

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



Desarrollo de un sistema de control de bajo costo para la asistencia de personas con movilidad severamente limitada

Trabajo Final de Graduación para optar por el título de Ingeniero en
Electrónica con el grado académico de Licenciatura

Allan Alberto Céspedes Sandí

Cartago, julio 2025

Declaratoria y Licencia

Declaro que el presente documento ha sido realizado en su totalidad por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado material bibliográfico, he procedido a indicar las fuentes mediante citas. En consecuencia, asumo responsabilidad total por el informe de anteproyecto de graduación realizado y por el contenido del correspondiente informe.



Allan Alberto Céspedes Sandí

Cédula: 116620885

Cartago, junio 2025

Desarrollo de un sistema de control de bajo costo para la asistencia de personas con movilidad severamente limitada © 2025 by Allan Alberto Céspedes Sandí is licensed under CC BY 4.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>



Resumen

El proyecto denominado "Look2Control" consiste en el desarrollo de un sistema de control accesible y de bajo costo para personas con movilidad severamente limitada, utilizando tecnologías como el rastreo ocular mediante cámara y la plataforma ESP8266.

El sistema proporciona una interfaz gráfica intuitiva que permite a los usuarios controlar dispositivos como luces, sonido y motores específicos mediante comandos visuales, transmitidos a través de Wi-Fi. La interfaz se adapta a las necesidades del usuario, garantiza un control eficiente y aborda una necesidad crítica en la accesibilidad tecnológica con una solución efectiva y fácil de usar para la gestión del entorno de personas con movilidad reducida.

Palabras Clave: accesibilidad, seguimiento ocular, interfaz gráfica, sistema de control, bajo costo.

Abstract

The project named "Look2Control" consists in the development of an accessible and low-cost control system for individuals with severe mobility limitations, using technologies such as eye tracking via camera and the ESP8266 platform.

The system provides an intuitive graphical interface that allows users to control devices such as lights, sound, and specific motors through visual commands transmitted over a Wi-Fi network. The interface is adapted to user needs, ensures efficient control, and addresses a critical need in technological accessibility through an effective and easy-to-use solution for managing the environment of individuals with reduced mobility.

Keywords: accessibility, eye tracking, graphical interface, control system, low cost.

Agradecimientos

Este proyecto representa no solo el cierre de una etapa académica, sino también la culminación de un proceso personal. Por eso, estas palabras son para quienes hicieron posible que hoy me encuentre aquí.

En primer lugar, a mi mamá, María —Yita—, la persona que más admiro en el mundo. Desde siempre ha sido el ejemplo de fortaleza que me guía. Su amor y resiliencia me han enseñado que no hay límites cuando uno cree en sí mismo. Nunca me soltó la mano, ni siquiera en mis momentos más oscuros. Ella estuvo ahí, demostrándome con su presencia que todo estaría bien. Su fe inquebrantable en mí me impulsó a seguir, a crecer, a sanar y a construir algo en mi vida de la cuál yo me siento orgulloso. Este logro es tan suyo como mío.

A mi hermano Mayron, mi compañero desde la infancia, mi espejo. Siempre vio en mí un potencial que a veces yo mismo dudaba tener. Me impulsó a ser el mejor en todo lo que hago, a no conformarme. Si alguien me dice que me parezco a él, lo tomo como un elogio, porque su corazón, su mente y su generosidad son rasgos que admiro profundamente.

A mi hermana Jennifer, por ser ese refugio protector en todo momento. Su preocupación constante, su capacidad de cuidarme, apoyarme y su fe en que yo podía completar este camino me hicieron sentir seguro y valioso en todo momento. Siempre deseando mi crecimiento y mejoría. Sé que todos en la familia nos sentimos así.

A mi sobrina Valentina, una chispa en mi vida. Su pensamiento hacia mí, con esa curiosidad y forma de admirar las cosas, me recordó que tengo una responsabilidad no solo como profesional, sino como ser humano. Ella me enseñó que puedo ser guía, ejemplo y motivación para quienes están comenzando a descubrir el mundo.

A mis amigos, quienes formaron parte de cada etapa de este recorrido, muchos desde la universidad, otros desde caminos distintos, pero todos con un lugar muy especial en mi vida. A quienes me escucharon hasta la madrugada, a quienes me impulsaron cuando quería detenerme, a quienes me recordaron que mi esfuerzo no era en vano. Gracias por confiar en mí incluso cuando mis propias fuerzas flaqueaban.

Y finalmente, pero no menos importante, a mí mismo. Porque no me rendí. Porque supe levantarme aun cuando el camino parecía deshecho bajo mis pies. Porque usé mi historia personal donde ni siquiera imaginaba que podía llegar hasta acá, como motivación para crear. Aquel recuerdo del pasado no solo me marcó físicamente, me hizo más consciente, más empático y determinado. Recordé por qué comencé este camino, por qué me apasiona, por qué amo mejorar. Este proyecto fue una forma de reconciliarme con mi pasado, de darle propósito al dolor, de transformar la adversidad en conocimiento. Estoy orgulloso de haber creído en mí, incluso cuando todo parecía incierto. Y sé, que mi yo de antes, el que apenas comenzaba a levantarse, estaría profundamente orgulloso de quien soy hoy.

Gracias, a todos.

Índice de Contenidos

Capítulo 1	Origen y contexto técnico de Look2Control.....	1
1.1	Accesibilidad tecnológica para movilidad reducida.....	1
1.2	Look2Control: solución accesible de ingeniería aplicada	2
1.3	Alcance funcional y limitaciones de Look2Control.....	4
1.4	Estructura del documento	5
Capítulo 2	Tecnologías y fundamentos aplicados al desarrollo de sistemas de asistencia 6	
2.1	Tecnologías de seguimiento ocular.....	6
2.1.1	Principios generales del seguimiento ocular	7
2.1.2	Detección de mirada y parpadeo mediante visión por computadora.....	8
2.1.3	Tecnologías implementadas: librería EyeGestures.....	9
2.2	Diseño de interfaces accesibles	10
2.2.1	Principios de accesibilidad visual.....	10
2.2.2	Pygame	11
2.3	Microcontrolador ESP8266 y entorno de hardware	11
2.4	Estructura de conectividad y comunicación	12
Capítulo 3	Diseño e implementación del sistema de control visual Look2Control	14
3.1	Diseño general del sistema Look2Control	14
3.1.1	Flujo general de funcionamiento	15
3.1.2	Componentes principales del sistema.....	17
3.1.3	Diagrama de bloques	17
3.2	Desarrollo del módulo de seguimiento ocular.....	18
3.2.1	Implementación de la librería EyeGestures.....	18
3.2.2	Detección de la mirada	19
3.2.3	Confirmación por parpadeo	21
3.2.4	Proceso de calibración	23
3.3	Construcción de la interfaz gráfica.....	25
3.3.1	Criterios visuales de accesibilidad.....	25
3.3.2	Retroalimentación visual y lógica de interacción.....	27
3.3.3	Descripción funcional de la interfaz.....	28

3.4	Implementación de la comunicación con el microcontrolador.....	31
3.4.1	Comunicación mediante protocolo HTTP.....	31
3.4.2	Cliente HTTP en Python y estructura de comandos.....	32
3.5	Programación del ESP8266 como servidor de control.....	34
3.6	Ensamblaje y configuración de prototipo.....	36
3.6.1	Organización del montaje físico y conexión de componentes.....	37
3.6.2	Asignación de pines y conexiones.....	38
Capítulo 4	Evaluación experimental del desempeño del sistema Look2Control.....	43
4.1	Metodología de validación.....	43
4.2	Pruebas por módulo funcional.....	44
4.2.1	Validación del seguimiento ocular y la interfaz visual.....	44
4.2.2	Validación de botones lógicos.....	46
4.2.3	Validación de módulos físicos.....	47
Capítulo 5	Análisis de resultados y oportunidades de mejora del sistema.....	50
5.1	Evaluación de respuesta del sistema de entrada visual.....	50
5.2	Análisis de la lógica interna y su interacción en la interfaz.....	51
5.3	Comportamiento funcional y eléctrico de los dispositivos físicos controlados....	52
5.4	Limitaciones detectadas y oportunidades de mejora técnica.....	53
Capítulo 6	Conclusiones generales y proyección del sistema Look2Control.....	54
6.1	Conclusiones.....	54
6.2	Recomendaciones.....	55
	Bibliografía.....	56

Índice de Figuras

Fig 2.1	Ejemplo de detección ocular.....	7
Fig 2.2	Face Mesh de MediaPipe.....	9
Fig 2.3	Diagrama de módulos del ESP8266.....	12
Fig 2.4	Estructura de funcionamiento HTTP.....	13
Fig 3.1	Diagrama de flujo de Look2Control.....	16
Fig 3.2	Diagrama de bloques de Look2Control.....	18
Fig 3.3	Ejemplo visual de punto estimado por algoritmo de regresión.....	21
Fig 3.4	Posición de la mirada con respecto a los ejes de la pantalla.....	24
Fig 3.5	Interfaz y estructura de botones para Look2Control.....	26
Fig 3.6	Cambio de borde al fijar la mirada.....	27

Fig 3.7 Borde de espera de confirmación.....	28
Fig 3.8 Mensaje de acción confirmada.....	28
Fig 3.9 Encabezado de la interfaz.....	29
Fig 3.10 Representación visual de la mirada en pantalla	31
Fig 3.11 Diagrama de flujo de comunicación por HTTP	32
Fig 3.12 Estructura de comandos HTTP tipo GET.....	33
Fig 3.13 Funcionamiento del decorador @en_fondo	34
Fig 3.14 Ejemplo de activación de dispositivo.....	35
Fig 3.15 Esquemático completo del sistema físico Look2Control.....	38
Fig 3.16 Divisor de voltaje para señal de temperatura	41
Fig 4.1 Puntos de calibración aceptado por el sistema	44
Fig 4.2 Posición estimada de mirada durante la prueba correspondiente al punto de enfoque	45
Fig 4.3 Proceso de lógica de interacción para botón	45
Fig 4.4 Retroalimentación visual de confirmación de activación	45
Fig 4.5 Encendido del led por medio de la interfaz.....	48
Fig 4.6 Proceso de intensidad de luz activada por la interfaz.....	48
Fig 4.7 Respuesta del servidor al activar botón en interfaz.....	49
Fig 4.8 Controlador encendido evidenciando el movimiento del motor	49

Índice de Tablas

Tabla 1.1 Comparación funcional y económica entre alternativas accesibles.....	3
Tabla 2.1: Técnicas de detección de mirada y parpadeo	8
Tabla 3.1: Componentes de Look2Control.....	17
Tabla 3.2: Variables clave utilizadas en Look2Control	18
Tabla 3.3: Variables para doble parpadeo	22
Tabla 3.4 Esquema de colores del sistema	25
Tabla 3.5 Funciones de botones en la interfaz.....	29
Tabla 3.6 Comandos y dispositivos controlados en ESP8266.....	36
Tabla 3.7 Desglose de componentes físicos utilizados en Look2Control	37
Tabla 3.8 Asignación de pines y funciones del ESP8266.....	38
Tabla 3.9 Entradas para controlador ULN2003	42
Tabla 4.1 Dispersión promedio y máxima por botón	46
Tabla 4.2 Pruebas de funcionalidad de botones lógicos del sistema	46

Capítulo 1

Origen y contexto técnico de Look2Control

1.1 Accesibilidad tecnológica para movilidad reducida

En la vida cotidiana, la mayoría de las personas realiza actividades básicas como alimentarse, comunicarse o desplazarse sin considerar la complejidad que estas implican en ausencia de movilidad física. Para personas con movilidad severamente limitada, estas tareas que parecen ser comunes pueden representar importantes que dependen de la asistencia de terceros, generando un impacto profundo en su autonomía, independencia y calidad de vida.

Una de las condiciones más extremas dentro de estas discapacidades es el síndrome del enclaustramiento (Locked-In Syndrome) [1], un trastorno neurológico en el que los pacientes pierden casi todas las funciones motoras voluntarias, conservando únicamente el movimiento ocular en algunos casos. Estas personas permanecen conscientes y cognitivamente funcionales, pero sin la capacidad física de interactuar con su entorno, lo que las hace totalmente dependientes de familiares o cuidadores incluso para las acciones más básicas.

Actualmente, existen dispositivos que permiten cierto grado de interacción mediante tecnologías de rastreo ocular o control por voz. No obstante, muchas de estas soluciones tienen precios muy elevados, como es el caso del Tobii Dynavox I-Series [2], un sistema comercial de seguimiento ocular con costos que pueden superar los \$15,000 USD, convirtiéndolos en imposibles de obtener para la mayoría de la población.

Alternativas más económicas, como el Tobii Eye Tracker 5 [3], tienen un precio base de aproximadamente \$270 USD, pero requieren software adicional especializado, lo que eleva el costo funcional final a más de \$1000 USD. Además, este tipo de dispositivo no incluye funciones esenciales como la detección de parpadeos, la confirmación visual por fijación sostenida, ni la posibilidad de controlar dispositivos físicos del entorno. Se limita a capturar el punto de atención visual en pantalla, siendo su uso principal en contextos de videojuegos, productividad o análisis visual, no en soluciones integrales para personas con movilidad severa.

Se sabe que, en Costa Rica, el 17.2%, de la población adulta presenta algún tipo de discapacidad, lo que representa más de 670.000 personas según la Encuesta Nacional sobre Discapacidad (ENADIS) en 2023 [4]. A nivel mundial, la Organización Mundial de la Salud (OMS) estima que 1 de cada 6 personas vive con algún tipo de discapacidad, lo que representa alrededor de 1300 millones de personas [5] y si focalizamos la discapacidad a personas con parálisis como el síndrome de enclaustramiento, se estima que afecta a más de 5 millones de

personas [6]. En este contexto, y respaldado por el marco legal nacional establecido en la Ley 7600 de Igualdad de Oportunidades para las Personas con Discapacidad, se reconoce como una responsabilidad social y profesional avanzar en el desarrollo de sistemas que promuevan la accesibilidad universal.

En años recientes, esfuerzos previos en el Instituto Tecnológico de Costa Rica demostraron la viabilidad de un sistema prototipo para personas con discapacidad motora. Sin embargo, dicho sistema presentaba limitaciones considerables en cuanto a su interfaz gráfica, portabilidad e integración real con dispositivos del entorno. Mostraba imprecisión y requería de periféricos poco adaptables a un uso cotidiano.

1.2 Look2Control: solución accesible de ingeniería aplicada

El proyecto reportado en este informe, ahora nombrado Look2Control, surge como respuesta directa a la necesidad de diseñar tecnologías accesibles que mejoren la independencia de personas con movilidad muy limitada. Como se expuso anteriormente, el acceso a dispositivos de asistencia avanzados se ve restringido por factores económicos, técnicos y de adaptabilidad, lo que excluye a una parte considerable de la población, particularmente en países como Costa Rica.

Desde un enfoque ingenieril, Look2Control destaca por su capacidad para brindar una solución funcional, de bajo costo y basada en tecnologías abiertas, sin comprometer la estabilidad ni la experiencia del usuario. A diferencia de otras propuestas existentes, esta solución plantea un equilibrio entre funcionalidad y accesibilidad, integrando las siguientes características clave:

- Seguimiento ocular utilizando una cámara convencional
- Sistema de confirmación sencilla por parpadeos
- Interfaz gráfica simple pero completa
- Conectividad inalámbrica con microcontroladores
- Activación física de dispositivos domóticos reales
- Estructura modular, escalable y flexible

La elaboración de este proyecto no se fundamenta únicamente en su bajo costo comparativo con las alternativas mencionadas, sino también en las limitaciones funcionales de esos dispositivos, como se evidencia en la Tabla 1.1.

Tabla 1.1 Comparación funcional y económica entre alternativas accesibles

Característica	Tobii Eye Tracker 5	Tobii Dynavox I-Series	Look2Control
Precio base	\$270 USD	\$10,000 USD	< \$500 USD
Costo total estimado	\$1000 USD	\$10,000 USD	< \$500 USD
Hardware requerido	Computadora	Integrado	Integrado
Sistema de confirmación	No	Sí	Sí
Control de dispositivos externos	No	Si (con módulos adicionales)	Sí (integrado y escalable)
Interfaz accesible	No	Sí	Sí
Aplicación principal	Videojuegos y productividad	Comunicación asistida	Control de entorno y accesibilidad

Cada opción presenta fortalezas y debilidades claras. El Tobii Dynavox I-Series, aunque ofrece una solución funcional con interfaz accesible y sistema de confirmación, tiene un costo extremadamente elevado, lo que lo vuelve inaccesible para la mayoría de los usuarios. Por otro lado, el Tobii Eye Tracker 5 representa una opción económica en cuanto a hardware, pero carece de funcionalidades clave, como la detección de parpadeo o el control de dispositivos físicos, lo que limita severamente su aplicabilidad como tecnología de asistencia. En contraste, Look2Control logra un equilibrio entre funcionalidad, costo y accesibilidad, integrando seguimiento ocular con control domótico en un sistema de fácil implementación. Esta combinación convierte al proyecto en una propuesta viable dentro del contexto de la ingeniería aplicada y la responsabilidad social.

Además de los factores técnicos y económicos que lo sustentan, la motivación para desarrollar Look2Control también se enmarca en una vivencia personal. Tras experimentar de forma temporal las limitaciones físicas derivadas de un grave accidente, surgió una mayor conciencia sobre los obstáculos cotidianos que enfrentan las personas con movilidad reducida. Durante ese periodo, tareas que suelen darse por sentadas, como encender una luz o solicitar asistencia, se convirtieron en desafíos complejos de afrontar. Esta experiencia no solo generó empatía hacia quienes enfrentan limitaciones permanentes, sino que también permitió identificar oportunidades reales para diseñar soluciones funcionales, accesibles y adaptadas al entorno local. Look2Control no nace únicamente como un proyecto académico, sino como una respuesta tangible a una necesidad sentida, desde la ingeniería y para la vida real.

El desarrollo de Look2Control se construyó a partir de la necesidad de integrar distintas tecnologías de forma coherente. Se abordaron aspectos fundamentales como la implementación de un sistema de seguimiento precisa, la creación de una interfaz gráfica

accesible para interacción visual, y la integración de un microcontrolador inalámbrico capaz de activar dispositivos físicos. Estas metas definidas, guiaron la toma de decisiones técnicas, el diseño modular del sistema y las pruebas funcionales que permitieron validar su utilidad y estabilidad en entornos reales.

1.3 Alcance funcional y limitaciones de Look2Control

El sistema Look2Control fue desarrollado con el objetivo de mejorar la interacción de personas con movilidad severamente limitada mediante un mecanismo de control accesible y económico, basado en seguimiento ocular y confirmación de acción por parpadeo simple. Durante su construcción, se definieron y lograron múltiples componentes fundamentales: el desarrollo de un algoritmo funcional de rastreo ocular con cámara integrada, una interfaz gráfica accesible programada en Python, y la integración con un microcontrolador ESP8266 para controlar dispositivos físicos del entorno. Además, el sistema se ejecuta completamente en una tablet con Windows estándar, sin requerir hardware especializado ni licencias de software comerciales, lo que refuerza su aplicabilidad en contextos reales y de bajo presupuesto.

El alcance del sistema incluye la capacidad de detectar la mirada del usuario, interpretar parpadeos como comandos de confirmación, activar botones en pantalla y enviar señales inalámbricas que controlan dispositivos como luces, ventiladores o alarmas. Todo esto se realiza mediante hardware de bajo costo, con arquitectura modular, conectividad Wi-Fi y una interfaz diseñada para personas con necesidades específicas de accesibilidad.

Sin embargo, también se definieron limitaciones que respondieron a las restricciones de tiempo, presupuesto y viabilidad técnica. El sistema fue diseñado principalmente para usarse en interiores, bajo condiciones de iluminación estable. Además, la precisión del rastreo ocular puede depender de la posición de la cámara y de la postura del usuario. Por otra parte, aunque se estableció un modelo funcional para controlar múltiples dispositivos, no se integraron todavía módulos de comunicación más avanzados como MQTT o plataformas de automatización residencial. Tampoco se realizó una validación clínica formal con usuarios finales reales debido a la naturaleza académica y exploratoria del proyecto.

Estas limitaciones no comprometen el funcionamiento del sistema como prueba de concepto, pero sí marcan oportunidades claras de mejora para futuras versiones, tales como optimizar la calibración del rastreo ocular, ampliar la integración domótica y probar el sistema en condiciones más diversas y controladas.

1.4 Estructura del documento

El presente documento se encuentra organizado en seis capítulos principales. El Capítulo 1 establece el contexto general del proyecto, su justificación técnica, delimita su alcance y principales restricciones. A lo largo del capítulo se plantea la necesidad de una solución accesible, así como las motivaciones que dieron origen a Look2Control como propuesta viable dentro del campo de la ingeniería electrónica aplicada.

El Capítulo 2 presenta el marco teórico y estado del arte, abordando el seguimiento ocular, los principios de diseño visual para interfaces interactivas, y los criterios utilizados para la selección del microcontrolador y arquitectura de comunicación.

En el Capítulo 3 se detalla el diseño e implementación del sistema, con énfasis en el enfoque modular adoptado, la arquitectura general, los componentes seleccionados y la integración entre el software de rastreo ocular, la interfaz gráfica y el microcontrolador responsable del control de dispositivos físicos.

El Capítulo 4 documenta las pruebas experimentales realizadas, evidenciando el funcionamiento del sistema, su precisión y estabilidad, así como su capacidad de control en tiempo real.

El Capítulo 5 presenta un análisis crítico de los resultados obtenidos, contrastándolos con los objetivos planteados inicialmente y valorando las limitaciones y oportunidades de mejora de la solución.

Finalmente, el Capítulo 6 expone las conclusiones generales del proyecto y plantea recomendaciones para futuras implementaciones y escalabilidad del sistema.

Capítulo 2

Tecnologías y fundamentos aplicados al desarrollo de sistemas de asistencia

El desarrollo de Look2Control requirió la integración de diversas disciplinas para construir una solución de asistencia accesible y estable. A diferencia de sistemas comerciales que dependen de hardware especializado o licencias propietarias, este proyecto se diseñó sobre tecnologías abiertas y componentes de bajo costo, priorizando la funcionalidad y escalabilidad.

Este capítulo presenta los fundamentos técnicos que respaldan la implementación del sistema, organizados en las siguientes áreas clave:

- **Tecnologías de seguimiento ocular:** se describen los principios del seguimiento ocular, con énfasis en métodos de detección de mirada y parpadeo mediante cámara especializadas, así como, en la evaluación de bibliotecas de software aplicadas al sistema.
- **Diseño de interfaces accesibles:** se analizan criterios técnicos de accesibilidad visual y estándares como las WCAG (Web Content Accessibility Guidelines) [7], aplicados al desarrollo de una interfaz gráfica programada en Pygame [8].
- **Componentes electrónicos utilizados:** se justifica la selección del microcontrolador ESP8266, junto con los módulos empleados, considerando criterios de compatibilidad, bajo consumo y facilidad de implementación.
- **Estructura de conectividad y comunicación:** se detalla la arquitectura de comunicación entre la interfaz gráfica y el microcontrolador, basada en protocolo HTTP, destacando las decisiones de diseño orientadas a simplicidad y estabilidad funcional.

Cada sección fundamenta las decisiones adoptadas durante el diseño e implementación, consolidando una base técnica sólida para comprender cómo Look2Control responde a un problema real desde la ingeniería.

2.1 Tecnologías de seguimiento ocular

Según Duchowski (2017), el seguimiento ocular es una técnica que permite registrar los movimientos de los ojos para obtener la dirección de la mirada y analizar la interacción

del usuario con su entorno [9]. A nivel técnico, se basa en la detección y seguimiento de puntos faciales clave, como el centro de las pupilas y los bordes de los párpados, mediante técnicas de visión por computadora

2.1.1 Principios generales del seguimiento ocular

De acuerdo con Holmqvist et al. (2011), el seguimiento ocular de alta precisión se ha basado históricamente en dispositivos que emplean iluminación infrarroja para generar reflejos en la córnea y la pupila. Estas reflexiones permiten estimar con gran exactitud el punto de fijación visual utilizando modelos geométricos calibrados [10]. Sin embargo, esta técnica requiere cámaras y emisores infrarrojos especializados, lo que incrementa significativamente el costo y la complejidad del sistema.

Una alternativa más accesible consiste en el uso de cámaras RGB convencionales, como las integradas en laptops o tablets. Tal como describe Hameed et al. (2024), los sistemas basados en cámaras RGB convencionales utilizan algoritmos de visión por computadora para detectar el rostro y estimar la mirada mediante puntos faciales de referencia (“**landmarks**”), es decir, coordenadas que identifican regiones específicas como ojos, nariz o boca para modelar su estructura [11]. Aunque esta aproximación implica ciertos desafíos en términos de precisión frente a variaciones de iluminación y posición del usuario, permite construir soluciones de bajo costo y accesibilidad, como la planteada en Look2Control. Este modelo de detección ocular permite inferir la dirección de la mirada mediante el cálculo de vectores relativos entre pupilas y nariz, como se ilustra en la **Fig 2.1**.

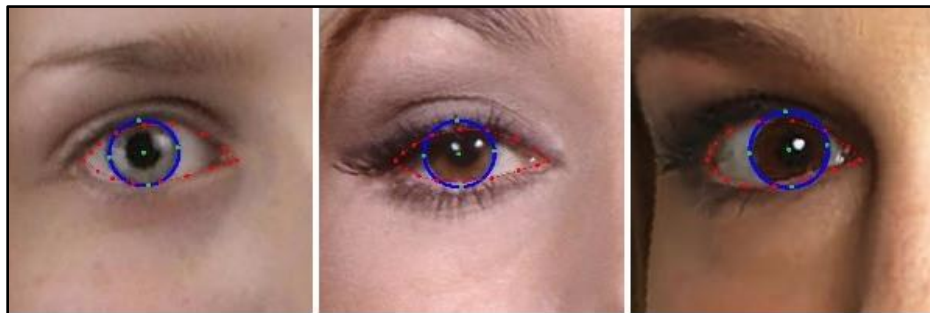


Fig 2.1 Ejemplo de detección ocular

La medición básica en eye tracking comprende dos principales indicadores:

- **Dirección de la mirada:** estimación del vector de atención visual basado en la posición relativa de pupilas y nariz.
- **Eventos de fijación:** análisis de los movimientos rápidos y fijaciones estables que realiza el ojo, información utilizada en sistemas que interpretan acciones basadas en la duración del enfoque.

2.1.2 Detección de mirada y parpadeo mediante visión por computadora

Según Duchowski (2017), la estimación de la dirección de la mirada en sistemas de seguimiento ocular se realiza a partir de la localización de landmarks faciales, identificando los bordes de los ojos y los centros aproximados de las órbitas oculares [9]. A partir de esta información, se calcula un vector de mirada (vector de gaze), el cual representa la dirección de atención visual del usuario con respecto al marco de referencia de la cámara. Este vector constituye la base para inferir hacia qué zona de la pantalla está enfocando el usuario.

La detección de parpadeos, por su parte, puede abordarse mediante distintas técnicas de análisis de la morfología ocular. De acuerdo con Holmqvist et al. (2011), existen dos métodos principales ampliamente utilizados [10]:

- **Detección por desaparición temporal de puntos faciales de referencia:** cuando el usuario parpadea, los algoritmos de visión por computadora registran una pérdida transitoria de los puntos faciales asociados a los bordes del ojo, interpretándolo como un evento de parpadeo.
- **Detección por relación de aspecto ocular (Eye Aspect Ratio, EAR):** como plantea Soukupová y Čech (2016), esta técnica consiste en calcular la relación entre la altura y el ancho del ojo en cada fotograma de video. Cuando esta relación disminuye por debajo de un umbral predeterminado, se considera que el ojo está cerrado [12].

Estos enfoques permiten no solo detectar parpadeos simples, sino también distinguir patrones más complejos como el doble parpadeo o el parpadeo prolongado, estrategias utilizadas en Look2Control como mecanismos de confirmación de comandos dentro de la interfaz accesible. La **Tabla 2.1** resume las principales técnicas utilizadas para la detección de la dirección de la mirada y eventos de parpadeo, junto con sus ventajas y limitaciones.

Tabla 2.1: Técnicas de detección de mirada y parpadeo

Técnica	Principio	Ventaja	Limitación
Vector de mirada	Estimación geométrica de dirección de mirada	No requiere hardware especializado	Sensible a iluminación variable
Región de aspecto ocular	Reducción de apertura ocular	Implementación ligera	Requiere calibración precisa
Pérdida de puntos faciales de referencia	Desaparición temporal de puntos oculares	Detección robusta de parpadeo	Afectado por oclusiones faciales

2.1.3 Tecnologías implementadas: librería EyeGestures

La implementación del módulo de seguimiento ocular en Look2Control se basa en el uso de la librería EyeGestures, un desarrollo de código abierto que combina algoritmos de visión por computadora para la detección de mirada y eventos de parpadeo utilizando cámaras RGB convencionales.

Según la documentación oficial de EyeGestures [13], el núcleo de su funcionamiento radica en la integración del modelo Face Mesh de MediaPipe, desarrollado por Google [14]. Este modelo realiza la localización precisa de 468 puntos faciales en tiempo real, incluyendo puntos específicos sobre los bordes de los ojos, la nariz y la boca. A partir de esta malla tridimensional, EyeGestures calcula vectores de atención visual y detecta gestos oculares como parpadeos mediante análisis geométrico de los puntos faciales de referencia .

De acuerdo con Grishchenko et al. (2020), el modelo Face Mesh emplea técnicas para ajustar los puntos faciales sobre la imagen capturada, proporcionando una representación precisa del rostro humano [15]. Esta representación es crítica para estimar la dirección de la mirada mediante la comparación de la posición relativa de las pupilas con respecto al centro de la cabeza.

En Look2Control, la librería EyeGestures permite realizar dos funciones principales:

- **Detección continua de la posición ocular:** seguimiento de la mirada mediante estimación geométrica basada en la malla de puntos faciales.
- **Identificación de eventos de parpadeo:** detección de cierre ocular mediante la variación de la distancia vertical entre landmarks de párpados.

La malla de puntos faciales utilizada por el modelo Face Mesh de MediaPipe puede observarse en la **Fig 2.1**, donde se representan las posiciones estratégicas para el análisis de movimientos oculares.



Fig 2.2 Face Mesh de MediaPipe

2.2 Diseño de interfaces accesibles

El diseño de la interfaz gráfica de Look2Control se fundamenta en principios de accesibilidad visual destinados a facilitar la interacción mediante la mirada, eliminando barreras para personas con movilidad severamente limitada. Esta sección describe los conceptos técnicos aplicados para garantizar un diseño correcto de la interfaz, así como la implementación práctica de estos principios utilizando la biblioteca Pygame.

2.2.1 Principios de accesibilidad visual

Según las Web Content Accessibility Guidelines (WCAG) 2.1, publicadas por el W3C (2024), el diseño de interfaces accesibles requiere garantizar que el contenido y la funcionalidad de un sistema sean perceptibles, operables, comprensibles y robustos para personas con diferentes tipos de discapacidades [7]. Aunque estas directrices fueron originalmente desarrolladas para entornos web, sus principios son aplicables de manera general al diseño de interfaces gráficas para aplicaciones.

Como mencionamos, entre los principios técnicos más relevantes se destacan:

- **Perceptibilidad:** los elementos de la interfaz deben ser fácilmente detectables por los usuarios, mediante el uso de contrastes adecuados, tamaños apropiados y colores distinguibles que faciliten la identificación de componentes interactivos.
- **Operabilidad:** los componentes deben poder ser activados y controlados sin requerir movimientos físicos complejos, ofreciendo métodos alternativos de interacción accesible.
- **Comprensibilidad:** la interfaz debe presentar elementos consistentes, con etiquetas claras, organización lógica y retroalimentación inmediata que facilite su comprensión para usuarios de diversos perfiles.
- **Robustez:** la estructura del sistema debe permitir su correcta interpretación en distintos dispositivos y entornos tecnológicos, asegurando compatibilidad y escalabilidad.

La aplicación de estos principios resulta fundamental para el diseño de soluciones accesibles y está respaldada no solo por estándares internacionales como las WCAG, sino también por marcos legales nacionales, como la Ley 7600 de Igualdad de Oportunidades para las Personas con Discapacidad vigente en Costa Rica.

2.2.2 Pygame

Pygame es una biblioteca de código abierto escrita en Python, diseñada para el desarrollo de aplicaciones multimedia, especialmente videojuegos y simulaciones interactivas en 2D. Permite manejar gráficos, sonido y entrada del usuario de manera eficiente y multiplataforma [8].

Desde una perspectiva técnica, Pygame proporciona funcionalidades para:

- La creación de ventanas y superficies gráficas.
- El manejo de eventos del usuario.
- El control de ciclos de actualización en tiempo real.
- La manipulación de imágenes de forma sencilla.

Esta biblioteca permite crear entornos visuales completamente adaptados a necesidades específicas, como lo requieren los sistemas de asistencia basados en interacción visual. Al estar hecha en Python, Pygame es altamente compatible con otras bibliotecas como OpenCV, NumPy o incluso módulos de comunicación por red. Esta integración facilita su uso en proyectos de ingeniería aplicada, donde se requiere combinar visión por computadora, interfaces de usuario y comunicación con hardware externo.

Pygame es también una herramienta de bajo consumo de recursos, lo que la hace adecuada para correr en dispositivos modestos, como tablets o laptops de gama media. Su portabilidad y soporte la convierte en una opción viable para el desarrollo de interfaces accesibles en sistemas experimentales o prototipos.

2.3 Microcontrolador ESP8266 y entorno de hardware

El ESP8266 es un microcontrolador de bajo costo desarrollado por Espressif Systems, ampliamente utilizado en sistemas embebidos e Internet de las Cosas (IoT) debido a su capacidad de conectividad Wi-Fi integrada, bajo consumo energético y tamaño compacto. Su arquitectura combina un procesador que tiene capacidades de comunicación y control digital, lo que lo convierte en una opción ideal para proyectos que requieren interacción con redes inalámbricas y control de periféricos en tiempo real [16].

Entre sus características más destacadas se encuentran:

- **Conectividad Wi-Fi:** permite establecer conexiones como cliente o servidor sin necesidad de módulos adicionales.
- Procesador a 80/160 MHz y hasta 4 MB de memoria flash (dependiendo del modelo).

- **Entradas/salidas digitales (GPIO):** incluye múltiples pines para control de relés, LEDs, sensores u otros actuadores.
- **Compatibilidad con el entorno Arduino IDE:** lo que facilita su programación utilizando bibliotecas ampliamente disponibles y una comunidad técnica activa.

En el contexto del sistema Look2Control, el ESP8266 cumple la función de interfaz entre el software de la interfaz gráfica y los dispositivos físicos a controlar, como luces o alarmas. Recibe comandos enviados desde la aplicación principal mediante el protocolo HTTP y ejecuta acciones específicas a través de sus salidas digitales.

Este microcontrolador ha sido particularmente valioso por su capacidad de operar de forma autónoma, su bajo costo y su amplia documentación técnica. Además, su integración directa con plataformas abiertas como Arduino lo hace ideal para prototipos funcionales, pruebas rápidas y despliegues en escenarios reales de bajo presupuesto. La estructura de los módulos integrados en el ESP8266 se muestra en la **Fig 2.3**.

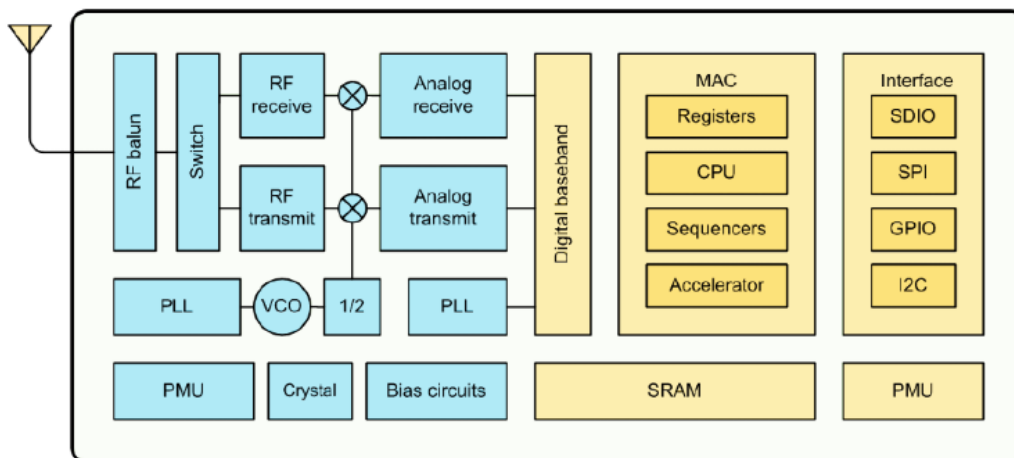


Fig 2.3 Diagrama de módulos del ESP8266

2.4 Estructura de conectividad y comunicación

Los sistemas de asistencia que integran software de interfaz con controladores físicos requieren una arquitectura de comunicación eficiente, estable y de baja latencia. En Look2Control, la interacción entre la interfaz gráfica desarrollada en Python y el microcontrolador ESP8266 se realiza mediante el protocolo HTTP (Hypertext Transfer Protocol), utilizando conexiones inalámbricas a través de Wi-Fi.

HTTP es un protocolo de capa de aplicación que permite el intercambio estructurado de datos entre un cliente (en este caso, la interfaz gráfica) y un servidor (el microcontrolador).

Su simplicidad, compatibilidad con múltiples plataformas y bajo requerimiento de procesamiento lo convierten en una opción viable para sistemas embebidos con capacidad de red [17].

En términos técnicos, la interfaz actúa como cliente HTTP, enviando solicitudes tipo GET a direcciones IP locales configuradas dentro de la red doméstica. Por su parte, el ESP8266 ejecuta un pequeño servidor HTTP que interpreta dichas solicitudes y activa o desactiva pines de salida según el endpoint accedido. Esta arquitectura permite la transmisión de comandos como apagar una luz de manera estructurada y sin necesidad de protocolos adicionales de comunicación en tiempo real. La Fig 2.4 representa la arquitectura de comunicación implementada, en la cual la tablet actúa como cliente HTTP y el ESP8266 como servidor que recibe y ejecuta comandos.

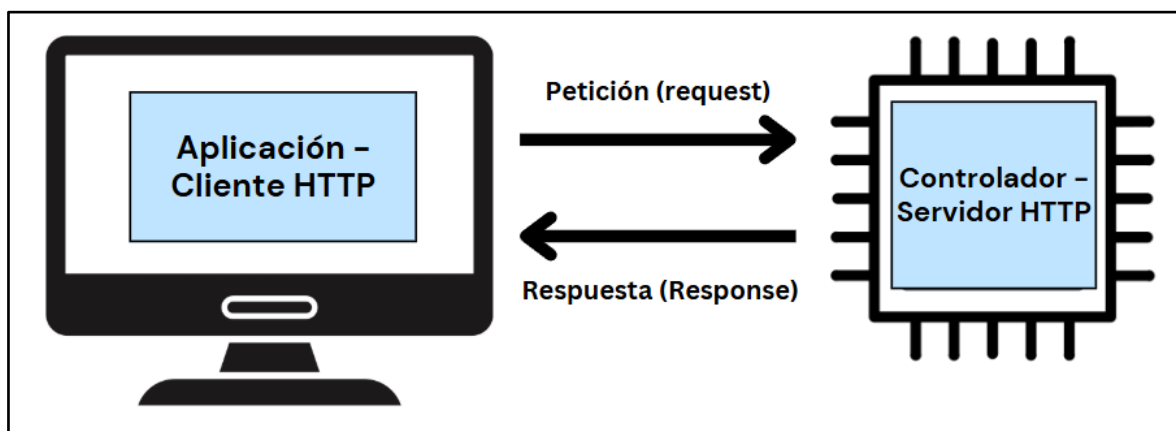


Fig 2.4 Estructura de funcionamiento HTTP

A diferencia de protocolos más avanzados como MQTT o WebSockets, HTTP no mantiene una conexión persistente ni requiere suscripción a tópicos. Esto reduce la complejidad de implementación, aunque limita algunas funcionalidades como el monitoreo continuo de estados o la comunicación bidireccional en tiempo real.

Capítulo 3

Diseño e implementación del sistema de control visual Look2Control

La implementación de Look2Control se planteó bajo un enfoque modular, en el que cada componente del sistema, desde la captura de la mirada hasta la activación física de dispositivos, fue diseñado como un bloque autónomo, pero compatibles entre sí. Esta estrategia permitió reducir la complejidad de integración, facilitar pruebas individuales y escalar el sistema de forma controlada.

Durante el desarrollo se aplicaron criterios propios del diseño de sistemas integrados, priorizando eficiencia y simplicidad. La elección del ESP8266 como microcontrolador, el uso de una interfaz gráfica ligera desarrollada en Pygame, y la decisión de usar HTTP como protocolo de comunicación, obedecen a este tipo de criterios.

Aquí se presenta la implementación desde la perspectiva práctica, partiendo del diseño global del sistema, seguido por cada una de las etapas funcionales: seguimiento ocular, interfaz gráfica, comunicación con el microcontrolador, programación del ESP8266, y ensamblaje físico del prototipo. Más que documentar qué se hizo, este capítulo analiza cómo y por qué se tomaron decisiones específicas a nivel técnico, con el objetivo de construir un sistema accesible y escalable.

3.1 Diseño general del sistema Look2Control

El diseño del sistema se concibió bajo un enfoque de arquitectura modular, donde cada bloque responde a una necesidad específica de accesibilidad tecnológica. Desde la entrada visual del usuario hasta la activación física de dispositivos, el sistema está compuesto por una secuencia clara de elementos que interactúan entre sí de manera organizada. Este está conformado por tres bloques principales:

- **Bloque de percepción e interacción visual:** compuesto por la cámara integrada de la tablet y la interfaz gráfica desarrollada en Pygame, la cual interpreta la mirada y los parpadeos del usuario como comandos de control.
- **Bloque de procesamiento y comunicación:** encargado de transformar los eventos visuales detectados en comandos específicos enviados mediante el protocolo HTTP al microcontrolador.
- **Bloque de control físico:** compuesto por el microcontrolador ESP8266 y los dispositivos electrónicos conectados, que reciben los comandos y ejecutan la acción correspondiente.

3.1.1 Flujo general de funcionamiento

El flujo Look2Control responde a una secuencia lógica basada en la interacción visual y la confirmación mediante parpadeos, la cual se encuentra representada en la **Fig 3.1**. Este sistema combina el seguimiento ocular en tiempo real con decisiones de activación controladas por el usuario, asegurando precisión y evitando activaciones accidentales.

El proceso inicia con la apertura de la interfaz gráfica, desarrollada en Pygame, la cual carga en pantalla los elementos interactivos necesarios para la calibración inicial. Una vez iniciada, el sistema procede a detectar el rostro del usuario mediante la cámara, activando los módulos de seguimiento ocular y detección de parpadeos, e iniciando la rutina de calibración visual, que permite al sistema mapear correctamente la posición relativa de los ojos sobre la pantalla.

Cuando la calibración ha finalizado exitosamente, se habilitan las opciones de control (botones visuales representando dispositivos domóticos). A partir de este momento, el sistema evalúa de forma continua la posición de la mirada del usuario, verificando si esta se encuentra enfocada en alguna de las opciones disponibles.

Al detectar que el usuario ha fijado su vista sobre una opción, el sistema espera la aparición de un primer parpadeo, interpretado como un intento de confirmación. Si no se detecta un segundo parpadeo en los siguientes tres segundos, la acción se cancela y el sistema retorna al estado de observación. En cambio, si se detecta un segundo parpadeo dentro del periodo de ventana de confirmación, la acción se considera válida y se procede a la activación del dispositivo correspondiente.

Esta activación se lleva a cabo mediante el envío de una petición HTTP al ESP8266, el cual interpreta el comando recibido y activa la salida digital apropiada. Finalmente, el dispositivo físico, como una luz, alarma o motor, responde con una acción concreta, cerrando así el ciclo de interacción del usuario.

El usuario tiene 3 opciones diferentes si lo separamos por la funcionalidad que realiza la aplicación. La más concreta es activar el control domótico, pero por obvias razones, es necesario que la aplicación sostenga la alternativa de recalibrar el sistema y también de cerrarlo sin necesidad de una ayuda física. Estas alternativas también se ven representados como botones dentro de la interfaz.

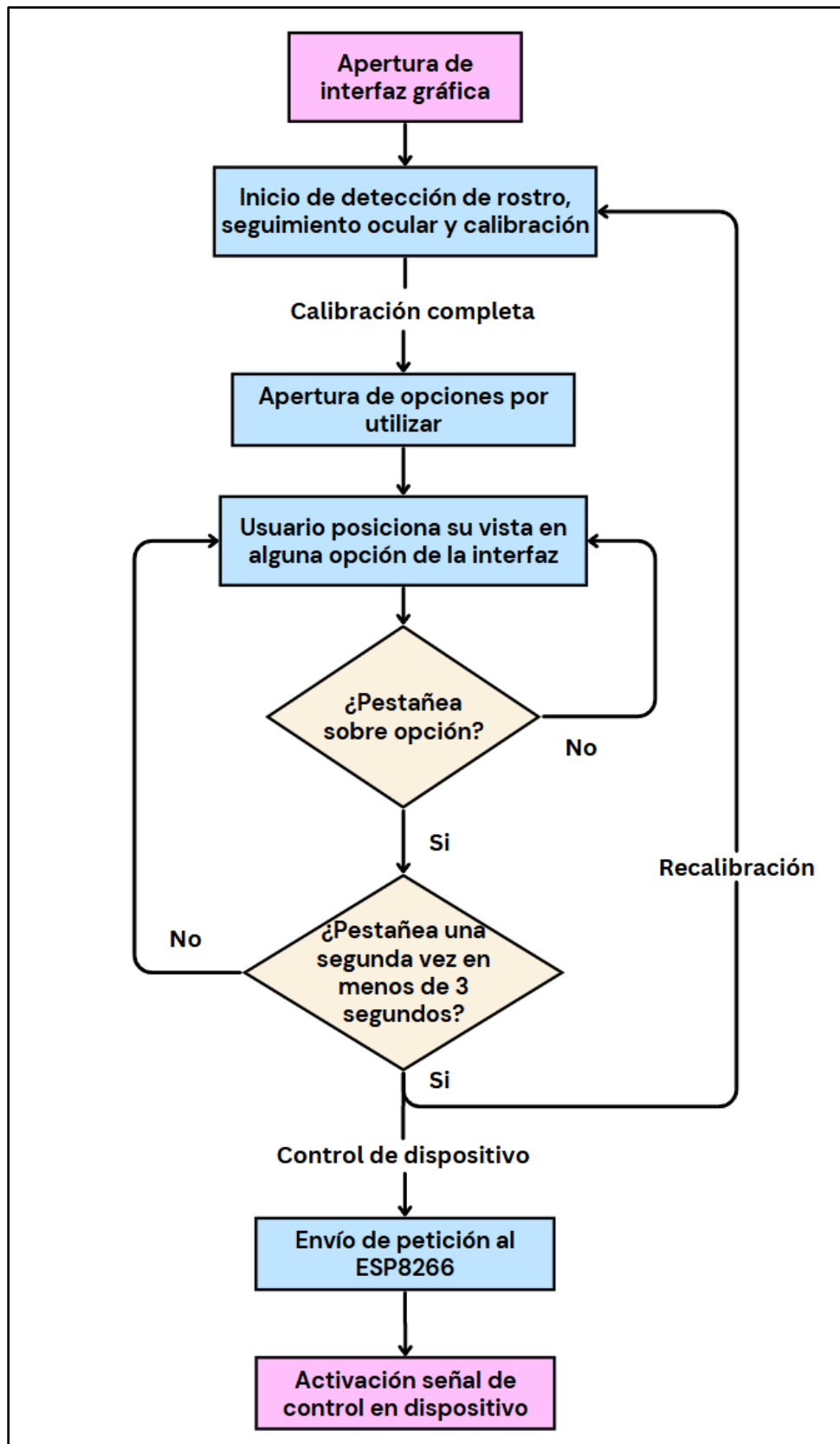


Fig 3.1 Diagrama de flujo de Look2Control

3.1.2 Componentes principales del sistema

A continuación, se describen brevemente los elementos clave involucrados en el sistema.

Tabla 3.1: Componentes de Look2Control

Componente	Función principal
Tablet Windows	Dispositivo principal de interacción. Ejecuta el sistema completo de seguimiento ocular e interfaz gráfica.
Cámara integrada	Permite la captura de video para el análisis de la mirada y los parpadeos del usuario.
Interfaz en Pygame	Genera una pantalla con botones accesibles que responden al seguimiento ocular.
Red Wi-Fi Local	Medio de comunicación entre la tablet y el ESP8266.
ESP8266	Microcontrolador que recibe comandos y controla los actuadores físicos.
Actuadores Físicos	Dispositivos conectados al ESP8266 (LEDs, buzzer, motor, etc.) que responden a los comandos.

3.1.3 Diagrama de bloques

El sistema Look2Control se basa en una arquitectura compuesta por dos bloques principales: el módulo de interfaz y procesamiento, y el módulo de control de dispositivos. Cada uno desempeña un rol específico dentro del flujo de interacción entre el usuario y los elementos físicos del entorno.

En la **Fig 3.2**, se presenta el diagrama de bloques general del sistema. A la izquierda, se observa el proceso de percepción del usuario, representado por el símbolo del ojo, que simboliza el inicio de la interacción. La tablet con cámara integrada actúa como núcleo del bloque de interfaz. Esta tablet ejecuta el software, que incluye los algoritmos de seguimiento ocular y la interfaz gráfica.

La mirada del usuario es captada por la cámara, procesada por el sistema y, si se confirma mediante doble parpadeo, genera una petición HTTP que es enviada a través de la red Wi-Fi. Esta señal es dirigida al microcontrolador ESP8266, que constituye el núcleo del segundo bloque: el control de dispositivos.

El ESP8266 funciona como un servidor HTTP embebido, encargado de recibir las solicitudes, interpretar los endpoints y activar las salidas digitales correspondientes. Estas salidas están conectadas a dispositivos físicos como luz, alarma y otros elementos domóticos, los cuales se activan o desactivan según el comando recibido.

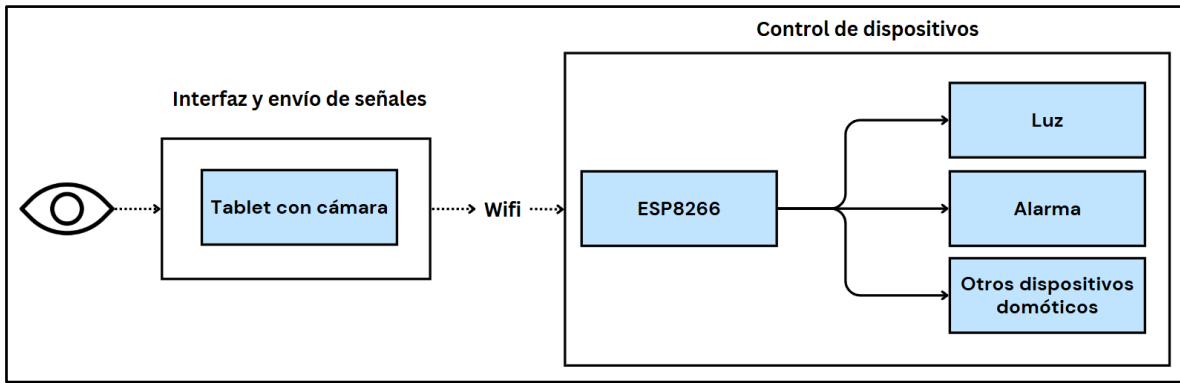


Fig 3.2 Diagrama de bloques de Look2Control

3.2 Desarrollo del módulo de seguimiento ocular

3.2.1 Implementación de la librería EyeGestures

El módulo de seguimiento ocular de Look2Control fue implementado a partir de la librería EyeGestures, seleccionada por sus ventajas de precisión, bajo consumo y compatibilidad con Python. En esta etapa del proyecto, se integraron sus módulos internos directamente al flujo del sistema.

La librería está organizada en varias clases clave. El archivo tracker.py actúa como el núcleo del sistema, coordinando la entrada visual y el análisis ocular. Se combinan técnicas de regresión y evaluación geométrica para emitir eventos de mirada y parpadeo en tiempo real. A nivel funcional, cada fotograma de video es procesado por la clase **EyeTracker**, que encapsula toda la lógica de actualización y exposición de datos. Dentro de este objeto se ejecuta el método **update()**, que evalúa la imagen actual, extrae los landmarks oculares y actualiza variables como se muestra en la **Tabla 3.2**.

Tabla 3.2: Variables clave utilizadas en Look2Control

Variable	Tipo de dato	Función en Look2Control
gaze_point	Tuple (x, y)	Posición estimada de la mirada en pantalla
blink	Boolean	Indica si se detectó un parpadeo
face_detected	Boolean	Confirmación de presencia facial en el cuadro
landmarks	Lista	Coordenadas faciales obtenidas de MediaPipe

Para estimar la mirada, EyeGestures implementa un modelo de regresión lineal múltiple, ajustado durante un proceso de calibración local. Durante el tema de desarrollo de Look2Control, se evaluaron distintos algoritmos de regresión incluidos en la librería, siendo

el modelo Ridge el que ofreció mayor estabilidad frente al ruido, superando alternativas como LassoCV. La elección de este modelo será justificada en la sección 3.2.4 sobre calibración.

Además, la librería incluye un sistema interno de "historial de eventos" que permitió a Look2Control desarrollar una lógica de confirmación por doble parpadeo robusta, basada en ventanas temporales de evaluación. Estos elementos, que serán descritos con más detalle en las siguientes subsecciones, fueron claves para lograr una interfaz accesible, sin falsas activaciones y con alta fiabilidad. Toda la integración se realizó sin modificar la lógica de núcleo de Eye Gestures, respetando la estructura original y los términos de su licencia MIT, que permite el uso con fines académicos con atribución adecuada [13].

A nivel estructural, la librería se compone de tres módulos esenciales: **video_input.py**, encargado de capturar los fotogramas desde la cámara y gestionar resolución y formato; **eye_landmarks.py**, que utiliza MediaPipe para detectar la malla facial completa y extraer los landmarks de los ojos; y **tracker.py**, el núcleo funcional que combina esta entrada con modelos de regresión para estimar la mirada y detectar eventos oculares.

Durante cada iteración, el sistema realiza los siguientes pasos:

- A. Captura de fotograma desde la cámara (con hasta 30 FPS, dependiendo del dispositivo).
- B. Detección de rostro y extracción de malla facial mediante MediaPipe, con más de 400 puntos clave.
- C. Selección de landmarks oculares (normalmente entre 12 y 16 por ojo), incluyendo párpados, iris y esclerótica.
- D. Estimación de la dirección de la mirada, utilizando modelos previamente calibrados.
- E. Evaluación de condiciones geométricas para determinar si hubo parpadeo (por ejemplo, cierre completo del ojo, pérdida de landmarks o reducción de relaciones como el Eye Aspect Ratio).

3.2.2 Detección de la mirada

La detección de la dirección de la mirada en Look2Control se basa en la combinación de geometría ocular y modelos matemáticos de predicción. El sistema analiza los movimientos del ojo captados por la cámara, extrae características relevantes de los landmarks faciales, y utiliza un modelo de regresión lineal múltiple para estimar el punto de atención visual del usuario sobre la pantalla.

En términos matemáticos, la regresión lineal múltiple es una técnica de aprendizaje supervisado que permite estimar una variable de salida continua (en este caso, la coordenada “x” o “y” de la mirada) a partir de un conjunto de variables predictoras relacionadas. Estas variables son características numéricas que describen el estado geométrico del ojo y su

relación con el rostro, y son obtenidas directamente de los landmarks faciales detectados por MediaPipe. Entre las variables predictoras utilizadas sobresalen y se utilizan:

- La distancia vertical entre el párpado superior e inferior (apertura del ojo).
- El desplazamiento horizontal y vertical del iris respecto al centro del ojo.
- La inclinación del eje del ojo respecto al eje horizontal del rostro.
- La posición relativa del iris normalizada según el tamaño del rostro.

Cada una de estas variables proporciona información parcial sobre la orientación de la mirada. El conjunto total de variables se representa como un vector de entrada al modelo de regresión. Este modelo ajusta una ecuación del tipo:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (1)$$

donde:

- \hat{y} es la coordenada estimada en pantalla (horizontal o vertical)
- x_1, x_2, \dots, x_n son las variables extraídas de la geometría ocular.
- y $\beta_1, \beta_2, \dots, \beta_n$ son los coeficientes del modelo, calculados durante el proceso de calibración para ajustar la salida del modelo a los datos reales del usuario.

Estos coeficientes representan el peso de cada variable predictoras en la estimación final. Por ejemplo, si la variable que representa el desplazamiento horizontal del iris tiene un coeficiente alto, eso indica que esa característica influye fuertemente en la estimación de la coordenada x. Lo mismo ocurre para la coordenada y.

Durante el desarrollo del proyecto, se evaluaron distintas técnicas de regresión, y se optó por utilizar **Ridge Regression**, una variante regularizada de la regresión lineal. Esta técnica incorpora una penalización adicional sobre los coeficientes del modelo para mejorar su estabilidad y evitar el sobreajuste. La función objetivo que se minimiza es:

$$Error = \sum (y_i - \hat{y}_i)^2 + \lambda \sum \beta_j^2 \quad (2)$$

Este segundo término en la ecuación (2), conocido como penalización L2, evita que los coeficientes del modelo crezcan excesivamente. Al penalizar la suma de los cuadrados de los coeficientes (β_j^2) el modelo se vuelve más resistente al ruido, especialmente útil en sistemas de visión por computadora donde las posiciones de los landmarks pueden variar ligeramente entre frames consecutivos. El parámetro λ controla la fuerza de esta penalización: si es muy bajo, el modelo se comporta como una regresión normal; si es muy alto, todos los coeficientes tienden a reducirse y el modelo se vuelve más conservador. Esta regularización L2 contribuye

a la estabilidad del sistema, asegurando que pequeñas fluctuaciones en la detección facial no causen saltos bruscos en la estimación de la mirada. En las pruebas realizadas, el modelo Ridge mantuvo un error bajo y estable incluso cuando el usuario se movía ligeramente, o bajo condiciones de iluminación no ideales.

Finalmente, las coordenadas estimadas (x, y) son utilizadas por la interfaz gráfica para determinar si la mirada del usuario coincide con alguno de los botones en pantalla. Esta evaluación se realiza en tiempo real, con una frecuencia promedio de entre 15 y 25 FPS, y permite activar acciones solo si la mirada permanece dentro del área de un botón durante varios frames consecutivos.

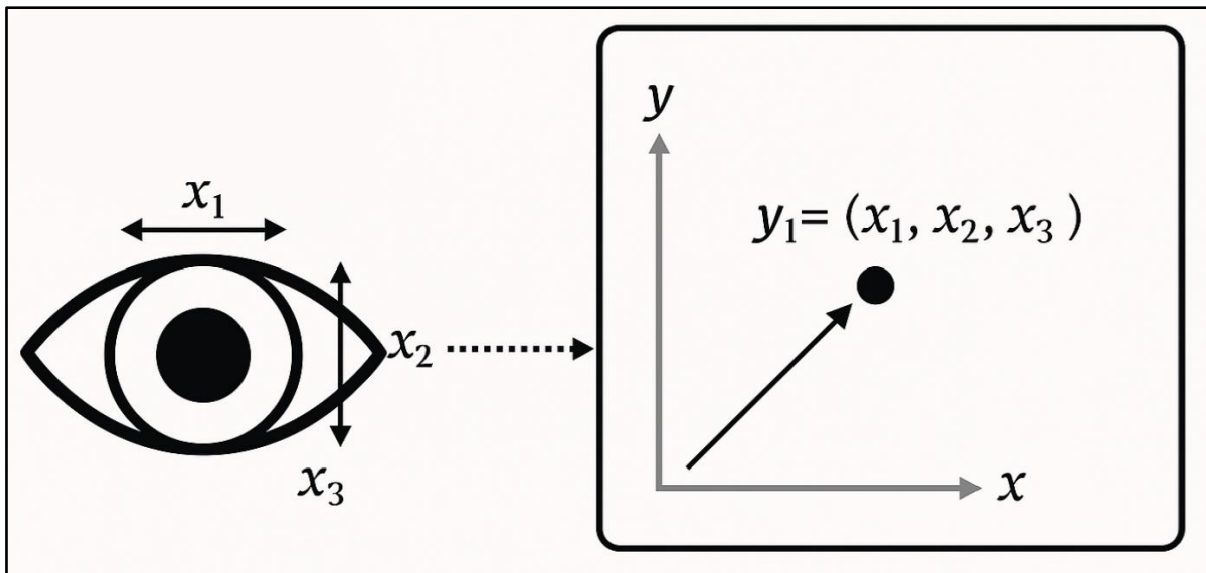


Fig 3.3 Ejemplo visual de punto estimado por algoritmo de regresión

3.2.3 Confirmación por parpadeo

Look2Control utiliza la acción voluntaria del parpadeo como método de confirmación para seleccionar opciones dentro de la interfaz gráfica. Esta técnica se adapta especialmente a usuarios que solo pueden interactuar con su entorno por medio de la mirada, y ha demostrado ser más precisa y accesible que los sistemas que dependen únicamente de fijación visual prolongada. Para evitar activaciones accidentales, se implementó una lógica de confirmación por doble parpadeo, estructurada sobre un modelo temporal de eventos.

La detección del parpadeo es realizada por EyeGestures utilizando dos mecanismos combinados:

- **Pérdida temporal de landmarks:** cuando los ojos se cierran completamente, la red de MediaPipe deja de identificar algunos puntos ya establecidos en los párpados, lo que se interpreta como un parpadeo.
- **Cálculo del Eye Aspect Ratio (EAR):** se trata de una medida geométrica que compara la altura vertical del ojo con su ancho. Cuando esta relación cae por debajo de un umbral definido (usualmente $EAR < 0.2$), se considera que el ojo está cerrado.

Ambos criterios deben cumplirse de manera consistente durante varios frames para validar el evento. Esta doble validación permite distinguir entre parpadeos reales y parpadeos involuntarios causados por fatiga ocular o interferencias.

Una vez detectado un primer parpadeo sobre un botón de la interfaz, el sistema activa un temporizador de 3 segundos durante el cual espera un segundo parpadeo. Si este ocurre dentro de ese intervalo, se interpreta como una confirmación de acción. En caso contrario, el sistema cancela la operación y devuelve el botón a su estado normal. Este segundo parpadeo puede ocurrir en cualquier lugar de la interfaz siempre y cuando sea dentro de ese intervalo de tiempo. Esto se debe a que, a la hora de parpadear, el seguimiento ocular se pierde ya que el ojo está totalmente cerrado. Esto perjudica un doble parpadeo sobre la misma zona, por lo que la lógica de tiempo viene siendo una alternativa mucho más accesible. Durante el periodo de confirmación, la interfaz gráfica indica la espera mediante una señalización visual en los botones, lo cual se explora en la subsección 3.3.

El mecanismo fue implementado mediante variables como se muestra en la **Tabla 3.3**:

Tabla 3.3: Variables para doble parpadeo

Variable	Función
waiting_for_second_blink	Bandera booleana que indica si el sistema espera un segundo parpadeo.
first_blink_time	Momento exacto del primer parpadeo.
first_blink_action	Identificador del botón observado en el primer parpadeo.
last_blink_processed_time	Registro del último parpadeo válido procesado.

Adicionalmente, para evitar errores de interpretación o múltiples activaciones por ruidos en la señal de entrada, se implementó un mecanismo similar al debouncing digital en electrónica. Esto implica ignorar eventos durante una ventana de seguridad posterior a un parpadeo válido, típicamente de 300 ms, lo cual estabiliza el comportamiento del sistema.

Este enfoque aporta las siguientes ventajas:

- **Reducción de falsos positivos:** se minimizan activaciones accidentales por movimientos involuntarios de los párpados.
- **Confirmación activa por parte del usuario:** se requiere una acción consciente para confirmar una selección.

- **Consistencia temporal y espacial:** al exigir que ambos parpadeos ocurran sobre el mismo botón, se refuerza la intención del usuario.

3.2.4 Proceso de calibración

Look2Control requiere una calibración previa para adaptar el seguimiento ocular a las características geométricas del rostro de cada usuario. Este proceso permite que el modelo de estimación de mirada funcione con alta precisión, sin necesidad de sensores especializados ni hardware externo como lo proponen otras alternativas comerciales de mayor precio.

Durante la calibración, el sistema recopila datos emparejando información visual con puntos de referencia en pantalla. A partir de estos datos se ajustan los parámetros del modelo de regresión ya descrito en la sección 3.2.2, el cual se basa en regresión lineal múltiple con penalización L2 (modelo Ridge). En este contexto, el conjunto de entrenamiento está formado por pares del tipo:

$$(x_i, y_i) \tag{3}$$

donde:

- x_i representa el vector de características geométricas del rostro en un instante i . Estas características incluyen:
 - Distancias entre puntos clave de los ojos (ej. entre párpado superior e inferior).
 - Ángulos formados entre líneas oculares y ejes de referencia.
 - Rezones de aspecto (ej. Eye Aspect Ratio - EAR).
 - Coordenadas normalizadas de landmarks proporcionadas por MediaPipe.
- y_i es la posición esperada en pantalla del punto que el usuario está mirando (coordenadas en píxeles dentro de la interfaz). Es también representado como $y_i = [u, v]$ donde u es la posición en el eje horizontal y v en el eje vertical.

En este modelo, u y v se estiman mediante funciones lineales del vector x_i , aplicando ecuaciones del tipo:

$$u = x_i \cdot \beta_u + \varepsilon_u ; v = x_i \cdot \beta_v + \varepsilon_v \tag{4}$$

donde β_u y β_v son vectores de coeficientes entrenados para cada coordenada, y ε representa el error residual. El objetivo es ajustar estos coeficientes para minimizar el error promedio entre las predicciones y las coordenadas reales durante la calibración.

El proceso de calibración se representa de la siguiente manera:

- A. **Presentación de puntos en pantalla:** Se muestra una cuadrícula con múltiples puntos secuenciales en distintas posiciones del área visible. En la versión actual, se utilizan 25 puntos distribuidos uniformemente, abarcando las zonas centrales y periféricas de la interfaz.
- B. **Fijación de la mirada en cada punto:** El usuario dirige su mirada a cada punto. Durante aproximadamente 1.5 segundos por posición, se registran múltiples capturas visuales por segundo (FPS), permitiendo recopilar gran cantidad de datos por punto.
- C. **Captura y asociación de datos:** En cada instante de observación, el sistema forma un par (x_i, y_i) . Estos pares son almacenados para entrenamiento.
- D. **Entrenamiento del modelo:** Al finalizar la secuencia de puntos, se ajusta un modelo Ridge para predecir u y v como funciones lineales de x_i . Los coeficientes resultantes se almacenan para su uso durante la detección de mirada en tiempo real.

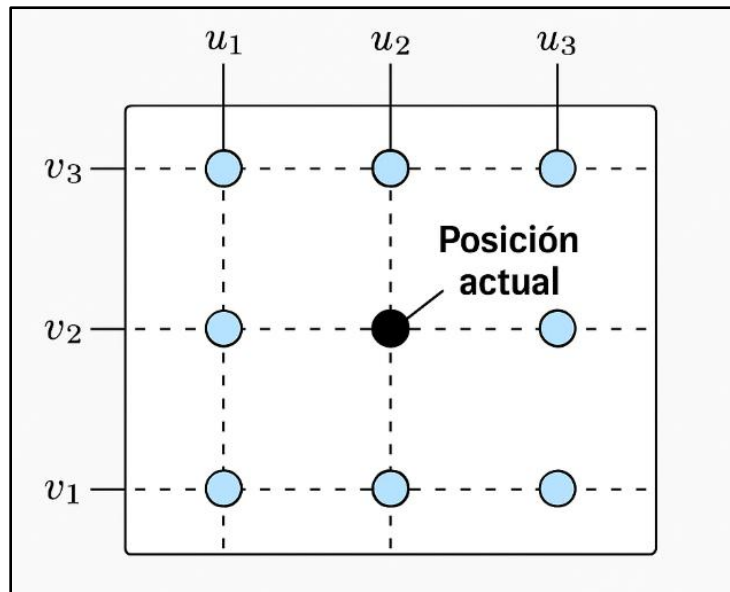


Fig 3.4 Posición de la mirada con respecto a los ejes de la pantalla

3.3 Construcción de la interfaz gráfica

Uno de los componentes fundamentales del sistema Look2Control es la interfaz gráfica, la cual actúa como medio de interacción entre el usuario y el entorno que desea controlar. Esta interfaz ha sido diseñada para que su uso no requiera movimiento físico alguno más allá de los ojos. A través de ella, el usuario puede emitir comandos mediante la confirmación por doble parpadeo.

El desarrollo de esta interfaz se construyó tomando como referencia las buenas prácticas y estándares internacionales de accesibilidad universal, particularmente las recomendaciones de las Web Content Accessibility Guidelines (WCAG) [7]. Estos lineamientos orientaron decisiones como el contraste de colores, el tamaño de los elementos visuales, la disposición de los botones, la claridad del lenguaje y la simplicidad de navegación. Se incorporó también un logo gráfico representativo en el encabezado, ubicado a la izquierda, reforzando la identidad visual del sistema al diseño.

3.3.1 Criterios visuales de accesibilidad

Uno de los pilares fundamentales de la accesibilidad visual es la perceptibilidad, entendida como la capacidad de los usuarios para identificar, distinguir y comprender los elementos presentes en pantalla. Para lograrlo, se implementaron contrastes adecuados entre el texto y el fondo, y se utilizaron colores uniformes para diferenciar tipos de acciones, como lo establece el criterio 1.4.3 de las WCAG 2.1 [7], que recomienda una relación mínima de contraste de 4.5:1 entre texto e imagen de fondo.

En Look2Control, el texto negro sobre botones de colores claros garantiza un contraste suficiente, como. Adicionalmente, se reservaron colores distintivos para funciones críticas: el botón “Botón de Emergencia” utiliza una base rosada con borde rojo oscuro y el botón “Salir” se diferencia con un azul celeste y borde azul profundo, lo que permite al usuario reconocerlas como opciones especiales. Este esquema de colores de muestra en la **Tabla 3.4**.

Tabla 3.4 Esquema de colores del sistema

Función	Color base	Color de borde	Descripción
Botones Generales	Gris claro	Negro	Botones estándares
Botón de Emergencia	Rosado pálido	Negro	Botón especial de emergencia
Botón Salir	Celeste	Azul oscuro	Botón especial de cierre
Confirmación activa	-	Rojo	Señal de confirmación de acción en espera
Enfoque de mirada	-	Azul	Señaliza la atención de la mirada

Las WCAG recomiendan que los componentes interactivos deben tener una zona activa mínima de 44 x 44 píxeles (criterio 2.5.5) para ser fácilmente operables por personas con movilidad limitada o utilizando ayudas técnicas [7]. En el caso de Look2Control, los botones fueron diseñados con un tamaño promedio de 300 x 250 píxeles, lo que supera ampliamente este umbral, esto debido a que se trata de una interacción por seguimiento ocular.

La disposición se realizó en una cuadrícula de 3 filas por 4 columnas, con márgenes constantes y separación simétrica, permitiendo que la mirada del usuario pueda desplazarse cómodamente entre los botones sin superposiciones ni falsas activaciones. Esta distribución evita aglomeración de información, respetando el principio de comprensibilidad espacial, como se muestra en la **Fig 3.5**.

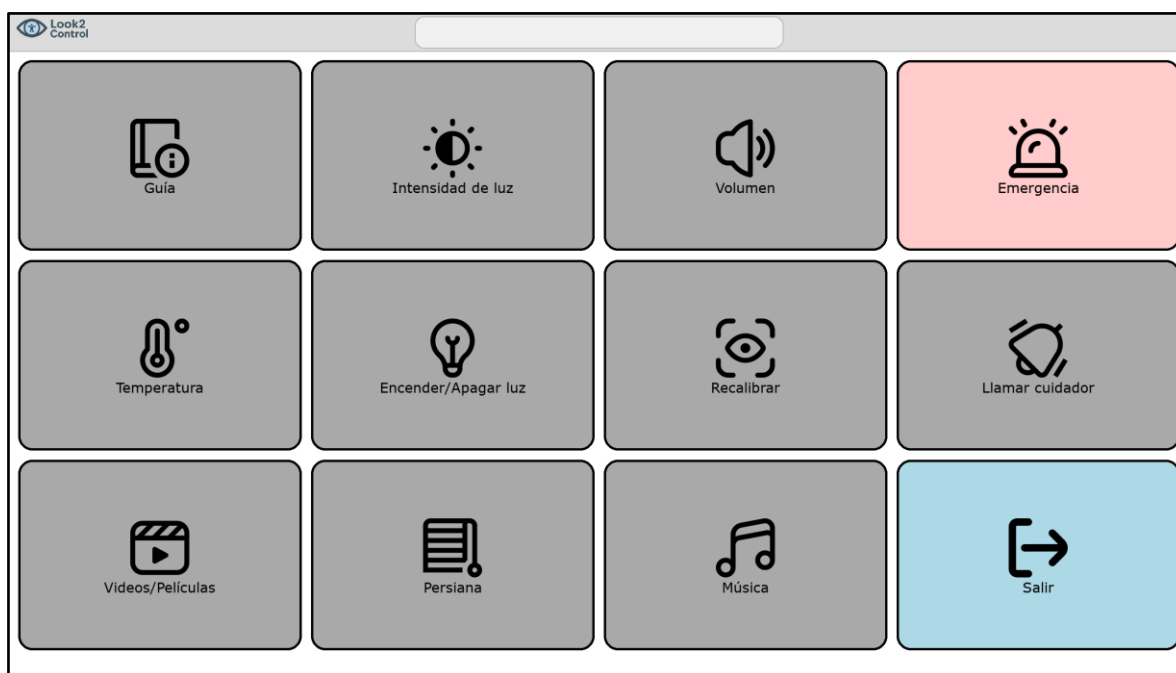


Fig 3.5 Interfaz y estructura de botones para Look2Control

Cada botón incluye un ícono de 128x128 píxeles, acompañado por un texto en fuente “Verdana” de 32 puntos, ubicado debajo del ícono. La elección de Verdana, una fuente “sans-serif” recomendada para accesibilidad por su forma clara y espaciado legible, responde a lo indicado por la WCAG sobre el uso de fuentes fácilmente distinguibles (recomendación general bajo el principio 3.1: comprensibilidad textual) [7]. Esta combinación de ícono y texto permite que los usuarios puedan reconocer el propósito de cada botón por redundancia semántica: reconocimiento visual del símbolo y confirmación textual.

3.3.2 Retroalimentación visual y lógica de interacción

La retroalimentación visual es un componente esencial en la interfaz del sistema, ya que cumple la función de confirmar al usuario que su acción ha sido reconocida correctamente por él. Dado que el único canal de salida del usuario es la mirada y el parpadeo, resulta crítico proporcionar respuestas claras e inmediatas que reduzcan la ambigüedad y validen su intención y refuercen su control sobre el entorno. El sistema implementa distintos tipos de retroalimentación visual, cada uno asociado a diferentes etapas del flujo de interacción:

- A. **Enfoque de la mirada:** Cuando el usuario fija la mirada sobre un botón, este cambia el color de su borde a azul. Esta señal indica al usuario que el sistema ha identificado su atención sobre un elemento interactivo, y que puede iniciar una acción si lo desea.

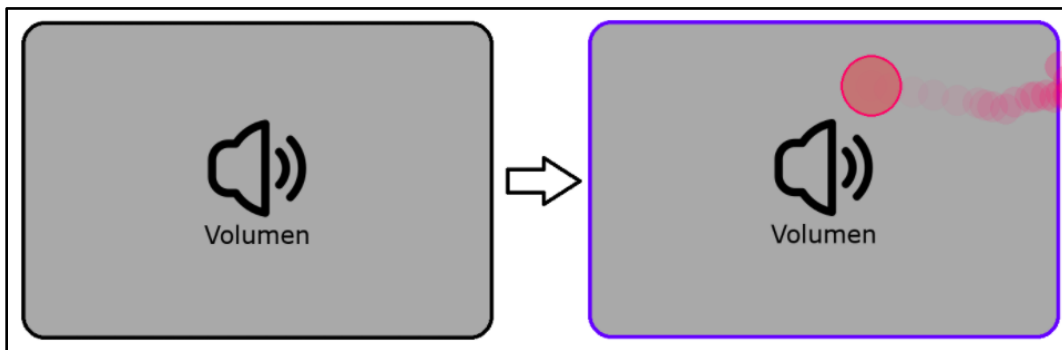


Fig 3.6 Cambio de borde al fijar la mirada

- B. **Primer parpadeo (espera de confirmación):** Una vez que se detecta un parpadeo mientras la mirada se encuentra sobre un botón, el sistema inicia un temporizador de tres segundos. Durante este periodo, el borde del botón cambia a rojo, indicando que se está esperando una confirmación por medio de un segundo parpadeo. Si el segundo parpadeo no ocurre dentro del tiempo establecido, el borde del botón vuelve a su color negro predeterminado y la acción se cancela. Esta lógica temporal imita el comportamiento de un sistema con “debouncing digital”, el cual resiste activaciones por error.



Fig 3.7 Borde de espera de confirmación

- C. **Activación exitosa:** Cuando el segundo parpadeo ocurre dentro del intervalo de confirmación, se ejecuta la acción correspondiente al botón y el sistema muestra un mensaje temporal centrado en la parte superior de la pantalla, dentro de un recuadro rectangular de fondo claro y bordes grises, indicando la acción realizada. Este mensaje tiene una duración de aproximadamente 4 segundos y está diseñado con contraste adecuado para que sea fácilmente legible sin interferir con el resto de la interfaz.



Fig 3.8 Mensaje de acción confirmada

Según los principios de operabilidad definidos por las Web Content Accessibility Guidelines (WCAG 2.1), es esencial que una interfaz proporcione retroalimentación visual clara e inmediata al usuario después de cualquier acción, para reforzar la confianza en el sistema y reducir el error por ambigüedad o falta de respuesta perceptible [7]. En Look2Control, este principio se implementa para compensar la falta de otros canales de interacción (como el tacto o el habla), asegurando que cada comando recibido tenga una respuesta visual.

3.3.3 Descripción funcional de la interfaz

La interfaz de Look2Control fue concebida como un entorno visual cuyo propósito es permitir al usuario ejecutar acciones concretas a través del seguimiento ocular y la confirmación por parpadeo. Su diseño está garantizado que cada área tenga un propósito específico. La organización visual se compone de tres bloques funcionales claramente diferenciados: encabezado, cuerpo principal y la representación visual del punto de mirada.

3.3.3.1 Encabezado superior

El encabezado ocupa una franja horizontal fija en la parte superior de la pantalla y cumple funciones informativas y visuales clave:

- **Identificación del sistema:** A la izquierda se ubica un logotipo representativo de Look2Control, dando identidad al sistema.
- **Área de mensajes temporales:** En el centro del encabezado se encuentra un rectángulo redondeado donde se muestran mensajes relacionados con acciones realizadas. Este elemento es permanente en pantalla (aunque su contenido solo se actualiza si se ejecuta una acción).
- **Separador visual:** Una línea gris tenue separa el encabezado del resto de la interfaz, sirviendo como límite visual entre zonas funcionales.











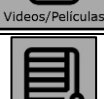



Fig 3.9 Encabezado de la interfaz

3.3.3.2 Cuerpo principal de interacción

El área central contiene los botones principales de control. Ya se ha descrito su estructura visual en la sección 3.3.1(incluyendo colores, fuentes, márgenes y disposición). Sin embargo, desde el punto de vista funcional, cada botón representa una instrucción concreta que activa una función física, digital o informativa. Estas funciones son gestionadas en segundo plano, y sus resultados se reflejan visualmente mediante retroalimentación (ver sección 3.3.2). A continuación, se presenta una tabla detallada con las acciones disponibles en la interfaz:

Tabla 3.5 Funciones de botones en la interfaz

Botón	Identificador	Función ejecutada
Guía		Muestra una ventana flotante con instrucciones detalladas del sistema-
Intensidad de Luz		Ajusta el brillo de un LED externo.
Volumen		Cambia entre niveles de volumen de la música.

Emergencia		Envía un correo electrónico de emergencia y notifica a través de sonido.
Temperatura		Muestra en pantalla el valor del sensor LM35 conectado al ESP8266.
Encender/Apagar Luz		Activa o desactiva el LED externo mediante un pulso digital.
Recalibrar		Inicia nuevamente la calibración del sistema del seguimiento ocular.
Emergencia Sonora		Activa una señal acústica utilizando un buzzer piezoeléctrico.
Videos/Películas		Abre el navegador en una página de entretenimiento
Persiana		Activa el motor paso a paso 28BYJ-48 + ULN2003.
Música		Inicia o detiene la reproducción de la música.
Salir		Cierra la aplicación Look2Control.

Nota: La función “Videos/Películas”, que originalmente abría YouTube como entretenimiento, fue deshabilitada por razones de seguridad, ya que su ejecución interrumpe el sistema de seguimiento ocular y requiere asistencia de un cuidador para retornar a la interfaz principal. Queda previsto en el sistema para futura activación supervisada.

3.3.3.3 Representación visual del punto de mirada

Durante el funcionamiento del sistema, se dibuja un círculo principal de color rojo en la posición estimada donde el usuario está mirando. Este círculo representa el resultado directo del proceso de cálculo descrito en la sección 3.2.2, donde el modelo de regresión lineal múltiple, entrenado con datos de calibración de cada usuario, predice las coordenadas visuales a partir de las variables geométricas faciales extraídas en tiempo real. El objetivo de este marcador visual es proporcionar una referencia inmediata y continua del punto de enfoque actual, como funcional los ratones de computadoras.

Esta predicción se actualiza para cada nuevo fotograma de la cámara (con una tasa promedio de entre 15 y 30 FPS), y permite observar de forma clara hacia qué región de la pantalla dirige su atención el usuario y ofrece retroalimentación directa del comportamiento del sistema. Además del punto actual, el sistema traza una estela visual de color magenta,

formada por una secuencia de círculos translúcidos que representan los últimos 20 puntos de mirada detectados. Esta estela permite analizar la estabilidad y comportamiento del seguimiento ocular. La opacidad de los puntos disminuye progresivamente conforme se alejan en el tiempo, simulando un efecto de rastro.

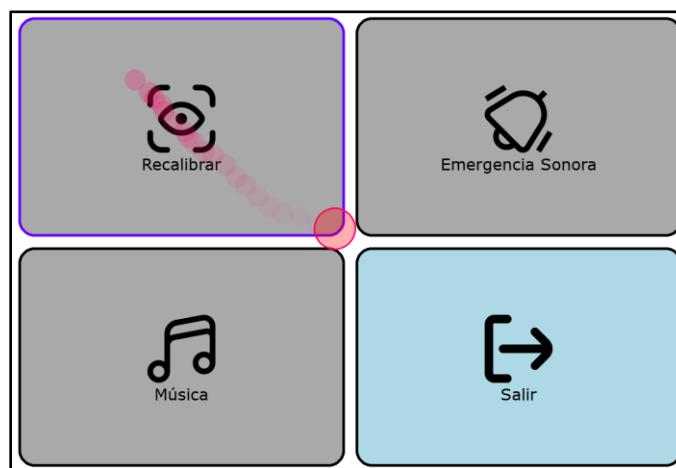


Fig 3.10 Representación visual de la mirada en pantalla

3.4 Implementación de la comunicación con el microcontrolador

Uno de los componentes fundamentales del sistema Look2Control es la capacidad de controlar dispositivos físicos reales a través de señales digitales. Para ello, fue necesario establecer un canal de comunicación entre la interfaz gráfica (ejecutada en una tablet con Windows) y el microcontrolador ESP8266, responsable de activar actuadores como luces, alarmas o motores.

3.4.1 Comunicación mediante protocolo HTTP

La comunicación entre la interfaz gráfica del sistema y el microcontrolador ESP8266 se basa en el uso del protocolo HTTP (Hypertext Transfer Protocol), bajo una arquitectura cliente-servidor. En esta configuración, la tablet que ejecuta la interfaz actúa como cliente HTTP, mientras que el ESP8266 opera como un servidor HTTP embebido, preparado para recibir solicitudes desde la red local y ejecutar comandos físicos en respuesta.

Este modelo permite establecer un canal de comunicación simple y directo y ampliamente soportado, evitando la necesidad de usar protocolos más complejos como

MQTT o WebSockets. HTTP es solución óptima para un entorno de control visual en tiempo real, donde cada comando debe ser enviado sin necesidad de mantener conexiones persistentes.

El flujo básico de comunicación del sistema es el siguiente:

- A. El usuario dirige su mirada hacia un botón de la interfaz y realiza un doble parpadeo para confirmar su intención.
- B. La interfaz detecta la confirmación y ejecuta la función correspondiente, que prepara una solicitud HTTP tipo GET con un comando específico.
- C. El ESP8266, funcionando como servidor, recibe esta solicitud, interpreta la ruta accedida y ejecuta la acción física asociada, como encender una luz o activar una alarma.

Este intercambio ocurre a través de una red Wi-Fi local, y la dirección IP del ESP8266 debe ser conocida y accesible desde la tablet. Una vez establecida la conexión, la comunicación es inmediata y sincrónica, con una latencia imperceptible para el usuario final.

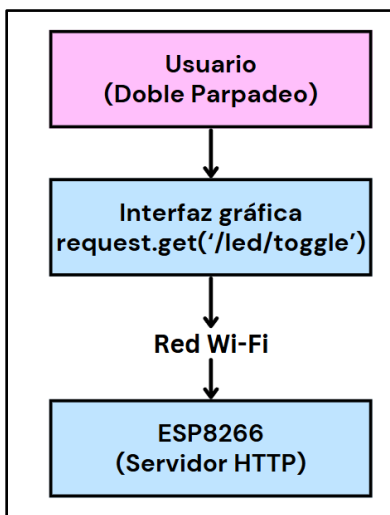


Fig 3.11 Diagrama de flujo de comunicación por HTTP

3.4.2 Cliente HTTP en Python y estructura de comandos

Como ha sido mencionado, la interfaz gráfica ejecutada en Python cumple la función de cliente dentro del modelo de comunicación. Su responsabilidad es enviar comandos que son transmitidos mediante solicitudes HTTP de tipo GET, que viajan a través de la red Wi-Fi local hasta el microcontrolador, el cual actúa como servidor web.

La implementación del cliente HTTP en Python se realiza utilizando la librería estándar **requests**. La documentación oficial describe requests como una librería "elegante y simple para HTTP, construida para uso humano" [18]. Toda acción seleccionada dentro de Look2Control está vinculada a un endpoint definido en el microcontrolador. Los comandos enviados al microcontrolador siguen un formato estructurado basado en endpoints URL simples. Cada botón de la interfaz, al ser activado mediante doble parpadeo, desencadena una solicitud HTTP con un path específico que identifica la acción deseada. Por ejemplo:

- `http://(dirección IP)/led/toggle`
- `http://(dirección IP)/alarm/on`

Este tipo de comandos se pueden ver representados en la **Fig 3.12** Este código define la dirección IP del ESP8266 y el endpoint deseado, envía una solicitud HTTP GET, verifica la respuesta e imprime mensajes de error si hay fallos en la red. Del lado del ESP8266, traduce cada endpoint para realizar una acción sobre un pin determinado del microcontrolador.

```
import requests

def toggle_led_externo():
    try:
        url = "http://(Your IP)/led/toggle"
        response = requests.get(url, timeout=1)
        if response.status_code == 200:
            print("LED toggled successfully.")
        else:
            print("Error: Status code", response.status_code)
    except Exception as e:
        print("Failed to send command:", e)
```

Fig 3.12 Estructura de comandos HTTP tipo GET

Para garantizar que la interfaz gráfica de Look2Control se mantenga fluida durante la ejecución de comandos, se implementó un mecanismo de ejecución en segundo plano mediante un decorador el cual se llamó **@en_fondo**. En Python, un decorador es una estructura que permite modificar el comportamiento de una función sin alterar su definición original.

Este decorador se apoya en un **ThreadPoolExecutor**, que crea un grupo de hilos donde se pueden lanzar tareas en paralelo. Gracias a esto, funciones como el envío de comandos al microcontrolador o la reproducción de audio pueden ejecutarse de forma independiente al hilo principal, evitando bloqueos o interrupciones visuales.

Al aplicar **@en_fondo** sobre una función, el sistema delega su ejecución a un hilo secundario. Esto permite que las acciones físicas se realicen de manera eficiente sin afectar la interacción continua del usuario con la interfaz. Esta técnica es clave para mantener una experiencia de uso responsiva, especialmente en contextos de accesibilidad donde los tiempos de respuesta son críticos.

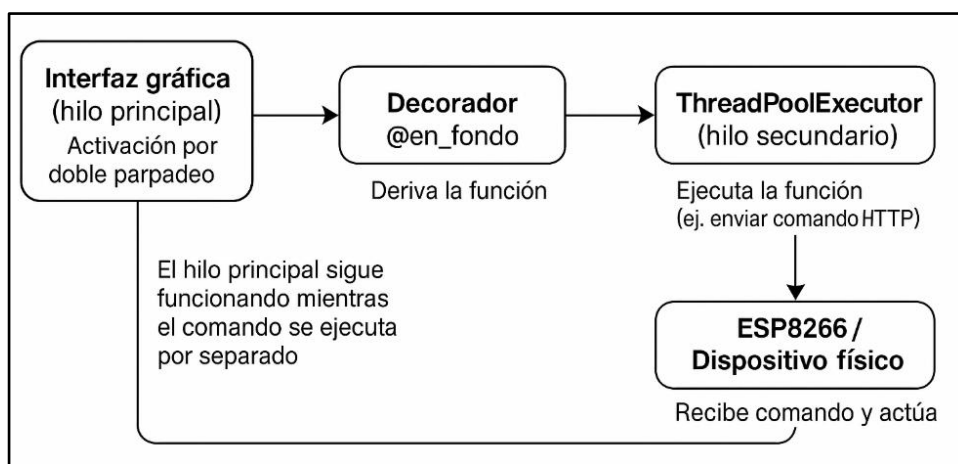


Fig 3.13 Funcionamiento del decorador **@en_fondo**

3.5 Programación del ESP8266 como servidor de control

El microcontrolador NodeMCU ESP8266 fue programado para desempeñar el rol de servidor embebido dentro del sistema Look2Control. Su función principal es recibir comandos provenientes de la interfaz gráfica, interpretarlos de acuerdo con su contenido y ejecutar acciones físicas sobre los dispositivos conectados a sus salidas digitales. Esta lógica de control embebido constituye el núcleo que permite la interacción entre el usuario y su entorno.

La elección del ESP8266 responde a múltiples factores, entre ellos: su bajo consumo energético, su capacidad de conexión Wi-Fi integrada, número de pines GPIO para el control de dispositivos y su bajo precio, lo cual lo hace ideal para la accesibilidad económica. La configuración inicial establece la conexión a una red Wi-Fi y lanza un servidor en el puerto 80. Una vez conectado, el ESP8266 queda a la espera de solicitudes del tipo GET. El sistema

interpreta las rutas (endpoints) incluidas en la URL para ejecutar acciones específicas, como encender una luz o emitir un sonido mediante un buzzer.

Una vez configurado en el entorno de desarrollo, el firmware cargado en el ESP inicia estableciendo la conexión a una red Wi-Fi predefinida. Tras lograr una conexión exitosa, se activa un servidor HTTP en el puerto 80. Este servidor se mantiene en escucha continua de peticiones entrantes desde la red local. La estructura del código sigue un patrón sencillo. En **el setup()**, se configura la conexión Wi-Fi, el inicio del servidor y la inicialización de los pines GPIO. Luego, dentro del **loop()**, se escucha continuamente por nuevas conexiones HTTP. Cada vez que se recibe una solicitud, se analiza la ruta solicitada y se ejecuta la lógica correspondiente al dispositivo asociado. Un ejemplo de código de activación se observa en la **Fig 3.14** Ejemplo de activación de dispositivo:

```
if (request.indexOf("/led/toggle") != -1) {
  estadoLEDInterno = !estadoLEDInterno;
  digitalWrite(ledInternoPin, estadoLEDInterno ? LOW : HIGH);
  Serial.print("LED interno ahora está: ");
  Serial.println(estadoLEDInterno ? "ENCENDIDO" : "APAGADO");
}
```

Fig 3.14 Ejemplo de activación de dispositivo

Cada vez que el servidor recibe una solicitud HTTP del tipo GET, se analiza el contenido de la URL para determinar la acción solicitada. El código embebido identifica comandos específicos como **/led/toggle**, **/buzzer/alarma** o **/led/pwm**, y ejecuta una acción correspondiente sobre el pin digital asociado. Estas acciones implican tanto activaciones binarias (encendido/apagado) como señales moduladas por ancho de pulso (PWM) en casos como el control de brillo .

La lógica se estructura mediante condicionales que interpretan la ruta de cada solicitud. Todos los comandos son acompañados por respuestas HTTP, adecuadas para su integración con la interfaz gráfica, como se puede ver en la **Tabla 3.6**, donde se resume la correspondencia entre los comandos HTTP implementados, las acciones internas ejecutadas por el ESP8266 y el dispositivo físico involucrado.

Tabla 3.6 Comandos y dispositivos controlados en ESP8266

Comando HTTP	Acción en ESP8266	Dispositivo controlado
/led/toggle	Cambia el estado del LED externo	LED externo
/led/pwm	Incrementa el brillo del LED por PWM	LED externo con control
/buzzer/alarma	Reproduce una secuencia sonora de alarma	Buzzer piezoeléctrico
/motor/toggle	Secuencia de pasos en sentido horario (CW) o antihorario (CCW)	Motor paso a paso
/temp	Devuelve temperatura en Celsius	Sensor LM35DZ

La lógica se estructura mediante condicionales que interpretan la ruta de cada solicitud. Por ejemplo, al recibir el comando **/led/toggle**, se invierte el estado del LED mediante una señal digital. En el caso del buzzer, el comando **/buzzer/alarma** activa una secuencia sonora específica. El comando **/led/pwm** incrementa progresivamente el nivel de brillo del LED, utilizando la función **analogWrite()** sobre un pin PWM. Para la lectura del sensor LM35DZ, el endpoint **/temp** devuelve la temperatura como un número decimal en grados Celsius.

En el caso del motor paso a paso 28BYJ-48, el firmware está preparado para ejecutar secuencias de pasos en sentido horario o antihorario (**/motor/toggle**) a través de un controlador ULN2003. Estas secuencias permiten simular el movimiento de apertura y cierre de una persiana controlada por el sistema.

3.6 Ensamblaje y configuración de prototipo

El prototipo del sistema Look2Control fue ensamblado mediante una arquitectura modular basada en una placa NodeMCU ESP8266, dispuesta sobre una protoboard de 840 puntos que facilita conexiones organizadas y sin soldadura. En esta etapa se realizó la interconexión de todos los módulos físicos definidos durante el diseño, incluyendo sensores, actuadores y salidas de control. La disposición del montaje sigue principios de depuración sencilla, permitiendo validar físicamente la lógica implementada por software.

La alimentación del sistema se divide en dos niveles. El ESP8266 se alimenta mediante su conector micro USB, utilizado tanto para programación como para operación estándar. Sin embargo, para ciertos actuadores que requieren mayor corriente, como el motor paso a paso, se integró una fuente externa de 5 V dedicada, conectada al módulo ULN2003 y con GND común con el ESP. Esta estrategia evita caídas de tensión y protege al microcontrolador frente a sobrecargas.

Las conexiones se realizaron con cables jumper tipo macho-macho y macho-hembra, procurando un enrutamiento limpio, señal GND compartida y protección por resistencias en los elementos sensibles. Los valores resistivos fueron seleccionados considerando las

características de los GPIOs (3.3 V máx., corriente nominal de salida <12 mA) y la caída de tensión de cada LED, buzzer, sensor o motor. El total de componentes utilizado para la realización completa del sistema se ve evidenciado en la Tabla 3.7.

Tabla 3.7 Desglose de componentes físicos utilizados en Look2Control

Componente	Cantidad	Costo
ESP8266 NodeMCU	1	\$10.35
Protoboard 840 puntos	1	\$15.08
Jumpers	1 kit	\$1.89
Motor 28BYJ-48	1	\$7.52
ULN2003 Driver	1	\$7.47
Sensor de temperatura LM35DZ	1	\$3.59
Buzzer piezoeléctrico	1	\$3.69
LEDs	1 kit	\$1.99
Resistencias variadas	1 kit	\$0.71
Fuente 5V	1	\$9.10
Tablet Lenovo Yoga Book	1	\$150.94
Total aproximado	-	\$ 211.32

3.6.1 Organización del montaje físico y conexión de componentes

La distribución del montaje fue diseñada para representar de forma fiel la arquitectura lógica implementada en software. El conjunto de componentes conectados incluye los siguientes actuadores, sensores y elementos de salida visual o sonora:

- Un LED externo encendido mediante señal digital.
- Un LED controlado por PWM (Modulación por Ancho de Pulso) para ajuste gradual de brillo.
- Un buzzer piezoeléctrico como elemento de retroalimentación acústica.
- Un sensor analógico de temperatura LM35DZ.
- Un motor paso a paso 28BYJ-48, gobernado por el controlador ULN2003 y una fuente externa.

Cada uno de estos módulos se conecta al ESP8266 a través de pines GPIO específicos, cuidadosamente seleccionados para evitar conflictos con los modos de arranque, mantener compatibilidad eléctrica y aprovechar la simetría en la disposición de pines del microcontrolador. En la **Fig 3.15** se presenta el esquemático general del sistema físico implementado, utilizado como referencia base para el ensamblaje del prototipo:

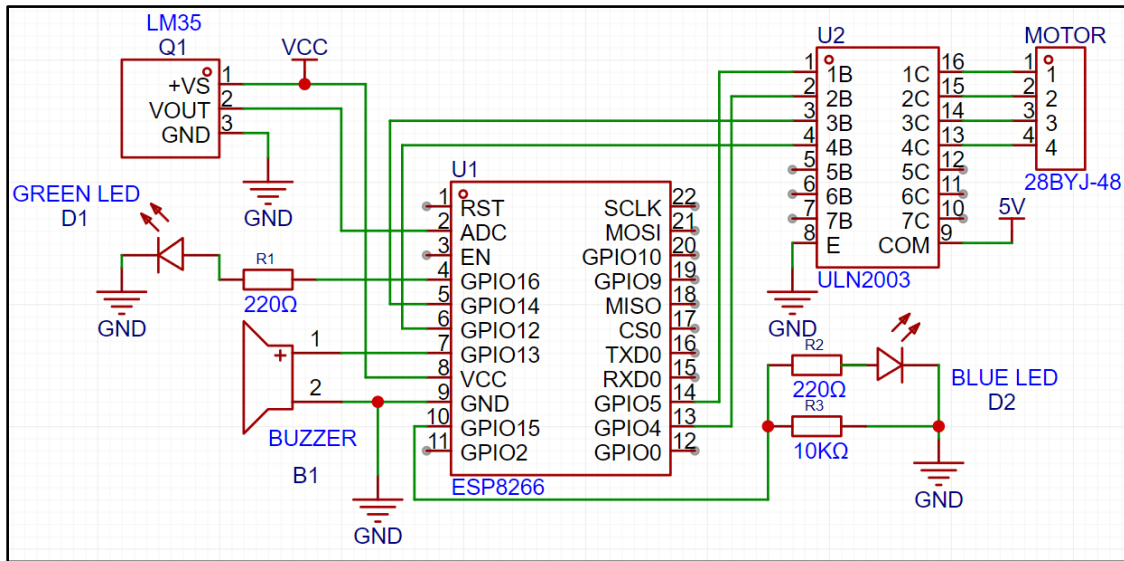


Fig 3.15 Esquemático completo del sistema físico Look2Control

3.6.2 Asignación de pines y conexiones

La asignación de pines del microcontrolador ESP8266 se realizó tomando en cuenta criterios de compatibilidad, disponibilidad y estabilidad. Cada componente fue conectado a un GPIO específico que garantizara su correcto funcionamiento sin interferir con funciones reservadas del sistema, como el arranque, la comunicación USB o el acceso al convertidor analógico.

La Tabla 3.8 resume la asignación completa de pines, funciones y características técnicas asociadas a cada conexión:

Tabla 3.8 Asignación de pines y funciones del ESP8266

Componente	GPIO	Alias	Tipo de señal	Función
Green LED	GPIO16	D0	Digital OUT	Toggle ON/OFF
Blue LED	GPIO15	D8	PWM OUT	Control de brillo
Buzzer	GPIO13	D7	Digital OUT	Secuencia de tonos de alarma
LM35DZ	A0	ADC	Analógica IN	Lectura de temperatura
Motor IN1	GPIO5	D1	Digital OUT	Activación bobina A
Motor IN2	GPIO4	D2	Digital OUT	Activación bobina B
Motor IN3	GPIO14	D5	Digital OUT	Activación bobina C
Motor IN4	GPIO12	D6	Digital OUT	Activación bobina D

3.6.2.1 Módulos de iluminación LED: digital y PWM

El LED digital consiste en un diodo emisor de luz color verde, conectado al pin GPIO16 (D0) del ESP8266 y dispuesto en la protoboard como representación funcional de un sistema de iluminación domótica básico, como una lámpara de habitación. Su activación se realiza mediante señal digital directa de tipo toggle, que cambia el estado lógico del pin entre niveles alto (HIGH) y bajo (LOW), reflejando visualmente el comando emitido desde la interfaz del sistema.

Desde el punto de vista eléctrico, el LED se conecta en serie con una resistencia de 220Ω , la cual limita la corriente de conducción conforme a los márgenes de seguridad del GPIO. El valor de la resistencia se obtuvo considerando una caída de tensión de 2.0 V en el LED y una corriente objetivo de 6 mA :

$$R = \frac{V_{GPIO} - V_{LED}}{I} = \frac{3.3\text{V} - 2.0\text{V}}{0.006\text{A}} \approx 217\Omega \quad (5)$$

Se seleccionó el valor comercial de 220Ω . La activación del pin GPIO16 es gestionada desde la interfaz de usuario mediante el botón “Encender/Apagar Luz”, el cual ejecuta una alternancia de estado cada vez que se detecta una secuencia de doble parpadeo sobre el área correspondiente. La respuesta del sistema es inmediata y visible en el entorno físico.

El segundo módulo de iluminación está compuesto por un LED azul conectado al pin GPIO15 (D8), también dispuesto sobre la protoboard en serie con una resistencia de 220Ω . En este caso, la activación no se realiza mediante una señal binaria fija, sino mediante una señal de modulación por ancho de pulso (PWM) generada por el microcontrolador. Esta señal alterna rápidamente entre niveles lógicos alto y bajo con un periodo constante, variando el tiempo activo dentro de cada ciclo para modificar la potencia media entregada al LED.

El parámetro relevante en esta configuración es el ciclo de trabajo (duty cycle D), el cual define la proporción del periodo en la que la señal permanece en nivel alto:

$$D = \frac{t_{on}}{T} \cdot 100\% \quad (6)$$

donde t_{on} es el tiempo de activación (HIGH) y T es el periodo total del ciclo. A medida que D aumenta, el brillo percibido del LED también se incrementa, debido a la persistencia de la visión humana. Esta estrategia permite simular una lámpara de intensidad variable, con eficiencia energética y sin requerir conversión digital-analógica adicional.

En sistemas digitales, un resistor pull-down es una resistencia conectada entre un pin de entrada y tierra (GND) que asegura que dicho pin mantenga un nivel lógico bajo (LOW) cuando no está activamente siendo impulsado por otra señal. Su función principal es evitar estados flotantes, los cuales pueden causar comportamientos erráticos en los microcontroladores durante el arranque. Este tipo de resistencias es fundamental en pines que determinan el modo de arranque (boot mode).

En este prototipo, se conectó una resistencia de 10 k Ω entre el pin GPIO15 (D8) y GND como pull-down, garantizando que el pin se mantenga en nivel bajo durante el encendido del microcontrolador. Esta medida es necesaria para asegurar un arranque correcto del ESP8266, ya que si el pin queda flotando o en nivel alto, el dispositivo podría no iniciar adecuadamente según lo menciona su ficha de datos [16]. Durante la operación, esta resistencia no interfiere con la señal PWM aplicada al LED azul conectado a ese pin.

El botón correspondiente en la interfaz gráfica es “Intensidad de Luz”. Cada secuencia válida de doble parpadeo sobre este botón incrementa el valor del ciclo de trabajo en una escala discreta predefinida. Una vez alcanzado el valor máximo, el sistema reinicia el ciclo de ajuste, permitiendo al usuario explorar distintos niveles de intensidad luminosa desde la interfaz de control.

3.6.2.2 Módulo buzzer piezoeléctrico

El buzzer piezoeléctrico fue incorporado al prototipo como actuador acústico, con el fin de simular una señal sonora de advertencia o emergencia. Este componente está conectado al pin GPIO13 (D7) del ESP8266, configurado como salida digital. La activación del buzzer se realiza mediante una señal pulsante generada por software, utilizando la función **tone()** del entorno Arduino, la cual permite aplicar una frecuencia específica sobre el pin de salida, haciendo vibrar el elemento piezoeléctrico del transductor.

El dispositivo responde adecuadamente dentro del rango de frecuencias audibles (20 Hz a 20 kHz), y en este caso, se seleccionó una alternancia entre 1000 Hz y 2000 Hz para producir un patrón de alarma reconocible. La corriente de activación del buzzer es suficientemente baja como para ser manejada directamente por el GPIO sin necesidad de etapa de amplificación intermedia.

El componente fue conectado entre el GPIO y GND sin polarización inversa, asegurando que el pin digital entregue la señal activa sobre el terminal positivo. Su respuesta acústica es inmediata y fue utilizada en el sistema para representar una alerta sonora. El evento se desencadena desde la interfaz mediante el botón “Llamar Cuidador”. Cada activación genera una secuencia sonora de ocho tonos alternados que emulan un sistema de alarma, útil para complementar señales visuales o para captar la atención de un asistente cercano.

3.6.2.3 Sensor de temperatura LM35DZ

El sensor analógico **LM35DZ** fue integrado al prototipo como módulo de medición ambiental para obtener información de temperatura en Celsius. Este dispositivo entrega una señal de salida lineal con una pendiente de 10 mV/C, lo cual lo hace compatible con entradas analógicas de microcontroladores. El pin de salida fue conectado al canal analógico A0 del ESP8266, cuya entrada admite un rango de 0 V a 1.0 V como máximo.

Dado que el LM35DZ puede entregar hasta 1.5 V (a 150 C), se implementó una red divisora de voltaje resistiva para adaptar su rango de salida al del conversor analógico-digital (ADC) interno del ESP8266. Esta red está formada por dos resistencias de 1 kΩ conectadas en serie entre la salida del sensor y GND, con el nodo intermedio conectado al pin A0. El escalamiento de voltaje se describe por la siguiente ecuación:

$$V_{ADC} = V_{OUT} \frac{R_2}{R_1 + R_2} = V_{ADC} = V_{OUT} \frac{1}{2} \quad (7)$$

Esta configuración permite extender el rango de lectura segura hasta 2.0 V en la salida del sensor, equivalente a 200 C, aunque el sistema está configurado para operar típicamente entre 0 y 100 C. El convertidor tiene una resolución de 10 bits, es decir, convierte el voltaje analógico de entrada en un valor numérico entre 0 y 1023.

Para convertir esa lectura en temperatura, mediante el código primero se obtiene el voltaje original dividiendo el valor leído entre 1023 y multiplicando por 1.0 V (que es el rango máximo de entrada permitido por el ESP8266). Luego, como se utilizó un divisor resistivo que reduce a la mitad el voltaje del sensor, se multiplica por 2 para recuperar el valor real de salida del LM35DZ. Finalmente, ese voltaje se transforma en temperatura multiplicando por 100, ya que el LM35DZ entrega 10 mV por cada Celsius. El circuito se ve reflejado en la **Fig 3.16**.

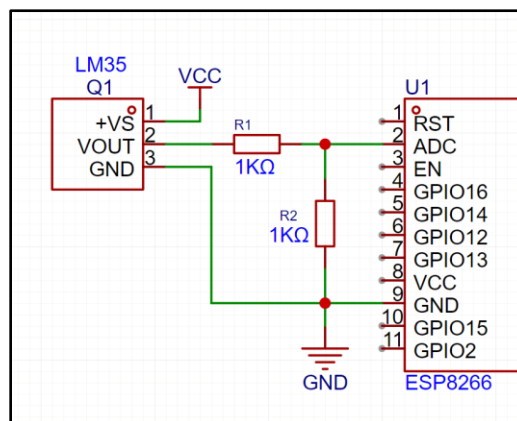


Fig 3.16 Divisor de voltaje para señal de temperatura

3.6.2.4 Motor paso a paso 28BYJ-48 con ULN2003

El sistema incluye un motor paso a paso **28BYJ**, de cinco hilos, alimentado a 5 V y controlado mediante un controlador **ULN2003A**. Este dispositivo fue seleccionado debido a la limitada capacidad de corriente de los pines del ESP8266 (aproximadamente 12 mA), insuficiente para excitar directamente las bobinas del motor. El ULN2003 actúa como una interfaz de potencia, entregando corriente suficiente a través de transistores internos y protegiendo al microcontrolador mediante diodos incorporados.

Cuatro pines digitales del ESP8266 fueron utilizados como salidas para controlar la secuencia de activación del motor, conectados a las entradas del ULN2003 según la correspondencia indicada en la Tabla 3.9.

Tabla 3.9 Entradas para controlador ULN2003

Entrada ULN2003	GPIO	Descripción
IN1	GPIO5 (D1)	Bobina A
IN2	GPIO4 (D2)	Bobina B
IN3	GPIO14 (D5)	Bobina C
IN4	GPIO12 (D6)	Bobina D

Las salidas OUT1–OUT4 del ULN2003 están conectadas directamente a las bobinas del motor. El pin COM del controlador se conectó a una línea de 5 V externa para habilitar la acción de los diodos internos de protección. La conexión de tierra se mantiene común entre el motor, el ULN2003 y el ESP8266, garantizando una referencia eléctrica estable en todo el sistema.

Dado que el ESP8266 no es capaz de suministrar la corriente necesaria para excitar las bobinas del motor directamente (recordemos que entrega un máximo de 12 mA), se optó por alimentar el ULN2003 mediante una fuente externa dedicada de 5 V y 1 A, conectada a la protoboard. Esta fuente es esencial para el correcto funcionamiento del motor paso a paso, ya que permite suministrar la corriente requerida sin provocar caídas de tensión ni sobrecargar el regulador interno del ESP. La conexión se realizó utilizando un cargador USB con salida de 5 V, se conectó a través de un adaptador a terminales, permitiendo llevar los 5 V y GND directamente a las líneas de alimentación de la protoboard. El voltaje medido fue de aproximadamente 5.55 V estables bajo carga, valor adecuado para alimentar tanto al ULN2003 como al motor 28BYJ, cuyo voltaje nominal también es de 5 V.

El control del motor se realiza mediante una secuencia de pasos controlada por software, que activa las bobinas en un orden predeterminado y permite alternar entre giro continuo o detención del eje, simulando una persiana motorizada. Esta lógica fue configurada desde el servidor web del ESP8266 mediante la URL `/motor/toggle`, y vinculada en la interfaz gráfica al botón “Persiana”. Al recibir la señal, el sistema alterna el estado del motor entre encendido y apagado, manteniéndose en funcionamiento hasta recibir una nueva instrucción.

Capítulo 4

Evaluación experimental del desempeño del sistema Look2Control

Una vez completado el ensamblaje del prototipo y desarrollados todos los módulos del sistema, se procedió a la validación funcional de Look2Control bajo condiciones reales de uso. Esta etapa permitió verificar el comportamiento del sistema tanto a nivel lógico como físico, evaluando si las funcionalidades implementadas respondían conforme a lo esperado y si existían inconsistencias, retardos o fallos en su ejecución.

Las pruebas incluyeron tanto componentes electrónicos conectados al ESP8266 como elementos lógicos dentro de la interfaz visual. Se documentaron observaciones cuantitativas (como tiempos de respuesta o lecturas analógicas) y cualitativas (como la fluidez visual, la estabilidad del rastreo y la percepción de retroalimentación). Cada funcionalidad fue activada mediante el sistema de control visual con parpadeo, replicando el uso real del sistema y permitiendo una validación completa de extremo a extremo.

4.1 Metodología de validación

La validación del sistema Look2Control se realizó mediante pruebas en condiciones reales, con el objetivo de confirmar que cada módulo del sistema respondiera correctamente a las instrucciones emitidas desde la interfaz, y que la interacción basada en seguimiento ocular se mantuviera operativa y estable durante el uso continuo.

Las primeras pruebas del sistema se enfocaron en evaluar la posibilidad de ejecutar la aplicación completa en una tablet con cámara integrada, sin requerir periféricos externos como cámaras USB ni sistemas embebidos adicionales. Por disponibilidad institucional, se utilizó una Lenovo Yoga Book YB1-X91X [19], un modelo con Windows 10, cámara integrada y 4 GB de RAM.

Se logró ejecutar exitosamente la interfaz gráfica, la calibración ocular y la activación de funciones, confirmando así la compatibilidad funcional del sistema con entornos portátiles. Sin embargo, se observó una tasa de actualización reducida, entre 7 y 9 FPS, atribuible a las limitaciones de procesamiento del dispositivo.

Posteriormente, todas las pruebas restantes se realizaron sobre una computadora con recursos más amplios, utilizada como entorno base para validar el comportamiento de cada módulo: detección ocular, interfaz gráfica, lógica de activación por parpadeo y respuesta de los dispositivos físicos conectados al ESP8266.

Las pruebas se organizaron en dos grandes grupos:

- **Funciones físicas:** aquellas que generan una acción tangible sobre un dispositivo conectado al microcontrolador (por ejemplo, encender un LED o activar un motor).
- **Funciones lógicas o informativas:** aquellas que operan únicamente dentro del entorno visual o de software (como mostrar una ventana, ajustar el volumen o abrir un navegador).

El criterio para considerar una prueba como satisfactoria fue el cumplimiento total de tres condiciones:

1. Que la acción se activara correctamente mediante parpadeo doble sobre la interfaz.
2. Que se ejecutara la función esperada sin errores ni retardos significativos.
3. Que la interfaz entregara retroalimentación visual coherente con el resultado.

4.2 Pruebas por módulo funcional

4.2.1 Validación del seguimiento ocular y la interfaz visual

El proceso de validación del sistema de control visual se realizó utilizando un monitor Gigabyte G32QC de 32 pulgadas con resolución 2560×1440 píxeles, bajo condiciones de iluminación controlada. Se utilizó una webcam genérica de 30 FPS y una distancia estimada de 50–60 cm entre el rostro y la cámara. El sistema desplegó un total de 25 puntos distribuidos uniformemente sobre la pantalla. Cada punto fue aceptado mediante fijación visual sostenida, utilizando un radio de aceptación de 40 píxeles definido en el código y registró correctamente los datos de mirada necesarios para el ajuste del modelo de regresión.

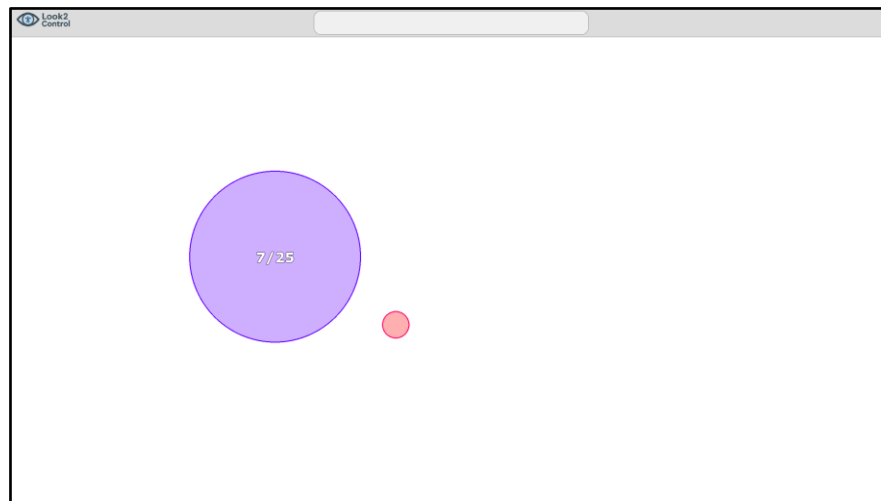


Fig 4.1 Puntos de calibración aceptado por el sistema

Una vez finalizado este proceso se verificó la estabilidad del modelo utilizando la posición estimada de la mirada, la cual se representa mediante un círculo de color rojo sobre la interfaz. Esta coordenada fue capturada y mostrada en consola como posición absoluta en pantalla (función `event.point()` del código).

```
Mirada estimada en pantalla: X=2362.70, Y=1145.78
Mirada estimada en pantalla: X=2369.86, Y=1150.77
Mirada estimada en pantalla: X=2375.49, Y=1156.11
Mirada estimada en pantalla: X=2395.70, Y=1143.97
Mirada estimada en pantalla: X=2417.37, Y=1122.83
```

Fig 4.2 Posición estimada de mirada durante la prueba correspondiente al punto de enfoque

Para validar la lógica de interacción, se ejecutaron activaciones por doble parpadeo sobre diferentes botones. La secuencia constó de:

- A. **Primer parpadeo** → borde rojo durante 3 segundos.
- B. **Segundo parpadeo** → activación de la función y mensaje correspondiente.

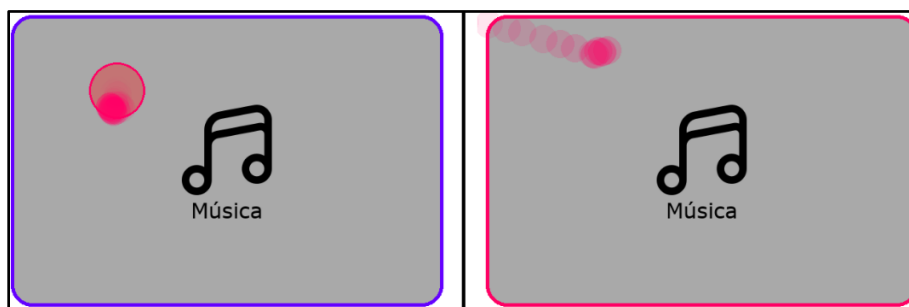


Fig 4.3 Proceso de lógica de interacción para botón



Fig 4.4 Retroalimentación visual de confirmación de activación

Durante las pruebas, la tasa de actualización del sistema se mantuvo entre 18 y 25 FPS, garantizando un funcionamiento fluido de la interfaz. Además, se implementó una métrica para evaluar la dispersión de la mirada durante fijaciones sostenidas. Esta medida fue calculada con base en los últimos 20 puntos de mirada estimada, registrando el error radial

de cada muestra respecto al centro medio de fijación. La fórmula utilizada corresponde con la distancia euclidiana estándar:

$$e_i = \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} \quad (8)$$

Con un promedio de:

$$E_{prom} = \frac{1}{n} \sum_{i=1}^n e_i \quad (9)$$

donde e_i representa el error radial del punto i , y (\bar{x}, \bar{y}) es el centro promedio de la fijación.

Esta metodología está fundamentada en los criterios de precisión definidos por Holmqvist et al. (2011), y por Duchowski (2017) [9]. Las mediciones se realizaron sobre tres botones representativos: “**Recalibrar**”, “**Encender/Apagar Luz**” y “**Música**”, obteniéndose los siguientes resultados:

Tabla 4.1 Dispersión promedio y máxima por botón

Botón	Dispersión promedio (px)	Dispersión máxima (px)
Recalibrar	7.28 px	26.98
Encender/Apagar Luz	9.47 px	27.32
Música	11.95 px	29.55 px

4.2.2 Validación de botones lógicos

Estas funciones operan exclusivamente dentro del entorno de software del sistema y no requieren una conexión directa con módulos electrónicos. Se validó que cada botón respondiera correctamente a la activación por doble parpadeo, ejecutando su acción prevista y proporcionando retroalimentación visual coherente. La **Tabla 4.2** resume los resultados obtenidos para cada función.

Tabla 4.2 Pruebas de funcionalidad de botones lógicos del sistema

Botón	Acción esperada	Resultado observado
Recalibrar	Reiniciar proceso de calibración visual.	Se mostraron nuevamente los puntos de guía visual.
Guía	Mostrar ventana con instrucciones.	Ventana flotante se desplegó y cerró bien.

Música	Reproducir melodía de fondo.	Sonido reproducido o detenido correctamente.
Volumen	Aumentar volumen progresivamente.	Intensidad aumentó en tres niveles cíclicos.
Llamar Cuidador	Activar sonido de alerta sin botón físico.	Se emitió un patrón sonoro intenso.
Emergencia	Enviar correo con ubicación al contacto registrado.	Mensaje enviado mediante API de “Mailjet”.
Video/Películas	Mostrar mensaje de función deshabilitada.	Sistema mostró notificación y no ejecutó acción.
Salir	Cerrar aplicación Look2Control.	Aplicación se cerró ordenadamente.

4.2.3 Validación de módulos físicos

Además de las pruebas realizadas sobre la interfaz visual y lógica, se validaron todas las funcionalidades físicas del sistema Look2Control. Estas pruebas se enfocaron en verificar que los botones vinculados a salidas electrónicas respondieran correctamente a los comandos generados mediante doble parpadeo sobre los botones, activando cada módulo de forma coherente.

Durante la fase de validación física del prototipo, se realizaron mediciones con un multímetro para verificar el comportamiento eléctrico de los circuitos conectados al ESP8266. Estas mediciones fueron chequeos de voltaje entre pines GPIO y tierra, así como pruebas de corriente en serie utilizando aperturas momentáneas en los cables de conexión. El objetivo fue asegurar que los consumos se mantuvieran dentro del rango de operación seguro de los pines digitales del microcontrolador (corriente máxima recomendada de 12 mA por GPIO), y que los valores de tensión no excedieran el umbral de 3.3 V. Las pruebas confirmaron que los elementos conectados, presentan comportamientos acordes a lo esperado, sin superar las capacidades del sistema.

4.2.3.1 LED verde (encendido/apagado)

El botón “**Encender/Apagar luz**” activó correctamente un LED verde conectado al pin GPIO16 (D0), funcionando en modo binario (ON/OFF). La activación fue inmediata y visible. El cambio de estado se reflejó también en el mensaje central de la interfaz.

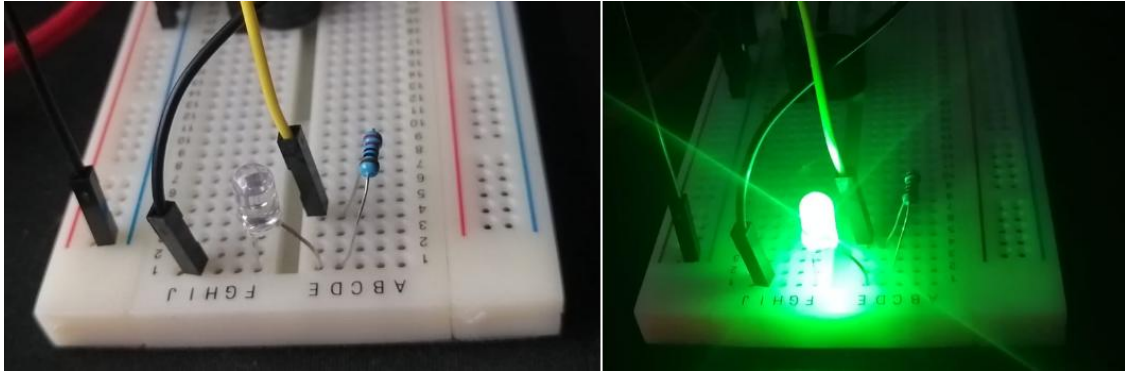


Fig 4.5 Encendido del led por medio de la interfaz

4.2.3.2 LED azul (control de brillo)

El botón “**Intensidad de luz**” permitió ajustar progresivamente el brillo de un LED azul conectado al pin GPIO15 (D8) mediante señal PWM. Con cada doble parpadeo, el ciclo de trabajo aumentó en pasos predefinidos hasta alcanzar el máximo, reiniciando luego al valor mínimo. El cambio de intensidad fue perceptible y cada uno se reflejó también en el mensaje central de la interfaz.

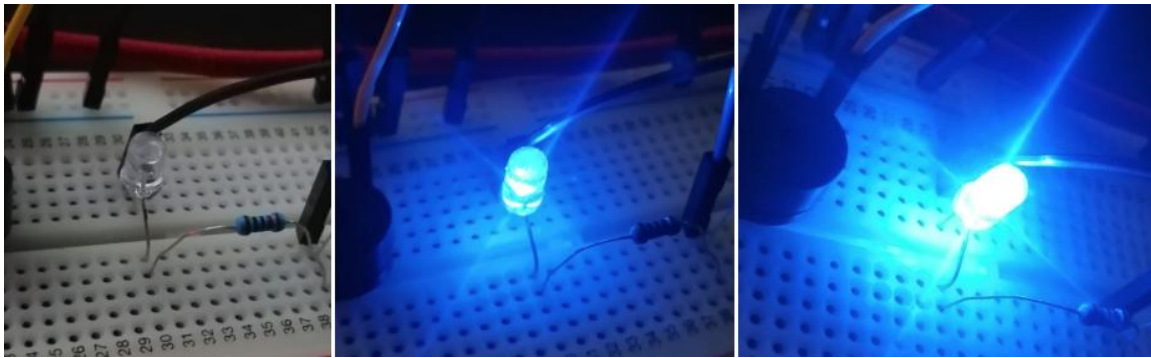


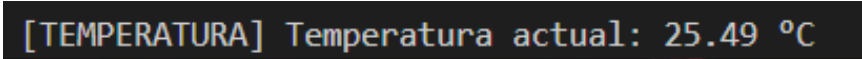
Fig 4.6 Proceso de intensidad de luz activada por la interfaz

4.2.3.3 Buzzer (alarma sonora)

El botón “**Llamar cuidador**” activó una secuencia alternante de tonos agudos mediante un buzzer piezoeléctrico conectado al pin GPIO13 (D7). El sonido generado fue claro y suficientemente audible como para simular una llamada de alerta. El buzzer respondió de forma inmediata y cesó correctamente tras completar la secuencia programada. Al activar el buzzer, esto se reflejó también en el mensaje central de la interfaz.

4.2.3.4 Sensor de temperatura (LM35DZ)

Durante las primeras pruebas, el sensor entregó valores en el rango de 23 °C a 27 °C al activar el botón de **“Temperatura”** en la interfaz. Estos fueron visualizados correctamente en el mensaje central de la pantalla y en la respuesta del servidor. Se verificó que la lógica de lectura y visualización permaneciera operativa ya que se consideró un posible mal funcionamiento del sensor en pruebas finales por sobreuso.



```
[TEMPERATURA] Temperatura actual: 25.49 °C
```

Fig 4.7 Respuesta del servidor al activar botón en interfaz

4.2.3.5 Motor paso a paso (persiana)

El botón **“Persiana”** activó la rotación continua de un motor paso a paso 28BYJ, controlado por el controlador ULN2003. La secuencia de pasos se inició al recibir la primera activación y se detuvo al recibir una segunda activación, simulando el comportamiento de una persiana controlada mediante interruptor. El movimiento fue estable, sin pérdida de pasos ni vibraciones anómalas. El motor fue alimentado por una fuente externa de 5 V, evitando caídas de tensión y garantizando su funcionamiento independiente del ESP8266.



Fig 4.8 Controlador encendido evidenciando el movimiento del motor

Capítulo 5

Análisis de resultados y oportunidades de mejora del sistema

5.1 Evaluación de respuesta del sistema de entrada visual

Los resultados obtenidos en la validación del sistema de seguimiento visual permiten afirmar que el seguimiento ocular implementado en Look2Control ofrece un nivel de precisión y estabilidad funcional adecuado para su propósito. Las medidas de dispersión registradas **Tabla 4.1** evidencian que la estimación de la mirada logra mantenerse dentro de márgenes razonables de error para interacción directa con elementos de gran tamaño, como los botones de la interfaz.

La menor dispersión promedio se obtuvo sobre el botón “Recalibrar” (7.28 px), lo cual sugiere que la zona central de la pantalla, donde habitualmente se ubican los puntos iniciales de calibración, conserva mayor estabilidad de fijación visual. Por otro lado, la mayor dispersión se presentó en el botón “Música” (11.95 px), lo cual podría atribuirse tanto a su ubicación más periférica como a una ligera desviación en el modelo de regresión para regiones laterales. Este comportamiento también puede estar asociado a una ligera degradación del modelo de calibración a lo largo del tiempo. Dado que el botón “Música” fue evaluado al final de la secuencia de pruebas, es posible que pequeñas variaciones en la postura, la iluminación o el cansancio ocular hayan introducido desviaciones en la estimación visual. En este contexto, la presencia del botón “Recalibrar” adquiere un rol clave para restaurar la precisión del sistema durante sesiones prolongadas de uso.

A pesar de las diferencias, el error radial máximo registrado (29.55 px) se mantiene significativamente por debajo del tamaño de los botones (200 px), asegurando que la detección de enfoque no se superponga entre zonas activables ni produzca falsos positivos.

La tasa de actualización registrada (entre 18 y 25 FPS) permitió mantener una experiencia fluida en la mayoría de las sesiones, sin saltos perceptibles en la trayectoria de la mirada ni retardos que comprometieran la interacción. Esta frecuencia resultó suficiente para detectar fijaciones estables y cambios de dirección oculares sin pérdida de eventos, gracias a la combinación de los filtros temporales y al tiempo de fijación ajustado mediante parámetros de la librería EyeGestures.

Si bien el sistema demostró un rendimiento óptimo en equipos de escritorio, se identificaron limitaciones de fluidez cuando se ejecutó en la tablet Lenovo Yoga Book, donde la tasa de cuadros cayó por debajo de los 10 FPS. Esta diferencia no compromete la validez

técnica del modelo de entrada visual, pero sí establece un límite práctico para su uso en dispositivos con capacidades gráficas reducidas.

En conjunto, el sistema de seguimiento ocular de Look2Control ha demostrado ser funcional, preciso y estable para entornos de interacción accesible, cumpliendo con los criterios técnicos definidos para estimación visual efectiva en interfaces de control de gran escala.

5.2 Análisis de la lógica interna y su interacción en la interfaz

La evaluación de los botones lógicos del sistema permite comprobar la solidez de la lógica interna de control basada en parpadeos y su adecuada integración con los elementos visuales de la interfaz. A través de las pruebas reportadas en la **Tabla 4.2**, se verificó que cada acción lógica fue correctamente activada mediante la secuencia de doble parpadeo, sin errores de interpretación ni eventos falsos.

Una de las características destacables del sistema es la implementación de una lógica de activación temporal segura, en la que el primer parpadeo cambia el borde del botón a color rojo durante tres segundos, habilitando un segundo parpadeo como confirmación. Esta mecánica reduce significativamente el riesgo de activaciones accidentales, especialmente en interfaces basadas en mirada continua.

El comportamiento de los botones “Recalibrar”, “Guía” y “Salir” evidenció la correcta ejecución de eventos que modifican la estructura visual del sistema, como el despliegue de ventanas o la finalización del programa. El botón “Guía” mostró ser particularmente estable, aislando correctamente la lógica base mientras la ventana flotante se encontraba activa.

Los botones “Música” y “Volumen” validaron la capacidad del sistema para gestionar estados internos persistentes. En el caso del volumen, se comprobó que el valor se incrementa de forma cíclica entre tres niveles predeterminados, mientras que el botón de música alterna entre reproducción y silencio, manteniendo la consistencia entre el estado sonoro y la retroalimentación visual.

El botón “Emergencia” cumplió con la acción crítica de enviar un mensaje con ubicación a través de la API de Mailjet, reflejando que el sistema puede interactuar con servicios externos en tiempo real sin afectar la estabilidad de la aplicación.

Finalmente, el botón “Video/Películas”, cuya funcionalidad fue deshabilitada por motivos de seguridad, mostró adecuadamente un mensaje informativo al usuario. En general, las pruebas evidencian que la lógica interna de Look2Control cumple con los principios de accesibilidad y coherencia. El sistema logró interpretar correctamente todas las intenciones de activación, presentar una respuesta visual clara y mantener la estabilidad operativa en todas las funciones lógicas evaluadas.

5.3 Comportamiento funcional y eléctrico de los dispositivos físicos controlados

Los módulos físicos integrados al sistema Look2Control fueron evaluados con el objetivo de comprobar su funcionamiento ante señales generadas por la interfaz gráfica y transmitidas al ESP8266 mediante conexión Wi-Fi. Los resultados obtenidos permiten validar que los dispositivos responden adecuadamente tanto desde el punto de vista lógico como eléctrico, manteniendo estabilidad y seguridad durante su activación.

El módulo de LED digital respondió correctamente a cada comando de activación. Se verificó que el encendido y apagado se efectuaban tras la detección de un doble parpadeo sobre el botón “Encender/Apagar Luz”. Desde el punto de vista eléctrico, el pin GPIO16 proporcionó un voltaje de salida cercano a 3.2 V, y una corriente medida experimentalmente de aproximadamente 3 mA al atravesar una resistencia de 220 Ω , valores que se mantienen dentro de los márgenes seguros de operación del microcontrolador.

El LED azul controlado por modulación por ancho de pulso (PWM) permitió simular con éxito un sistema de iluminación de intensidad variable. Las pruebas confirmaron que, tras cada doble parpadeo sobre el botón “Intensidad de Luz”, el ciclo de trabajo aumentaba en un patrón cíclico, generando un cambio visible en el brillo percibido. La corriente suministrada en este caso fue similar a la del LED digital, ya que se utilizó también una resistencia limitadora de 220 Ω . El control PWM no presentó parpadeos ni inestabilidad, lo cual indica una generación continua y precisa de la señal por parte del ESP8266.

El buzzer, activado desde la interfaz mediante el botón “Llamar Cuidador”, emitió correctamente la secuencia sonora programada. El sonido fue claramente perceptible, y la duración de la señal concordó con el patrón definido en el código del ESP8266. El consumo de corriente no es mayor a 12mA, permitiendo su conexión directa al GPIO sin necesidad de un transistor intermedio. Este resultado respalda la viabilidad del módulo para alertas o avisos auditivos inmediatos en entornos de asistencia.

El motor fue activado desde el botón “Persiana” mediante una solicitud HTTP implementada en el servidor web del ESP8266. La lógica de activación alterna entre detener o iniciar el movimiento rotacional, simulando la operación de una persiana. El controlador ULN2003 permitió manejar la corriente requerida por el motor (220 mA por fase) sin que los pines del ESP8266 quedaran comprometidos. Se validó además que la fuente de alimentación externa conectada suministraba un voltaje estable de 5 V con suficiente corriente, sin presentar caídas de tensión durante la operación del motor. Esta decisión fue crítica, ya que alimentar el motor directamente desde el puerto USB del ESP8266 generaba inestabilidad, debido a que el consumo total superaba la capacidad del regulador interno.

El sensor LM35DZ entregó valores proporcionales a la temperatura ambiente, que fueron correctamente leídos a través del canal analógico A0 del ESP8266, mediante un divisor de voltaje. Las mediciones obtenidas durante las pruebas realizadas en el cantón de

Santa Ana se encontraron entre 24 °C y 27 °C, lo cual coincidió con la temperatura ambiente estimada obtenida por fuentes de internet para dicho cantón de San José, Costa Rica.

5.4 Limitaciones detectadas y oportunidades de mejora técnica

Se observó una leve pérdida de precisión en el seguimiento ocular durante sesiones prolongadas, especialmente en las regiones periféricas de la pantalla. Esta disminución en la exactitud del modelo de calibración podría estar asociada a ligeros cambios en la postura del usuario o a variaciones en la iluminación del entorno. Por este motivo, se recurrió al botón de recalibración como solución práctica durante la operación.

Adicionalmente, se identificaron limitaciones importantes en el rendimiento gráfico de la tablet Lenovo Yoga Book. A pesar de cumplir con los requisitos mínimos, su capacidad de procesamiento no fue suficiente para ejecutar la aplicación de forma fluida, limitando la experiencia de usuario a tasas de actualización bajas.

Por último, se documentó que, tras largos periodos de inactividad, el ESP8266 puede requerir múltiples intentos para restablecer la comunicación Wi-Fi. Si bien no afecta la integridad del sistema, sí introduce una pequeña latencia en la respuesta inicial luego de un tiempo sin interacción.

Estas observaciones no comprometen la funcionalidad general de Look2Control, pero ofrecen un panorama claro de los retos técnicos enfrentados y superados durante el desarrollo.

Capítulo 6

Conclusiones generales y proyección del sistema Look2Control

6.1 Conclusiones

El desarrollo del sistema Look2Control ha permitido demostrar la viabilidad de una solución tecnológica accesible, funcional y de bajo costo para asistir a personas con movilidad severamente limitada en la gestión de su propio entorno. Mediante la implementación de técnicas de seguimiento ocular utilizando hardware disponible en el mercado y una interfaz diseñada con criterios de accesibilidad universal, se logró construir un sistema que facilita el control de dispositivos del hogar y la comunicación básica de emergencia.

Se consolidó un modelo de interacción visual preciso y estable, capaz de interpretar con fiabilidad la dirección de la mirada y los eventos de parpadeo como comandos. Esta capacidad se integró exitosamente con una interfaz gráfica clara y reactiva, que respondió a cada activación esperada, tanto en acciones lógicas internas como en el control físico de módulos externos como luces, alarmas o motores.

Los resultados obtenidos evidencian que el sistema cumple con las funcionalidades esenciales para su propósito: detección confiable de la mirada, control por parpadeo, activación de alertas y manejo de dispositivos habituales en un entorno doméstico. El sistema opera con un consumo moderado de recursos y sin requerir hardware especializado, lo que permite su implementación en contextos con restricciones económicas.

Si bien se identificaron ciertas limitaciones en el rendimiento de equipos con bajo poder de procesamiento y se reconocieron áreas técnicas susceptibles de mejora, estas no impiden el funcionamiento general del sistema ni comprometen su utilidad. El sistema se comportó de manera robusta bajo las condiciones previstas, entregando resultados acordes con los lineamientos funcionales establecidos.

Look2Control representa un ejemplo tangible de la aplicación de tecnologías modernas como la visión por computadora, la electrónica digital y los principios de diseño inclusivo para mejorar la calidad de vida de personas con movilidad reducida. Su estructura modular, su bajo costo y su potencial de escalamiento permiten proyectarlo como una base viable para futuras adaptaciones o ampliaciones en el campo de la asistencia tecnológica.

6.2 Recomendaciones

A partir de los resultados obtenidos durante el desarrollo e implementación del sistema, se identifican las siguientes recomendaciones para futuras versiones del proyecto:

- **Mejoras en la calibración dinámica del seguimiento ocular:** Sería conveniente implementar un sistema de recalibración continua o adaptativa que mantenga la precisión del modelo ante cambios de postura, iluminación o fatiga visual durante sesiones prolongadas.
- **Rediseño de hardware con circuitería integrada:** Se sugiere desarrollar una placa PCB personalizada que integre todos los componentes y conexiones necesarias, eliminando el uso de breadboards y cables. Esto permitiría mayor estabilidad, portabilidad y durabilidad del sistema.
- **Ampliación de funcionalidades con nuevos dispositivos controlados:** El sistema puede extenderse para incluir dispositivos como televisores, cortinas automatizadas o asistentes por voz. Para ello, se recomienda incorporar acoplamiento galvánico mediante optoacopladores o relés con aislamiento, especialmente si se conectan circuitos de mayor potencia. Esto protege el microcontrolador y mejora la estabilidad del sistema ante futuras expansiones.
- **Evaluación de requerimientos mínimos del dispositivo:** Dado que tablets con procesadores limitados como la Lenovo Yoga Book presentaron una tasa de cuadros por segundo insuficiente para una interacción fluida, se recomienda definir requerimientos mínimos de hardware para ejecutar el sistema con rendimiento aceptable (por ejemplo, CPU de al menos 2 núcleos a 2.0 GHz, 8 GB de RAM y cámara de al menos 30 FPS), considerando siempre mantener la accesibilidad económica.
- **Evaluación con usuarios reales:** Dado que el sistema está destinado a una población vulnerable como personas con movilidad severamente limitada, es importante aclarar que cualquier estudio futuro que involucre usuarios reales debe ser sometido a revisión y aprobación por un Comité Ético-Científico (CEC) [20] y por el Consejo Nacional de Investigación en Salud (CONIS) de Costa Rica [21]. La validación realizada en este proyecto se limitó a pruebas técnicas y funcionales, debido a que los tiempos establecidos en el marco del proyecto de graduación no permiten tramitar las aprobaciones requeridas por estas entidades. Se recomienda, en futuras investigaciones, considerar este proceso formal si se busca evaluar directamente el impacto en personas usuarias finales.

Estas recomendaciones permiten visualizar una ruta clara de evolución técnica para Look2Control, manteniendo su enfoque en la accesibilidad, el bajo costo y la adaptabilidad a las necesidades de personas con movilidad severamente limitada.

Bibliografía

- [1] J. M Das, K. Anosike y R. M. Asuncion, Locked-in Syndrome, StatPearls Publishing, 2023.
- [2] Tobii Dynavox US, «Tobii Dynavox US,» 2025. [En línea]. Available: <https://us.tobiidynavox.com/products/td-i-series>. [Último acceso: 2025].
- [3] Tobii, «Tobii Gaming,» 2025. [En línea]. Available: <https://gaming.tobii.com/product/eye-tracker-5/>. [Último acceso: 2025].
- [4] INEC, «Encuesta Nacional sobre Discapacidad 2023,» 2024. [En línea]. Available: https://admin.inec.cr/sites/default/files/2024-12/reSocialPobDis_Enadis2023.pdf. [Último acceso: 2025].
- [5] OMS, «Organización Mundial de la Salud,» 2023. [En línea]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/disability-and-health>. [Último acceso: 2025].
- [6] CORDIS, «The first mental joystick based on Artificial Intelligence for Multi-Modal Human Machine Interaction,» Italia, 2023.
- [7] W3C, «Web Content Accessibility Guidelines (WCAG) 2.1,» 2024. [En línea]. Available: <https://www.w3.org/TR/WCAG21/>. [Último acceso: 2025].
- [8] Pygame, «About - Wiki,» 2025. [En línea]. Available: <https://www.pygame.org/wiki/about>. [Último acceso: 2025].
- [9] A. Duchowski, Eye Tracking Methodology: Theory and Practice, 2017.
- [10] K. Holmqvist, M. Nyström, R. Andersson y R. Dewhurst, Eye Tracking: A Comprehensive Guide to Methods and Measures, OUP Oxford, 2011.
- [11] A. Hameed, T. Abbas y A.-W. Sami, «Best low-cost methods for real-time detection of the eye and gaze tracking,» *De Gruyter Brill*, 2024.
- [12] Soukupová y Čech, «Real-Time Eye Blink Detection using Facial Landmarks,» 2016.
- [13] EyeGestures, «Documentation - EyeGestures Library,» 2025. [En línea]. Available: <https://github.com/NativeSensors/EyeGestures/tree/main>.
- [14] Google, «MediaPipe Face Mesh,» 2025. [En línea]. Available: https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/face_mesh.md.

- [15] I. Grishchenko, A. Ablavatski y K. Yury, «Attention Mesh: High-fidelity Face Mesh Prediction in Real-time,» Seattle, 2020.
- [16] Espressif Systems, «ESP8266EX Datasheet,» 2024.
- [17] R. Fielding, J. Gettys, J. Mogul y H. Masinter, «Hypertext Transfer Protocol -- HTTP/1.1,» 1999. [En línea]. Available: <https://www.rfc-editor.org/rfc/rfc2616.html>. [Último acceso: 2025].
- [18] Requests, «Requests: HTTP for Humans™,» [En línea]. Available: <https://requests.readthedocs.io/en/latest/>. [Último acceso: 2025].
- [19] Lenovo, «YOGA Book (YB1-X91F, YB1-X91L, YB1-X90F, YB1-X90L),» [En línea]. Available: <https://pcsupport.lenovo.com/cr/es/products/tablets/yoga-series/yoga-book/downloads/ds119816>. [Último acceso: 2025].
- [20] UCR, «CEC Comité Ético-Científico,» [En línea]. Available: <https://cec.ucr.ac.cr/>. [Último acceso: 2025].
- [21] CONIS, «Consejo Nacional de Investigación de la Salud,» [En línea]. Available: <https://conis.ministeriodesalud.go.cr/>. [Último acceso: 2025].