

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**CREACIÓN DE UN PROTOTIPO DE UN SISTEMA DE
SEGUIMIENTO DE MIRADA PARA EL CONTROL DE
DISPOSITIVOS IOT**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO/A EN
INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN**

EDWIN FABIÁN COCHA TOBANDA

edwin.cocha@epn.edu.ec

MARÍA CRISTINA ERAZO CUCÁS

maria.erazo@epn.edu.ec

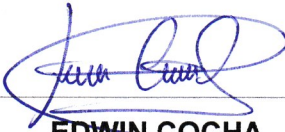
DIRECTOR: PhD. San Guun Yoo

san.yoo@epn.edu.ec

Quito, agosto 2022

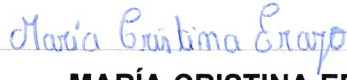
CERTIFICACIONES

Yo, EDWIN FABIÁN COCHA TOBANDA declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



EDWIN COCHA

Yo, MARÍA CRISTINA ERAZO CUCÁS declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



MARÍA CRISTINA ERAZO

Certifico que el presente trabajo de integración curricular fue desarrollado por EDWIN COCHA Y MARÍA CRISTINA ERAZO, bajo mi supervisión.



SANG GUUN YOO, Ph.D.
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

EDWIN COCHA

MARÍA CRISTINA ERAZO

SANG GUUN YOO, Ph.D.

DEDICATORIA

Dedico este trabajo a mis padres por su cariño y apoyo incondicional a lo largo de la vida. A ustedes, por cultivar en mí valores de respeto, humildad, esfuerzo y perseverancia; valores que han sido fundamentales para realizar esta tesis. También quiero dedicar este trabajo a mis hermanos por brindarme su apoyo constante a lo largo de este camino de estudios.

Edwin

DEDICATORIA

Dedico esta tesis a mi madre, por su amor infinito y apoyo incondicional. Gracias a usted mamá he podido conseguir uno de los anhelos más grandes de mi corazón.

María Cristina

AGRADECIMIENTO

Agradezco a Dios por cuidar de mí y de mi familia. A mis padres Raúl Cocha y Fabiola Tobanda, por ser un ejemplo de vida, por su amor, apoyo, consejos y bendiciones. A mis hermanos, Joffre, pilar económico fundamental; Doris, que me acogido en su morada y a Kelly por su apoyo.

Además, agradezco a mi amiga y gran compañera de tesis, María Cristina. A mi tutor Ph.D. Sang Guun Yoo por guiarnos con sus conocimientos a lo largo de este proyecto.

Edwin

AGRADECIMIENTO

Le agradezco a Dios por ser la luz y guía principal de mi vida.

A mi madre Pilar por su amor, por sus consejos y sus palabras de aliento. A mis hermanos Karen y Joseph por el cariño que nos tenemos.

También quiero agradecerle a Edwin por su constancia, entrega y trabajo en este proyecto.

Y finalmente, a nuestro tutor Ph.D. Sang Guun Yoo por la enseñanza y guía brindada durante el desarrollo de este proyecto.

María Cristina

ÍNDICE DE CONTENIDO

RESUMEN	IX
ABSTRACT	X
1 INTRODUCCIÓN	1
1.1 Descripción del componente desarrollado	1
1.2 Objetivo general	2
1.3 Objetivos específicos	2
1.4 Alcance	2
1.5 Marco teórico	2
Introducción	2
Soluciones de seguimiento de mirada	5
Tipos de dispositivos	5
Tipos de librerías	7
Tipos de Aplicaciones	8
2 METODOLOGÍA	10
3 DESARROLLO DE LA SOLUCIÓN	12
3.1 Selección de herramientas para el seguimiento de mirada	12
Dispositivos	12
Librerías.....	13
3.2 Incremento 1	19
Análisis de requerimientos	19
Diseño de interfaces de usuario.....	19
Desarrollo de la interfaz de usuario	20
Pruebas	20
3.3 Incremento 2	21
Análisis de requerimientos	21
Diseño de interfaces de usuario.....	21
Desarrollo de la interfaz de usuario	22
Pruebas	22
3.4 Incremento 3	23
Análisis de requerimientos	23
Diseño.....	23

Desarrollo de la interfaz de usuario	25
Pruebas	25
3.5 Incremento 4	27
Análisis de requerimientos	27
Diseño.....	27
Desarrollo	28
Pruebas	28
3.6 Pruebas del Prototipo Integrado	29
Planificación.....	29
Experimentación	29
Resultados	30
4 ANÁLISIS DE RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	30
4.1 Análisis de Resultados	30
4.2 Conclusiones.....	32
4.3 Recomendaciones	33
5 REFERENCIAS BIBLIOGRÁFICAS	34
6 ANEXOS	I
6.1 Anexo I. Dispositivos de seguimiento de mirada	I
6.2 Anexo II. Librerías de seguimiento de mirada	II
6.3 Anexo III. Requerimientos del prototipo	IV
6.4 Anexo IV. Mockups de las interfaces de usuario	IV

RESUMEN

El seguimiento de mirada es una disciplina tecnológica que ofrece interfaces alternativas dentro del área de interacción humano-computadora. Esta forma de interacción, por su originalidad, ha permitido que esté presente en distintas áreas del Internet de las Cosas. Una solución común basada en seguimiento de mirada hace uso de dos elementos: un dispositivo hardware para obtener datos de la mirada del usuario, y una librería para procesar los datos y obtener la predicción de la mirada. En este proyecto, se hace uso de esta disciplina tecnológica para crear una solución que permite controlar dispositivos de Internet de las Cosas. A lo largo de este documento se detallan los pasos que se realizaron para cumplir con este objetivo. En primer lugar, se presenta el análisis realizado para seleccionar el dispositivo y librería de seguimiento de mirada que fueron utilizados en el prototipo. De manera similar, se describen la metodología de desarrollo utilizada y los incrementos que permitieron obtener el prototipo final. Además, se muestran los resultados de las pruebas del prototipo integrado, para posteriormente realizar el análisis de estos y presentar las conclusiones y recomendaciones de este proyecto.

PALABRAS CLAVE: Internet de las Cosas, IoT, Seguimiento de miradas, MQTT, WebGazer.

ABSTRACT

Gaze tracking is a technological discipline that offers alternative interfaces within the area of human-computer interaction. This form of interaction, due to its originality, has been applied to different areas of the Internet of Things. A common solution based on gaze tracking has two elements: a hardware device to obtain gaze data from the user, and a software library which process the gathered data and obtain gaze prediction. In this project, the technology of gaze tracking is used to create a solution that allows controlling different Internet of Things devices. Throughout this document, the steps that were carried out to reach the proposed objective have been detailed. First, the analysis steps executed to select the gaze tracking device and software library used in the prototype is presented. Similarly, the development methodology used in this project is described, as well as the increments that allowed to achieve the final prototype. In addition, the results of the tests of the integrated prototype are shown. After that, the results are analyzed, to finally, deliver the conclusions and recommendations of this project.

KEYWORDS: Internet of Things, IoT, Gaze Tracking, MQTT, WebGazer

1 INTRODUCCIÓN

1.1 Descripción del componente desarrollado

El desarrollo de este componente consta de dos partes principales: (1) selección del dispositivo y librería para realizar el seguimiento de mirada, (2) creación de un prototipo de un sistema de seguimiento de mirada para el control de dispositivos IoT.

La primera parte del componente inicia con el planteamiento de la pregunta de investigación, la cual fue formulada de la siguiente manera: ¿Qué dispositivos y librerías son utilizados en la implementación de soluciones tecnológicas de IoT basadas en el seguimiento de mirada? En base a esta pregunta de investigación, se realizó un estudio del estado del arte, que consistió en una investigación de proyectos similares, con el fin de conocer el estado y uso actual de esta tecnología. Este estudio permitió tener una lista de dispositivos y librerías que posteriormente fueron estudiadas. Luego, en base a este conocimiento, se seleccionó el mejor dispositivo y las mejores librerías para el seguimiento de mirada. Los criterios de selección estuvieron enfocados en las librerías de código abierto y en la compatibilidad con cámaras web convencionales. Estas librerías pasaron a una fase de instalación y prueba, para finalmente seleccionar la librería que se utilizó en la creación del prototipo. El estudio del estado del arte también permitió tener un primer acercamiento a los requerimientos para la creación del prototipo.

La segunda parte del componente consistió en desarrollar una aplicación web donde el usuario puede interactuar con sus interfaces utilizando la mirada. En estas interfaces el usuario puede seleccionar y enviar varios comandos a un dispositivo IoT, como por ejemplo el comando de encender un foco. Para que las interfaces puedan comunicarse con el dispositivo IoT se utilizó un servidor MQTT (Message Queuing Telemetry Transport) y un Bróker de comunicación. Para que este prototipo pueda ser desarrollado se dividió en cuatro etapas:

1. Análisis de requerimientos: el resultado de este proceso es una lista de los requerimientos funcionales del prototipo.
2. Diseño: en esta etapa se obtuvo los mockups de las interfaces y los diagramas de comunicación entre las interfaces y los dispositivos IoT.
3. Desarrollo: el entregable de esta etapa es el prototipo funcional.
4. Pruebas: de este proceso se obtiene los resultados de las pruebas realizadas al prototipo.

El Prototipo y la Documentación asociada, que son los entregables de este proyecto, se realizó siguiendo una Metodología de desarrollo Incremental y la Metodología Investigación–Acción.

1.2 Objetivo general

Crear un prototipo de un sistema de seguimiento de mirada para el control de dispositivos IoT, orientado a personas con discapacidad motriz severa y del habla.

1.3 Objetivos específicos

1. Entender el estado del arte de dispositivos y librerías para el seguimiento de mirada.
2. Comprender cuáles son las mejores herramientas de seguimiento de mirada.
3. Crear un prototipo de seguimiento de mirada que permita el control de dispositivos de IoT.

1.4 Alcance

El presente proyecto consiste únicamente en la Creación de un prototipo de un sistema de seguimiento de mirada para el control de dispositivos IoT y no en un sistema real. El prototipo permitirá controlar los siguientes dispositivos IoT: foco y televisión. El diseño de las interfaces está enfocado en computadoras de escritorio y laptops con pantallas de 14” o superior.

1.5 Marcoteórico

Introducción

El Internet de las Cosas (IoT) es una tecnología que está presente en varias disciplinas y que puede ser combinada con otras tecnologías [1] . Esto ha permitido que IoT se esparza llegando así a cambiar la forma de realizar las actividades cotidianas de las personas. Lo mencionado anteriormente se afianza con el hecho de que para el 2025, habrá alrededor de 77 millones de dispositivos IoT en el mundo [2]. No hay duda de que el Internet de las cosas está evolucionando rápidamente, no sólo en el campo de la informática y la comunicación, sino también en áreas como la logística, la medicina, etc. [3].

El Internet de las Cosas se refiere a la red interconectada de dispositivos, sensores, actuadores, software, etc. que almacenan e intercambian información [4]. El objetivo principal del IoT es permitir que cosas u objetos se conecten y se comuniquen desde cualquier lugar, en cualquier momento [5].

El Internet de las Cosas ha permitido que las personas puedan interactuar fácilmente con dispositivos inteligentes haciendo uso de la voz o el cuerpo [1] [6]. También se ha visto

como las personas hacen uso de interfaces web o móviles para controlar dispositivos de IoT [7] [8]. Sin embargo, este tipo de soluciones excluye a personas que sufren discapacidad motriz severa y del habla, ya que no pueden hacer uso de esta tecnología debido a las condiciones físicas en las que se encuentran. Dentro de este grupo de personas se encuentran aquellas que sufren de discapacidad motriz causada por Lesión de la Médula Espinal (LME) cuyo número es de aproximadamente 500 mil personas [9]. Se conoce también que la Esclerosis Lateral Amiotrófica (ELA) produce discapacidad motriz y se estima que la tasa de incidencia promedio de ELA en todo el mundo es una de cada 50 mil personas por año [10]. Otras enfermedades que producen este tipo de discapacidad son la Esclerosis Múltiple, Parálisis Cerebral, Ataque cerebro vascular, Distrofia Muscular, etc. [11] [12] [13] [14].

Basado en lo expuesto anteriormente, se puede indicar que la mayoría de las soluciones de IoT no son totalmente inclusivas. Por lo que como una de las tecnologías complementarias aparece el seguimiento de miradas como una interfaz para la interacción con dispositivos IoT. El enfoque de esta tecnología es muy grande y puede ser aplicado en diferentes soluciones, como el de ayudar a personas con discapacidad [15]. El seguimiento de miradas es una disciplina tecnológica en desarrollo que ha ido tomando importancia en varias áreas debido a que puede facilitar la implementación de soluciones tecnológicas que se utilizan en la vida diaria. Un ejemplo de estas soluciones puede ser encender o apagar un foco a través de movimientos ligeros de los ojos [1]. Por este motivo, entender el funcionamiento de esta tecnología se vuelve importante.

El ojo humano está formado por la retina, la pupila, el iris y otros componentes. Estos componentes, en especial la pupila, pueden ser utilizados para realizar el seguimiento de miradas. Un proceso común es capturar imágenes de la pupila, procesarlas mediante algoritmos y predecir la dirección de la mirada. El seguimiento de miradas es el proceso de seguir el movimiento de los ojos para determinar exactamente dónde está mirando una persona y por cuánto tiempo [16]. Existen diferentes dispositivos y algoritmos que permiten predecir la posición y la dirección de la mirada. Tanto hardware como software han sido implementados por diferentes grupos de investigación o empresas afines a esta tecnología, los cuales serán analizados en las siguientes secciones del presente trabajo.

Existen varias soluciones que hacen uso del seguimiento de mirada. A continuación, se da a conocer algunas de estas: En [17], se evaluó la experiencia de usuarios cotidianos al interactuar con una página web, utilizando el seguimiento de mirada. En [18] [19], se controla una silla de ruedas, utilizando la técnica de seguimiento ocular, a través del algoritmo de transformada de Hough; para ello, utiliza una cámara montada que se alinea con el ojo del paciente y captura la posición de los ojos, luego se aplica técnicas de

procesamiento de imagen en tiempo real. En [20], se monitorea, evalúa y rastrea la participación de un alumno en situaciones y contextos de aprendizaje para proporcionar retroalimentación en tiempo real y posterior a la acción, con el fin de ayudar a los instructores a modular la participación del alumno. En [21], se desarrolla un sistema de seguimiento del movimiento ocular y sensor de movimiento de la cabeza, conectado a una infraestructura IoT para facilitar el esfuerzo diario de un paciente con Esclerosis Amiotrófica Lateral (ELA); este sistema fue creado para ayudar a pacientes con ELA a comunicarse con los demás mediante el uso de sus movimientos oculares. En [22], se desarrolló E-ACE (Intercambio de comunicación alternativa basado en ojos), el cual es un sistema basado en teléfonos inteligentes que permite al usuario utilizar nueve movimientos oculares, donde el usuario puede llamar, enviar mensajes y alertar a las personas involucradas con solo dos movimientos oculares.

Con el fin de llevar a cabo implementaciones similares a las descritas anteriormente, se ha hecho uso de varias herramientas que ofrece tanto IoT y el Seguimiento de Mirada. En [1] [23] [24] [25] [26] [27], se hizo uso de un dispositivo que incluye tanto cámaras como un algoritmo para realizar el seguimiento de mirada. En [1] [26], además se utilizó un protocolo de comunicación de Internet y varios dispositivos IoT. En [23], se usa un dispositivo de seguimiento de mirada basado en pantalla para seleccionar pictogramas de una interfaz. Además del algoritmo, en [28] [29] [30] [31], se encontró que hacían uso de un dispositivo con cámara infrarroja (IR) para obtener datos de la mirada. También, se encontró que en [32] [33] [18] [34], se usó una cámara web y una librería de código abierto para realizar el seguimiento de mirada. Adicional, en [19] [35], se utilizó lentes que contenía una cámara de video y en [36] [15] [21], se hace uso de una cámara IR. En [37], los lentes contenían los dos tipos de cámara.

De lo anterior se observa que las herramientas indispensables para realizar el seguimiento de mirada son: un dispositivo para obtener información de la mirada del usuario en tiempo real y un algoritmo/librería que procese dicha información y prediga la posición de la mirada en la pantalla.

Debido a la importancia de la tecnología de seguimiento de miradas, este trabajo tiene la intención de realizar un estudio del arte de la misma de los últimos 5 años (desde el 2016). Con este estudio se analizará los diferentes tipos de hardware y software que se pueden utilizar para la implementación de soluciones basados en el seguimiento de la mirada y las diferentes aplicaciones en las cuales se puede aplicar esta tecnología. Creemos que el presente trabajo será de gran contribución para las personas que inician su investigación en el área y pueda entender la situación y la tendencia de la tecnología de seguimiento de miradas.

Soluciones de seguimiento de mirada

IoT es una tecnología que se encuentra en auge y al ser combinada con el Seguimiento de mirada ha permitido crear nuevas interfaces para que las personas puedan interactuar con sistemas y dispositivos de una forma diferente. Las soluciones basadas en estas dos tecnologías también están haciendo posible la creación de interfaces más inclusivas, permitiendo que la tecnología llegue a un mayor número de personas como, por ejemplo, personas con discapacidad motora severa y del habla. Por ende, es necesario conocer los elementos que hacen posible el seguimiento de mirada.

Una solución común basada en seguimiento de mirada hace uso de dos elementos principales: un dispositivo para capturar datos de la mirada en tiempo real y una librería para procesar los datos y determinar la dirección de la mirada. Existen diferentes tipos de dispositivos y librerías para realizar el seguimiento de mirada, los cuales se detallan a continuación.

Tipos de dispositivos

Existe una gran cantidad de dispositivos que son empleados en el seguimiento de mirada. Lo común de estos dispositivos es que permiten recolectar datos sobre la mirada del usuario. Para cumplir con este objetivo, los dispositivos hacen uso de otros elementos como cámaras de video, cámaras infrarrojas y sensores de ondas cerebrales, empleando diferentes métodos. Entre los métodos principales, se encuentran el de pupila oscura y pupila brillante. Estos equipos trabajan con diferentes frecuencias de muestreo que dependen de las características del hardware. La precisión y exactitud es otra diferencial. Algunos dispositivos ofrecen un margen de error aceptable, incluso cuando el usuario realiza movimientos de la cabeza. Dependiendo de los elementos empleados, los equipos trabajan a diferentes condiciones de iluminación. El peso y longitud varía para cada dispositivo y son compatibles con diferentes sistemas operativos. Los dispositivos de seguimiento de mirada se pueden clasificar por la forma en la que interactúan con el usuario y se clasifican en: basados en pantalla, lentes o montados en la cabeza y estabilizados en la cabeza [38].

Basado en pantalla

La característica principal de estos dispositivos es que están ubicados en frente del usuario, ya sea arriba o debajo de la pantalla donde se encuentra la interfaz de interacción. La información que capturan estos dispositivos para realizar el seguimiento de mirada está enfocada en el centro de la pupila, el reflejo de la córnea y la posición de la cabeza. Las soluciones desarrolladas con estos dispositivos son amplias, entre ellas podemos

encontrar soluciones para controlar dispositivos IoT [23], interactuar con un computador [35], evaluar experiencias de usuario [17], entre otras. Dentro de esta categoría se encontraron los siguientes dispositivos ¹.

- Tobii Pro Fusion
- Tobii Pro Nano
- Smart Eye AI-X
- EyeTech VT3 Mini
- Eyegaze Edge Encore Camera
- Eyegaze Edge Prime Camera
- GazePoint 3 (GP3)
- Webcams

Lentes o montadas en la cabeza

Este dispositivo generalmente es usado en forma de anteojos, los componentes que forman este dispositivo son una cámara ubicada en el camino visual de los ojos y una cámara adicional que se encarga de grabar el campo de visión. La aplicación de estos dispositivos en el mundo real es muy amplia. Pueden ser usados para realizar compras en tiendas, entrenamiento deportivo, simuladores, etc. [38]. Por ejemplo, en [39] se planteó la creación de un sistema de alerta para detectar la fatiga de un conductor, usando lentes de seguimiento ocular; En [40] se realizó una investigación sobre el registro y análisis de la atención de un piloto durante los procedimientos de despegue y aterrizaje; En [41] se utilizó para recopilar datos de entrenamiento para redes neuronales con lentes de seguimiento ocular.

Los sistemas de seguimiento ocular basados en lentes son sencillos y cómodos de utilizar, además son menos invasivos para las personas a diferencia de otros dispositivos que dan poca libertad de movimiento y que causan fatiga al usuario. Sin embargo, este tipo de dispositivos tienen algunas limitaciones, como el movimiento de los ojos fuera del campo de visión que capturan las cámaras, así como también, el rastreo de la mirada cuando la luz solar genera un brillo excesivo [38]. Dentro de esta categoría se encontraron los siguientes dispositivos ¹.

- SMI Eye Tracking Glasses
- Tobii Pro Glasses 3

¹ Los detalles de cada dispositivo se muestran en el Anexo I.

Estabilizados en la cabeza

Algunas áreas de investigación, como el campo de la Resonancia Magnética Funcional, requiere de precisión y exactitud. Para ello se debe estabilizar la cabeza del participante al momento de efectuar el experimento, dejando en segundo plano el confort del usuario. Para estas áreas de investigación, existen los dispositivos de seguimiento de mirada montados en la cabeza. Estos dispositivos ofrecen mayor precisión a cambio de minimizar la buena experiencia de usuario. Una limitación importante de estos dispositivos es que no pueden ser empleados en escenarios del mundo real [38] o en soluciones que puedan facilitar la realización de tareas de la vida diaria. Un representante de esta categoría es el EyeLink Plus 1000 ¹.

Tipos de librerías

Existe una amplia variedad de librerías empleadas en el seguimiento de mirada. Estas librerías permiten realizar el procesamiento del flujo de datos proveniente de los dispositivos y determinar la posición y dirección de la mirada [42]. Estas herramientas de seguimiento de mirada están escritas en diferentes lenguajes de programación y son compatibles con diferentes sistemas operativos. Durante la fase de revisión se encontraron librerías de código abierto y de pago. También se encontró que algunas de ellas eran compatibles con dispositivos de seguimiento ocular específicos. A continuación, se listan las librerías encontradas ².

- Eye Like
- WebGazer
- OpenCV
- Turker Gaze
- Gaze Tracking
- Pupil Core
- OGAMA (Open Gaze And Mouse Analyzer)
- Gaze Detection
- OpenGazer
- iMotions
- Tobii Pro SDK
- Tobii Pro Glasses 3 API
- SMI BeGaze
- SmartGaze
- Eyeware – GazeSense
- openEyes
- GazeParse
- GazePointer
- GazePoint
- PyGaze

¹ Los detalles de cada dispositivo se muestran en el Anexo I.

² Las características de cada librería se muestran en el Anexo II.

Tipos de Aplicaciones

El seguimiento de mirada y IoT están siendo aplicadas en distintas áreas y subáreas del conocimiento. La clasificación de estas áreas resulta extensa, por lo que se ha utilizado las áreas del IoT para determinar en cuales de estas se aplica más el seguimiento de miradas. Una clasificación formal de las áreas de IoT se obtiene de [43], en donde se encuentran las siguientes: ciudades inteligentes, salud, educación, deporte, interacción humano-computador, psicología, agricultura, industria automotora y casas inteligentes.

Salud y Bienestar

Dentro del área de la Salud se encuentran aplicaciones que ayudan a obtener un mejor diagnóstico de enfermedades relacionadas con la visión y también, aquellas destinadas a ayudar a personas con discapacidad motora. Por ejemplo, en [1] se creó un sistema para controlar electrodomésticos a través del seguimiento ocular utilizando un dispositivo Tobii, dirigido a personas adultas mayores y con necesidades especiales. Otra aplicación orientada a ayudar a este grupo de personas se desarrolló en [19]. En este artículo se pudo controlar una silla inteligente a través del seguimiento de mirada, la silla podía moverse en diferentes direcciones derecha, izquierda o pare. El seguimiento de mirada en el área de salud ha sido utilizado en personas con problemas de lenguaje, discapacidades motoras, personas con lesiones de la médula espinal, Esclerosis Lateral Amiotrófica, pacientes con demencia frontotemporal, pacientes con Alzheimer, etc.

Psicología

En el campo de la Psicología, las aplicaciones han sido utilizadas para determinar el comportamiento humano a través de las expresiones faciales y también para ayudar en el diagnóstico de enfermedades mentales. En el reconocimiento de expresiones faciales se analizan varios elementos del rostro como la boca, los labios, y en especial los ojos. A través del análisis de estos elementos, se pudo identificar expresiones faciales comunes de las personas cuando son expuestas a diferentes estímulos, con el fin de crear un modelo de aprendizaje de máquina que reconozca las expresiones faciales humanas [26]. En [44], se pudo diferenciar pacientes que tenían esquizofrenia mediante la medición de los movimientos de los ojos como la longitud de la ruta de escaneo del entorno, la duración de las fijaciones y la diferencia de la posición de la mirada con respecto a un punto de referencia. Si adicionalmente se mide otros movimientos de los ojos como movimientos sacádicos superpuestos, movimientos horizontales repetidos y movimientos circulares, se podría determinar si una persona padece de Delirio. En [45], se analizaron estas

mediciones para tratar de determinar pacientes con Delirio en el Hospital General de Massachusetts.

Interacción Humano-Computador (HCI)

En HCI, el seguimiento de mirada ha sido utilizado para crear nuevas interfaces de interacción, las cuales son manipuladas a través del movimiento de los ojos. Estas interfaces permiten que los usuarios puedan acceder a Internet e incluso al correo electrónico utilizando el seguimiento de mirada [36]. Estas dos tecnologías pueden resultar muy atractivas en museos, donde las personas pueden interactuar con interfaces de exposición de arte [23]. También existen aplicaciones de uso común en las actividades diarias de los usuarios como el escribir un mensaje con un teclado virtual [46].

Industria Automotora

En la industria Automotora, el seguimiento de mirada está siendo utilizada para generar un ambiente de conducción más seguro. Por ejemplo, se ha utilizado carros inteligentes que alertan al usuario cuando este se encuentra en estado de somnolencia, es decir, el sistema del vehículo detectará el parpadeo prolongado y enviará una alerta al usuario para evitar accidentes de tránsito [39]. En [47], se evaluó la atención visual que tienen los usuarios en respuesta a colisiones cuando conducían vehículos semiautónomos.

Casas Inteligentes

En esta área de aplicación, el seguimiento de mirada ha permitido implementar casas inteligentes que puedan ser controladas a través de movimientos oculares. Dentro de las casas, varios son los dispositivos IoT que pueden ser controlados con la mirada, entre los más comunes se encuentran focos, ventiladores, puertas, TV, campanas, timbres y radio [48] [27]. En su mayoría, este tipo de soluciones están orientadas para personas con discapacidad motora y del habla, para que ellos puedan controlar los dispositivos de IoT [1] [31].

Educación

En el área de la Educación, las aplicaciones se centran en evaluar el aprendizaje y en generar una educación inclusiva. Por ejemplo, en [49], se utilizó un rastreador ocular para asignar la atención de estudiantes a diferentes valencias de retroalimentación, es decir que el estudiante debía completar una tarea y según su atención visual se clasificaba como retroalimentación positiva o negativa. El seguimiento de mirada en la educación también permite que se pueda evaluar la participación del estudiante en situaciones y contextos de aprendizaje y a su vez proporcionar retroalimentación en tiempo real [20].

2 METODOLOGÍA

La Figura 1 muestra las etapas del proyecto y metodologías que se van a emplear para su desarrollo. Para las etapas uno y dos se utilizó la Metodología Investigación-Acción, mientras que la Metodología de Desarrollo Iterativo e Incremental fue la guía para el desarrollo de las etapas tres, cuatro y cinco.

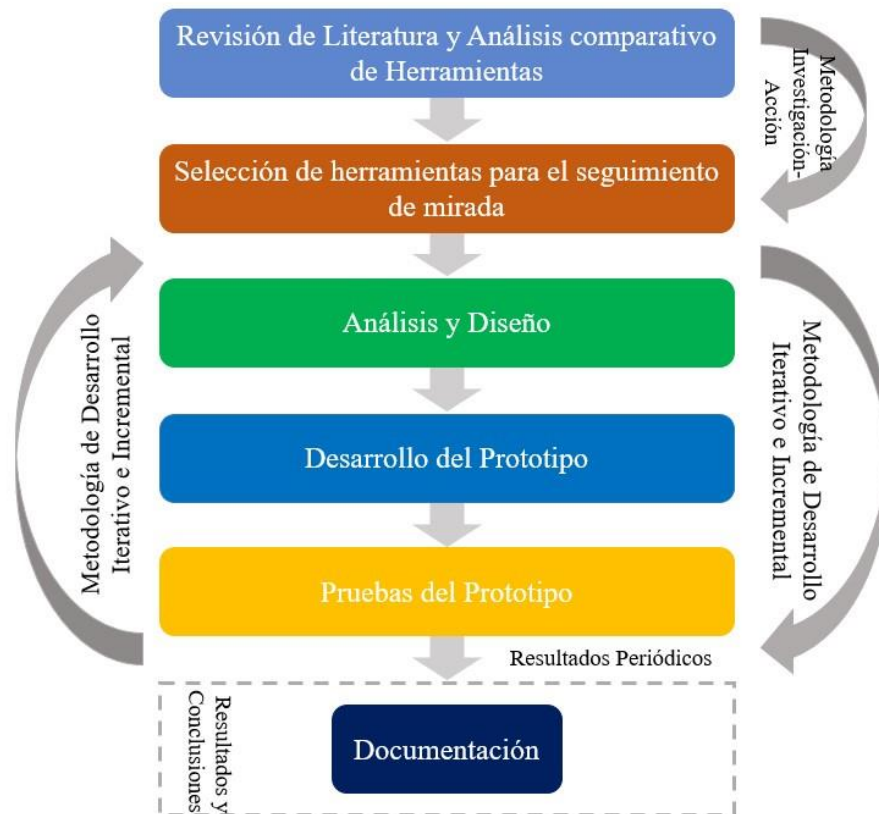


Figura 1. Etapas del Proyecto y metodologías utilizadas

La Metodología Investigación-Acción [2] [50] permitió analizar la tendencia de los dispositivos hardware y librerías utilizadas en el seguimiento de mirada, desde una perspectiva técnica, y las aplicaciones en las cuales se pueden utilizar. Esta metodología se combinó con una revisión sistemática de literatura [51] [52], dando como resultado un método compuesto por tres fases principales (ver Figura 2).

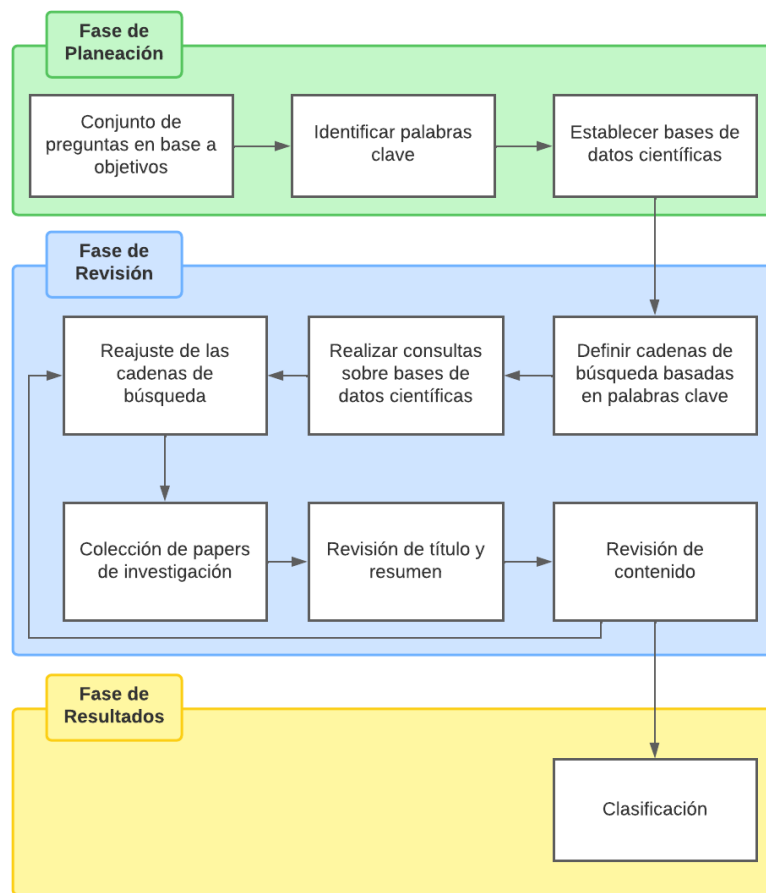


Figura 2. Método de Investigación-Acción.

Una vez completado las etapas uno y dos de la Figura 1, la Metodología de Desarrollo Iterativo e Incremental fue puesta en marcha. Esta metodología es un marco de trabajo que permite la gestión de proyectos y que permite un crecimiento progresivo e iterativo de un proyecto hasta alcanzar el resultado esperado [53]. Como resultado de aplicar esta metodología para el desarrollo de este prototipo, se obtuvo 4 incrementos con 4 etapas cada uno (Figura 3). En la etapa de Análisis, se obtuvieron los requerimientos funcionales y características operacionales del prototipo. En la etapa de Diseño se empleó mockups para proponer una solución a los requerimientos obtenidos. En la etapa de Desarrollo se implementó la solución obtenida en la etapa anterior. En la cuarta etapa se realizaron pruebas a la solución obtenida y los errores encontrados fueron solucionados en el siguiente incremento.

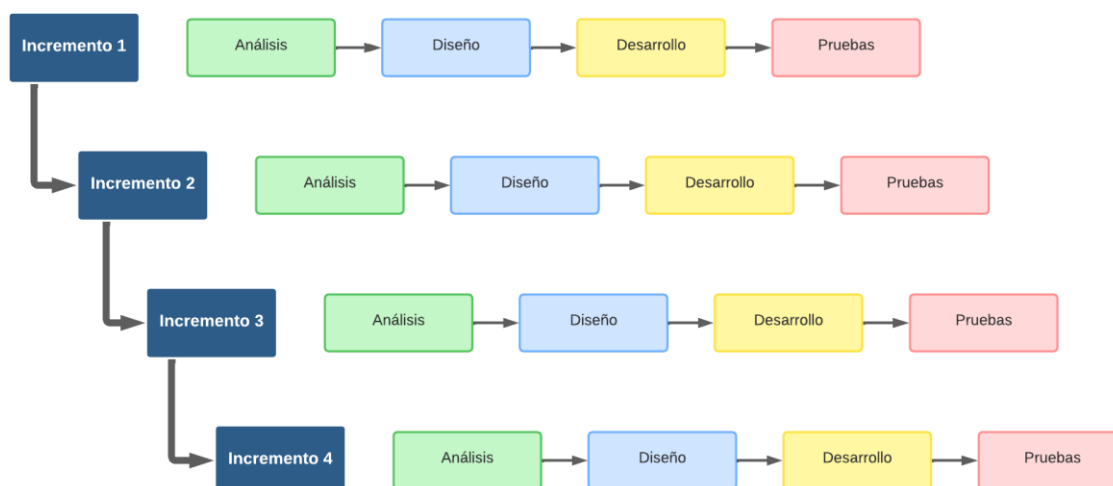


Figura 3. Método Iterativo e Incremental aplicado en el prototipo.

Como última etapa de la Figura 1, se tiene la documentación asociada a la creación del prototipo, en la cual se incluye los resultados y conclusiones.

3 DESARROLLO DE LA SOLUCIÓN

3.1 Selección de herramientas para el seguimiento de mirada

Dispositivos

Es importante indicar que la selección de un dispositivo para el seguimiento de mirada dependerá del uso y aplicación en el que se lo va a emplear. Si la precisión del dispositivo no afecta drásticamente los resultados de la investigación, o si se cuenta con pocos recursos para la misma, se puede utilizar un dispositivo de bajo costo como una Webcam. Para mayor detalle se muestra la Tabla 1, la cual contiene los dispositivos ordenados de menor a mayor precio. En esta tabla también se puede observar que GazePoint 3 es un dispositivo promedio en cuanto a costos y especificaciones. Por otro lado, si la precisión es muy importante para la investigación, más allá del costo, se podría utilizar un dispositivo con las mejores especificaciones técnicas. Un ejemplar de este grupo podría ser Tobii Pro Fusion, que tiene la mejor exactitud y una excelente frecuencia de muestreo.

En general, la Tabla 11 (ver Anexo I) contiene varias características de los dispositivos que pueden ser usadas como criterios de selección en función de los objetivos del trabajo a realizar. Por ejemplo, si se requiere un dispositivo que ofrezca la mayor libertad de movimiento, SMI Eye Tracking Glasses podría ser la mejor selección; por otro lado, si se requiere que la persona esté lo más alejada de la pantalla, Tobii Pro Fusion o Smart Eye AI-X son los candidatos ideales.

Tabla 1. Dispositivos ordenados de menor a mayor precio.

Dispositivo	Precio [USD]	Exactitud [grados]	Procesamiento Interno	Frecuencia de muestreo [Hz]
Webcams	\$ 15+	4.3	No	60
GazePoint 3 (GP3)	\$ 845	1.0	No	60
Smart Eye AI-X	\$ 3450	0.5	No	60
Eye Tech VT3 Mini	\$ 4000	0.5	Sí	200
Tobii Pro Fusion	\$ 13900	0.3	Sí	120
Tobii Pro Nano	\$ 10000+	0.3	Sí	60
Eyegaze Edge Prime Camera	\$ 7500	0.45	Sí	60
Eyegaze Edge Encore Camera	\$ 7500	0.45	Sí	50
Tobii Pro Glasses 3	\$ 10000+	0.6	No	100
Eye Link 1000	\$ 10000+	0.5	No	2000
SMI Eye Tracking Glasses	\$ 11900	0.5	No	120

Librerías

A diferencia de los dispositivos de seguimiento de la mirada, seleccionar una biblioteca puede ser un trabajo más complejo. Por esta razón, para brindar un criterio de selección de librerías de seguimiento de la mirada, se propuso un método que consistía en seleccionar la mayor cantidad de características de la Tabla 12 (ver Anexo II) y asignarles una ponderación. El criterio de ponderación resultante se muestra en la Tabla 2, en donde el peso más alto (35%) se dio al Grado de prestaciones, debido a que esta característica determina la compatibilidad con los diferentes dispositivos de seguimiento de mirada que fueron estudiados previamente; además, permite conocer si la librería realiza seguimiento de mirada o solo seguimiento de ojos. La segunda característica más importante es el Número de artículos ya que indica el reconocimiento que tiene la librería dentro de las investigaciones científicas, por lo que se le asignó una ponderación del 30%. También, es importante conocer las condiciones de uso de la librería, es decir, si la librería puede ser usada, modificada o distribuida libremente, o si esta requiere ser pagada para su utilización; la característica que ofrece esta información es el Tipo de licencia, por lo que se le asignó el 15%. Debido a que existen máquinas virtuales que permiten simular cualquier sistema operativo, la característica de Compatibilidad con sistemas operativos no sería una limitación, sin embargo, el rendimiento es inferior al de un sistema operativo nativo, por lo que se le asignó el 10% a esta característica. La facilidad de instalación permite conocer que tan fácil o difícil es la instalación de la librería; una vez completada la instalación, indistintamente del grado de complejidad, esta característica dejará de ser importante, por lo que solo se le asignó una ponderación del 10%. Finalmente, el Soporte es una

característica que permite conocer las actualizaciones recientes de la librería para su correcto funcionamiento; sin embargo, una librería que no ha recibido actualizaciones en los últimos años podría funcionar correctamente. Por esto, el Soporte no sería una limitación al momento de elegir o probar una librería, por lo que se le ha asignado el 5%. En conjunto, todas estas características suman el 100%. Los resultados de evaluar las librerías con el método propuesto se muestran en la Tabla 3.

Adicionalmente, para ponderar los valores alto, medio y bajo de cada característica (Tabla 2), se decidió asignar el 46%, 34% y 20% a cada uno de los valores respectivamente. El porcentaje se tomó del valor total de la característica. De manera similar, se ponderaron los valores de la característica Tipo de Licencia. Para obtener la ponderación del número de artículos de cada librería, se multiplicó el número de artículos por 3, para garantizar que la librería con más artículos obtenga una ponderación del 30%. Finalmente, para la característica Soporte se realizó una clasificación binaria donde al Sí se le asigna todo el peso (5%) y el NO tiene peso de cero.

Tabla 2. Criterio de ponderación a las características.

CARACTERÍSTICA	PONDERACIÓN CARACTERÍSTICA [%]	VALOR	PONDERACIÓN DEL VALOR [%]
Grado de prestaciones	35	Alto	16
		Medio	12
		Bajo	7
Número de artículos	30	Cantidad	Cantidad * 3
Tipo de licencia	15	Open Source	7
		Freeware	5
		Pago	3
Compatibilidad con sistemas operativos	10	Alta	5
		Media	3
		Baja	2
Facilidad de instalación	10	Alta	5
		Media	3
		Baja	2
Soporte	5	Si	5
		No	0

Como se puede observar en la Tabla 3, OpenCV ocupa el primer lugar, seguido por OGAMA, Webgazer, iMotions y Pupil Core. Para completar la etapa de selección de librerías, se propuso hacer un paso adicional, el cual consistió en instalar y probar 5 librerías de esta tabla. Para seleccionar las librerías se consideraron las 5 primeras librerías que sean open source y que además sean compatibles con cámaras web. Esta

consideración se hizo debido a que no se disponía de dispositivos de seguimiento de mirada especiales, ni recursos económicos para este fin. Por lo tanto, las librerías instaladas y probadas fueron: OpenCV, WebGazer, Pygaze, GazeTracking y GazeParser. Para evaluar OpenCV se utilizó el proyecto Eyes-Tracking-Opencv-and-Dlib cuyo código fuente está disponible en el siguiente enlace: <https://github.com/Asadullah-Dal17/Eyes-Tracking-Opencv-and-Dlib>

Tabla 3. Evaluación de las librerías en base a sus características.

Librería	Grado de prestaciones	Número de artículos	Tipo de licencia	Complejidad de instalación	Compatibilidad con sistemas operativos	Soporte	Puntaje /100
OpenCV	16	24	7	3	5	5	60
OGAMA	7	30	7	2	2	5	53
WebGazer	16	12	7	3	5	5	48
iMotions	12	18	3	5	-	5	43
Pupil Core	7	15	7	2	5	5	41
PyGaze	12	6	7	5	5	0	35
Gaze Tracking	16	0	7	5	0	5	33
SMI BeGaze	12	6	3	5	2	5	33
GazeParse	16	3	7	2	5	0	33
Eyeware – GazeSense	16	0	3	5	3	5	32
GazePointer	12	6	5	5	2	0	30
Turker Gaze	12	0	7	5	5	0	29
GazePoint	16	0	5	2	5	0	28
Eye Like	7	6	7	2	5	0	27
Tobii Pro-SDK	7	0	5	5	5	5	27
openEyes	12	0	7	3	5	0	27
Tobii Pro Glasses 3 API	7	0	5	3	5	5	25
OpenGazer	12	0	7	2	3	0	24
Gaze-Detection	7	0	7	3	3	0	20
SmartGaze	7	0	3	5	-	5	20

La instalación y prueba de las librerías, se realizó en dos computadoras diferentes con las siguientes características: 1) Windows 10, Intel Core I7, 16 GB y cámara web incrustada en el computador; 2) Windows 10, Intel Core I7, 16 GB y cámara web externa. Un resumen de los resultados obtenidos se muestra en la Tabla 4.

Tabla 4. Resumen de los resultados obtenidos de las cinco librerías probadas.

Librería	Resumen
OpenCV (Eyes- Tracking- Opencv-and- Dlib)	<ol style="list-style-type: none"> 1. Para la instalación es necesario clonar un repositorio e instalar algunas dependencias como: Dlib, Cmake y Numpy. 2. Es una librería que funciona offline. 3. Detecta la mirada cuando el usuario mira hacia la derecha, izquierda, centro. No detecta la mirada arriba ni abajo. Adicionalmente detecta el parpadeo de ojos. 4. Se requiere que el usuario gire mínimo 22° y máximo 50° para detectar la posición hacia la izquierda o derecha. 5. Si el usuario se encuentra a una distancia menor a 26 cm, ya no realiza la detección de ojos, ya que pasa por un procesamiento previo el cual consiste en detectar el rostro y luego los ojos. 6. Cuando el usuario se encuentra a una distancia mayor a 90 cm, ya no realiza la detección de ojos o lo hace con baja precisión. 7. Se requieren condiciones de luz óptimas. Esto es, utilizar la luz del día, o también el uso de luz artificial, como una linterna, para una buena detección. Se probó en condiciones con poca luminosidad y no se realizó la detección de ojos. 8. Se requiere que el movimiento de los ojos sea pronunciado para saber dónde está mirando el usuario, caso contrario no detecta la posición y reconoce como si estuviera observando al centro de la pantalla.
Webgazer	<ol style="list-style-type: none"> 1. Webgazer nos da el punto (x, y) que el usuario está mirando en la pantalla. 2. Requiere una calibración previa para una mejor precisión a partir de clics y de movimientos de los ojos. 3. Es necesario que los ojos y el rostro se visualicen en el cuadro (verde) delimitador para que se puede hacer un correcto procesamiento de imágenes. Por lo tanto, la distancia a la que se probó y funciona correctamente es de 33 a 43 cm. A una distancia mayor a 50 cm sí detecta, pero se tiene una baja precisión. 4. El usuario no requiere realizar un movimiento pronunciado o exagerado de los ojos para la detección. 5. El usuario puede girar su cabeza máximo 28° horizontalmente, para que la librería pueda realizar una correcta detección de ojos. 6. Se requiere buenas condiciones de luz, como la luz del día o uso de una linterna, ya que en condiciones oscuras o con baja iluminación no realiza la detección de los ojos. 7. Es una librería que detecta los dos ojos, por lo tanto, es binocular.

	<p>8. Para que la librería funcione offline, se siguió los pasos descritos en la página oficial de la librería, pero no se obtuvo resultados satisfactorios.</p>
GazeParser	<ol style="list-style-type: none"> 1. La iluminación es algo importante para que la librería pueda detectar el ojo, se utilizó la linterna del celular para tener una condición de luz aceptable y la librería pueda detectar el ojo correctamente. 2. Por defecto la librería detecta un solo ojo, pero esto es configurable. 3. La librería detecta los ojos escaneando la parte más oscura del rostro que es la pupila, por lo cual tiende a confundir con otras zonas oscuras que se encuentran alrededor. 4. La librería está configurada para trabajar con una cámara ubicada en la parte inferior de la pantalla. Por lo que es obligatorio tener una cámara web externa. 5. No se obtuvo una prueba satisfactoria de la librería debido a que utiliza otra herramienta para recolectar datos (Simple Gazer Parser) y la comunicación de los dos es mediante el protocolo TCP/IP y no pudo conectarse. 6. Depende de otras librerías: pshycopy, pil, matplotlib, wxPython, scipy. 7. Se requieren condiciones de luz óptimas. Esto es, utilizar la luz del día, o también el uso de luz artificial, como una linterna, para una buena detección.
GazeTracking	<ol style="list-style-type: none"> 1. Para la instalación es necesario clonar un repositorio e instalar algunas dependencias como: Dlib, OpenCV y Numpy. 2. Además, Dlib, requiere la instalación de Boost, Boost.Python, Cmake y X11. Boots y Cmake se lo instaló automáticamente mediante el comando pip. Mientras que Boost.Python requiere instalación manual. 3. Es una librería que funciona offline. 4. Detecta la mirada cuando el usuario mira hacia la derecha, izquierda y centro. No detecta la mirada arriba ni abajo. Adicionalmente detecta el parpadeo de ojos. 5. Nos indica la posición exacta de la pupila en coordenadas x e y con respecto a la pantalla. 6. El movimiento de los ojos debe ser exagerado para que la librería detecte si el usuario mira hacia la derecha o hacia la izquierda. Cuando el movimiento de los ojos es normal, es decir no hay esfuerzo para mirar a la derecha o izquierda, la librería tiene una baja precisión al detectar la posición. 7. También, se pudo observar que cuando el usuario mira hacia la derecha, la librería tiene problemas para detectar esa posición y tiende a confundir con la posición del centro.

	<p>8. La posición de la cámara debe estar frontal al usuario, caso contrario si la cámara se encuentra en la parte inferior de la pantalla, la librería no realiza la detección.</p> <p>9. Es necesario tener buenas condiciones de iluminación para que la librería funcione correctamente. Se realizó una prueba con poca iluminación y cualquier movimiento de los ojos lo detectaba como parpadeo.</p> <p>10. Es una librería binocular que detecta los dos ojos.</p> <p>11. La posición ideal del usuario frente a la cámara es de 60cm. Cuando el usuario se encuentra a una distancia de 35 cm, ya no realiza la detección.</p> <p>12. Luego de un tiempo prolongado de funcionamiento (aproximadamente 15 minutos), la librería se cerraba automáticamente, y no permitía seguir utilizando. Para solucionar este problema se debía limpiar la caché del programa para que vuelva a funcionar.</p> <p>13. Es necesario que se visualice el rostro completo del usuario para que la librería pueda detectar la pupila, caso contrario la librería no detecta ninguna posición de la mirada.</p>
Pygaze	<p>1. Depende de las librerías PyGame, Numpy, Matplotlib, PIL, VideoCapture, OpenCV y Scipy.</p> <p>2. La librería permite detectar únicamente un ojo.</p> <p>3. Detecta únicamente la pupila, utilizando la técnica dark pupil, que es la parte más oscura de la imagen. Por esta razón tiende a confundir la pupila con otras zonas oscuras de la imagen como las cejas.</p> <p>4. Las salidas son la posición de la pupila, el color de la pupila y las medidas del rectángulo que encierra la pupila. Es una librería que no realiza el seguimiento de la mirada.</p> <p>5. Debido a que esta librería no tiene soporte, no es compatible con la versión de Python 3 o superior. Tampoco con las nuevas versiones de las librerías de las cuales depende.</p> <p>6. Requiere de una pre-calibración que consiste en decirle al programa cual es la zona del ojo.</p> <p>7. Las condiciones de iluminación deben ser normales, esto es, no se utilizó una linterna adicional como en las anteriores.</p> <p>8. A una distancia menor a 20 cm y mayor a 40 cm la librería no realiza la detección de la pupila.</p>

Al finalizar este proceso el dispositivo y la librería seleccionados para desarrollar este prototipo fueron:

- Cámara web: Es un dispositivo que ofrece características técnicas similares a la mayoría de los dispositivos de la Tabla 11 (ver Anexo I) y es el más económico de

todos. Debido a su facilidad y bajo costo de adquisición, la cámara web es ideal para el desarrollo de este proyecto que está orientado a personas con discapacidad, las cuales pertenecen mayormente a la clase media y baja [54]. Otra característica que permitió su elección fue el número de artículos en la que fue referenciada.

- WebGazer: Es una librería de código abierto y compatible con cámaras web. Esta librería se diferencia de otras similares por ser binocular, por su capacidad de realizar el seguimiento de mirada y predecir la posición de la mirada del usuario en la pantalla en tiempo real. Además, esta librería no requiere movimiento pronunciado o exagerado de los ojos, lo cual disminuye el cansancio visual del usuario. Es una librería que está escrita en JavaScript lo que facilita la integración con aplicaciones web. Estas son las características principales por las que WebGazer fue seleccionada para este proyecto.

A continuación, se muestran las actividades realizadas para el desarrollo del prototipo.

3.2 Incremento 1

Análisis de requerimientos

El análisis de requerimientos permite tener una visión clara del funcionamiento operacional del prototipo. En esta fase se analizó los requerimientos básicos para la funcionalidad del prototipo y como resultado se obtuvieron 15 requisitos en el formato Historias de Usuario (ver Anexo III).

Diseño de interfaces de usuario

En base a los requisitos obtenidos en la fase anterior y con la ayuda de una herramienta de diagramación, se realizó los mockups del prototipo en bajo nivel. En la Figura 4 se muestra el mockup de la interfaz principal y en el Anexo IV se muestran los mockups de las interfaces restantes.

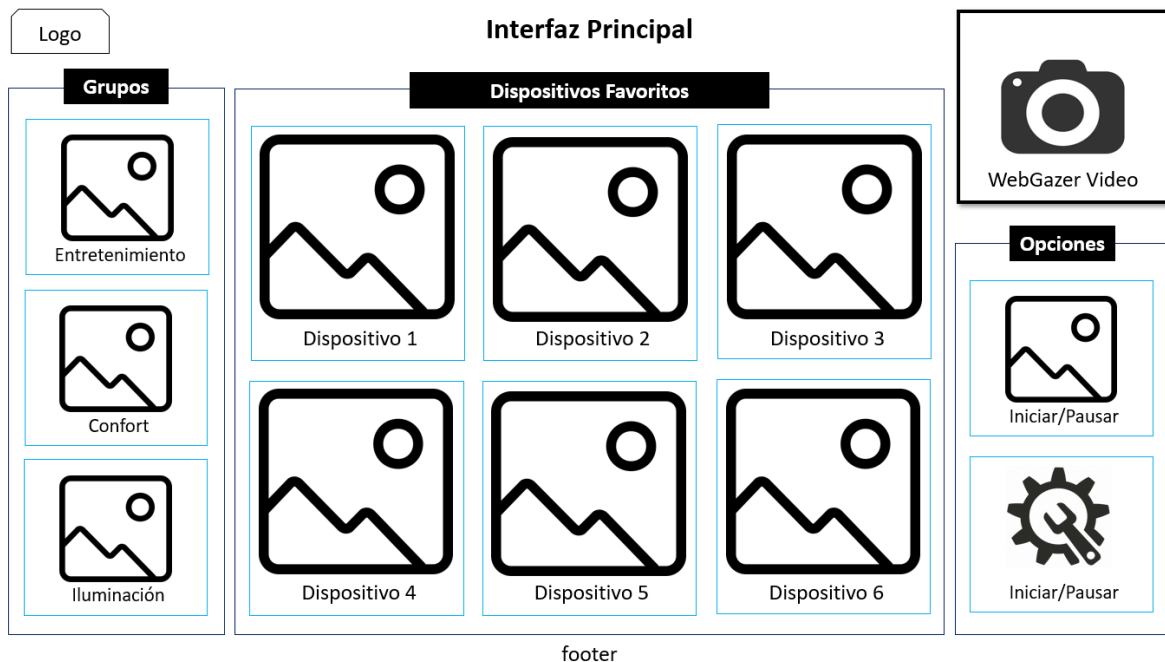


Figura 4. Mockup Interfaz Principal

Desarrollo de la interfaz de usuario

Una vez definidas las interfaces de usuario en bajo nivel, se continuó con el desarrollo de las interfaces en alto nivel. En esta fase se espera la navegación funcional entre todas las interfaces, para lo cual se utilizó los siguientes lenguajes:

- HTML: se utilizó para darle un diseño al contenido de las interfaces.
- CSS: se utilizó para dar estilos a las etiquetas HTML.
- JavaScript: fue usado para implementar la lógica de navegación entre las interfaces y su funcionalidad.

La navegación entre interfaces se la realizó utilizando el seguimiento de mirada, para lo cual se utilizó la librería WebGazer, la misma que se encuentra escrita en JavaScript.

Pruebas

Lógica de navegación

Estas pruebas se hicieron con el objetivo de verificar la navegación correcta entre interfaces y la funcionalidad de los íconos. Como resultado se obtuvo que el ícono de Añadir Favorito mostraba un comportamiento erróneo.

Precisión de Webgazer

Para evaluar la precisión de Webgazer, se tomaron 100 datos tanto para la interfaz principal, televisión y foco. Para cada una de estas interfaces se evaluó los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. También, se registró el tiempo que el usuario tardaba en completar una tarea, la misma que consiste en seleccionar un ícono. Estas pruebas fueron realizadas por 2 usuarios en dos equipos diferentes, obteniendo una precisión promedio de 86.7% y 92.05% respectivamente.

Tabla 5. Precisión de Webgazer

	Usuario 1	Usuario 2
Tiempo de fijación [s]	1.5	2.5
Interfaz Principal	61.22 %	83.67 %
Foco	100 %	99.00 %
Televisión	98.89%	93.47 %
Precisión Promedio	86.7 %	92.05 %

3.3 Incremento 2

Para mejorar la precisión obtenida en el Incremento 1, se realizó una nueva iteración la cual permitió añadir nuevos requerimientos, modificar las interfaces y realizar nuevas pruebas.

Análisis de requerimientos

Se optó por añadir una ventana de calibración para mejorar la precisión de WebGazer.

Diseño de interfaces de usuario

De acuerdo al análisis anterior, se diseñó el mockup de la interfaz calibración, la cual contenía 17 puntos de calibración distribuidos uniformemente dentro de la pantalla.

Adicionalmente se redujo el número de íconos de las interfaces propuestas en el Incremento 1, dando como resultado la interfaz ilustrada en la Figura 5. Para visualizar las demás interfaces ver el Anexo IV.

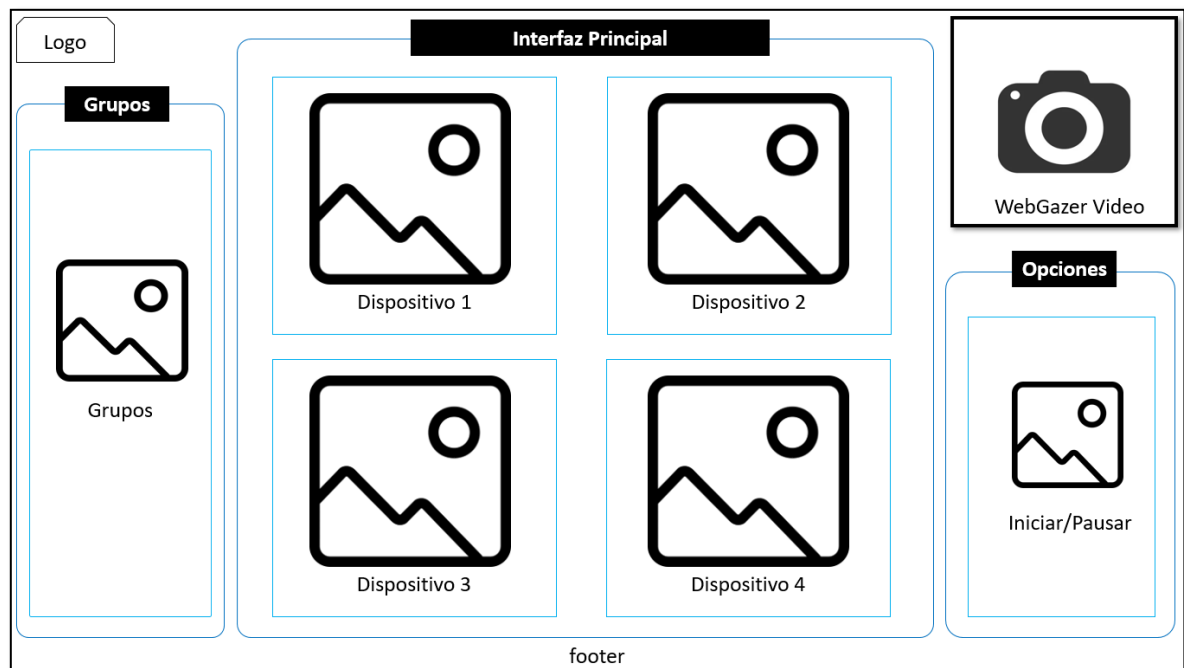


Figura 5. Mockup Interfaz Principal Mejorada

Desarrollo de la interfaz de usuario

La implementación de la interfaz calibración, se la realizó en base a la plantilla de calibración propuesta por el equipo de Webgazer, la cual contenía 9 puntos de calibración con 5 clics cada uno y un menú de opciones. Sin embargo, para este prototipo se consideró utilizar 17 puntos de calibración con 3 clics cada uno, esto con el fin de mejorar la precisión del seguimiento de mirada de la librería. También se añadió el submenú “Interfaz Principal”, el cual permitía acceder a la Interfaz Principal. Luego, se modificaron las interfaces resultantes del Incremento 1 acorde a los mockups obtenidos en la fase de diseño. Finalmente, se corrigió el error encontrado en el incremento anterior.

Pruebas

Precisión Webgazer

Para evaluar la precisión de Webgazer, se tomaron 100 datos tanto para la interfaz principal, televisión y foco. Para cada una de estas interfaces se evaluó los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. También, se registró el tiempo que el usuario tardaba en completar una tarea, la misma que consiste en seleccionar un ícono. Estas pruebas fueron realizadas por 3 usuarios en dos equipos diferentes, obteniendo una precisión promedio de 98.99%, 96.59% y 96.97% respectivamente, como se muestra en la Tabla 6.

Tabla 6. Precisión Webgazer, Incremento 2

	Usuario 1	Usuario 2	Usuario 3
Tiempo de fijación [s]	1.5	2	2.5
Tiempo promedio en completar una tarea [s]	3.14	6.29	6.86
Interfaz Principal	100%	96%	99%
Foco	100%	98.99%	96.97%
Televisión	96.97%	94.79%	94.95%
Precisión Promedio	98.99%	96.59%	96.97%

3.4 Incremento 3

Análisis de requerimientos

Se añadió los requisitos R14, R15 (ver Anexo III), los cuales permitieron, crear una cuenta de usuario e iniciar sesión con nombre de usuario y contraseña. Además, se optó por crear un canal en un servidor MQTT para el usuario.

Diseño

Diseño de interfaces de usuario

El diseño utilizado como interfaz Login, fue descargado de Colorlib templates [55] y posteriormente personalizado y modificado para cumplir con los requisitos propuestos.

Diseño de la arquitectura

Para establecer la comunicación entre la interfaz de usuario y los dispositivos IoT, se planteó utilizar un servidor MQTT en el cual, el cliente publica un mensaje y el Bróker de comunicación lee el mensaje. Para ello, el cliente debe tener un canal en el servidor MQTT y un Bróker de comunicación que debe estar suscrito a ese canal. Con la finalidad de dar solución a estos requerimientos se propuso el diagrama de secuencia de la Figura 6. En este diagrama, el cliente crea una cuenta en el servidor web, con nombre de usuario y contraseña. El servidor web registra los datos del cliente en la base de datos y a su vez crea un canal para el cliente en el servidor MQTT. La base de datos notifica al servidor web el registro satisfactorio del cliente, y el servidor MQTT envía el token del canal creado al servidor web. Luego, el servidor web genera un archivo con los datos necesarios para que el Bróker de comunicación se suscriba al servidor MQTT; este archivo es enviado al cliente. Posteriormente, el cliente envía el archivo de credenciales al Bróker de comunicación, el cual hace uso de estos datos para suscribirse al servidor MQTT. Finalmente, el servidor MQTT envía una respuesta al Bróker de comunicación con el estado de la suscripción, el cual puede ser exitoso o erróneo.

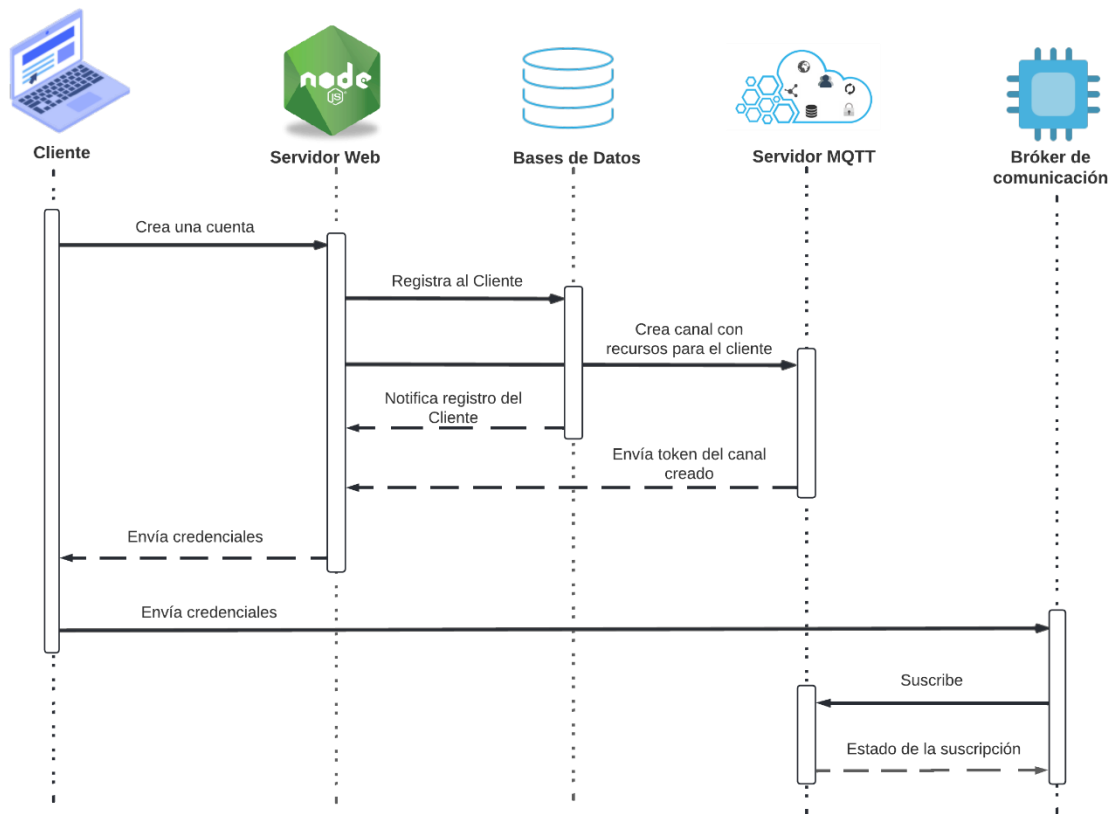


Figura 6. Proceso de obtención de credenciales y suscripción en el servidor MQTT.

Una vez completado el proceso de la Figura 6, el cliente podrá enviar instrucciones al dispositivo IoT empleando el diagrama de comunicación de la Figura 7. En este diagrama, (1) el cliente con sesión iniciada envía el nombre del recurso y el mensaje (instrucción) al servidor web; (2) El servidor web recibe el mensaje y lo publica en el servidor MQTT, dentro del canal del usuario y dentro del recurso especificado; (3) El Bróker de comunicación que se encuentra suscrito a ese canal puede leer el comando enviado por el cliente; (4) el Bróker de comunicación envía el comando al dispositivo IoT.

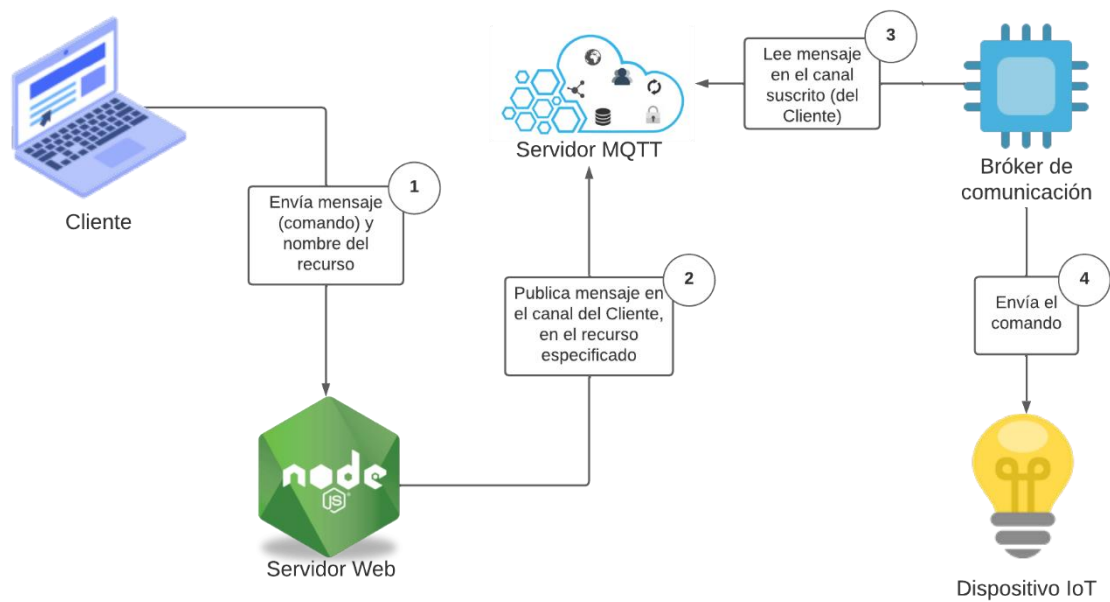


Figura 7. Proceso de comunicación entre cliente y dispositivo IoT

Desarrollo de la interfaz de usuario

Como se mencionó anteriormente la interfaz Login fue desarrollada por Colorlib. En esta interfaz se realizaron algunas modificaciones con la finalidad de que el usuario además de iniciar sesión pueda crear una cuenta. También se implementó el submenú “Descargar Credenciales” en la interfaz Calibración, a la cual se puede acceder una vez que el usuario haya iniciado sesión o haya creado una cuenta.

Debido a la implementación de la interfaz Login, fue necesario conectar el servidor a una base de datos, con el fin de almacenar datos de los usuarios y el controlar el inicio de sesión. De manera similar, se conectó el servidor web a un servidor MQTT para de esta forma crear un canal para el usuario, a medida que este se registra. Previo a este paso se creó una cuenta principal en Beebotte, que es el servidor MQTT utilizado en este prototipo.

Pruebas

Crear Cuenta

Se realizaron 3 casos de pruebas los mismos que se muestran en la Tabla 7. Se pudo verificar que los tres casos fueron satisfactorios.

Tabla 7. Casos de Prueba Crear Cuenta

N°	Prueba	Acción	Resultado
1	Nombre de usuario inexistente y Contraseña igual a Confirmar contraseña	<ul style="list-style-type: none"> - Registro del usuario en la base de datos. - Creación de un canal con recursos en Beebotte para el usuario. - Redirección a la Interfaz Calibración. - Generación del archivo de credenciales 	Exitoso
2	Nombre de usuario existente y Contraseña igual a Confirmar contraseña	<ul style="list-style-type: none"> - Muestra el mensaje: "El nombre de usuario ya existe, ingrese otro nombre de usuario" 	Exitoso
3	Nombre de usuario inexistente y Contraseña diferente a Confirmar contraseña	<ul style="list-style-type: none"> - Botón crear cuenta se deshabilita. 	Exitoso

Iniciar Sesión

Se realizaron 3 casos de pruebas los mismos que se muestran en la Tabla 8. Se pudo verificar que los tres casos fueron satisfactorios.

Tabla 8. Casos de Prueba Iniciar Sesión

N°	Prueba	Acción	Resultado
1	Usuario existente y Contraseña correcta	<ul style="list-style-type: none"> - Redirección a la interfaz Calibración - Acceso al archivo de credenciales 	Exitoso
2	Usuario incorrecto	<ul style="list-style-type: none"> - Muestra el mensaje: "Usuario incorrecto, intente nuevamente o cree una cuenta" 	Exitoso
3	Usuario existente y Contraseña incorrecta	<ul style="list-style-type: none"> - Muestra mensaje: "La contraseña es incorrecta, ingrese nuevamente" 	Exitoso

Conexión Cliente - Servidor Web - Beebotte

La prueba consistió en enviar un mensaje desde la interfaz de un dispositivo, ubicada en el lado del cliente. Luego, verificar que el mensaje y nombre del dispositivo lleguen al servidor web. Finalmente, constatar que en Beebotte el mensaje haya sido publicado en el canal del usuario dentro del recurso (dispositivo) especificado. Esta prueba se realizó varias veces desde la interfaz foco y televisión con diferentes mensajes. En todos los casos el resultado de las pruebas fue exitoso.

3.5 Incremento 4

Análisis de requerimientos

Se hizo énfasis en los requisitos R2 – R9 (ver Anexo III), los cuales están enfocados en la interacción entre el usuario y los dispositivos IoT.

Diseño

En el Incremento 3, se estableció satisfactoriamente la comunicación entre el Cliente y el Servidor MQTT (ver Figura 7). En el incremento actual, se realizó la comunicación entre el Servidor MQTT y el dispositivo IoT, empleando el diagrama de comunicación de la Figura 7. Para llevar a cabo lo anterior, se utilizó el diagrama de flujo de la Figura 8. En este diagrama, se puede visualizar el proceso que se estableció para controlar un dispositivo IoT. Para la conexión con Beebotte es necesario el token y nombre del canal del usuario. Si la conexión es exitosa se procederá a seleccionar el tipo de dispositivo, buscarlo en la red y configurarlo. Una vez establecida la comunicación con los dispositivos, se inicia el proceso de escucha de mensajes en Beebotte. Si existe una nueva publicación, el mensaje es transformado a un comando entendible por el dispositivo y enviado al mismo.

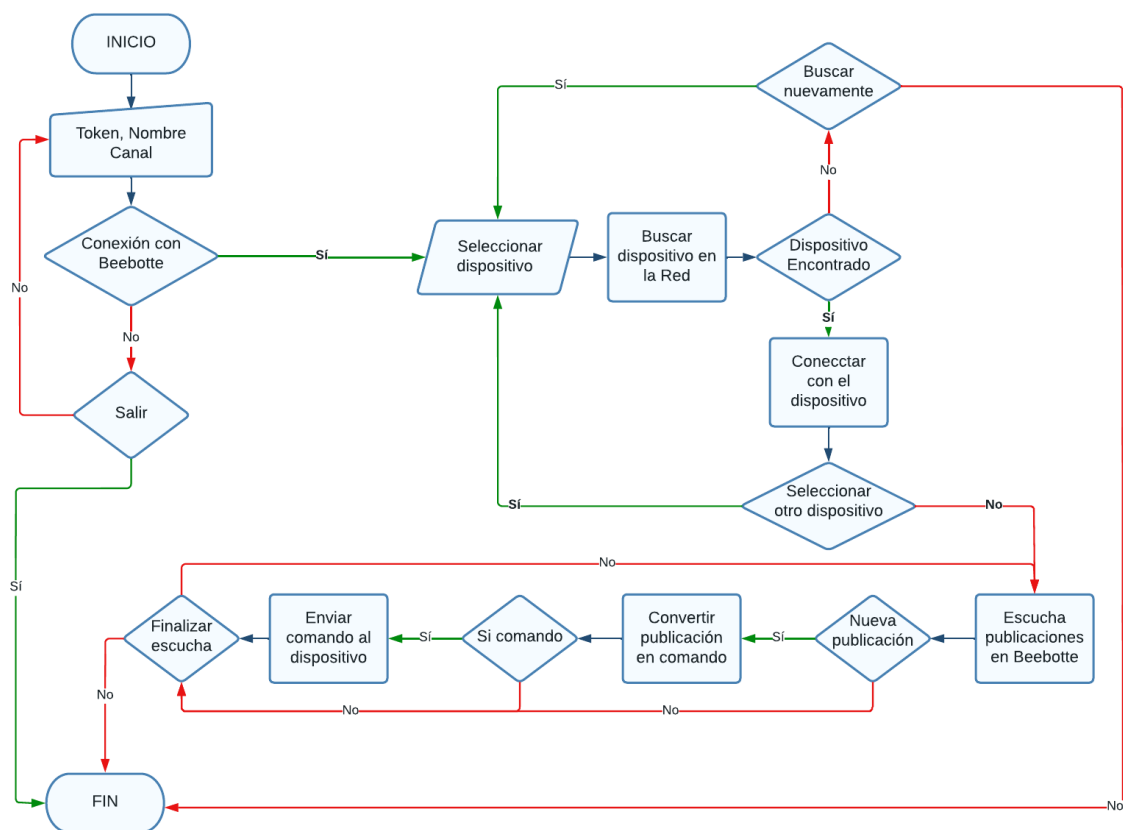


Figura 8. Diagrama de Flujo para la comunicación entre Beebotte y los dispositivos IoT

Desarrollo

Para la comunicación entre Beebotte y el dispositivo IoT, se implementó el diagrama de flujo de la Figura 8 utilizando el lenguaje de programación Python. Además, de las librerías propias de este lenguaje, se utilizaron librerías adicionales que facilitaron la búsqueda y control de los dispositivos, como se detalla a continuación:

- Beebotte: establece la conexión entre Beebotte y el Bróker de comunicación.
- Nmap y Netifaces: estas librerías se utilizaron para buscar los dispositivos en la red.
- Kasa-Python: permite controlar un foco inteligente de la marca TP-Link.
- SamsungTVWS: permite controlar una televisión de la marca Samsung.

Pruebas

Para realizar las pruebas de este incremento, se utilizó una computadora como Bróker de comunicación, sin embargo, para mayor portabilidad se podría utilizar una Raspberry Pi.

Conexión Beebotte – Bróker de comunicación – Dispositivo IoT

La prueba consistió en leer un mensaje en Beebotte desde el Bróker de comunicación y transformar el mensaje leído en un comando entendible por el dispositivo. Luego, se envió este comando al dispositivo y se verificó que este realice la acción solicitada. Esta prueba se realizó varias veces con el foco y la televisión enviando diferentes mensajes. En todos los casos los resultados de las pruebas fueron exitosos.

3.6 Pruebas del Prototipo Integrado

Planificación

Para realizar las pruebas del prototipo completo, se establecieron tres secuencias de pruebas:

- (1) Encender y apagar el foco: para cumplir con esta prueba el usuario debe estar en la Interfaz Foco y con su mirada seleccionar el ícono de encendido y luego el de apagado.
- (2) Encender Tv, cambiar el canal al siguiente, subir el volumen y apagar Tv: en esta prueba el usuario debe estar en la Interfaz Televisión y con su mirada seleccionar el ícono de encendido, luego el ícono de más controles, canal siguiente, subir volumen, volver atrás y finalmente apagar TV.
- (3) Encender foco, encender Tv, bajar volumen, apagar Tv y apagar foco: esta secuencia es la más completa, para ello el usuario debe estar en la interfaz Foco y luego con la mirada seleccionar el ícono de encender foco, volver a inicio, escoger el ícono de televisión, encender Tv, más controles, volumen -, atrás, apagar Tv, volver a inicio, escoger el ícono de foco y finalmente apagar.

Experimentación

Para este proceso, se contó con la colaboración de 17 personas de las cuales ninguna sufría de discapacidad motora o del habla. Este grupo de personas se encontraba entre los 20 y 34 años de edad, 11 personas no usaban lentes y dos personas contaban con experiencia previa en el uso del prototipo. Para iniciar las pruebas, primero se explicó brevemente al usuario en qué consistía el prototipo. Segundo, se procedió a ubicar al usuario en una posición adecuada frente al computador para realizar las pruebas. Tercero, el usuario debía realizar una calibración previa para el uso de la aplicación. Cuarto, el usuario se familiarizaba con el uso de la aplicación, para posteriormente iniciar con las secuencias de prueba establecidas en la planificación. A la par que se realizaron las

pruebas, se registró en una hoja de Excel el tiempo que tardó el usuario en completar una secuencia de prueba, los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Finalmente, cada participante realizó cinco pruebas por cada secuencia.

Resultados

Como se puede visualizar en la Tabla 9, se obtuvo un tiempo promedio de 8.37 segundos para la secuencia 1, de 31.62 segundos para la secuencia dos y de 77.14 segundos para la secuencia 3. Otra medida calculada fue la precisión, la cual fue del 100% para la secuencia 1, 97.94% para la secuencia 2 y de 98.97% para la secuencia 3, obteniendo así que la precisión promedio del prototipo es de 98.97%.

Tabla 9. Resultados de las pruebas del Prototipo final

	Secuencia 1	Secuencia 2	Secuencia 3
Tiempo mínimo [s]	6.26	23.02	59.30
Tiempo máximo [s]	13.08	42.90	114.40
Tiempo promedio [s]	8.37	31.62	77,14
Precisión	100%	97.94%	98.97%

4 ANÁLISIS DE RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

4.1 Análisis de Resultados

En esta sección se realizó el análisis de los resultados de todas las pruebas del prototipo. De las pruebas realizadas en el Incremento 2 se obtuvo que el tiempo óptimo de fijación del prototipo es de 1.5 segundos. Con este tiempo de fijación el tiempo en completar una tarea fue 3.14 segundos y el prototipo alcanzó una precisión del 98.99%, que es la mayor precisión con respecto a las precisiones obtenidas con otros tiempos de fijación. Cabe mencionar que tiempos menores a 1.5 segundos incrementan la tasa de falsos negativos, debido a que el prototipo tiene un tiempo de retraso ± 1 segundo, por lo que estos tiempos fueron descartados.

Con los datos obtenidos en el apartado 3.6, se realizó la matriz de confusión (Ver Tabla 10), la cual permitió calcular la precisión y exactitud del prototipo final. La precisión es el resultado de dividir el número de verdaderos positivos para la suma de los verdaderos positivos y verdaderos negativos. Por otro lado, la exactitud es el resultado de la suma de los verdaderos positivos y verdaderos negativos dividido para el total de casos [56]. Después de realizar estas operaciones se obtuvo una precisión del 98.97% y una exactitud

del 97%. Esta información garantiza que el prototipo final es funcional y que se puede controlar un dispositivo IoT con la mirada.

Tabla 10. Matriz de confusión

	Positivo	Negativo
Verdadero	288	3
Falso	9	0

Algunos participantes que realizaron las pruebas usaban lentes, esto fue muy importante ya que permitió conocer si la librería tenía algún inconveniente para predecir el seguimiento de la mirada mientras usaban los mismos. A continuación, se muestra la Figura 9, donde se puede observar que no existe diferencia entre una persona que usa lentes de una que no lo usa; por lo que este prototipo podría ser utilizado por ambos tipos de personas.

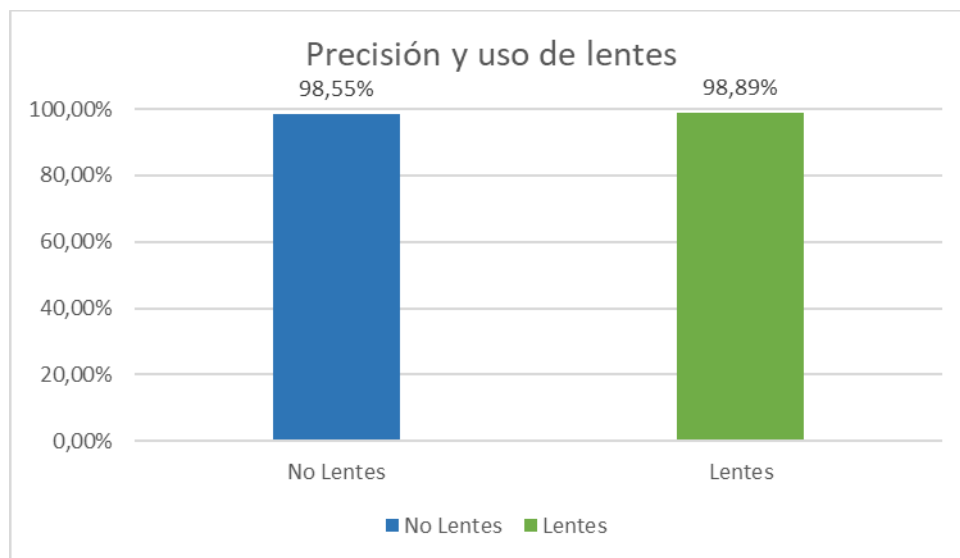


Figura 9. Precisión y uso de lentes

Otro de los resultados observados fue que no se requiere de una capacitación extensa en el uso del prototipo o una experiencia previa. Esto se pudo evidenciar cuando el menor tiempo para completar las secuencias de pruebas fue alcanzado por usuarios no experimentados.

Finalmente, se pudo observar que el uso prolongado de la aplicación causa cansancio visual en el usuario, en especial en aquellos participantes que no usan lentes. No obstante, este efecto puede ser disminuido con el uso de lentes y con el empleo de colores fríos y neutros en las interfaces del prototipo.

4.2 Conclusiones

Se logró crear un prototipo funcional que permitió controlar dispositivos IoT utilizando la mirada. Este prototipo podría ser utilizado por personas con discapacidad motora y del habla, y en general por cualquier persona sin ningún inconveniente. De acuerdo a las pruebas realizadas, se obtuvo una precisión del 98.97%, una exactitud del 97% y tiempo de selección de un ícono de 3.14 segundos. Además, no se requiere experiencia previa en el uso de la aplicación para interactuar con la misma. No obstante, es un proyecto que puede ser mejorado en un futuro, donde la predicción de la mirada sea más precisa y también se pueda controlar otros dispositivos inteligentes.

En la revisión de literatura, se puede ver que la tecnología de seguimiento de miradas ha aportado significativamente al desarrollo de diferentes soluciones tecnológicas de múltiples áreas. Y esto ha sido gracias a la aparición de diferentes dispositivos y librerías para el desarrollo de dichas soluciones.

Varios son los dispositivos de seguimiento de mirada disponibles en el mercado. Dependiendo de la formalidad de la investigación y los recursos, se puede optar desde dispositivos económicos, pero con poca precisión o dispositivos de gran precisión a un costo elevado. Debido a la gran variedad, también es posible obtener dispositivos que traen consigo un algoritmo de seguimiento de mirada, por lo que ya no es necesario utilizar librerías como las revisadas en las secciones 1.5 y 3.1. Estas librerías son indispensables si se requiere realizar análisis adicionales de los datos obtenidos con el dispositivo de seguimiento de mirada. También, son necesarias si el dispositivo no cuenta con un algoritmo interno de procesamiento. Tanto las librerías de código abierto como las de pago ofrecen buenas características, pero no todas son compatibles con todos los dispositivos, en especial con las cámaras web.

Con respecto a las 5 librerías probadas, se debe analizar los requisitos del proyecto en los que se va a utilizar este software, es decir, si solo se requiere conocer la posición de la pupila, se deberá utilizar una librería para ese fin. Pero si se necesita realizar seguimiento de la mirada, se debe elegir la librería que realice esa actividad. Una ventaja importante de estas librerías es la compatibilidad que tienen con cámaras web, por lo que, si los recursos del proyecto son bajos, estas librerías son una buena opción.

4.3 Recomendaciones

Si se requiere mejorar la precisión o mejorar el tiempo de selección de un ícono, se recomienda trabajar en mejorar el modelo de calibración o buscar otros métodos de selección como, por ejemplo: el parpadeo de ojos.

Debido al uso prolongado con la aplicación, se sugiere diseñar las interfaces con colores fríos y neutros. Esto con el fin de evitar que el usuario sufra cansancio visual. Además, el usuario podría hacer uso de lentes, ya que el prototipo es compatible con el uso de estos accesorios.

Cabe indicar que este prototipo es compatible únicamente con focos inteligentes TP-Link y con televisores inteligentes de la marca Samsung con sistema operativo Tizen. Por lo que se recomienda utilizar este tipo de marcas para su correcto funcionamiento.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] F. Klaib, O. Alsrehin, Y. Melhem y O. Bashtawi, «IoT Smart Home Using Eye Tracking and Voice Interfaces for Elderly and Special Needs People,» *Journal of Communications*, vol. 14, nº 7, p. 8, 2019.
- [2] J. Barriga, J. Sulca, J. L. León, A. Ulloa, D. Portero, R. Andrade y S. G. Yoo, «Smart Parking: A Literature Review from the Technological Perspective,» *MDPI*, p. 34, 2019.
- [3] W. Zhaoa y L. Yi, «Research on the evolution of the innovation ecosystem of the Internet of Things: A case study of Xiaomi(China),» *ScienceDirect*, pp. 56-62, 2022.
- [4] M. Humayuna, N. Jhanjhi, A. Alsayat y V. Ponnusamy, «Internet of things and ransomware: Evolution, mitigation and prevention,» *ScienceDirect*, vol. 22, pp. 105-117, 2021.
- [5] G. Siddesh, S. Manjunath y K. Srinivasa, «Application for assisting mobility for the visually impaired using IoT infrastructure,» *IEEE Xplore*, 2017.
- [6] A. Alanwar, M. Alzantot, B. Ho, P. Martin y M. Srivastava, «SeleCon: Scalable IoT Device Selection and Control Using Hand Gesture,» de *Internet-of-Things Design and Implementation (IoTDI 2017)*, Pennsylvania, 2017.
- [7] T. B. Tedla, M. N. Davydkin y A. M. Nafikov, «Development of an Internet of Things based Electrical Load Management System,» *Journal of Physics: Conference Series*, vol. 1886, nº 1, 2021.
- [8] I. Kirsh y H. Ruser, «Phone-Pointing Remote App: Using Smartphones as Pointers in Gesture-Based IoT Remote Controls,» *HCI International 2021 - Posters*, pp. 14-21, 2021.
- [9] W. H. Organization, «Spinal cord injury,» 19 November 2013. [En línea]. Available: <https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury>. [Último acceso: 22 February 2022].
- [10] A. N. Today, «How Common Is ALS?,» 18 September 2019. [En línea]. Available: <https://alsnewstoday.com/how-common-is-als/>. [Último acceso: 22 February 2022].
- [11] C. Metz, M. Jaster, E. Walch, A. Sarpong-Bengelsdorf, A. Kaindl y J. Schneider, «Clinical Phenotype of Cerebral Palsy Depends on the Cause: Is It Really Cerebral Palsy? A Retrospective Study,» *Journal of Child Neurology*, vol. 37, nº 2, pp. 112-118, 2021.
- [12] A. Randolph, S. Petter, V. Storey y M. Jackson, «Context-aware user profiles to improve media synchronicity for individuals with severe motor disabilities,» *Wiley*, pp. 130-163, 2021.
- [13] E. M. A. El-Kafy, M. A. Alshehri, A. A. R. El-Fiky y M. A. Guerhazi, «The Effect of Virtual Reality-Based Therapy on Improving Upper Limb Functions in Individuals

With Stroke: A Randomized Control Trial,» *Frontiers in Aging Neuroscience*, vol. 13, pp. 1-8, 2021.

- [14] Ö. Yayıcı, Ö. Kucur, C. Taşkıran, Ü. Öztoprak, Ç. Genç, E. Aksoy, A. Aksoy, T. Yoldaş y D. Yüksel, «Clinical Features and Quality of Life in Duchenne and,» *The Journal of Current Pediatrics*, nº 19, pp. 15-22, 2021.
- [15] N. Tarek, M. A. Mandour, N. El-Madah, R. Ali, S. Yahia, B. Mohamed, D. Mostafa y S. El-Metwally, «Morse glasses: an IoT communication system based on Morse code for users with speech impairments,» *Springer*, 2021.
- [16] A. F.Klaib, N. O.Alsrehin, W. Y.Melhem, H. O.Bashtawi y A. A.Magableh, «Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies,» *ScienceDirect*, vol. 166, 2021.
- [17] A. Papoutsaki, N. Daskalova, P. Sangkloy y J. Huang, «WebGazer: Scalable webcam eye tracking using user interactions,» *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pp. 3839-3845, 2016.
- [18] R. Veerati, E. Suresh, A. Chakilam y S. P. Ravula, «Eye Monitoring Based Motion Controlled Wheelchair for Quadriplegics,» *Springer*, pp. 41-49, 2018.
- [19] B. H. Malini, R. Supritha, N. V. Prasad, Vandana y Y. R., «Eye And Voice Controlled Wheel Chair,» *IEEE Xplore*, 2020.
- [20] M. Carroll, M. Ruble, M. Dranias, S. Rebensky, M. Chaparro, J. Chiang y B. Winslow, «Automatic Detection of Learner Engagement Using Machine Learning and Wearable Sensors,» *Scientific Research*, p. 14, 2020.
- [21] K. Wankhede y S. Pednekar, «Aid for ALS Patient Using ALS Specs and IOT,» *IEEE Xplore*, 2019.
- [22] S. Pai y A. Bhardwaj, «Eye Gesture Based Communication for People with Motor Disabilities in Developing Nations,» *IEEE Xplore*, 2019.
- [23] P. Dondi, M. Porta, A. Donvito y G. Volpe, «A gaze-based interactive system to explore artwork imagery,» *Journal on Multimodal User Interfaces*, vol. 16, nº 1, pp. 55-67, 2022.
- [24] Z. Su, X. Zhang, N. Kimura y J. Rekimoto, «Gaze+Lip: Rapid, Precise and Expressive Interactions Combining Gaze Input and Silent Speech Commands for Hands-free Smart TV Control,» *Eye Tracking Research and Applications Symposium (ETRA)*, vol. PartF169257, 2021.
- [25] E. Arias Soto, G. López Herrera, L. Quesada Quirós y L. A. Guerrero Blanco, «Web accessibility for people with reduced mobility: A case study using eye tracking,» *Springer*, 2016.

- [26] A. S. Abdallah, L. J. Elliott y D. Donley, «Toward Smart Internet of Things (IoT) Devices: Exploring the Regions of Interest for Recognition of Facial Expressions using Eye-gaze Tracking,» *IEEE Xplore*, 2020.
- [27] A. Bissoli, D. Lavino-Junior, M. Sime, L. Encarnaçã y a. T. Bastos, «A Human–Machine Interface Based on Eye Tracking for Controlling and Monitoring a Smart Home Using the Internet of Things,» *MDPI*, 2019.
- [28] B. Brousseau, J. Rose y M. Eizenman, «SmartEye: An Accurate Infrared Eye Tracking System for Smartphones,» *2018 9th IEEE Annual Ubiquitous Computing*, p. 951–959, 2018.
- [29] A. Brunete, E. Gambao, M. & Hernando y R. Cedazo, «Smart Assistive Architecture for the Integration of IoT Devices, Robotic Systems, and Multimodal Interfaces in Healthcare Environments,» *Sensors*, vol. 21, nº 6, pp. 1-25, 2021.
- [30] S. Chandra, G. Sharma, S. Malhotra, D. Jha y A. P. Mittal, «Eye tracking based human computer interaction: Applications and their uses,» *Proceedings - 2015 International Conference on Man and Machine Interfacing*, 2016.
- [31] S. Heravian, N. Nouri, M. Behnam Taghadosi y S. M. H. Seyedkashi, «Implementation of Eye Tracking in an IoT-Based Smart Home for Spinal Cord Injury Patients,» *In Modern Care Journal*, vol. 16, nº 4, 2019.
- [32] M. T. Chew y K. Penver, «Low-cost eye gesture communication system for people with motor disabilities,» *I2MTC 2019 - 2019 IEEE International Instrumentation and Measurement Technology Conference*, pp. 1-5, 2019.
- [33] T. Robal, «Spontaneous Webcam Instance for User Attention Tracking,» *Technology Management in the World of Intelligent Systems*, 2019.
- [34] L. Xia, B. Sheng, W. Wu, L. Ma y P. Li, «Accurate gaze tracking from single camera using gabor corner detector,» *Multimedia Tools and Applications*, vol. 71, nº 1, p. 221–239, 2016.
- [35] P. Salunkhe y A. R. Patil, «A device controlled using eye movement,» *International Conference on Electrical, Electronics, and Optimization Techniques*, p. 732–735, 2016.
- [36] R. G. Lupu, R. G. Bozomitu, A. Păsărică y C. Rotariu, «Eye tracking user interface for Internet access used in assistive technology,» de *2017 E-Health and Bioengineering Conference*, Sinaia, Romania, 2017.
- [37] Y. Tamura y K. Takemura, «Estimating point-of-gaze using smooth pursuit eye movements without implicit and explicit user-calibration,» *Eye Tracking Research and Applications Symposium (ETRA)*, pp. 1-4, 2020.
- [38] M. A. Mento, «Different Kinds of Eye Tracking Devices,» Bitbrain, 12 June 2020. [En línea]. Available: <https://www.bitbrain.com/blog/eye-tracking-devices>. [Último acceso: 28 March 2022].

- [39] W. Wu et al, «Faster Single Model Vigilance Detection Based on Deep Learning,» *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, n° 3, pp. 621-630, 2021.
- [40] Z. Gomolka, B. Twarog, E. Zeslawska y D. Kordos, «Registration and Analysis of a Pilot's Attention Using a Mobile Eyetracking System,» *Advances in Intelligent Systems and Computing*, vol. 987, pp. 215-224, 2020.
- [41] K. Siivonen, J. Sainio, M. Viitanen, J. Vanne y T. D. Hamalainen, «Open framework for error-compensated gaze data collection with eye tracking glasses,» *Journal of Signal Processing Systems*, vol. 93, pp. 1457, 2021, 2018.
- [42] A. Papoutsaki, A. Gokaslan, I. D. Smet, J. Tompkin y J. Huang, «Webgazer.js,» [En línea]. Available: <https://webgazer.cs.brown.edu/>. [Último acceso: 25 03 2022].
- [43] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal y B. Sikdar, «V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar,» *IEEE Access*, vol. 7, pp. 82721-82743, 2019.
- [44] K. Morita, K. Miura, M. Fujimoto y H. Yamamori, «Eye movement abnormalities and their association with cognitive,» *ScienceDirect*, p. 7, 2018.
- [45] CHING y WINNIE, «DETECTION OF DELIRIUM THROUGH EYE-TRACKING METHODS,» *ProQuest*, p. 11, 2021.
- [46] J. Acharjee y S. Deb, «Identification of Significant Eye Blink for Tangible Human Computer Interaction,» *Scopus*, p. 5, 2021.
- [47] S. Baee, E. Pakdamanian, V. Ordonez, I. Kim y L. Feng, «EyeCar: Modeling the Visual Attention Allocation of Drivers in Semi-Autonomous Vehicles,» *arxiv*, 2019.
- [48] M. Aqel, A. Alashqar y A. Badra, «Smart home automation system based on eye tracking for quadriplegic users,» *Scopus*, 2020.
- [49] C. M. Turgeon, T. Saiyera, C. S. L. G. Esparza, R. MacDonald y V. Kokhan, «Eye Tracking the Feedback Assigned to Undergraduate Students in a Digital Assessment Game,» *Scopus*, p. 13, 2019.
- [50] J. Drummond y M. Themessl-Huber, «The cyclical process of action research,» *Action Res*, vol. 5, pp. 430-448, 2007.
- [51] S. Chauhan, N. Agarwal y A. Kar, «Addressing big data challenges in smart cities,» *A systematic literature review*, vol. 18, pp. 73-90, 2016.
- [52] C. de Moraes, D. Sadok y J. Kelner, «An IoT sensor and scenario survey for data researchers,» *Journal of the Brazilian Computer Society*, vol. 25, p. 4, 2019.
- [53] M. SM y S. CB, «A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review,» *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009.

- [54] O. M. d. I. Salud, «Organización Mundial de la Salud,» 24 11 2021. [En línea]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/disability-and-health>. [Último acceso: 04 08 2022].
- [55] Colorlib, «colorlib,» 02 2022. [En línea]. Available: <https://colorlib.com/>. [Último acceso: 01 06 2022].
- [56] F. Izco, «bookdown,» 27 11 2018. [En línea]. Available: https://bookdown.org/f_izco/BDC-POC/metricas.html. [Último acceso: 04 08 2022].
- [57] L. L. Russell, C. V. Greaves, R. S. Convery, J. Nicholas, J. D. Warren, D. Kaski y J. D. Rohrer, «Novel instructionless eye tracking tasks identify emotion recognition deficits in frontotemporal dementia,» *Springer*, p. 11, 2021.
- [58] T. C. M, T. Saiyera, L. G. E. C. S, R. MacDonald y V. Kokhan, «Eye Tracking the Feedback Assigned to Undergraduate Students in a Digital Assessment Game,» *Scopus*, p. 13, 2019.
- [59] L. Bender, I. Guerra, G. Ito, I. Vizcarra y Alejandro, «Mirada, tiempo y acción: visualizaciones alternativas de experimentos de seguimiento ocular con escenas dinámicas,» p. 19, 2017.
- [60] M. A. H. Ameen, C. M. Aldridge, Y. Zhuang y X. Yin, «Investigating the Need for Calibration to Track Eye,» *Research Square*, p. 14, 2021.
- [61] M. Artemi y H. Liu, «A user study on user attention for an interactive content-based image search system,» *Scopus*, 2021.
- [62] R. Florent, G. Cavanaugh y C. LLerena, «Measuring the Engagement of Children with Autism Spectrum Disorder Using Eye-Tracking Data,» *NSUWorks*, 2020.
- [63] M. R. Saleema, A. Straus y R. Napolitano, «Interpretation of Historic Structure for Non-Invasive Assessment Using Eye Tracking,» *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 46, p. 8, 2021.
- [64] M. L. Mele y D. Millar¹, «Explicit and Implicit Measures in Video Quality Assessment,» *SCITEPRESS*, p. 12, 2019.
- [65] P. Contero, C. Torrecilla, C. Escribá y M. Contero, «Understanding Fashion Brand Awareness Using Eye-Tracking: The Mix-and-Match Approach,» *Lecture Notes in Networks and Systems*, vol. 276, pp. 432-440, 2021.
- [66] P. Sulikowski y T. Zdziebko, «Deep learning-enhanced framework for performance evaluation of a recommending interface with varied recommendation position and intensity based on eye-tracking equipment data processing,» *Electronics (Switzerland)*, vol. 9, nº 2, p. 266, 2020.
- [67] J. Katona et al, «Recording Eye-Tracking Parameters during a Program Source-code Debugging Example,» *10th IEEE International Conference on Cognitive Infocommunications*, pp. 335-338, 2019.

- [68] R. S. Hessels, M. Nystrom y I. T. Hooge, «Do pupil-based binocular video eye trackers reliably measure vergence?,» *ScienceDirect*, p. 9, 2019.
- [69] C. Costescu, A. Rosan, N. Brigitta, A. Hathazi, A. Kovari, J. Katona, R. Demeter, I. Heldal, C. Helgesen, S. Thill y I. Efrem, «Assessing Visual Attention in Children Using GP3 Eye Tracker,» *10th IEEE International Conference on Cognitive Infocommunications*, pp. 343-348, 2019.
- [70] J. Iskander, I. Hettiarachchi, S. Hanoun, M. Hossny, S. Nahavandi y A. Bhatti, «A classifier approach to multi-screen switching based on low cost eye-trackers,» *2018 Annual IEEE International Systems Conference*, pp. 1-6, 2018.
- [71] P. Mannaru, B. Balasingam, K. Pattipati, C. Sibley y J. Coyne, «Performance evaluation of the gaze point GP3 eye tracking device based on pupil dilation,» *Lecture Notes in Computer Science*, vol. 10284, pp. 166-175, 2017.
- [72] U. Saisara, P. Boonbrahm y A. Chaiwiriya, «Strabismus screening by Eye Tracker and games,» *2017 14th International Joint Conference on Computer Science and Software Engineering*, pp. 1-5, 2017.
- [73] A. Kovari, «Analysis of Gaze Fixations Using an Open-source Software,» *10th IEEE International Conference on Cognitive Infocommunications*, pp. 325-328, 2019.
- [74] D. Yadav, N. Kohli, E. Kalsi, M. Vatsa, R. Singh y A. Noore, «Unraveling Human Perception of Facial Aging Using Eye Gaze,» *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2221-22217, 2018.
- [75] S. Seha, G. Papangelakis, D. Hatzinakos, A. S. Zandi y F. J. Comeau, «Improving Eye Movement Biometrics Using Remote Registration of Eye Blinking Patterns,» *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2562-2566, 2019.
- [76] J. Thampan, F. Mohammed, M. Tijin, P. Prabhu y K. M. Rince, «Eye Based Tracking and Control System,» *International Journal of Innovative Science and Modern Engineering (IJISME)*, vol. 4, n^o 10, pp. 13-17, 2017.
- [77] I. Mohanraj y S. Siddharth, «A framework for tracking system aiding disabilities,» *2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, pp. 1-7, 2017.
- [78] Höffner y Sebastian, *Gaze Tracking Using Common Webcams*, 2018, p. 89.
- [79] Liu y Yuanzheng, «Real-time Pupil Localization Using 3D Camera,» *escholarship*, p. 19, 2016.
- [80] C. Huan, M. Osman y J. Ying, «An innovation-driven approach to specific language impairment diagnosis,» *Scopus*, 2021.
- [81] P. Alexandra, G. Aaron, J. Tompkin, Y. He y J. Huang, «The Eye of the Typer: A Benchmark and Analysis of Gaze Behavior during Typing,» *ACM*, p. 9, 2018.

- [82] A. Papoutsaki, J. Laskey y J. Huang, «SearchGazer: Webcam Eye Tracking for Remote Studies of Web Search,» *ACM*, p. 10, 2017.
- [83] K. Kathpal, S. Negi y S. Sharma, «IChat: Interactive Eyes for Specially Challenged People Using OpenCV Python,» *Scopus*, 2021.
- [84] M. Hashmi, N. Priya, S. Reddy, G. Vakula y D. Usha, «Drowsiness Detection System Using Raspberry Pi and OpenCV,» *Scopus*, 2021.
- [85] K. K. y. K. Son, «Eyeball tracking and Object Detection in Smart Glasses,» *Scopus*, p. 3, 2020.
- [86] E. Shlyamova, K. Ezhova y D. Fedorenko, «The capabilities of developing eye tracking for AR systems on the base of a microcontroller Raspberry Pi,» *Scopus*, 2020.
- [87] A. Kwiatkowska y D. Sawicki, «Eye tracking as a method of controlling applications on mobile devices,» *Scopus*, 2018.
- [88] N. Wanluk, S. Visitsattapongse, A. Juhong y C. Pintavirooj, «Smart wheelchair based on eye tracking,» *Scopus*, p. 4, 2017.
- [89] T. Samuel, P. Jami, W. R. M y L. Otto, «Visual anticipation of the future path: Predictive gaze and steering,» *ArvoJournals*, p. 23, 2021.
- [90] T. S. MS, T. Wyss y R. Gilgen-Ammann, «Eyeblink Detection in the Field: A Proof of Concept,» *Research Gate*, p. 7, 2021.
- [91] A. Velisar y N. M. Shnidze, «Noise in the Machine: Sources of Physical and Computation,» *ResearchGate*, p. 4, 2021.
- [92] P. Hausamann, C. Sinnott y P. R. MacNeilage, «Positional head-eye tracking outside the lab: an open-source solution,» *ACM*, p. 14, 2020.
- [93] A. Caspi, M. P. Barry, U. K. Patel y M. A. Salas, «Eye movements and the perceived location of phosphenes generated by intracranial primary visual cortex stimulation in the blind,» *Science Direct*, p. 10, 2021.
- [94] U. Patayona, J. M. Gallegos, P. Mackb, R. Bacabis y Celesamae, «Signaling and Pacing: A Comparative Study on Evidence Based Stimuli using an Eye Tracking Device,» *ScienceDirect*, p. 8, 2021.
- [95] L. Guimaraes, G. Schirlo, G. Gasparello, S. Bastos, M. Pithon y O. Tanaka, «Visual facial perception of postsurgical cleft lip scarring assessed by laypeople via eye-tracking,» *Scopus*, 2021.
- [96] A. Al-Lahham, P. Souza, C. Miyoshi, S. Ignácio, T. Meira y O. Tanaka, «An eye-tracking and visual analogue scale attractiveness evaluation of black space between the maxillary central incisors,» *Scopus*, 2021.

- [97] B. Banire, D. A. Thani, M. Qaraqe, B. Mansoor y M. Makki, «Impact of mainstream classroom setting on attention of children with autism spectrum disorder: an eye-tracking study,» *Springer*, 2020.
- [98] R. Pierdicca, M. Paolanti, R. Quattrini, M. Mameli y E. Frontoni, «A Visual Attentive Model for Discovering Patterns in Eye-Tracking Data—A Proposal in Cultural Heritage,» *pubmed*, 2020.
- [99] Khan y Y. Sendi, «A new approach towards evaluating the performance of maritime officers by the utilization of mobile eye tracking system and facial electromyography,» 2021.
- [100] B. A. Zardari, Z. Hussain, A. A. Arain, W. H. Rizvi y M. S. Vighio, «QUEST e-learning portal: applying heuristic evaluation, usability testing and eye tracking,» *Springer*, 2018.
- [101] N. Du, X. J. Yang y F. Zhou, «Psychophysiological responses to takeover requests in conditionally automated driving,» p. 35, 2020.
- [102] R. Burdzik, I. Celiński y J. Młyńczak, «Study of the Microsleep in Public Transport Drivers,» *Advances in Intelligent Systems and Computing*, vol. 1032, pp. 63-73, 2020.
- [103] S. Popelka, J. Dolezalova y M. Beitlova, «New features of scangraph - A tool for revealing participants' strategy from eye-movement data,» *Eye Tracking Research and Applications Symposium*, nº 103, pp. 1-2, 2018.
- [104] A. Sergio, Navarro-Tuch, J. Gammack, D. Kang y S.-G. Kim, «Axiomatic design of a man-machine interface for Alzheimer's patient care,» *Conference Series: Materials Science and Engineering*, vol. 1174, nº 1, 2021.
- [105] K. J. Li, A. R. Chowdhury, K. Fawaz y Younghyun, «Real-Time Privacy Control for Eye-Tracking Systems,» *USENIX Association*, pp. 1793--1810, 2021.
- [106] M. Hanke, S. Mathôt, E. Ort, J. Stadler y A. Wagner, «A practical guide to functional magnetic resonance imaging with simultaneous eye tracking for cognitive neuroimaging research,» *Neuroinformatics*, vol. 151, pp. 291-305, 2020.
- [107] S. Karthick, K. Madhav y K. Jayavidhi, «A Comparative Study of Different Eye Tracking System,» *AIP Conference Proceedings 2112*, 2019.

6 ANEXOS

6.1 Anexo I. Dispositivos de seguimiento de mirada

Tabla 11. Dispositivos de seguimiento de mirada.

Dispositivo	Artículos de referencia	Rango de pantalla [pulgadas]	Exactitud [grados]	Frecuencia de muestreo [Hz]	Distancia dispositivo - usuario [cm]	Tipo de Dispositivo	Libertad de movimiento de la cabeza [grados] H: Horizontal V: Vertical	Número de artículos de referencia	Precio [USD]
EyeLink 1000 Plus	[41] [44] [57] [58] [59]	24"	0.5	2000	40 -70	Estabilizado en la cabeza	-	5	\$ 10000+
Tobii Pro Nano	[60] [61] [62]	24"	0.3	60, 120, 250	50 -80	Basado en pantalla	23.63-34.99 H 20.55-30.96 V	3	\$10000+
Tobii Pro Fusion	[38] [45]	19"	0.3	120	45 -85	Basado en pantalla	29.36-31.59 H 20.55-21.01 V	2	\$ 13900
Smart Eye AI-X	-	24"	0.5	60	45 -85	Basado en pantalla	-	0	\$ 3450
EyeTech VT3 Mini	[19] [63]	22"	0.5	200	50 -70	Basado en pantalla	16.07-21.97 H 8.53-11.86 V	2	\$ 4000
Eyegaze Edge Encore Camera	-	22"	0.45	50	40 - 73	Basado en pantalla	13.66-23.98 H 9.26-16.57 V	0	\$ 7500
Eyegaze Edge Prime Camera	-	19"	0.45	60	43 - 83	Basado en pantalla	5.23-10.02 H 4.40-8.46 V	0	\$ 7500
GazePoint 3 (GP3)	[64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75]	24" Resolución: 1366 x 768	0.5	60	50 - 80	Basado en pantalla	17.01-26.10 H 7.83-12.40 V	11	\$ 845

SMI Eye Tracking Glasses	[39]	-	0.5	120	-	Montado en la cabeza	80 H 60 V	1	\$ 11900
Tobii Pro Glasses 3	[40]	-	0.6	100	-	Montado en la cabeza	-	1	\$ 10000+
Webcam	[18] [19] [27] [29] [32] [33] [34] [35] [76] [77]	23"	-	60	28.5 – 60	Basado en pantalla	22-39 H	10	\$ 15+

6.2 Anexo II. Librerías de seguimiento de mirada

Tabla 12. Librerías de seguimiento de mirada.

Librería	Grado de compatibilidad	Artículos de referencia	Tipo de Licencia	Complejidad de instalación	Lenguajes utilizados	Compatibilidad con Sistema Operativo	Soporte	Última actualización
Eye Like	Bajo	[78] [79]	Open Source	Alto	C++, C	Alto	No	2019-03-31
WebGazer	Alto	[17] [80] [81] [82]	Open Source	Medio	JavaScript	Alto	Si	202-11-16
OpenCV	Alto	[46] [83] [84] [85] [48] [86] [87] [88]	Open Source	Medio	C++, C, Python, Java	Alto	Si	2022-02-03
Turker Gaze	Medio	None	Open Source	Bajo	JavaScript	Alto	No	2016-04-06
Gaze Trackin	Alto	None	Open Source	Bajo	Python	Medio	Si	2022-01-19
Pupil Core	Medio	[89] [90] [91] [92] [93]	Open Source	Alto	Python, C++	Alto	Si	2022-01-31

OGAMA	Medio	[65] [69] [94] [95] [96]	Open Source	Alto	C#, C++	Bajo	Si	2021-03-30
Gaze-Detection	Bajo	None	Open Source	Medio	JavaScript	Alto	No	2021-06-24
iMotions	Medio	[97] [98] [99] [100] [101] [47]	Comercial	Bajo	No especificado	No especificado	Si	No especificado
OpenGazer	Medio	None	Open Source	Alto	C++, Python	Medio	No	2010-11-16
Tobii Pro-SDK	Bajo	None	Freeware	Bajo	Python, Matlab y .NET.	Alto	Si	2021-09-15
Tobii Pro Glasses 3 API	Bajo	None	Freeware	Medio	Python, Matlab y .NET.	Alto	Si	2020
SMI BeGaze	Medio	[102] [103]	Comercial	Bajo	Proprietary language	Bajo	Si	2020
SmartGaze	Bajo	None	Comercial	Bajo	N/S	N/S	Si	N/S
Eyeware-GazeSense	Alto	None	Comercial	Bajo	Just user interface	Medio	Si	N/S
openEyes	Medio	None	Open Source	Medio	Matlab	Alto	No	N/S
GazeParse	Alto	[46]	Open Source	Alto	Python, C++	Alto	No	2021-03-22
GazePointer	Medio	[104] [105]	Freeware	Bajo	N/S	Bajo	No	2017
GazePoint	Alto	None	Freeware	Alto	Proprietary commands, HTML	Alto	No	2018
PyGaze (Webcam Eye Tracker)	Medio	[106] [107]	Open Source	Bajo	Python	Alto	No	2016

6.3 Anexo III. Requerimientos del prototipo

<https://drive.google.com/file/d/139yVoMrUeruKTdWJBJALVthMthSkKvkm/view?usp=sharing>

6.4 Anexo IV. Mockups de las interfaces de usuario

<https://drive.google.com/file/d/1byBP3s7osYIPiChZKdiYRAHqhoTe6lZi/view?usp=sharing>