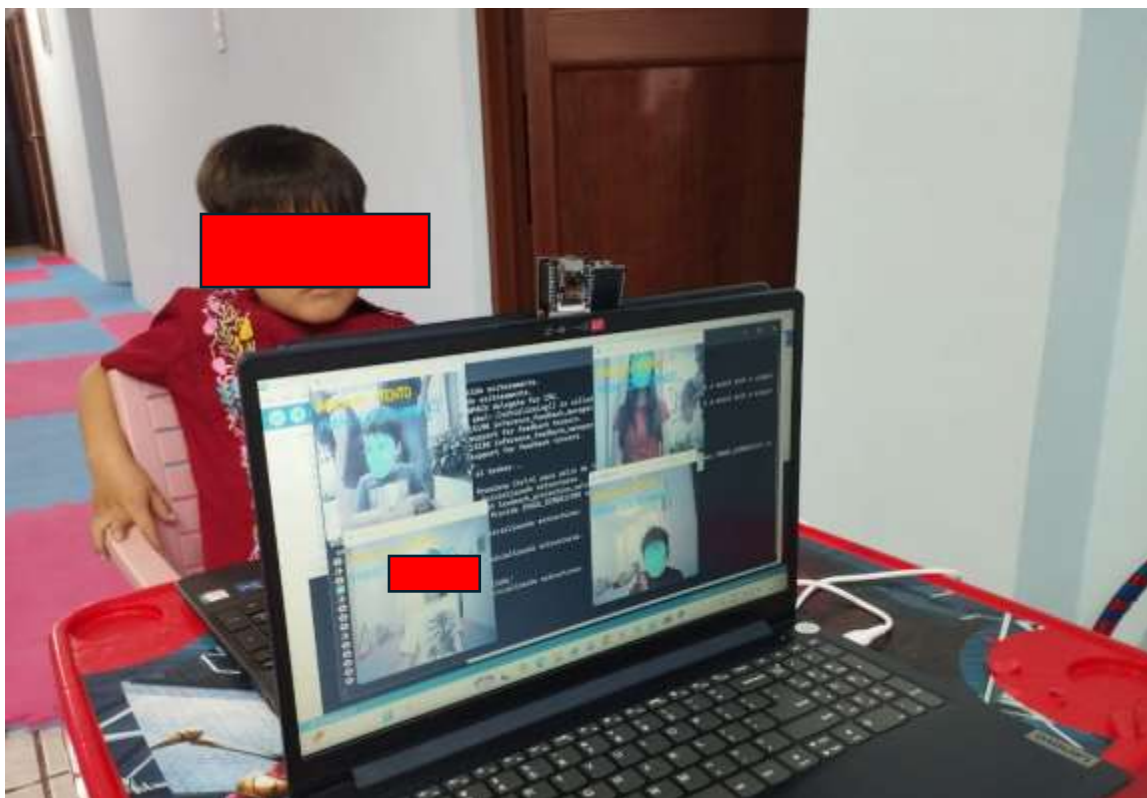
Anexo 10. Ajustes de los equipos instalados



Anexo 11. Verificando captación de audio y video



Anexo 12. Sistema en funcionamiento



Anexo 14. Sistema de alertas en funcionamiento



Anexo 13. Explicación de alertas



Anexo 15. Miembros encargados de la enseñanza



Anexo 16. Código para el ESP32 - WROOM con el sensor de sonido INMP441

```
#define MQTT_MAX_PACKET_SIZE 8192

#include <WiFi.h>
#include <PubSubClient.h>
#include "driver/i2s.h"
#include "base64.h"

// --- CONFIGURACIÓN WIFI Y MQTT ---
const char* WIFI_SSID      = "H2024";
const char* WIFI_PASS      = "FELIX2024";
const char* MQTT_BROKER_IP = "192.168.2.6"; // IP de PC
const int   MQTT_PORT      = 1883;

// --- IDENTIFICADOR DEL ESTUDIANTE ---
const char* DEVICE_ID      = "ESP32-WROOM-Estudiante-01-AUDIO";
const char* MQTT_TOPIC_AUDIO = "institucion/aula/3B/est_01/audio";

// --- CONFIGURACIÓN DEL MICRÓFONO I2S ---
#define I2S_WS_PIN 25
#define I2S_SCK_PIN 26
#define I2S_SD_PIN 22
#define I2S_PORT I2S_NUM_0
```

```

// --- PARÁMETROS DE AUDIO ---
#define SAMPLE_RATE    16000
#define BUFFER_SIZE    512
#define AMPLIFICATION_FACTOR 2.5

// --- Variables Globales ---
WiFiClient espClient;
PubSubClient mqttClient(espClient);

// --- Declaración de funciones ---
void setup_wifi_and_mqtt();
void reconnect_mqtt();
void setup_microphone();
void capture_and_send_audio();

void setup() {
    Serial.begin(115200);
    delay(1000);
    Serial.println("\n=== Iniciando Nodo de Audio (v2.4 - Amplificado) ===");

    setup_wifi_and_mqtt();
    setup_microphone();
}

void loop() {
    if (!mqttClient.connected()) {
        reconnect_mqtt();
    }
    mqttClient.loop();
    capture_and_send_audio();
}

void setup_wifi_and_mqtt() {
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    Serial.print("Conectando a WiFi...");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi conectado!");
    Serial.print("Dirección IP: ");
    Serial.println(WiFi.localIP());

    mqttClient.setServer(MQTT_BROKER_IP, MQTT_PORT);
    mqttClient.setBufferSize(MQTT_MAX_PACKET_SIZE);
    Serial.println("Cliente MQTT configurado.");
}

```

```

void reconnect_mqtt() {
    while (!mqttClient.connected()) {
        Serial.print("Intentando conexión MQTT...");
        if (mqttClient.connect(DEVICE_ID)) {
            Serial.println(" ¡conectado!");
        } else {
            Serial.print(" falló, rc=");
            Serial.print(mqttClient.state());
            Serial.println(" | Intentando de nuevo en 5 segundos");
            delay(5000);
        }
    }
}

void setup_microphone() {
    i2s_config_t i2s_config = {
        .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX),
        .sample_rate = SAMPLE_RATE,
        .bits_per_sample = I2S_BITS_PER_SAMPLE_32BIT,
        .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
        .communication_format = I2S_COMM_FORMAT_STAND_I2S,
        .intr_alloc_flags = 0,
        .dma_buf_count = 8,
        .dma_buf_len = 256,
        .use_apll = true
    };

    i2s_pin_config_t pin_config = {
        .bck_io_num = I2S_SCK_PIN,
        .ws_io_num = I2S_WS_PIN,
        .data_out_num = I2S_PIN_NO_CHANGE,
        .data_in_num = I2S_SD_PIN
    };

    i2s_driver_install(I2S_PORT, &i2s_config, 0, NULL);
    i2s_set_pin(I2S_PORT, &pin_config);
    Serial.println("Micrófono I2S iniciado correctamente.");
}

void capture_and_send_audio() {
    int32_t i2s_read_buffer[BUFFER_SIZE];
    int16_t samples_16bit[BUFFER_SIZE];
    size_t bytes_read = 0;

    i2s_read(I2S_PORT, &i2s_read_buffer, BUFFER_SIZE * 4, &bytes_read,
portMAX_DELAY);
}

```

```

if (bytes_read > 0) {
    int samples_read = bytes_read / 4;
    for (int i = 0; i < samples_read; i++) {
        // Extraer la muestra de 16 bits
        int16_t sample = (i2s_read_buffer[i] >> 16);

        // [NUEVO] Aplicar amplificación digital
        int32_t amplified_sample = (int32_t)sample * AMPLIFICATION_FACTOR;

        // [NUEVO] Control de saturación (clipping) para evitar distorsión
        if (amplified_sample > 32767) {
            amplified_sample = 32767;
        } else if (amplified_sample < -32768) {
            amplified_sample = -32768;
        }
        samples_16bit[i] = (int16_t)amplified_sample;
    }

    String encodedAudio = base64::encode((uint8_t*)samples_16bit, samples_read
* 2);

    String json_payload = "{\"timestamp\":\"" + String(time(NULL)) + "\", " +
        "\"device_id\":\"" + DEVICE_ID + "\", " +
        "\"payload_type\":\"audio\", " +
        "\"format\":\"pcm_base64\", " +
        "\"data\":\"" + encodedAudio + "\"}";

    if (!mqttClient.publish(MQTT_TOPIC_AUDIO, json_payload.c_str())) {
        // El reconnect se gestionará en el bucle principal
    }
}
}
}

```

Anexo 17. Código para el ESP32 - CAM

```

#define MQTT_MAX_PACKET_SIZE 20480

#include "esp_camera.h"
#include <WiFi.h>
#include <PubSubClient.h>
#include "base64.h"

// --- CONFIGURACIÓN WIFI Y MQTT ---
// Reemplaza con tus credenciales
const char* WIFI_SSID = "H2024";
const char* WIFI_PASS = "FELIX2024";
const char* MQTT_BROKER_IP = "192.168.2.6"; // <-- IP de tu PC
const int MQTT_PORT = 1883;

```

```

const char* DEVICE_ID = "ESP32-CAM-Estudiante-03-VIDEO";
const char* MQTT_TOPIC_VIDEO = "institucion/aula/3B/est_01/video";

// Definición de pines para el modelo AI-THINKER
#define PWDN_GPIO_NUM      32
#define RESET_GPIO_NUM     -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM      26
#define SIOC_GPIO_NUM      27
#define Y9_GPIO_NUM        35
#define Y8_GPIO_NUM        34
#define Y7_GPIO_NUM        39
#define Y6_GPIO_NUM        36
#define Y5_GPIO_NUM        21
#define Y4_GPIO_NUM        19
#define Y3_GPIO_NUM        18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM     25
#define HREF_GPIO_NUM      23
#define PCLK_GPIO_NUM      22

// --- Variables Globales ---
WiFiClient espClient;
PubSubClient mqttClient(espClient);
unsigned long lastMsg = 0;
const int msgInterval = 100; // Enviar una imagen cada 5 segundos

// --- Declaración de funciones ---
void reconnect_mqtt();
void capture_and_send_image();

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;

```

```

config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sccb_sda = SIOD_GPIO_NUM;
config.pin_sccb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 10000000;
config.pixel_format = PIXFORMAT_JPEG;
config.frame_size = FRAMESIZE_UXGA; // Inicia con alta resolución
config.jpeg_quality = 20;
config.fb_count = 1;
config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
config.fb_location = CAMERA_FB_IN_PSRAM;

// Configuración avanzada
if (psramFound()) {
    config.jpeg_quality = 15;
    config.fb_count = 2;
    config.grab_mode = CAMERA_GRAB_LATEST;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.fb_location = CAMERA_FB_IN_DRAM;
}

// Iniciar la cámara
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Fallo al iniciar la cámara! Error 0x%x\n", err);
    return;
}
Serial.println("Cámara iniciada correctamente.");

// Bajar la resolución para el envío
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA);

// Conectar a WiFi
WiFi.begin(WIFI_SSID, WIFI_PASS);
WiFi.setSleep(false);
Serial.print("Conectando a WiFi...");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("\nWiFi conectado!");
Serial.print("IP: ");
Serial.println(WiFi.localIP());

```

```

// Configurar MQTT
mqttClient.setServer(MQTT_BROKER_IP, MQTT_PORT);
mqttClient.setBufferSize(MQTT_MAX_PACKET_SIZE);
Serial.println("Cliente MQTT configurado.");
}

void loop() {
    if (!mqttClient.connected()) {
        reconnect_mqtt();
    }
    mqttClient.loop();

    unsigned long now = millis();
    if (now - lastMsg > msgInterval) {
        lastMsg = now;
        Serial.printf("Memoria libre: %d bytes\n", ESP.getFreeHeap());
        capture_and_send_image();
        Serial.println("-----");
    }
    delay(1);
}

void reconnect_mqtt() {
    while (!mqttClient.connected()) {
        Serial.print("Intentando conexión MQTT...");
        if (mqttClient.connect(DEVICE_ID)) {
            Serial.println("conectado!");
        } else {
            Serial.print("falló, rc=");
            Serial.print(mqttClient.state());
            Serial.println(" intentando de nuevo en 5 segundos");
            delay(5000);
        }
    }
}

void capture_and_send_image() {
    camera_fb_t * fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Fallo en la captura de la cámara");
        return;
    }

    char json_header[200];
    sprintf(json_header,
    "{\"timestamp\":\"%lu\", \"device_id\":\"%s\", \"payload_type\":\"video\", \"format\":\"jpeg_base64\", \"data\":\"\", (unsigned long)time(NULL), DEVICE_ID);

```

```

size_t header_len = strlen(json_header);
size_t base64_len = ((fb->len + 2) / 3) * 4;
size_t footer_len = 2; // "}"
size_t total_len = header_len + base64_len + footer_len;

if (mqttClient.beginPublish(MQTT_TOPIC_VIDEO, total_len, false)) {
    mqttClient.print(json_header);

    const int CHUNK_SIZE = 1023;
    uint8_t *input_buffer = fb->buf;
    size_t input_len = fb->len;

    for (size_t i = 0; i < input_len; i += CHUNK_SIZE) {
        size_t current_chunk_size = (i + CHUNK_SIZE < input_len) ? CHUNK_SIZE :
input_len - i;
        String chunk_encoded = base64::encode(input_buffer + i,
current_chunk_size);
        mqttClient.print(chunk_encoded);
    }

    mqttClient.print("\n");

    if (mqttClient.endPublish()) {
        Serial.println("-> Mensaje de video publicado exitosamente.");
    } else {
        Serial.println("-> Fallo al finalizar la publicación de video.");
    }
} else {
    Serial.println("-> Fallo al iniciar la publicación de video.");
}

esp_camera_fb_return(fb);
}

```

Anexo 18. Código del servidor local (Python) de procesamiento de los datos

```

import sys
import os
import json
import base64
import time
import queue
import re
import threading
from datetime import datetime
import traceback

```

```

# --- IMPORTAR LIBRERÍAS REQUERIDAS ---
try:
    from scipy.spatial import distance as dist
    import paho.mqtt.client as mqtt
    import numpy as np
    import cv2
    import firebase_admin
    from firebase_admin import credentials, db
    import mediapipe as mp
    import sounddevice as sd
    from vosk import Model, KaldiRecognizer
    from transformers import pipeline
    import torch
except ImportError as e:
    print(f"❌ Error: Falta una librería esencial -> {e}"); sys.exit()

# --- CONFIGURACIÓN GENERAL Y RUTAS ---
BROKER_IP = "localhost"
BROKER_PORT = 1883
SUSSCRIPTION_TOPIC = "institucion/aula/3B/+/+"
CLIENT_ID = f"pc_analizador_main_{int(time.time())}"
AUDIO_DEVICE_ID = 3
SCRIPT_DIR = os.path.dirname(os.path.abspath(__file__))
print(f"📁 Sistema de análisis de habla ")

# --- CONFIGURACIÓN DE FIREBASE ---
try:
    json_key_path = os.path.join(SCRIPT_DIR, "serviceAccountKey.json")
    cred = credentials.Certificate(json_key_path)
    firebase_admin.initialize_app(cred, {
        'databaseURL': 'https://atencion-lectora-default-rtdb.firebaseio.com'
    })
    print(f"✅ Conexión con Firebase establecida exitosamente.")
except Exception as e:
    print(f"❌ ERROR CRÍTICO AL CONECTAR CON FIREBASE: {e}"); sys.exit()

# --- CONFIGURACIÓN DE MODELOS DE IA ---
try:
    model_folder_name = "vosk-model-es-0.42"
    model_path = os.path.join(SCRIPT_DIR, model_folder_name)
    if not os.path.exists(model_path):
        print(f"❌ Error: El modelo de Vosk no se encontró en la ruta '{model_path}'"); sys.exit()
    vosk_model = Model(model_path)
    print(f"✅ Modelo Lingüista (Vosk) cargado exitosamente.")
except Exception as e:
    print(f"❌ ERROR CRÍTICO AL CARGAR MODELO VOSK: {e}"); sys.exit()

```

```

try:
    print("⌚ Cargando modelo Psicólogo (Hugging Face)...")
    prosody_classifier = pipeline("audio-classification",
model="superb/wav2vec2-base-superb-er")
    print("✅ Modelo Psicólogo (Wav2Vec2) cargado exitosamente.")
except Exception as e:
    print(f"❌ ERROR CRÍTICO AL CARGAR MODELO DE HUGGING FACE: {e}");
sys.exit()

# --- PARÁMETROS DE CALIBRACIÓN Y LÓGICA ---
# Calibración Visual
EAR_THRESHOLD = 0.18
HORIZONTAL_GAZE_THRESHOLD = 0.15
VERTICAL_GAZE_THRESHOLD = 0.12
EYE_CLOSED_CONSEC_FRAMES = 5
FACE_LOSS_THRESHOLD = 15
GAZE_OFF_CONSEC_FRAMES = 5
# Calibración Auditiva
HESITATION_WORDS = {"eh", "este", "pues", "o sea", "bueno", "um", "mm"}
HIGH_HESITATION_RATE = 0.25
HIGH_REPETITION_RATE = 0.20
LOW_WPM_RATE = 80.0
LOW_TTR_RATE = 0.7
LOW_SENTENCE_LEN_RATE = 3.0
PAUSE_THRESHOLD_S = 0.8
AUDIO_ENERGY_THRESHOLD = 300
LOW_COHERENCE_THRESHOLD = 0.05
SILENCE_THRESHOLD_S = 5.0

# --- ESTRUCTURAS DE DATOS Y ESTADO ---
student_data = {}; data_queue = queue.Queue(); firebase_queue = queue.Queue()
audio_playback_queue = queue.Queue(); running = True; connected = False

# --- PARÁMETROS DE AUDIO ---
SAMPLE_RATE = 16000; CHANNELS = 1

# --- INICIALIZACIÓN DE MEDIAPIPE ---
mp_face_mesh = mp.solutions.face_mesh
mp_drawing = mp.solutions.drawing_utils # <-- AÑADIDO PARA DIBUJAR
face_mesh = mp_face_mesh.FaceMesh(max_num_faces=1, refine_landmarks=True,
min_detection_confidence=0.5, min_tracking_confidence=0.5)

# --- ÍNDICES DE PUNTOS CLAVE ---
LEFT_EYE_INDICES = [362, 385, 387, 263, 373, 380]
RIGHT_EYE_INDICES = [33, 160, 158, 133, 153, 144]
LEFT_IRIS_INDICES = [474, 475, 476, 477]
RIGHT_IRIS_INDICES = [469, 470, 471, 472]

```

```

# --- PIPELINE DE ANÁLISIS DE HABLA ---
def extract_linguistic_features(transcribed_text, duration_s):
    text_clean = re.sub(r'^\w\s', '', transcribed_text.lower())
    words = text_clean.split()
    num_words = len(words)
    if num_words == 0: return None

    coherence_score = 1.0
    if num_words > 3:
        bigrams = list(zip(words, words[1:]))
        if bigrams:
            COMMON_SPANISH_BIGRAMS = {
                ('de', 'la'), ('de', 'el'), ('en', 'la'), ('en', 'el'), ('a',
                'la'), ('a', 'el'), ('con', 'la'), ('con', 'el'),
                ('por', 'que'), ('para', 'que'), ('que', 'es'), ('lo', 'que'),
                ('es', 'que'), ('yo', 'soy'), ('yo', 'tengo'),
                ('yo', 'quiero'), ('yo', 'estoy'), ('el', 'es'), ('ella',
                'es'), ('es', 'un'), ('es', 'una'), ('me', 'gusta'),
                ('no', 'me'), ('no', 'es'), ('no', 'se'), ('se', 'fue'),
                ('se', 'ha'), ('voy', 'a'), ('vamos', 'a'),
                ('tiene', 'que'), ('habia', 'una'), ('era', 'un'),
            }
            common_found = sum(1 for bg in bigrams if bg in
COMMON_SPANISH_BIGRAMS)
            coherence_score = common_found / len(bigrams)

    return {
        "transcripcion": transcribed_text,
        "ppm": round((num_words / duration_s) * 60 if duration_s > 0 else 0),
        "tasa_hesitacion": round(sum(1 for w in words if w in
HESITATION_WORDS) / num_words, 3),
        "tasa_repeticion": round((num_words - len(set(words))) / num_words,
3),
        "riqueza_lexica_ttr": round(len(set(words)) / num_words if num_words >
0 else 0, 3),
        "longitud_frase": num_words,
        "coherence_score": coherence_score
    }

def analyze_prosody(audio_bytes):
    try:
        audio_array = np.frombuffer(audio_bytes,
dtype=np.int16).astype(np.float32) / 32768.0
        result = prosody_classifier({"raw": audio_array, "sampling_rate":
SAMPLE_RATE}, top_k=1)
        return result[0]['label'], result[0]['score']
    except Exception: return "N/A", 0.0

```

```

def unified_diagnostic_model(linguistic_features, prosody_label):
    if not linguistic_features or linguistic_features["longitud_frase"] < 2:
        return "HABLA MUY CORTA"

    if linguistic_features["longitud_frase"] > 4 and
linguistic_features["coherence_score"] < LOW_COHERENCE_THRESHOLD:
        return "LENGUAJE DESESTRUCTURADO"

    if linguistic_features["tasa_repeticion"] > HIGH_REPETITION_RATE:
        return f"DIFICULTAD DE FLUIDEZ (Repeticiones)"
    if linguistic_features["tasa_hesitacion"] > HIGH_HESITATION_RATE:
        return f"DIFICULTAD DE FORMULACIÓN (Muletillas)"
    if linguistic_features["ppm"] < LOW_WPM_RATE and
linguistic_features["longitud_frase"] > 4:
        return f"HABLA LENTA (Tono: {prosody_label})"
    if linguistic_features["riqueza_lexica_ttr"] < LOW_TTR_RATE and
linguistic_features["longitud_frase"] > 5:
        return f"POSIBLE POBREZA LÉXICA (Tono: {prosody_label})"
    if linguistic_features["longitud_frase"] < LOW_SENTENCE_LEN_RATE:
        return f"HABLA SIMPLIFICADA (Frases cortas)"
    return f"HABLA FLUIDA (Tono: {prosody_label})"

# --- Hilo de trabajo para el análisis ---
def analysis_worker(student_id, audio_to_analyze, recognizer_instance):
    try:
        final_result = json.loads(recognizer_instance.FinalResult())
        transcribed_text = final_result.get('text', '').strip()

        if not transcribed_text:
            return

        duration = 0
        word_info = final_result.get('result', [])
        if word_info:
            start_time = word_info[0]['start']
            end_time = word_info[-1]['end']
            duration = end_time - start_time
        else:
            duration = len(audio_to_analyze) / (SAMPLE_RATE * 2)

        print(f"🔊 Pausa detectada. Analizando frase de {student_id}:
'{{transcribed_text}}' (dur: {{duration:.2f}}s)")

        linguistic_features = extract_linguistic_features(transcribed_text,
duration)
        prosody_label, prosody_score = analyze_prosody(audio_to_analyze)
        final_status = unified_diagnostic_model(linguistic_features,
prosody_label.upper())

```

```

s_data = student_data.get(student_id)
if not s_data: return

s_data['current_speech_status'] = final_status
s_data['speech_metrics'] = linguistic_features if linguistic_features
else {}
s_data['speech_metrics']['prosody'] = {"label": prosody_label,
"score": round(float(prosody_score), 3)}

payload_fb = {
    "attention_status": s_data['current_attention_status'],
    "speech_status": final_status,
    "speech_metrics": s_data.get('speech_metrics', {}),
    "timestamp": datetime.now().isoformat()
}
firebase_queue.put((student_id, payload_fb))
print(f"🔊 ALERTA DE HABLA {student_id}:
A: '{s_data['current_attention_status']}', H: '{final_status}'")

s_data['last_speech_sent'] = final_status

except Exception as e:
    print(f"⚠️ Error en el hilo de análisis para {student_id}: {e}")

# --- FUNCIONES DE CÁLCULO VISUAL ---
def calculate_ear(eye_landmarks):
    A = dist.euclidean(eye_landmarks[1], eye_landmarks[5]); B =
dist.euclidean(eye_landmarks[2], eye_landmarks[4])
    C = dist.euclidean(eye_landmarks[0], eye_landmarks[3]); return (A + B) /
(2.0 * C)

def calculate_gaze_ratios(eye_points, iris_center):
    eye_width = dist.euclidean(eye_points[0], eye_points[3])
    if eye_width == 0: return 0, 0

    eye_center_x = (eye_points[0][0] + eye_points[3][0]) / 2
    gaze_ratio_h = (iris_center[0] - eye_center_x) / eye_width

    eye_height = dist.euclidean(eye_points[1], eye_points[4])
    if eye_height == 0: return gaze_ratio_h, 0

    eye_center_y = (eye_points[1][1] + eye_points[5][1]) / 2
    gaze_ratio_v = (iris_center[1] - eye_center_y) / eye_height

    return gaze_ratio_h, gaze_ratio_v

# --- HILOS Y MQTT ---

```

```

def firebase_thread_func():
    while running:
        try:
            student_id, payload_to_send = firebase_queue.get(timeout=1)
            db.reference(f'analysis/{student_id}').set(payload_to_send)
            db.reference(f'analysis_logs/{student_id}').push(payload_to_send)
        except queue.Empty: continue
        except Exception: pass

def audio_playback_thread():
    stream = None
    try:
        stream = sd.OutputStream(samplerate=SAMPLE_RATE, channels=CHANNELS,
dtype='int16', device=AUDIO_DEVICE_ID)
        stream.start()
        while running:
            try:
                _, audio_data = audio_playback_queue.get(timeout=1)
                if running: stream.write(audio_data)
            except queue.Empty: continue
        except Exception: pass
    finally:
        if stream:
            stream.stop(); stream.close()

def on_connect(client, userdata, flags, rc, properties=None):
    global connected
    if rc == 0: client.subscribe(SUSCRIPTION_TOPIC); connected = True;
    print(f"✅ Analizador conectado al broker.")
    else: print(f"❌ Falló la conexión al broker con código: {rc}")

def on_message(client, userdata, msg):
    try:
        topic_parts = msg.topic.split('/')
        if len(topic_parts) >= 5:
            student_id, data_type = topic_parts[3], topic_parts[4]
            data_queue.put((student_id, data_type,
json.loads(msg.payload.decode('utf-8'))))
        except Exception: pass

def create_recognizer():
    rec = KaldiRecognizer(vosk_model, SAMPLE_RATE)
    rec.SetWords(True)
    return rec

def initialize_student_data(student_id):
    if student_id not in student_data:

```

```

        print(f"👤 Nuevo estudiante detectado: {student_id}. Inicializando estructuras.")
        student_data[student_id] = {
            'id': student_id,
            'no_face_counter': 0,
            'eye_closed_counter': 0,
            'gaze_off_counter': 0,
            'current_attention_status': 'Iniciando...', 'last_attention_sent':
'Iniciando...',
            'recognizer': create_recognizer(),
            'last_speech_time': time.time(), 'is_speaking': False,
            'current_speech_status': 'ESCUCHANDO...', 'last_speech_sent':
'ESCUCHANDO...',
            'speech_metrics': {}, 'accumulated_audio': b''
        }
        cv2.namedWindow(f'Video - {student_id}', cv2.WINDOW_NORMAL)

# --- FUNCIÓN PRINCIPAL ---
def main():
    global running
    client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2,
client_id=CLIENT_ID)
    client.on_connect = on_connect; client.on_message = on_message

    try: client.connect(BROKER_IP, BROKER_PORT, 60)
    except Exception as e: print(f"❌ No se pudo conectar al broker. Error:
{e}"); return

    client.loop_start()
    threading.Thread(target=firebase_thread_func, daemon=True).start()
    threading.Thread(target=audio_playback_thread, daemon=True).start()

    print("⌚ Analizador esperando conexión al broker...")
    time.sleep(2)
    if not connected: print("❌ No se pudo conectar al broker. Terminando.");
running = False; return

    print("\n▶️ Análisis multimodal iniciado. Presiona Ctrl+C para salir de
forma segura.")

    try:
        while running:
            try:
                student_id, data_type, payload = data_queue.get(timeout=0.01)
                initialize_student_data(student_id)
                s_data = student_data.get(student_id)
                if not s_data: continue

```

```

        if data_type == 'video':
            img =
cv2.imdecode(np.frombuffer(base64.b64decode(payload['data']), np.uint8),
cv2.IMREAD_COLOR)
            if img is not None:
                results = face_mesh.process(cv2.cvtColor(img,
cv2.COLOR_BGR2RGB))

                previous_attention_status =
s_data['current_attention_status']

                if results.multi_face_landmarks:
                    s_data['no_face_counter'] = 0
                    face_landmarks_list =
results.multi_face_landmarks[0] # <-- AÑADIDO
                    landmarks = np.array([(lm.x * img.shape[1], lm.y *
img.shape[0]) for lm in face_landmarks_list.landmark], dtype=np.int32)
                    avg_ear =
(calculate_ear(landmarks[LEFT_EYE_INDICES]) +
calculate_ear(landmarks[RIGHT_EYE_INDICES])) / 2.0
                    left_iris_center =
landmarks[LEFT_IRIS_INDICES].mean(axis=0).astype(np.int32)
                    right_iris_center =
landmarks[RIGHT_IRIS_INDICES].mean(axis=0).astype(np.int32)
                    left_gaze_h, left_gaze_v =
calculate_gaze_ratios(landmarks[LEFT_EYE_INDICES], left_iris_center)
                    right_gaze_h, right_gaze_v =
calculate_gaze_ratios(landmarks[RIGHT_EYE_INDICES], right_iris_center)
                    avg_gaze_h = (left_gaze_h + right_gaze_h) / 2.0
                    avg_gaze_v = (left_gaze_v + right_gaze_v) / 2.0

                    if avg_ear < EAR_THRESHOLD:
                        s_data['eye_closed_counter'] += 1
                        s_data['gaze_off_counter'] = 0
                        if s_data['eye_closed_counter'] >=
EYE_CLOSED_CONSEC_FRAMES:
                            s_data['current_attention_status'] = "OJOS
CERRADOS"

                        elif abs(avg_gaze_h) > HORIZONTAL_GAZE_THRESHOLD
or abs(avg_gaze_v) > VERTICAL_GAZE_THRESHOLD:
                            s_data['gaze_off_counter'] += 1
                            s_data['eye_closed_counter'] = 0
                            if s_data['gaze_off_counter'] >=
GAZE_OFF_CONSEC_FRAMES:
                                s_data['current_attention_status'] = "SIN
ATENCIÓN VISUAL"

                    else:
                        s_data['eye_closed_counter'] = 0

```

```

        s_data['gaze_off_counter'] = 0
        s_data['current_attention_status'] = "ATENTO"

        # --- [VISUALIZACIÓN DE PUNTOS CLAVE ACTIVADA] ---
        mp_drawing.draw_landmarks(
            image=img, landmark_list=face_landmarks_list,
            connections=mp_face_mesh.FACEMESH_TESSELATION,
            landmark_drawing_spec=None,
            connection_drawing_spec=mp_drawing.DrawingSpec
            (color=(255, 255, 0), thickness=1, circle_radius=1))
        cv2.circle(img, tuple(left_iris_center), 3, (0, 0,
255), -1)

        cv2.circle(img, tuple(right_iris_center), 3, (0,
0, 255), -1)

        # --- [FIN DE LA VISUALIZACIÓN] ---

    else:
        s_data['no_face_counter'] += 1
        s_data['eye_closed_counter'] = 0
        s_data['gaze_off_counter'] = 0
        if s_data['no_face_counter'] >=
FACE_LOSS_THRESHOLD: s_data['current_attention_status'] = "DISTRACCION"

        if s_data['current_attention_status'] !=
s_data['last_attention_sent']:
            s_data['last_attention_sent'] =
s_data['current_attention_status']
            payload_fb = {
                "attention_status":
s_data['current_attention_status'],
                "speech_status":
s_data['current_speech_status'],
                "speech_metrics": s_data.get('speech_metrics',
{}),
                "timestamp": datetime.now().isoformat()
            }
            firebase_queue.put((student_id, payload_fb))
            print(f"👁 ALERTA VISUAL {student_id}:
'{s_data['current_attention_status']}'")

            cv2.putText(img, f"Atencion:
{s_data['current_attention_status']}", (10, 30), cv2.FONT_HERSHEY_SIMPLEX,
0.7, (0, 255, 255), 2)

            cv2.putText(img, f"Habla:
{s_data['current_speech_status']}", (10, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.7,
(255, 150, 0), 2)

            cv2.imshow(f'Video - {student_id}', img)
            if cv2.waitKey(1) & 0xFF == ord('q'): running = False

```

```

        elif data_type == 'audio':
            audio_data_bytes = base64.b64decode(payload['data'])
            audio_playback_queue.put((student_id,
np.frombuffer(audio_data_bytes, dtype=np.int16)))
            s_data['accumulated_audio'] += audio_data_bytes

            audio_chunk_np = np.frombuffer(audio_data_bytes,
dtype=np.int16)

            rms = np.sqrt(np.mean(audio_chunk_np.astype(float)**2))

            if rms > AUDIO_ENERGY_THRESHOLD:
                s_data['is_speaking'] = True
                s_data['last_speech_time'] = time.time()

                s_data['recognizer'].AcceptWaveform(audio_data_bytes)

    except queue.Empty:
        pass

    current_time = time.time()
    for sid, s_data_loop in list(student_data.items()):
        time_since_last_speech = current_time -
s_data_loop.get('last_speech_time', 0)

        if s_data_loop.get('is_speaking') and time_since_last_speech >
PAUSE_THRESHOLD_S:

            s_data_loop['is_speaking'] = False

            audio_to_analyze = s_data_loop['accumulated_audio']
            recognizer_to_analyze = s_data_loop['recognizer']

            s_data_loop['accumulated_audio'] = b''
            s_data_loop['recognizer'] = create_recognizer()

            if len(audio_to_analyze) > SAMPLE_RATE * 0.5:
                worker = threading.Thread(target=analysis_worker,
args=(sid, audio_to_analyze, recognizer_to_analyze), daemon=True)
                worker.start()

        elif not s_data_loop.get('is_speaking') and
time_since_last_speech > SILENCE_THRESHOLD_S:
            if s_data_loop['current_speech_status'] !=
'ESCUCHANDO...':
                print(f"🔊 Silencio prolongado detectado para {sid}.
Reseteando estado de habla.")
                s_data_loop['current_speech_status'] = 'ESCUCHANDO...'

```

```

        s_data_loop['last_speech_sent'] = 'ESCUCHANDO...'
        s_data_loop['speech_metrics'] = {}

        payload_fb = {
            "attention_status":
s_data_loop['current_attention_status'],
            "speech_status": "ESCUCHANDO...",
            "speech_metrics": {},
            "timestamp": datetime.now().isoformat()
        }
        firebase_queue.put((sid, payload_fb))

        time.sleep(0.01)

    except KeyboardInterrupt:
        print("\n🛑 Interrupción de teclado detectada. Iniciando apagado
seguro...")
        finally:
            running = False
            print("⌚ Esperando a que los hilos finalicen...")
            time.sleep(2)
            client.loop_stop()
            cv2.destroyAllWindows()
            print("✅ Analizador finalizado correctamente.")

if __name__ == "__main__":
    main()

```

Anexo 19. Carta de consentimiento

CARTA DE CONSENTIMIENTO INFORMADO Y CONFIDENCIALIDAD

Fecha: Santo Domingo, 02 de octubre de 2025 **Institución:** Centro de Atención Integral “ROUSSBRIS” **Investigador Responsable:** Félix Rafael Gracia Guato (Universidad Técnica Estatal de Quevedo)

TÍTULO DEL PROYECTO: “Sistema de Internet de las Cosas con algoritmos de aprendizaje automático para la detección temprana de dificultades lectoras en niños con autismo de educación básica”.

OBJETIVO Y PROCEDIMIENTO: Por medio de la presente, se informa a los padres de familia y representantes legales sobre la realización de una prueba piloto de un sistema tecnológico de apoyo pedagógico. El objetivo es monitorear indicadores de atención y fluidez durante actividades de lectura mediante el uso de sensores no invasivos (cámaras y micrófonos) conectados a un sistema de procesamiento local.

CONFIDENCIALIDAD Y USO DE DATOS: Se garantiza estricta confidencialidad en el manejo de la información:

1. **Uso Académico:** Los datos recopilados serán utilizados exclusivamente para fines de investigación y validación de la tesis de grado.
2. **Anonimato:** La identidad de los estudiantes será protegida. En los informes y documentos públicos (tesis), los niños serán identificados únicamente mediante códigos (ej. "Estudiante 01", "Estudiante 02"), sin revelar nombres ni rostros.
3. **Seguridad:** Las grabaciones de video y audio crudos serán procesadas en el laboratorio y eliminadas posteriormente, conservando solo los datos estadísticos.

ACEPTACIÓN: Al firmar este documento, los abajo firmantes declaran haber sido informados de los objetivos del estudio y autorizan la participación de sus representados en las sesiones de prueba del sistema, comprendiendo que la participación es voluntaria y no conlleva riesgos para la integridad de los menores.

FIRMAS DE AUTORIZACIÓN DE LOS REPRESENTANTES LEGALES



Sr. Carlos Eduardo Méndez



Sra. María Fernanda López



Sra. Lucía Estefanía Torres



Sr. Jorge Luis Andrade