
Centinela: Sistema de ayuda para patologías visuales derivadas de la fatiga

Por
David Pacios Izquierdo



UNIVERSIDAD COMPLUTENSE
MADRID

Grado en Ingeniería Informática
FACULTAD DE INFORMÁTICA

Dirigido por
José Luis Vázquez Poletti

**Sentinel: Help system for visual pathologies
derived from eye fatigue**

MADRID, 2019–2020

Centinela

Sistema de ayuda para patologías visuales derivadas de la fatiga

Memoria que se presenta para el Trabajo de Fin de Grado

David Pacios Izquierdo

Dirigido por

José Luis Vázquez Poletti

Departamento de Arquitectura de Computadores y Automática
Facultad de Informática
Universidad Complutense de Madrid

Madrid, 2019

para Sara

Agradecimientos

En primer lugar a mi mujer, Sara, por toda la ayuda y tiempo que ha invertido en mí durante estos 10 años de convivencia. Me has dado fortaleza en momentos más oscuros donde otros han fallado.

A mis padres, que también han tenido que soportar y cuidar a una persona enferma durante todo el proceso de aprobar la carrera. Tal y como decía mi padre, estudiar es la mejor herencia que te podemos dejar y he intentado aprovecharlo con toda la luz de mis ojos.

Un gran y fuerte agradecimiento a mi tutor de TFG, José Luis Vázquez-Poletti, al cual conocí en el 2016 y durante estos 3 años me ha tenido que aguantar con todas las locuras que se me han ocurrido. Es otro de los motivos por el cuál he continuado la carrera.

A toda la asociación ASCII, que contra viento y marea lucha para demostrar todo lo que vale. A todos sus socios, los cuales, me han abierto su corazón muchísimas veces y han demostrado ser los mejores socios del mundo.

A la junta de ASCII que he conocido desde que me metí en responsabilidad. Siempre tendré a esa secretaria y a ese presidente en mi mente con buenos recuerdos.

A mi núcleo de amigos de toda la vida que forjé juntando locuras de universidad con locuras del día a día, Barullo, Judith, Dani, Selo, Carlos, Víctor, Cesar incluso a Roberto, que jamás será médico pero que ha conseguido conducir antes de que me sacara la carrera sorprendiendo a todos los que le conocían.

A ella, Blanca, que me alegró el año, a su hermana, Paloma y a sus padres que me dieron más de lo que creen.

A Estefanía y Lauri, vinisteis en el mejor momento, para demostrar que podía ser un buen año.

A Cristina, a Esther, por ser un gran aliciente para que me interese por dar clase.

A Nieves, por todos esos lunes de familia que guardaré con cariño y que me animaban a comenzar la semana.

A Andrea, la lingüista. La incorporación más reciente a la familia, por todo ese invierno de Pokémon y locuras de juegos, que nunca termine.

Muchas gracias a todos.

Abstract

This work aims to finally detect the incomplete flicker and relate it to pathologies. Initially it was thought to use several sensors in the same frame, but this idea was dismissed as it was too heavy and unreliable. Next, it was thought to use an infrared sensor to take advantage of the Doppler effect of wave interference to better determine flicker. To do this, it was chosen to reduce the frame to only detect the eyelids well and put an infrared sensor on the bottom of it. With this project you want to prediagnose the following pathologies:

- Allergic conjunctivitis.
- Dry eye.
- Pathologies linked to the ocular fatigue.

To reach the final project they went through four phases. The first phase was the creation of the mount, since, as stated above it was not granted as in the final phase. Next, it was thought like saving the data. First it was thought to perform offline, but that idea was dismissed to be exposed the data. So it was decided to save to an AWS or ThingSpeak-like cloud, because you could update the data and report in real time.

Finally, it is planned to apply this project in the pre-diagnosis of medicine and optometry.

Key words: Doppler, interference, infrared, pre-diagnosis, Centinel , dry eye, conjunctivitis, AWS, ThingSpeak, optometry.

Resumen

Este trabajo tiene como objetivo final conseguir detectar el parpadeo incompleto y relacionarlo con patologías. En un principio se pensó en utilizar varios sensores en una misma montura, pero esta idea se descartó al ser demasiado pesada y poco fiable. Seguidamente, se pensó en utilizar un sensor infrarrojo para aprovechar el efecto Doppler de interferencia de ondas para determinar mejor el parpadeo. Para ello, se optó por reducir la montura para sólo detectar bien los párpados y poner un sensor infrarrojo en la parte inferior de la misma. Con este proyecto se quiere prediagnosticar las siguientes patologías:

- Conjuntivitis alérgica.
- Ojo seco.
- Patologías relacionadas con la fatiga ocular.

Para llegar al proyecto final se pasaron por cuatro fases. La primera fase fue la creación de la montura, ya que, como dicho anteriormente no se concedió como en la fase final. Seguidamente, se pensó como realizar el guardado de los datos. Primero se pensó realizar de manera offline, pero esa idea fue desechada a quedar expuestos los datos. Por lo que se decidió guardar en una nube tipo AWS o ThingSpeak, ya que, se podría actualizar los datos y dar un informe en tiempo real.

Finalmente, se tiene pensado aplicar este proyecto en el prediagnóstico de la medicina y de la optometría.

Palabras clave: Doppler, interferencia, infrarrojo, prediagnóstico, Centinela, ojo seco, conjuntivitis, AWS, ThingSpeak, optometría.

Sobre TEF_LON^NX

TEFLON X(CC0 1.0(DOCUMENTACIÓN) MIT(CÓDIGO))ES UNA PLANTILLA DE L^AT_EX CREADA POR DAVID PACIOS IZQUIERDO CON FECHA DE ENERO DE 2018. CON ATRIBUCIONES DE USO CC0.

Esta plantilla fue desarrollada para facilitar la creación de documentación profesional para Trabajos de Fin de Grado, Trabajos de Fin de Máster o Doctorados. La versión usada es la X

V:X OVERLEAF V2 WITH XE_LA_TE_X, MARGIN 1IN, BIB

Contacto

Autor: DAVID PACIOS IZQUIERO

Correo: dpacios@ucm.es

ASCII: ascii@ucm.es

DESPACHO 110 - FACULTAD DE INFORMÁTICA

Índice general

	Página
A. Introducción y conceptos básicos	1
1. Introduction	1
1.1. Motivation	1
1.2. Goal to be realized	2
1.3. About documentation and style	2
1. Introducción	5
1.1. Motivación	5
1.2. Objetivo a realizar	6
1.3. Sobre la documentación y estilo	6
2. Estado del arte	9
2.1. Aplicaciones y proyectos similares	9
2.1.1. OpenFace	9
2.1.2. Facefirst	10
2.1.3. herta	10
2.1.4. Snapchat	10
2.1.5. WebGazer	11
2.1.6. “Blink to Talk” usando Charlieplexing	11
2.1.7. Sistema de detección de parpadeo mediante unas gafas smart	14
2.1.8. Sensor de parpadeo y Arduino	16
2.1.9. Método eficiente de detección de parpadeo para ayudar a personas discapacitadas	16
2.1.10. Captura de imágenes usando la detección de parpadeo	19
2.1.11. Detección del parpadeo basado en las señales oculares y el filtro Savitzky-Golay	23
2.1.12. Detección del parpadeo para interacción persona-ordenador	28
2.1.13. EVA facial mouse	32
2.1.14. Eye Tribe	33
3. Metodología y tecnologías	35
3.1. Metodología Kanban	35
3.1.1. ¿Qué es?	35
3.1.2. Las tres reglas de Kanban	35
3.1.3. ¿Cómo funciona?	37

3.1.4.	Ventajas de la metodología Kanban	37
3.1.5.	Comparación con la metodología Scrum	37
3.1.6.	¿Por qué escoger Kanban?	38
3.2.	Desarrollo del proceso y plan del proyecto	38
3.2.1.	Aplicación Taiga.io	38
3.3.	Tecnologías	39
3.3.1.	Taiga	39
3.3.2.	Latex-Overleaf	40
3.3.3.	Python	41
3.3.4.	Arduino Studio	41
3.3.5.	AWS	42
3.3.6.	Ultimaker Cura	42
3.3.7.	ThingSpeak	42
3.3.8.	CircuitLab	42
3.3.9.	EasyEDA	43
 B. Diseño y arquitectura		 47
4. Diseño de la solución		47
4.1.	La importancia del parpadeo	47
4.2.	Patologías a detectar	48
4.2.1.	Luminosidad y calidad visual	48
4.2.2.	Temperatura y sequedad ocular	49
4.2.3.	Ojo seco	50
4.3.	Explicación de términos ópticos/optométricos	51
4.4.	Cómo escanear la cara con el sistema Kinect	52
4.5.	Construcción de la herramienta	53
4.5.1.	Escáner 3D con kinect	53
4.6.	Hardware diseñado y estudio	56
4.6.1.	Efecto Doppler	56
4.6.2.	Implementación del efecto Doppler en el trabajo	58
4.7.	Fases del proyecto	60
4.7.1.	Investigación y estudio	60
4.7.2.	Prototipo - Alpha: HAN SOLO	61
4.7.3.	Prototipo - 4 sensores	63
4.7.4.	Prototipo - MARK1	64
4.7.5.	Centinela Offline	65
4.7.6.	Integración con pulsera de actividad	65
4.7.7.	Centinela - Merovingio	67
4.7.8.	Gestión de riesgos activa	72
4.7.9.	Riesgos del planteamiento del proyecto	72
4.7.10.	Riesgos de la creación	73
4.7.11.	Riesgos con el proyecto finalizado	73
 5. Arquitectura e implementación		 75
5.1.	Esquemas de implementación Centinela - Final	75
5.1.1.	Modulo ESP8266	75

5.2.	Escenarios con AWS	77
5.2.1.	Amazon Web Service	77
5.2.2.	Conectar el módulo ESP8266 a AWS	78
5.2.3.	Coste AWS	80
5.3.	Escenarios con ThingSpeak	80
5.3.1.	ThingSpeak	80
5.3.2.	Conectar el módulo ESP8266 a ThingSpeak	81
5.3.3.	Coste ThingSpeak	81
5.4.	Esquema de Merovingio	82
5.5.	Esquema de Centinela	82
5.6.	Diagrama de conexiones offline	83
5.7.	Diagrama de conexiones en Cloud	83
6.	Mediciones y resultados	85
6.1.	Pruebas en sujetos con patologías	85
6.2.	Prueba de 72 horas	88
6.3.	Pruebas públicas	93
6.4.	Documento médico generado	98
7.	Conclusiones y futuras aplicaciones del proyecto	101
7.1.	Conclusiones	101
7.2.	Futuras aplicaciones del proyecto	102
7.2.1.	Aplicaciones ópticas	102
7.2.2.	Aplicaciones optometristas	102
7.2.3.	Otros futuros estudios	102
7.3.	Futuros prototipos	102
7.3.1.	Ojo sólido	102
7.4.	Futuras aplicaciones en Cloud	103
7.	Conclusions and future applications of the project	105
7.1.	Conclusions	105
7.2.	Future project applications	106
7.2.1.	Optical applications	106
7.2.2.	Optometrist applications	106
7.2.3.	Other future studies	106
7.3.	Future prototypes	106
7.3.1.	Solid eye	106
7.3.2.	Future Cloud Applications	107
8.	Repercusión	109
8.1.	Cadena Ser	109
8.2.	Telemadrid	111
8.3.	Concurso Universitario de Software Libre	113
8.4.	Presentación del TFG ante Óptica y Optometría	113
8.4.1.	Entrevista de la Facultad de Óptica y Optometría	115
9.	Bibliografía y enlaces de referencia	122

C. Anexos	129
A. Montaje de la impresora 3D	129
1.1. Planos y montaje de la PRUSA i3	129
1.1.1. Montaje de estructura	130
1.2. Mosfet externo para impresora 3D	147
B. Software usado y final	149
1.1. Software y resultados	149
1.2. Creaciones propias y códigos fuente	149

Parte A

Introducción y conceptos básicos

Capítulo 1

Introduction

—*No one gives it to you. You have to take it.*
Jack Nicholson as Frank Costello,
The Departed

1.1. Motivation

You look all around without thinking that one day the light that makes you see it shuts down, the eyes. Just as bricklayer makes use of his talented hands and that a footballer makes use of his legs, the main tool of an engineer in computer are his eyes. While overcoming each day makes more people with visual deficit doing computer studies, it is understandable and empathetic to look methods to help these people.

With the increasing of IOT into systems that we might consider dystopics (for example now the fridge warns me of lack of food in it), it makes sense to guide these technologies to the common good for medical and scientific progress.

It is our duty to see and observe what we can change and improve with technology. A responsibility that has to be done from scratch, with free research and creation and free. Eye problems, overexerting of the human eye can lead to a decrease in performance at work and, most importantly, in our daily lives and health. The use of such technology has to be the use of knowledge and skills to devise and build projects those health needs. Eye overexertion is often not taken into account much, as most of the time we are looking at our screens and we don't realize what's in front of us with what it entails. This overexertion causes us to encounter problems in close vision such as asthenopia, accommodating problems and problems of convergence[1]. In addition to these problems, we may encounter fatigue ocular[2] associated with it.

In conclusion, we can stay that eye strain leads to a degradation of the visual system and leading to pathologies associated with their degraded.

1.2. Goal to be realized

It is intended to create a child-themed device for therapeutic use with different features:

- Use and distinction of flicker independently and flawlessly, with detection active from false flickering in real time.
- Temperature change sensor and sudden changes in luminosity.
- Child theme for therapeutic use in hospitals and children's areas.

To perform the following software operations:

- Blink calculation and prediction and distinction to facilitate diagnosis of eye fatigue.
- Calculation of fatigue caused by abrupt variation in luminosity or temperature.
- Real-time calculations with bluetooth connection to mobile app to perform online calculations or locally to predict derived behaviors fatigue.
- Calculation via PC for behaviors.
- Specialist presets to facilitate the collection of medical data for further analysis.

This way, the tool will have three modes:

- How to collect information to collect data for medical use.
- Active medical mode, to detect pathologies and stop eye strain.
- Mixed mode, both at the same time.

A simple mechanism will be provided so that the project can be advanced by minimizing the impact of size on the subject.

1.3. About documentation and style

Document license

The document is under the CC-0 license.

Document license

All software is under MIT license.

Manuals and annexes

All manuals are made of public domain.

Documentation for all

Documentation is presented in english and spanish. It will also be presented with a version with tool use manual, assembly guide so that anyone you can mount it and a version of everything mentioned above for low vision. From free and free form.

Capítulo 1

Introducción

—*Nadie te da nada, tienes que cogerlo.*
Jack Nicholson como Frank Costello,
Infiltrados

1.1. Motivación

Observas todo alrededor sin pensar que algún día puede que la luz que hace que lo vea se apague, los ojos. Del mismo modo que un albañil hace uso de sus talentosas manos y que un futbolista hace uso de sus piernas, la principal herramienta de un ingeniero en informática son sus ojos. Si bien la superación cada día hace que haya más personas con déficit visual haciendo los estudios de informática, es comprensible y empático buscar métodos para ayudar a estas personas.

Con la creciente integración de IOT en sistemas que podríamos considerar distópicos (por ejemplo ahora la nevera me avisa de falta de alimentos en ella), es lógico orientar estas tecnologías al bien común al progreso médico y científico.

Es nuestro deber el ver y observar aquello que podamos cambiar y mejorar con tecnología. Una responsabilidad que se tiene que hacer desde cero, con investigación y creación libres y gratuitas.

Los problemas oculares, el sobre esfuerzo del ojo humano puede llevar a tener una disminución del rendimiento en el trabajo y, lo que es más importante, en nuestra vida diaria y salud. El uso de esa tecnología tiene que ser el uso de esos conocimientos y destrezas para idear y construir proyectos para satisfacer esas necesidades de salud.

El sobre esfuerzo ocular no se suele tener en cuenta mucho, ya que, la mayor parte del tiempo estamos mirando a nuestras pantallas y no nos damos cuenta de lo que tenemos delante con lo que ello conlleva. Este sobre esfuerzo hace que encontremos problemas en visión próxima como astenopía, problemas de acomodación y problemas de convergencia[1]. Además de estos problemas, podemos encontrar problemas de fatiga ocular[2] y patologías asociado a él.

En conclusión, podemos decir que el sobre esfuerzo ocular conlleva una degradación del sistema visual y llevando a patologías asociadas a su degradado.

1.2. Objetivo a realizar

Se pretende crear un dispositivo con temática infantil para uso terapéutico con las distintas características:

- Uso y distinción de parpadeos de forma independiente y sin fallos, con detección activa de falsos parpadeos en tiempo real.
- Sensor de cambios de temperatura y de cambios bruscos de luminosidad.
- Temática infantil para uso terapéutico en hospitales y zonas infantiles.

Para realizar las siguientes operaciones sobre software:

- Cálculo y predicción de parpadeos y distinción para facilitar el diagnóstico de fatiga ocular.
- Cálculo de fatiga producida por variación brusca en la luminosidad o temperatura.
- Cálculo en tiempo real con conexión vía bluetooth a aplicación de móvil para realizar los cálculos online o de forma local para predecir comportamientos derivados de la fatiga.
- Cálculo vía PC para los comportamientos.
- *Preset*s para especialistas para facilitar la recopilación de datos médicos para su análisis posterior.

De esta forma, la herramienta tendrá tres modos.

- Modo de recopilación de información para recaudar datos para uso médico.
- Modo médico activo, para detectar las patologías y frenar el cansancio ocular.
- Modo mixto, ambos a la vez.

Se proporcionará un mecanismo sencillo para que se pueda avanzar el proyecto minimizando el impacto de tamaño en el sujeto.

1.3. Sobre la documentación y estilo

Licencia del documento

El documento está bajo la licencia CC-0.

Licencia de todo el software

Todo el software está bajo licencia MIT.

Manuales y anexos

Todos los manuales son de dominio público.

Documentación para todos

La documentación se presenta en inglés y castellano. También se presentará una versión con manual de uso de la herramienta, guía de montaje para que cualquiera pueda montarlo y una versión de todo lo citado anteriormente para baja visión. De forma libre y gratuita.

Capítulo 2

Estado del arte

—*I amar prestar aen*
Galadriel (Sindarín)

2.1. Aplicaciones y proyectos similares

Se ha procedido a realizar una búsqueda de aplicaciones similares, hemos encontrado desde aplicaciones de reconocimiento facial a detección de parpadeo.

2.1.1. OpenFace

OpenFace¹ es un código abierto en Github que nos permite descargarnos un código para el reconocimiento facial. Además, no sólo nos permite realizar el reconocimiento facial sino que nos permite recortar la cara.

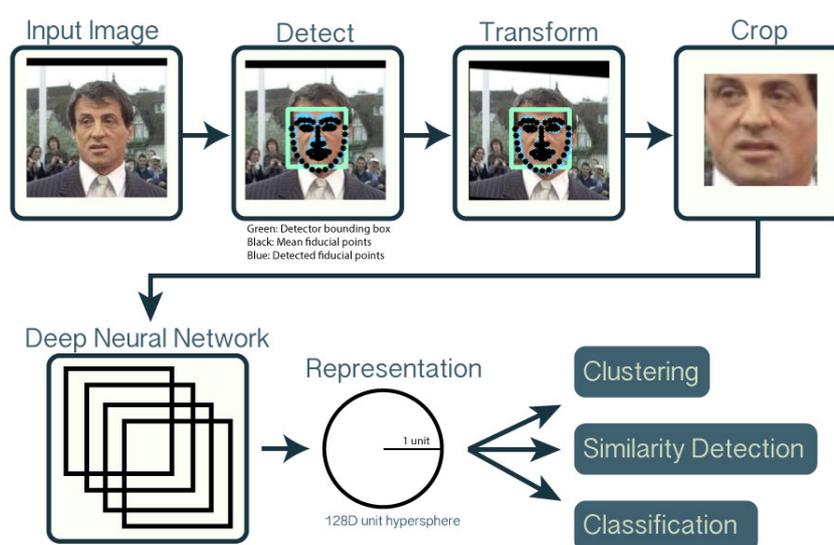


Figura 2.1: Funcionamiento de la aplicación OpenFace

¹Dirección del proyecto: <https://cmusatyalab.github.io/openface/>

2.1.2. Facefirst

Facefirst² es una aplicación privada que da servicios a comercios y a empresas. Promociona el reconocimiento facial para evitar hurtos en tiendas en comercios y empresas.



Figura 2.2: Facefirst en aeropuertos

2.1.3. herta

herta³ es otro software privado de reconocimiento facial. Es utilizado en empresas, aeropuerto u otras empresas que quieran obtener un servicio de ella. Utiliza el reconocimiento facial para aplicaciones de seguridad.

2.1.4. Snapchat

Es una aplicación móvil⁴ muy reconocida, realiza un reconocimiento facial para poder poner los emoticonos en la cara de las personas.



Figura 2.3: Captura de Snapchat

²Dirección de la aplicación: <https://www.facefirst.com/>

³Dirección de herta: <http://hertasecurity.com/es>

⁴Dirección web: <https://www.snapchat.com/1/es/>

Su uso es muy sencillo, sólo tiene que utilizar la cámara del móvil y mediante el reconocimiento facial, colocará los distintos emoticonos sobre la cara del usuario.

2.1.5. WebGazer

Se utiliza⁵ para seguir la dirección de los ojos, con este software se puede ver la predisposición que tiene un usuario por un contenido o por otro.

2.1.6. “Blink to Talk” usando Charlieplexing

Este proyecto⁶ está basado en que, los pacientes que son incapaces de comunicarse por una parálisis u otras patologías sean capaces de comunicarse mediante el parpadeo de sus ojos.

La funcionalidad del proyecto está limitado y no está comparado con las funciones de los eyewriter. Pero el concepto de este proyecto es realizar un proyecto accesible y barato.

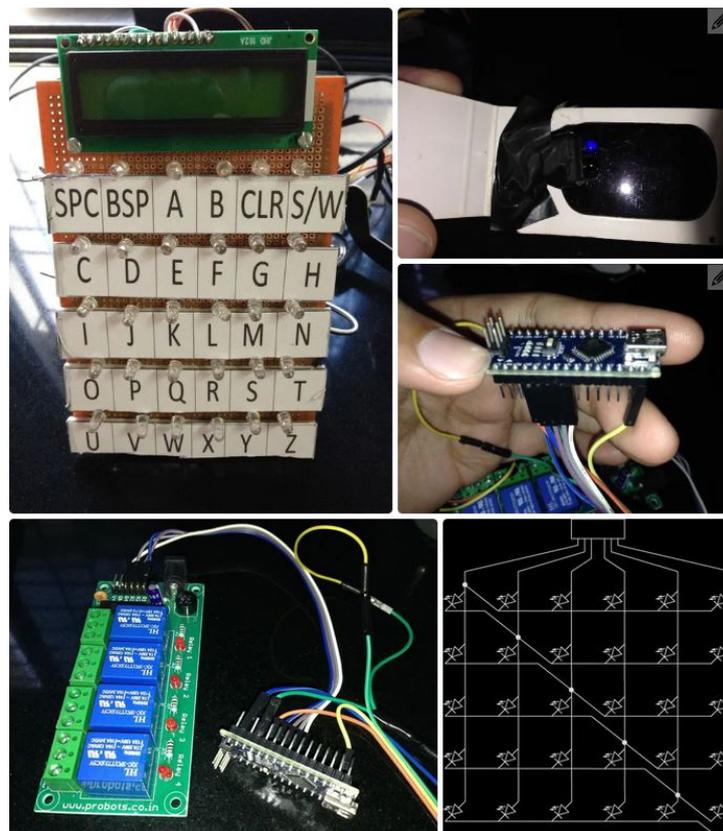


Figura 2.4: Imágenes del proyecto

La funcionalidad del proyecto es sencilla. El sensor está formado por un LED IR o un fotodiodo montado en unas gafas. El valor que vuelve al fotodiodo depende de cómo la

⁵Información obtenida de la propia web de la aplicación: <https://webgazer.cs.brown.edu/>

⁶Información obtenida del proyecto original: <https://www.instructables.com/id/Arduino-Blink-to-Talk-Using-Charlieplexing/>

luz IR es reflejada en el párpado o en la esclera. Esto es usado para obtener un valor límite para la detección del parpadeo.

El sistema está formado por 26 LEDs por cada una de las letras del alfabeto. Para que el coste sea barato utiliza ATmega328, sin embargo, este sistema se queda corto. Para arreglarlo utiliza el método de multiplexación denominado Charlieplexing. En este sistema hay 30 LED que están conectados en una cuadrícula como se muestra en la figura 3.3. Cada fila de líneas está formada por una serie de LEDs dispuestas en 4 líneas como un “High Impedance State”, lo que significa que en el sistema Arduino estas 4 líneas son INPUT.

Cuando se enciende el sistema se requerirá un calibrado del sistema. Este calibrado se obtiene mediante los valores límites que se obtienen al abrir y cerrar los ojos. El sistema empieza a escanear las filas en intervalos de 1-2 segundos mientras se realiza el parpadeo. Cuando el parpadeo se detecta en una columna, el sistema empieza a escanear las columnas de LED mientras se realiza el segundo parpadeo. Cuando este segundo parpadeo se detecta es cuando se ilumina el LED y se ve en la imagen LCD. Esta señal puede ser transmitida de una manera opcional mediante una señal bluetooth o de manera inalámbrica.

L	Get me a doctor
M	I am not hungry
N	I want to sleep
O	I need medication
P	What time is it?
Q	I am thirsty
R	I need the newspaper
S	I need assistance
T	I want to read
U	EMERGENCY
V	SWITCH OFF
W	TOGGLE 1
X	TOGGLE 2
Y	TOGGLE 3
Z	TOGGLE 5

SPC	Hello
BSP	Goodbye
A	Thank you
B	I am in pain
CLR	I need a doctor
S/W	Put on the television
C	What day is it?
D	I am hungry
E	I am sick
F	Draw the curtains
G	Open the curtains
H	Call my family
I	Turn down the volume
J	Turn up the volume
K	Take me outdoors

Figura 2.5: Traducción de las letras por órdenes

Cuando el segundo parpadeo llega a cada una de las letras y es verificado por el sistema se mostrará la orden en la pantalla LCD.

Seguidamente, vamos a proceder a describir el diagrama utilizado para este proyecto. Para ello, vamos a introducir 6 LEDs en una fila en el diagrama. En la siguiente figura se va a mostrar las uniones realizadas con el Arduino y el sistema:

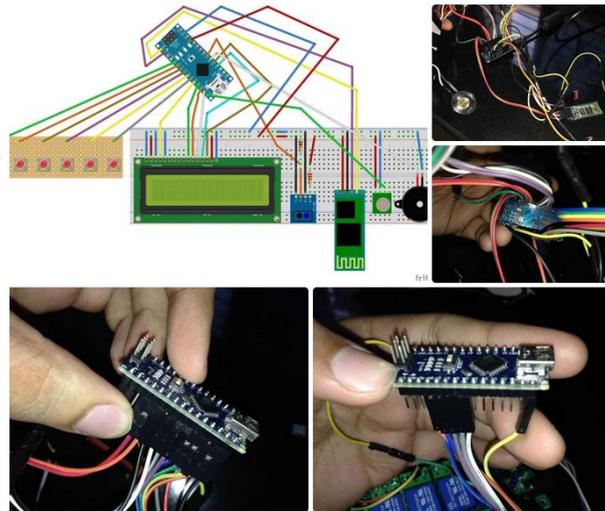


Figura 2.6: Uniones Arduino

A este sistema se le puede agregar un sistema wireless para transmitir la señal vía inalámbrica.

El sistema de detección del parpadeo está separado del Arduino y del wireless y tiene su propio módulo. Como este proyecto va a ser utilizado por pacientes con problemas de movilidad, la fila de LED debería estar lo más cerca posible del paciente para que tenga una mayor visibilidad y se solucionen futuros problemas.

Posteriormente, vamos a comentar cómo se construyó el detector de parpadeo de este proyecto.



Figura 2.7: Gafas 3D utilizadas en este proyecto

Como se puede observar para montar el sensor del parpadeo en este proyecto se utilizó una gafa 3D y se montó el sensor en la parte posterior de la gafa, enfrente del ojo.

Finalmente, comentaremos que en este proyecto se van dando las instrucciones para el calibrado del sensor al paciente mediante una pantalla LCD.

2.1.7. Sistema de detección de parpadeo mediante unas gafas smart

En este proyecto se han basado en la alineación del sistema con el ojo. Es muy importante tener en cuenta que el parpadeo se basa en la apertura y el cierre rápido de los párpados[3]. Detecta el cierre de los párpados usando el método del gradiente. El método utiliza un algoritmo que suprime los máximos para detectar el parpadeo.

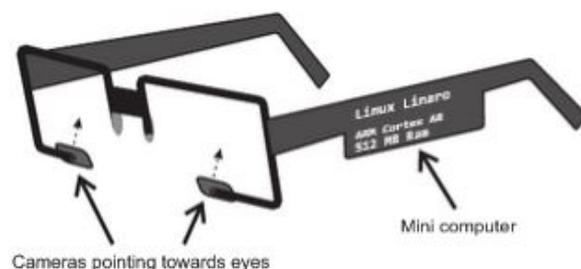


Figura 2.8: Configuración de la gafa smart

En este método se toma como medida el cierre del párpado en vídeo.

Los componentes de este proyecto son económicos. Lo primero va a ser una montura estándar, se pone una cámara en cada uno de los ojos para capturar cada uno de sus movimientos. Estas cámaras se posicionan en el centro del ojo y en su parte inferior. Para el procesamiento se conecta a una CPU mediante un MK802 y basado en Linux.

El sistema de detección es similar al utilizado en personas con poca movilidad para que puedan interactuar con el ordenador[4]. O también es similar al utilizado en el cine[5] para animar el parpadeo.

Una vez hemos visto en qué proyectos similares está basado este proyecto, vamos a ver cómo recoge los datos.

La complicación que tiene este sistema es la captura en vídeo del sistema de parpadeo, ya que, la calidad del vídeo es mala y es inestable. Para mejorar la captura de datos con el ojo cerrado se va a utilizar un sistema de alineación con el ojo.

Para mejorar en la detección al cerrar el ojo se ha utilizado un software similar al utilizado en vídeo para el reconocimiento facial[6]. Primero empieza recogiendo las imágenes de los ojos abiertos y cerrados. Seguidamente, aplican un componente de análisis de imágenes para obtener un set de imágenes básicas[7]. Se reconstruye el ojo como una combinación lineal de estas imágenes básicas. Después, obtienen un vector para representar cada imagen. Seguidamente, usan un “Gradient Boosting” (GB) para clasificar entre ojo abierto y ojo cerrado en la base de datos[8]. Se encarga de clasificar cada frame con el ojo cerrado y lo utilizará posteriormente para la detección del parpadeo.

El sistema requiere una alineación con el ojo para evitar una desalineación de las imágenes. Para evitar futuros problemas por las distintas formas de cara, este proyecto realiza un calibrado para poder ajustar la imagen a la posición de los ojos basado en el método de Haar desde un OpenCV⁷.

El parpadeo es detectado como un pequeño número de imágenes consecutivas capturadas al abrir o cerrar los ojos. Para ahorrar energía y obtener unos resultados acorde con los

⁷Dirección web: <http://opencv.org/>

datos científicos según Human Vision Research obtuvieron 6 frames. Por cada frame, se obtiene una imagen del ojo cerrado para contar el parpadeo.

Posteriormente, recogen como datos positivos como parpadeo y como datos negativos como no parpadeo. Utilizarán los datos del parpadeo como vector. Vuelven a utilizar el GB para comprobar si hay parpadeo o no.

El método de detección del parpadeo a veces puede obtener duplicidades de parpadeo durante la captura de vídeo. Para evitar las duplicidades utilizan el método de los máximos en la que en el momento de cerrar el ojo se obtiene un máximo en el frame. Para verlo el método vamos a visualizarlo con una imagen del proyecto.

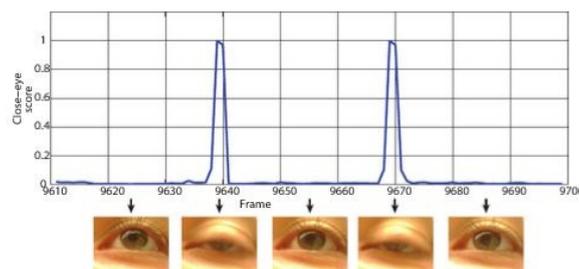


Figura 2.9: Detección del ojo cerrado

En este proyecto han participado 4 pacientes de distintas razas en distintos entornos luminosos.

Para representar los datos los colectan con el PCA para extraer las funciones y usar el GB para que detecte el cierre de los ojos y detecte el parpadeo. Además, testearon con el PCA y el Componente Independiente de Análisis (ICA)[9] para la función de extracción del vector y testean con “Support Vector Machine” (SVM)[10], “Gradient Boosting” (GB) y Gaussian Naive Bayes (GNB) para la detección de los distintos pasos. En total han utilizado 6 diferentes de combinaciones de algoritmos en total.

Recolectaron un total de 3861 imágenes como ejemplos positivos con el ojo cerrado y otras 3861 imágenes como ejemplos negativos con el ojo cerrado. También obtuvieron 1419 vídeos de 6 frames, unos en los que se detectaba parpadeo y otros en los que no. De los cuales el 75 % fueron a una base de datos y el 25 % fueron evaluados a través de las distintas funciones y métodos.

El experimento muestra los distintos métodos para conseguir las imágenes. La velocidad del algoritmo desciende con el tamaño de la imagen o los números de las funciones. Por lo que el tamaño del frame será de 16×12 para que mediante los métodos PCA o ICA puedan extraer al menos 30 funciones y puedan usar el “Gradient Boost” como método de detección para obtener una buena relación entre la velocidad y la precisión.

En conclusión este proyecto ha obtenido unas gafas smart por las cuales se ha podido detectar la frecuencia del parpadeo a través de distintas funciones. Destaca que tiene una fuente de potencia de poca potencia y que la calidad del vídeo es baja.

2.1.8. Sensor de parpadeo y Arduino

Es un pequeño proyecto hindú⁸, en el que destaca su código abierto. Es un sensor sencillo montado en una gafa, tiene un pequeño LED que en un primer momento se ilumina cuando parpadea el paciente y finalmente, detecta el cansancio cuando disminuye la frecuencia de parpadeo y avisa con un ruido.

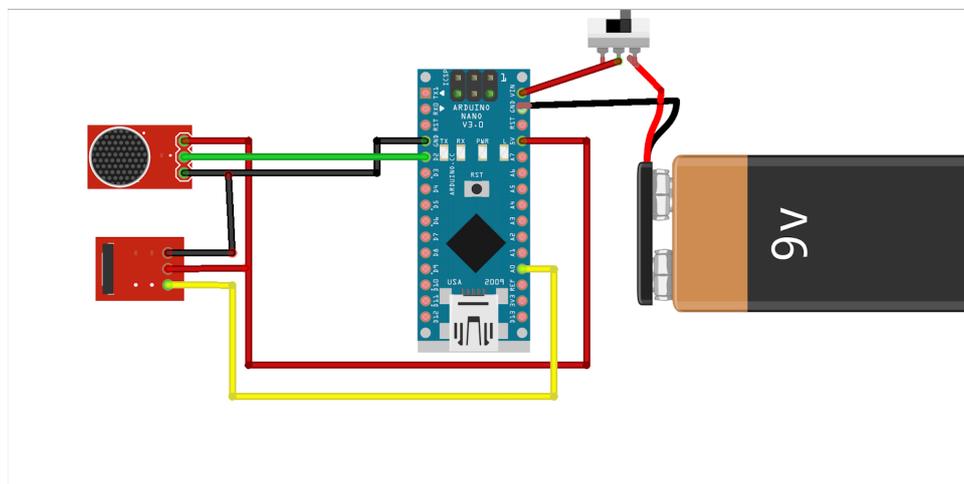


Figura 2.10: Esquema sensor utilizado

2.1.9. Método eficiente de detección de parpadeo para ayudar a personas discapacitadas

Este proyecto[11] utiliza un procesamiento de imágenes en vídeo para detectar el parpadeo mediante algoritmia. Se basa en el cerrar y abrir los párpados para controlar los móviles. Esta aplicación es usada en tiempo real para estudiar el efecto de la luz, la distancia entre los ojos, el uso del móvil para evaluar la detección y la precisión del sistema en general. La detección del parpadeo es importante componente en varios de los dominios de este estudio, como la interacción de la persona con el ordenador, salud visual y conducir de una manera segura. Por ejemplo, ha sido utilizado una modalidad para personas discapacitadas puedan interactuar con ordenadores y teléfonos móviles[12].

Para este proyecto se ha utilizado una cadena de funciones filtro únicas tipo cascada de Haar[13] para identificar la parte inferior de la imagen. Con esto podemos realizar el primer cálculo de la imagen integral.

Mediante el seguimiento ocular puede haber una interacción con la interfaz de una manera rápida sin ser invasiva. Se ha utilizado una plantilla como ojo abierto y ojo cerrado para hacer de patrón en la medida[14]. Es un método muy útil para personas con parálisis. La detección del parpadeo en tiempo real basado en la función SIFT siguiendo con la implementación en GPU[15].

El método eficiente[16] basado en el procesamiento de imágenes para detectar el parpadeo y generar unos intervalos de parpadeo es el clasificador cascada de Haar y el algoritmo Camshift.

⁸Enlace al vídeo del proyecto:<https://www.youtube.com/watch?v=XpfINvWDnCo>

Utiliza una cámara colocada en la parte anterior del teléfono[17] para detectar los movimientos del ojo y aprovecharlos para dirigirlos en el móvil. Se controla en el móvil a través de un control móvil EBCM.

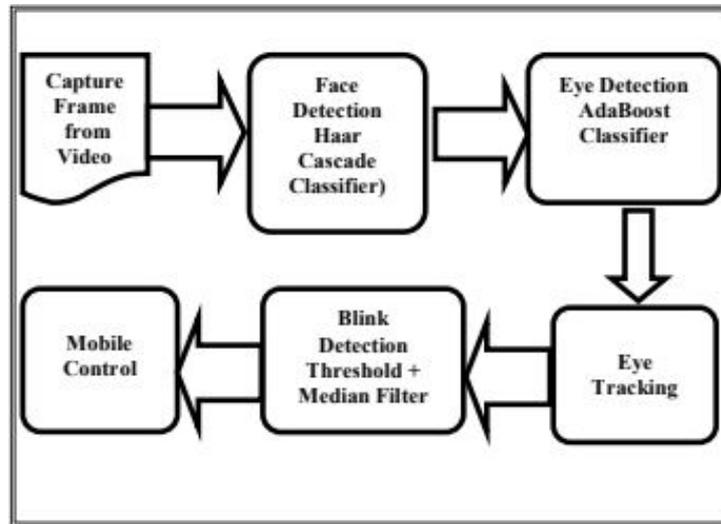


Figura 2.11: Diagrama EBCM

Lo primero que realiza el EBCM es convertir el vídeo corto en frames. Estos frames coloreados se pasarán a frames en escala de grises.

Seguidamente, el clasificador Haar detecta los objetos que necesitan localizar, por ejemplo, la cara. Reorganiza la área para eliminar los falsos positivos. Como resultado de esto, se marca la cara con una región rectangular y los ojos con una región cuadrada.

Para detectar al ojo deben estar implementadas las funciones Haar y AdaBoost, porque una contiene la imagen y otra tiene el objeto.

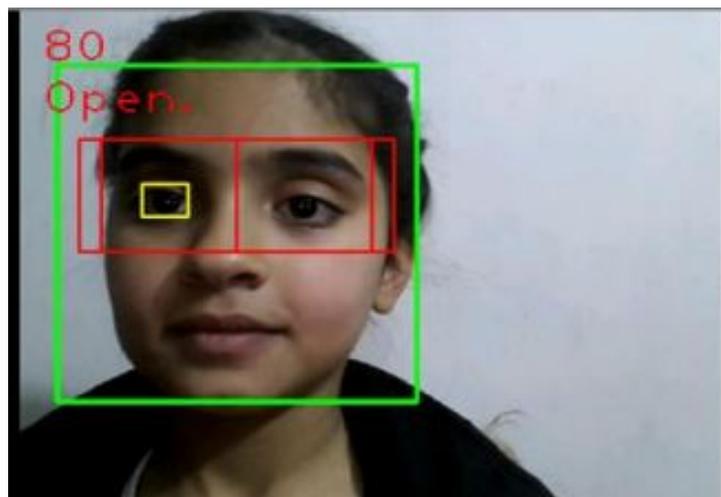


Figura 2.12: Detección ojos

El método EBCM utiliza distintas funciones para detectar el movimiento del ojo mediante la reflexión corneal y el centro de la pupila.

El parpadeo y los movimientos oculares pueden ser detectados con alta confianza mediante

técnicas que no son obstructivas.

Primero, vamos a mostrar el esquema en la que detectan el límite del parpadeo:

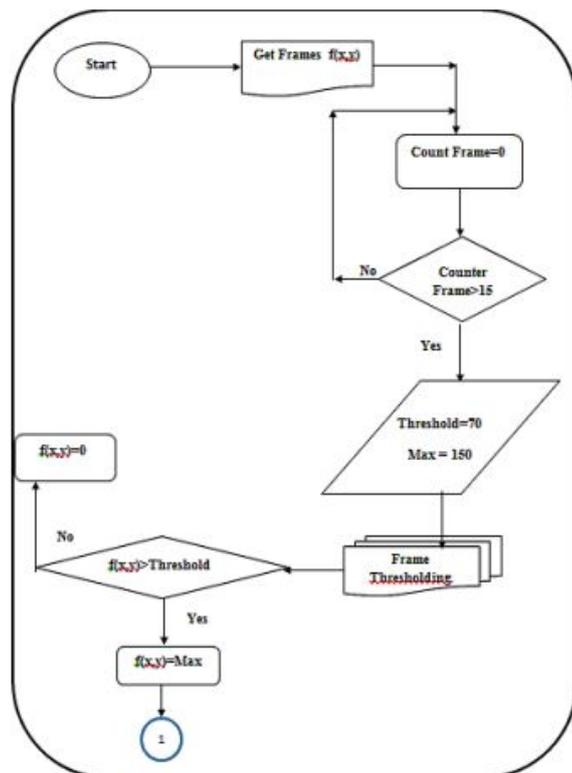


Figura 2.13: Esquema límite

Seguidamente, se va a mostrar el esquema que utilizan para detectar el parpadeo:

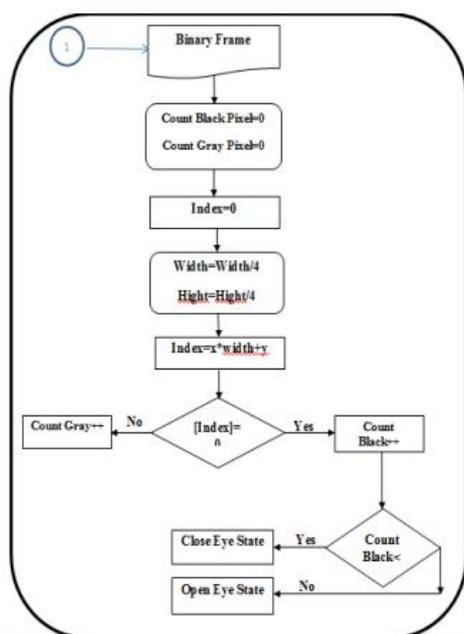


Figura 2.14: Esquema detección parpadeo

Como se pueden ver en ambos esquemas, tratan las imágenes con un preprocesado para eliminar el ruido de la imagen y mejorar la precisión de la detección.

Después de aplicar el algoritmo, verifica si el límite se supera en algunos de los píxeles de la imagen y les devuelve el valor del píxel en ese punto.

El último paso que sigue el algoritmo EBCM es controlar la actividad del móvil haciendo una llamada telefónica dependiendo si el ojo está abierto o cerrado.

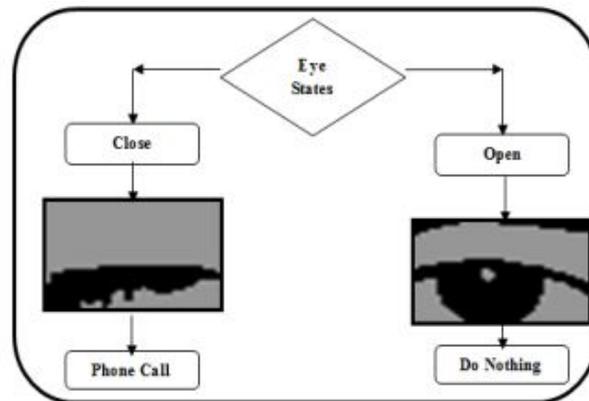


Figura 2.15: Esquema algoritmo EBCM

Los parámetros de este algoritmo son la distancia al sensor y la iluminación.

Para valorar si era óptimo el sensor lo probaron a distintas distancias con distintas condiciones de iluminación. Y obtuvieron los siguientes resultados:

Cuadro 2.1: Resultados del estudio

Distancia (cm)	VP	FP	FN	VN	General %	Detección %
13	0	0	0	0	0	0
20	0	0	18	53	75	0
25	1	0	99	2489	96	1
30	3	0	265	2300	90	1
35	221	77	4	1355	95	98
40	142	16	30	2671	98	83
45	2	0	10	26	74	17

Como conclusión este estudio destaca al complicación de ajustar el parpadeo al control del móvil. En un futuro quieren mejorarlo añadiendo un reconocimiento de voz y adaptarlo con una aplicación para evitar la detección de la cara de la persona en la cámara frontal.

2.1.10. Captura de imágenes usando la detección de parpadeo

Este proyecto[18] está inspirado en las “Google Glass”, se trata de realizar capturas de imágenes usando la detección de parpadeo.

Utiliza un microcontrolador para analizar la señal y detectar el parpadeo. La cámara se encarga de capturar las imágenes y enviarlas vía bluetooth al módulo, que envía la

información al móvil. En el proyecto se han ayudado de dispositivos espías. Para que detecten el parpadeo tienen las siguientes opciones de tecnología:

- a). Procesado de Imagen.
- b). Electrooculograma.
- c). Tecnología infrarroja.

En el proyecto se ha optado con la tecnología del procesado de imagen[11] para detectar el parpadeo. Utiliza el clasificador cascada de Haar. La técnica sigue los siguientes pasos:

- a). Captura de imágenes.
- b). Detección de la cara.
- c). Detección del ojo.
- d). Seguimiento del ojo.
- e). Detección del parpadeo.

Lo primero que realiza es la captura de imágenes en color, seguidamente son convertidas en escalas de grises, posteriormente detectan la cara usando el clasificador de Haar y mediante una función se encarga de detectar y reorganizar una área.

La detección de los ojos se realiza mediante los clasificadores de Haar. Una vez se ha detectado la cara, los algoritmos de Haar y AdaBoost se encargan de realizar dos sets de imágenes. El primero contiene la imagen y la segunda no contiene el objeto. Una vez que el clasificador de Haar detecta los ojos, los señalará bordeando el ojo.

Para detectar el ojo se utilizan distintas funciones, es muy importante detectar el reflejo corneal y el centro pupilar. A través de distintas imágenes detectan el parpadeo. Este proceso depende de la distancia al sensor y las condiciones de iluminación. Si la distancia es mayor, peor será el reconocimiento. También le añadirán una serie de filtros para eliminar el ruido de la imagen para su preprocesado y mejorar la precisión de la detección de parpadeo.

Por otro lado utilizan el electrooculograma[19] para detectar el parpadeo. El parpadeo puede ser espontáneo, reflexivo o realizarlo de una manera voluntaria. Cuando el ojo parpadea genera una actividad eléctrica en el eje vertical y horizontal del electrooculograma. Este método utiliza las señales que genera el parpadeo, identifica su amplitud y determina el tipo de parpadeo.

Para la detección del parpadeo colocan cuatro electrodos, uno en la parte superior, otro en la parte inferior, otro en la parte derecha y otro en la parte izquierda. A continuación vamos a mostrar los electrooculogramas obtenidos de los distintos parpadeos de este proyecto:

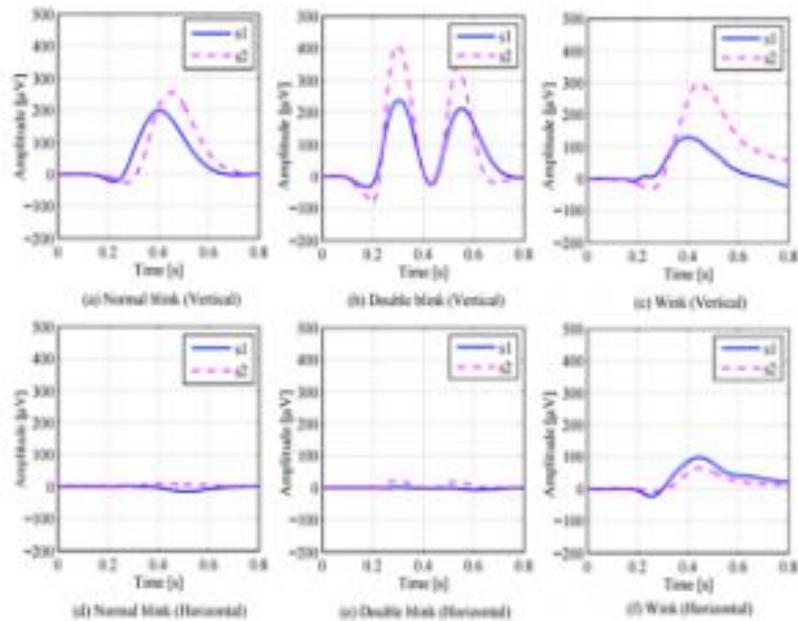


Figura 2.16: Electrooculograma de los distintos tipos de parpadeo

Se puede ver que las gráficas tienen distinta amplitud según el tipo de parpadeo. Pero este método tiene muchas desventajas, ya que, los sensores a veces pueden detectar unas señales fallidas, que el equipamiento no es cómodo de colocar y el coste de este equipamiento es muy alto.

La última herramienta utilizada es la tecnología infrarroja. Sus componentes son los LEDs y los detectores IR. Están situados a ambos lados de la montura de la gafa. El pico de transmisión del IR es de $880 \mu\text{m}$.

Los detectores se colocan en la parte nasal y temporal de cada ojo. El rayo infrarrojo detecta el parpadeo del párpado superior e inferior.

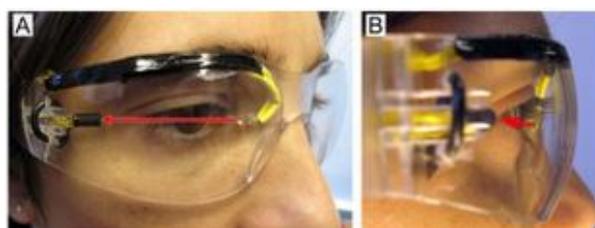


Figura 2.17: Detección parpadeo IR

Las desventajas de este método son las fluctuaciones del rayo IR, el circuito es muy delicado y que los detectores necesitan un ángulo específico.

A continuación se va a mostrar el diagrama utilizado para este proyecto:

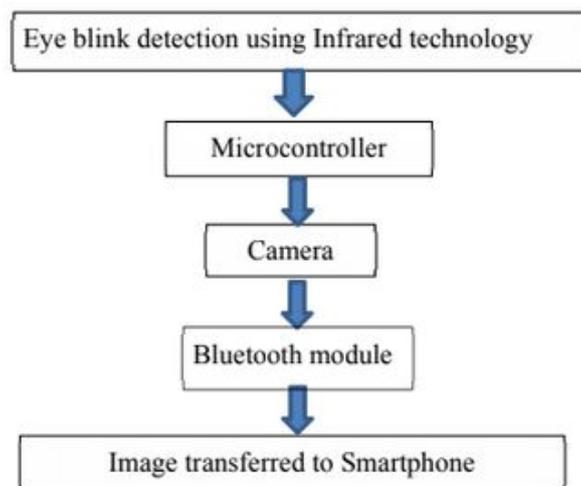


Figura 2.18: Diagrama del proyecto

El microcontrolador captura las imágenes del parpadeo cada dos segundos. Si detecta más de un parpadeo en dos segundos envía una señal a la cámara, pasa por el LED y vuelve a la cámara.

Después que envíe la señal a la cámara, esta captura la imagen y la envía vía bluetooth al móvil. El módulo de la cámara está conectado al Arduino.



Figura 2.19: Cámara utilizada

Seguidamente, la captura de imágenes pasa al módulo bluetooth. Este módulo tiene que estar conectado al móvil para transmitir imágenes sin error. También tiene que estar conectado a la cámara y al microcontrolador al mismo tiempo.

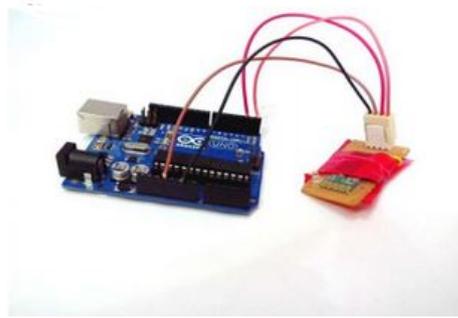


Figura 2.20: Módulo bluetooth

Con el proyecto han desarrollado un dispositivo capaz de ayudar a personas con problemas de movilidad. Serán capaces de utilizar la cámara sin ningún problema.

2.1.11. Detección del parpadeo basado en las señales oculares y el filtro Savitzky-Golay

Este proyecto se basa en otros proyectos similares para personas con problemas de movilidad y ordenadores[20], detección del sueño[21] y problemas cognitivos[22]. En este método tienen en cuenta la resolución de la imagen, la iluminación y los movimientos de la cara. Primero tienen en cuenta las señales en la cara, seguidamente se tiene en cuenta los ojos abiertos midiendo la distancia entre los párpados. Le aplican el filtro de Savitzky-Golay (SG) para obtener una buena señal y reducir el ruido. Con esto se puede detectar el parpadeo, ya que, detecta cuando se abren y cierran los ojos. Terminan acotando la duración del parpadeo mediante la función FSM.

Destacan que los algoritmos más empleados son los de Viola y Jones para detectar la cara y los ojos[23]. Sin embargo, estos algoritmos no eran óptimos cuando se cambiaba la iluminación. Para este proyecto se ha utilizado un histograma de patrones locales binarios (LBP)[24] para detectar el parpadeo. Primero, al abrir el ojo se crea una plantilla con muchas imágenes donde el ojo está abierto y sin moverse. Por cada plantilla se usa una medida de divergencia de Kullback-Leiber. Seguidamente, se detectan los máximos usando el test de Grubb y que considera que el ojo parpadea. Se le añade un gradiente pesado descriptor (WGD)[25] para tener un valor de retorno por los modelos cascada. Con ello se crean vectores que localizan la región de cada ojo con un pixel. Los máximos y los mínimos representan la apertura y el cierre del ojo. Se le añade un nuevo detector en la cámara de Basler y un nuevo dataset.

Posteriormente pasan a detectar la cara, para ello utilizan unos sensores que evalúan la posición, la iluminación y la rotación de la cabeza. Después se pasan a evaluar los falsos positivos y los falsos negativos. A partir de esto, se obtienen la frecuencia, la amplitud y la duración.

Seguidamente, vamos a describir el método utilizado en este proyecto. Utilizan Zface para determinar la posición de los ojos y de la cara. Y a partir de una serie de filtrados y cálculos detectan el parpadeo siguiendo los siguientes pasos:

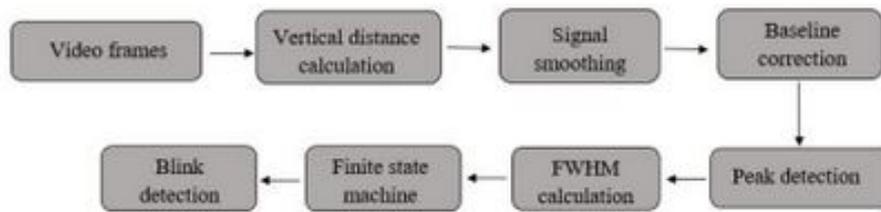


Figura 2.21: Técnica utilizada

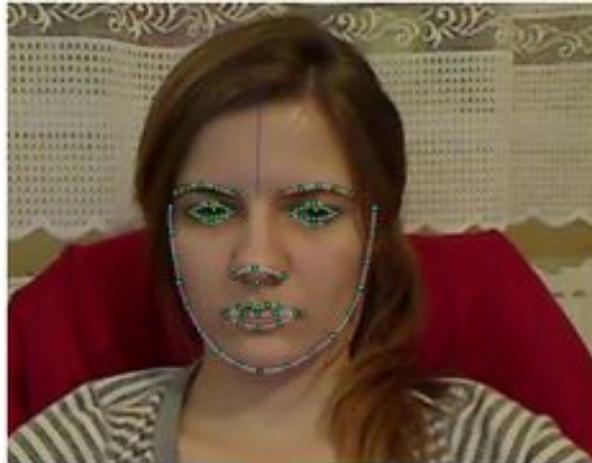


Figura 2.22: Detección cara

Seguidamente, para quitar el ruido de la señal utilizarán el filtro SG. Con el filtro obtendrán un polinomio que separará el ruido de la imagen.

Posteriormente, obtendrán los picos de la señal. Para ello, escanearán 30 fps de vídeo en una ventana de una duración de $500 \mu\text{s}$. Y le vuelven a aplicar el filtro SG para mejorar la señal y la detección.

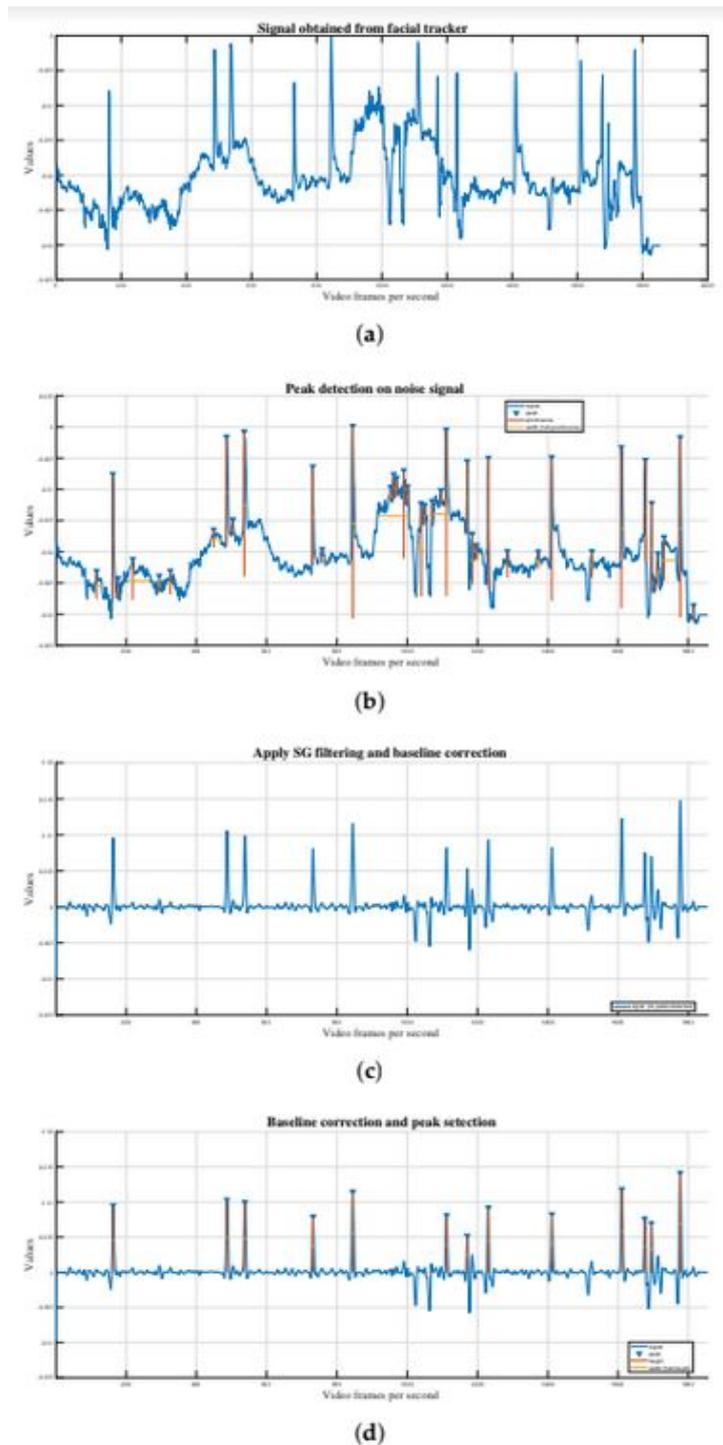


Figura 2.23: Distintas gráficas del proceso para la detección del parpadeo: a) Señal obtenida tras marcar la cara en una sesión de vídeo, b) picos aplicados a la señal con ruido, c) filtro SG aplicado a la señal con la base mal, d) detección de picos aplicado a la señal filtrada

Después de que la señal filtrada es obtenida pasa por una máquina de estados finito (FSM) para detectar los picos en caso de realizar o no el parpadeo. En caso de este proyecto se consideró la duración del parpadeo de 100 a 500 μs . El ancho de los picos representa el tiempo en el que se ha detectado el parpadeo.

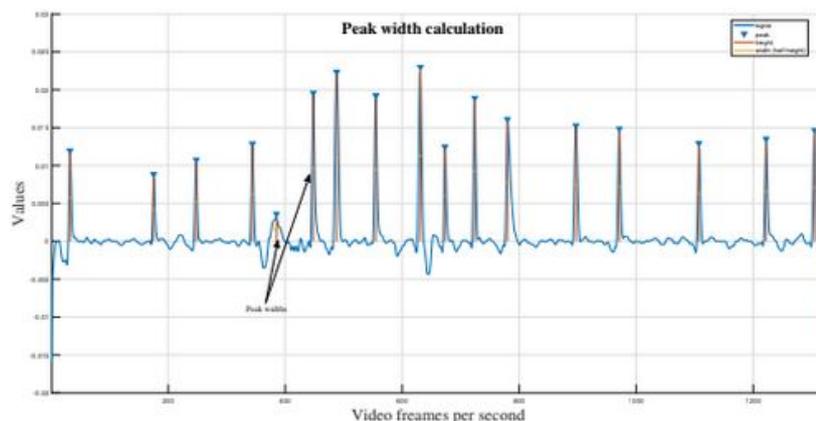


Figura 2.24: Anchura picos de parpadeo

Una vez se detectan los picos pasarán por la FSM para calcular la duración del parpadeo.

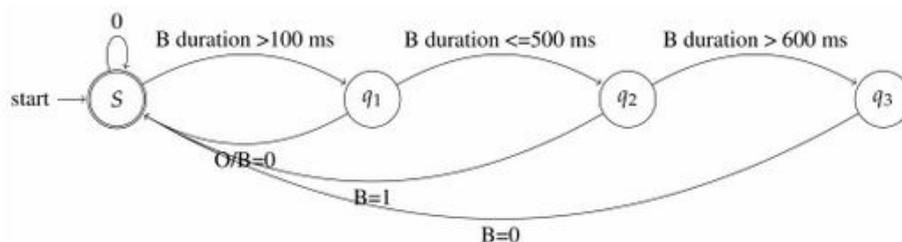


Figura 2.25: FSM para la duración estimada del parpadeo

Posteriormente, se almacenan 80 vídeos de 20 individuos en una base de datos ZJU[26], las expresiones faciales se almacenan en la base de datos Eyeblink8[27] y también almacenaron los vídeos del paciente mirando a la cámara.

A continuación calcularán la precisión y el recordado de los verdaderos positivos y verdaderos negativos:

$$\text{Precisión} = \frac{VP}{VP + FP}$$

$$\text{Recordado} = \frac{VP}{VP + FN}$$

Y les dieron los siguientes resultados:

Cuadro 2.2: Resultados obtenidos

Base de datos	Precisión	Recordado
ZJU	100 %	98.08 %
ZJU	94.4 %	91.7 %
ZJU	91 %	73.1 %
ZJU	100 %	98.01 %
Mirando a cámara	95 %	93.44 %
Mirando a cámara	83.3 %	91.2 %
Mirando a cámara	92.2 %	96.7 %
Mirando a cámara	98.38 %	98.38 %
Eyeblink8	94.69 %	91.91 %
Eyeblink8	96.65 %	98.78 %

Finalmente, se mide la duración, la amplitud y el ratio del parpadeo. El ratio de parpadeo indica el número de parpadeos por minuto y se mide mediante el parámetro ZJU. Durante la conversación, el ratio de parpadeo aumenta. Seguidamente, se mide la duración del parpadeo. Esta propiedad mide de una manera indirecta el estado de ánimo de una persona, ya que parpadear más o menos puede reflejarlo. La amplitud del parpadeo se mide cuando el ojo está abierto y entre los distintos pacientes.

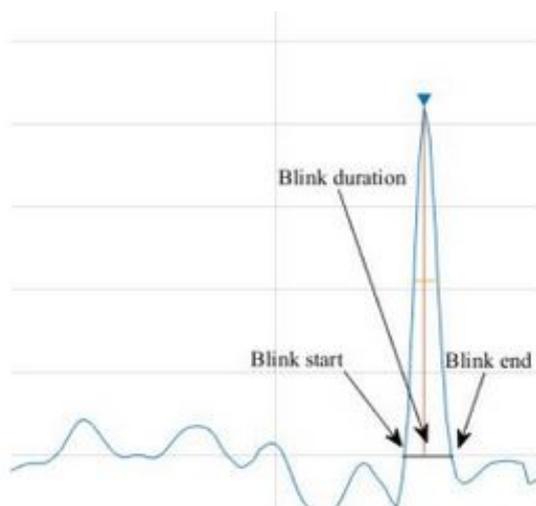


Figura 2.26: Esquema duración parpadeo

Cuadro 2.3: Estadísticas del proyecto donde GT (verdaderos parpadeos), DB (parpadeos detectados), RGT (ratio de parpadeo), RD (ratio de parpadeo detectado) y duración media

Base de datos	Video	GT	DB	RGT	RD	Duración media (ms)
Eyeblink	1	38	43	4.4	5.1	395
	2	88	89	14.3	14.4	297
	3	65	67	12.7	13.1	222
	4	31	31	10.3	10.3	245
	8	30	34	5.3	6.1	190
	9	41	41	16.2	16.2	224
	10	72	72	14.2	14.2	177
	11	43	44	17.6	18.1	254
Mirando a cámara	1	61	61	24.7	24.7	184
ZJU	261	256	61	48.93	48.03	190

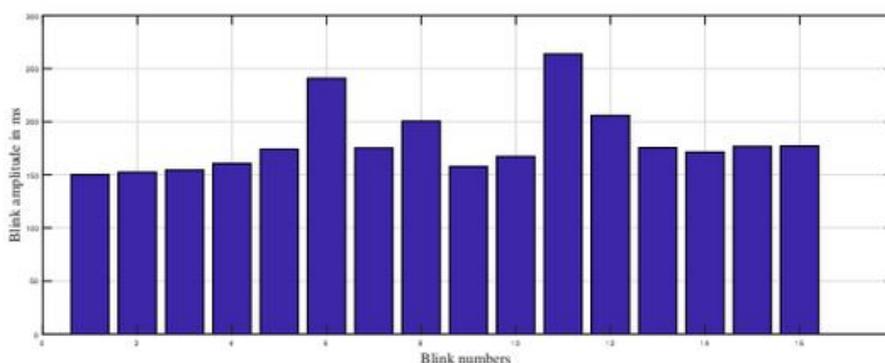


Figura 2.27: Amplitud de parpadeo medida

El proyecto concluye que la detección del parpadeo es esencial para la prevención de la fatiga ocular y ayudar a personas con problemas de movilidad. Ha utilizado técnicas distintas respecto de otros proyectos para evitar el ruido en el procesado de imagen y ha medido el ratio del parpadeo, su amplitud y frecuencia.

2.1.12. Detección del parpadeo para interacción persona-ordenador

En este proyecto⁹ se van a encargar de comunicar al ordenador con la máquina mediante el parpadeo.

El sistema es capaz de procesar una secuencia de imágenes de la cara a una velocidad de 30 fps. El propósito de este proyecto es detectar los parpadeos largos y cortos e interpretarlos. Separando los parpadeos cortos, que la mayoría suelen ser involuntarios, para poder diseñar un código para el parpadeo.

Con este proyecto pretenden que personas con problemas de parálisis o movilidad puedan controlar el ordenador a través de la mirada.

La mayor parte de sistemas que controlan la dirección de la mirada lo realizan a través de

⁹Dirección al artículo de este proyecto: <https://link.springer.com/article/10.1007/s10209-011-0256-6>

sensores IR[28] o electrooculogramas[29]. Con estos sistemas se puede distinguir el parpadeo voluntario del involuntario, pueden dar lugar a estrategias para controlar programas y si se añade control facial para decir palabras.

La detección del parpadeo se puede dividir en dos grupos, una de manera activa y otra de manera pasiva. La activa requiere de una iluminación especial mientras que la pasiva no requiere de una iluminación concreta, ya que, trata de medir la luz que devuelve retina. Un sistema de visión similar al teclado podría utilizar dos diodos apuntando a pupila, que controlan la dirección de la mirada y que reemplazarían al ratón a la hora de interaccionar con el ordenador.

Un sistema[30] más eficaz para realizarlo es mediante el uso de dos webcam, una que siga a la pupila y otra que siga a la cabeza.

Los métodos pasivos de detección de parpadeo no requieren luz adicional. Pero requieren de otros elementos como de plantillas[31], modelos de color[32], proyecciones[33], transformadas de Hough[34], múltiples ondas respuestas de Gabor[35] o la detección del ojo usando Haar[36].

La manera más sencilla de detectar el parpadeo es mediante la substracción de imágenes para crear una imagen diferente.

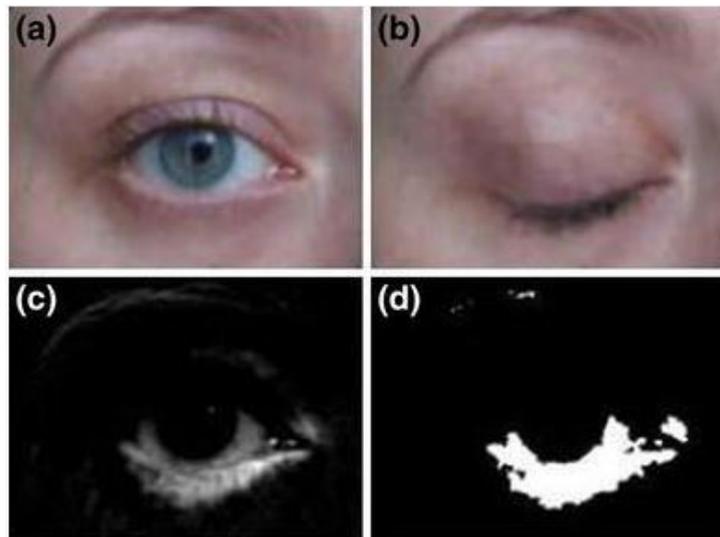


Figura 2.28: Detección del parpadeo mediante la diferenciación de imágenes: a) imagen del ojo abierto, b) imagen del ojo cerrado, c) diferencia de imágenes, d) limite de la diferencia de imágenes

El método basado en la plantilla consiste en comparar una serie de imágenes con una imagen de referencia. Se utiliza para la detección de objetos utilizando una función de Haar[37].

Un ejemplo de este tipo de detección ha sido aplicado por los laboratorios “Applied Science”¹⁰. Pero se ha comprobado que el sistema no es cómodo para todos los usuarios aunque dé medidas de confianza.

Un ejemplo de sistema no intrusivo es uno que utiliza una cámara USB[38] para realizar los seguimientos del ojo basados en la simetría entre el ojo derecho e izquierdo. Se utiliza para jugar en el ordenador o para buscar por Internet.

¹⁰Artículo del estudio: <http://www.a-s-l.co/>

El sistema propuesto está formado por un PC y por una webcam de una calidad media. La detección del parpadeo se hace en cuatro pasos: primero se detecta la cara, después se extrae una región del ojo, se detecta el parpadeo y se clasifica.

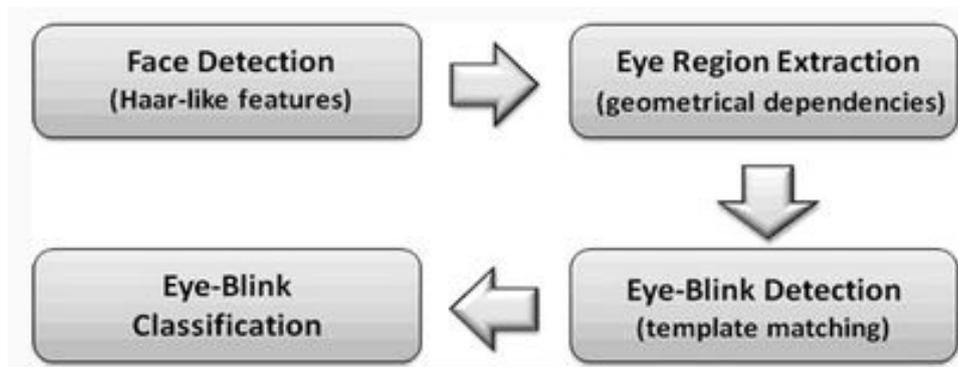


Figura 2.29: Esquema propuesto para detectar el parpadeo

La detección de la cara es esencial para desarrollar el algoritmo de la detección del parpadeo. El método utilizado para detectar la cara se puede dividir en cuatro métodos: método de las funciones para valorar la simetría de la cara[39], funciones para detectar la boca, los ojos, la nariz o el color de la piel[40], correspondencia de plantillas utilizando la correlación[41] y algoritmos usando redes neuronales[42]. Para el desarrollo del algoritmo han utilizado el de Viola y Jones[43] y modificado por Leinchart y Maydt[44].

Para detectar correctamente las caras es necesario utilizar un clasificador de cascada, denominado “Adaptative Boosting” o AdaBoost[45].

El siguiente paso que realizan es localizar la región de los ojos en la imagen.

Seguidamente, pasarán a detectar el parpadeo y clasificarlo. Para clasificarlo se basarán en los límites del parpadeo.

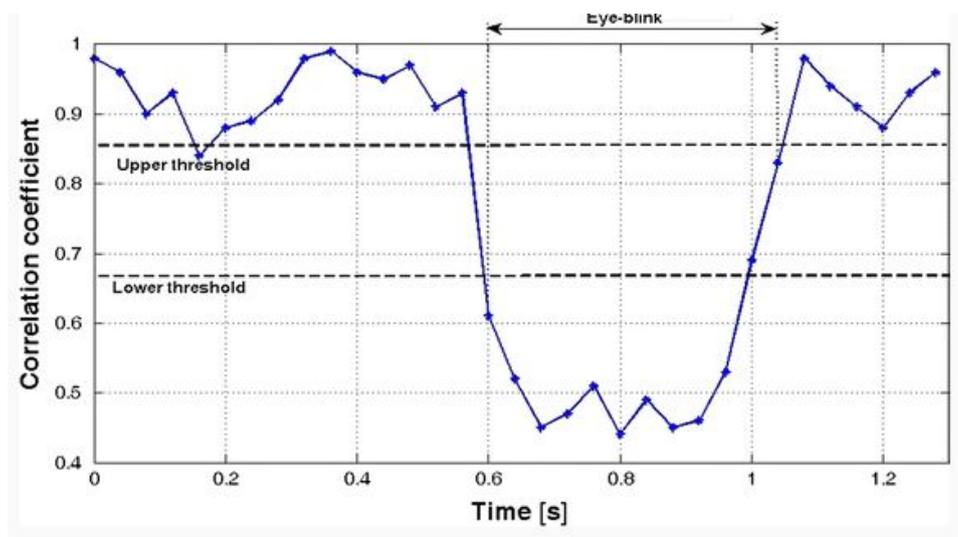


Figura 2.30: Cambio en el valor de la correlación del parpadeo

El sistema está formado por una CPU y una cámara USB. La persona debe estar situado enfrente al monitor, en el cual verán las instrucciones para el parpadeo dispuestas. Durante la sesión se podrán identificar las falsas detecciones y los parpadeos fallidos.

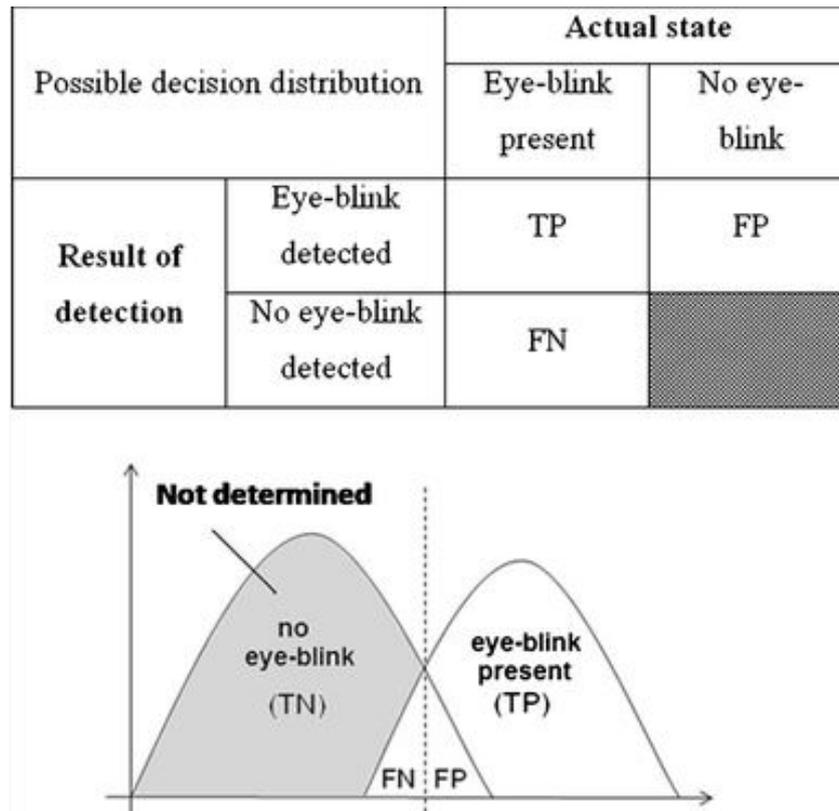


Figura 2.31: Posible distribución del detector del parpadeo

Van a utilizar las siguientes fórmulas para calcular la precisión y el recordado.

$$\text{Precision} = \frac{VP}{VP + FP}$$

$$\text{Precision medida} = \frac{TP}{TP + FP + FN}$$

$$\text{Recordado} = \frac{VP}{VP + FN}$$

Y obtuvieron los siguientes resultados con poca y mucha iluminación:

Cuadro 2.4: Medidas realizadas con buena iluminación

Medida	Precisión (%)	Recordado (%)	Precisión medida (%)
Largo tiempo de parpadeo	96.91	98.13	95.17
Poco tiempo de parpadeo	96.99	98.50	95.53
Máximo sistema de precisión	96.95	98.31	95.35

Cuadro 2.5: Medidas hechas con poca iluminación

Medida	Precisión (%)	Recordado (%)	Precisión medida (%)
Largo tiempo de parpadeo	96.91	98.13	95.17
Poco tiempo de parpadeo	96.99	98.50	95.53
Máximo sistema de precisión	96.95	98.31	95.35

La aplicación de este proyecto ha sido realizada utilizando C++ en Visual Studio y librería OpenCV.



Figura 2.32: Sistema detección

El sistema es capaz de detectar entre los parpadeos voluntarios (largos) y de los involuntarios (cortos):

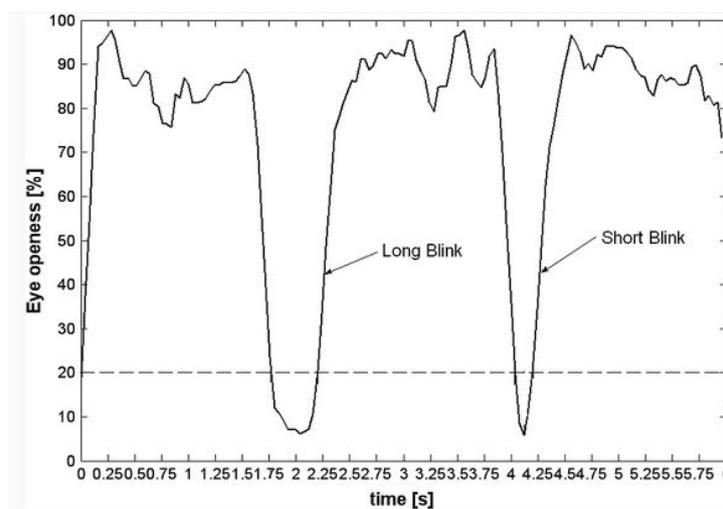


Figura 2.33: Diferenciación parpadeo detectado

Los resultados se testaron con 49 usuarios: 12 con discapacidades. Todos completaron la prueba en 10-15 minutos.

El proyecto concluye con que el 99 % de los usuarios han sido capaces de mover la interfaz del ordenador usando el sistema dispuesto en este proyecto.

2.1.13. EVA facial mouse

EVA facial mouse¹¹ es una aplicación gratuita y de código abierto que permite controlar el móvil a través de los ojos.

¹¹Enlace a la aplicación: <http://www.fundacionvodafone.es/app/eva-facial-mouse>

No sólo se puede utilizar en el móvil, también se puede utilizar en ordenador. Con el uso de este software los ojos del paciente simulan ser un puntero, por lo que mirando las distintas aplicaciones puede meterse en ellas sin necesidad de moverse. Es una aplicación muy útil para personas con necesidades especiales de movilidad.

2.1.14. Eye Tribe

Es un software móvil¹² que permite la navegación del móvil a través de los ojos. Para ello, se conectará una cámara al monitor para calibrarla. El paciente se posicionará e instalará los trackers para el calibrado. Una vez lo tengamos todo ya se puede proceder a calibrarlo. Cuando se ha realizado el calibrado aparecerá lo siguiente en el monitor:



Figura 2.34: Captura de pantalla del software EyeTribe

¹²Dirección de la aplicación: <http://theyetribe.com/theeyetribe.com/about/index.html>

Capítulo 3

Metodología y tecnologías

—*Hazlo o no lo hagas, pero no lo intentes.*
Maestro Yoda

3.1. Metodología Kanban

3.1.1. ¿Qué es?

Kanban[46] es una metodología de proyectos, de metodología ágil.

La palabra Kanban, es de origen japonés y se compone de dos términos: Kan podría traducirse como «visual» y ban podría traducirse como «tarjeta», la traducción de esta palabra podría ser «tarjeta visual». Por lo tanto, se podría decir que es un método visual de distribución del trabajo.

3.1.2. Las tres reglas de Kanban

El método Kanban se guía por tres reglas[47]:

- a). Ver el trabajo.
- b). Limitar el trabajo en curso.
- c). Optimizar el flujo de trabajo.

Ver el trabajo

Esta regla consiste en que el trabajo y su desarrollo tienen que ser mostrados en todo momento. Y la forma de mostrarlo es mediante un tablero físico. El objetivo de realizarlo así es porque:

- Entender el proceso del trabajo.
- Conocer los problemas que puedan surgir y tomar decisiones.

- Predecir los futuros procesos.

Limitar el trabajo en curso

Esta regla dictamina que hay acordar anticipadamente cada uno de los procesos para evitar cuellos de botella.

Los cuellos de botella representan el estancamiento de un proceso determinado. A continuación se va a mostrar una tabla¹³ de ejemplo de un proceso:

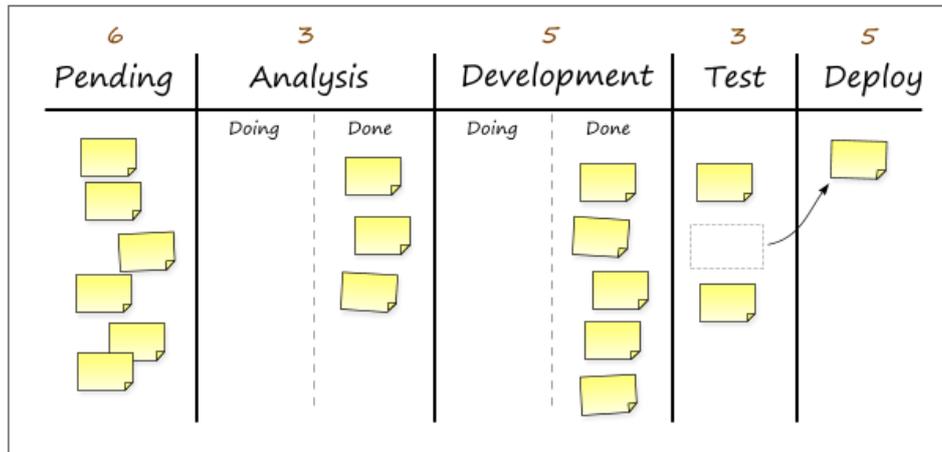


Figura 3.35: Tabla de ejemplo

En la tabla mostrada anteriormente se pueden ver tanto situaciones de cuello de botella como de trabajo fluido. En un punto posterior de este capítulo se explicará cómo funciona la metodología Kanban.

Optimizar el flujo de trabajo

Para optimizar el trabajo se requiere que la producción del mismo sea estable, continua y previsible. Midiendo el tiempo que el ciclo completo de ejecución del proyecto demandado se obtiene el CycleTime.

Al dividir el CycleTime por el WIP[48] se obtiene el “rendimiento de trabajo” también denominado Throughput, es la cantidad de ítems que un equipo puede terminar en un determinado período de tiempo.

$$\text{Throughput} = \text{CycleTime} / \text{WIP}$$

Por lo tanto, la optimización consistirá en:

- Minimizar el CycleTime.
- Maximizar el Throughput.
- Una variabilidad mínima entre CycleTime y Throughput.

¹³Dirección web de la tabla: <https://medium.com/@marvin.soto/la-metodolog%C3%ADa-kanban-6ab002502831>

3.1.3. ¿Cómo funciona?

Lo primero que se deberá realizar es un tablero[49] visible y accesible. En las columnas se anotará el estado del flujo de las tareas y se determinará el estado de cada proyecto. Es un tablero continuo en el que las tarjetas no se desplazan, sino que a medida que se avanza por él, las nuevas funcionalidades, mejoras o incidencias se acumulan al inicio. Así, se pueden desplazar por las secciones según las distintas prioridades.

Fases del ciclo de producción

Este método está basado en dividir el trabajo en distintas partes para que haya un desarrollo incremental.

Cada tarjeta que se utilice para aplicar el método se añade al tablero en la fase que corresponda. Seguidamente, se va añadiendo la información. Y finalmente, se da cualquier incidencia sobre la misma.

Así, el trabajo queda distribuido por secciones donde se ven claramente las prioridades y objetivos.

3.1.4. Ventajas de la metodología Kanban

En este punto vamos a describir las ventajas[50] de la metodología Kanban y en el siguiente punto se comparará con la metodología Scrum:

- a). **El trabajo a vista de águila:** Todo el mundo ve de una manera visual el trabajo en general, con sus tiempos de entrega y con una gran fiabilidad.
- b). **Evitar tareas inútiles:** Se evita la sobreproducción y la limitación de los recursos.
- c). **Control del trabajo:** El tiempo de producción es más rápido, por lo que se reduce el esfuerzo y se mejora la planificación. Hay un control mayor del entorno de trabajo.
- d). **Tiempo de respuesta mayor:** Si surge algún imprevisto hay una capacidad de respuesta que permite arreglarlo.

3.1.5. Comparación con la metodología Scrum

En este punto se va a mostrar una tabla[51] comparando las dos metodologías. De estas dos metodologías se van a comparar:

- Los roles.
- Los tiempos de entrega.
- La priorización y la delegación.
- Modificaciones.
- Medida de la productividad.
- Aplicaciones.

Cuadro 3.6: Comparación de Kanban y Scrum

	Kanban	Scrum
Roles	No hay roles definidos. Figura del encargado del trabajo.	Cada componente del grupo tiene un rol definido.
Tiempos de entrega	Los productos y los procesos están en constante movimiento.	Las entregas son determinadas por sprints y se dan unos tiempos de entrega que deben ser completados.
Priorización y delegación	Sistema pull: Se ponen nuevas tareas una vez se han completado las anteriores.	También utiliza el sistema pull, pero para cada iteración.
Modificaciones	Permite cambios durante el desarrollo.	No es recomendable hacer cambios durante los sprints.
Medida de la productividad	La productividad se mide por un ciclo de tiempo.	La productividad se mide mediante los sprints.
Aplicaciones	Proyectos con prioridades variadas.	Proyectos con prioridades estables, que no cambien en el tiempo.

3.1.6. ¿Por qué escoger Kanban?

Es un tipo de metodología donde se puede ver claramente cómo se ha ido desarrollando el proyecto, ya que, se ve bien que tarea hay que priorizar y permite administrar el tiempo de una manera óptima.

3.2. Desarrollo del proceso y plan del proyecto

3.2.1. Aplicación Taiga.io

Es una aplicación de software libre que se utiliza para organizar los proyectos o trabajos de una manera cómoda y accesible con el usuario.

La hemos elegido por ser de software libre y por que nos permite organizar el trabajo mediante una metodología Kanban.

Nuestro proyecto se ha dispuesto de la siguiente forma:

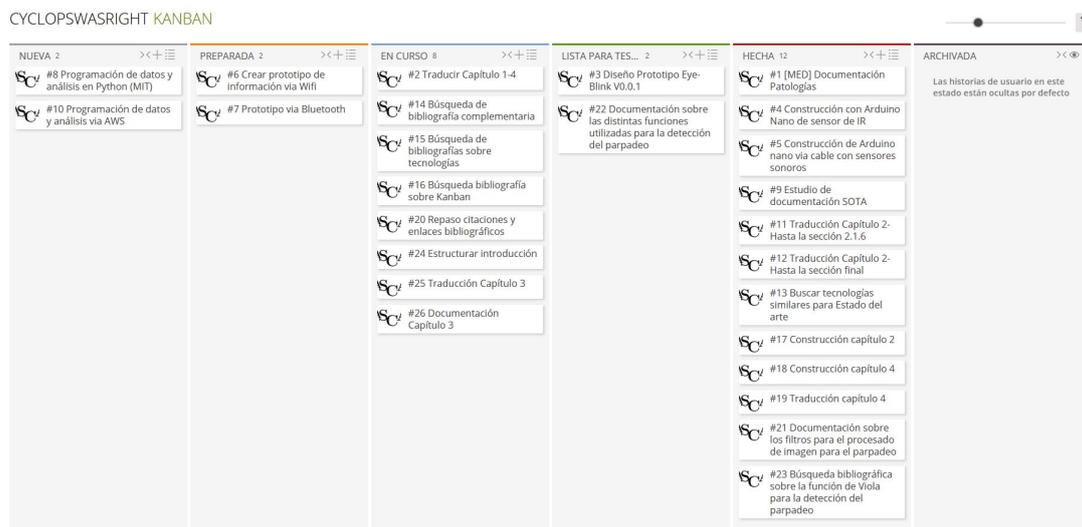


Figura 3.36: Captura del proyecto

En nuestro proyecto encontramos los siguientes apartados:

- Búsqueda bibliográfica:** En este apartado nos hemos dedicado a la búsqueda bibliográfica de tecnologías similares para el estado del arte y las tecnologías.

- **Construcción de capítulos:** En este apartado una vez tenemos la búsqueda bibliográfica, leemos la información bibliográfica y la referenciamos en los capítulos.
- **Traducción:** En este apartado nos hemos dedicado a traducir los distintos capítulos del trabajo.
- **Versiones:** En este apartado se prueban las distintas versiones del trabajo.
- **Traducción de capítulos:** En este apartado se fueron traduciendo los distintos capítulos.
- **Presentación de prototipos:** En los distintos apartados se han ido desarrollando los distintos prototipos.
- **Simulación de circuitos:** Se han simulado los distintos circuitos utilizados en el proyecto.
- **Repercusión:** En este apartado se aportó la repercusión que va a tener este proyecto en otros ámbitos.

3.3. Tecnologías

3.3.1. Taiga

Es una aplicación de software libre¹⁴ que nos permite organizar nuestro proyecto mediante la metodología Kanban. Lo primero que vemos nada más entrar al enlace es lo siguiente:

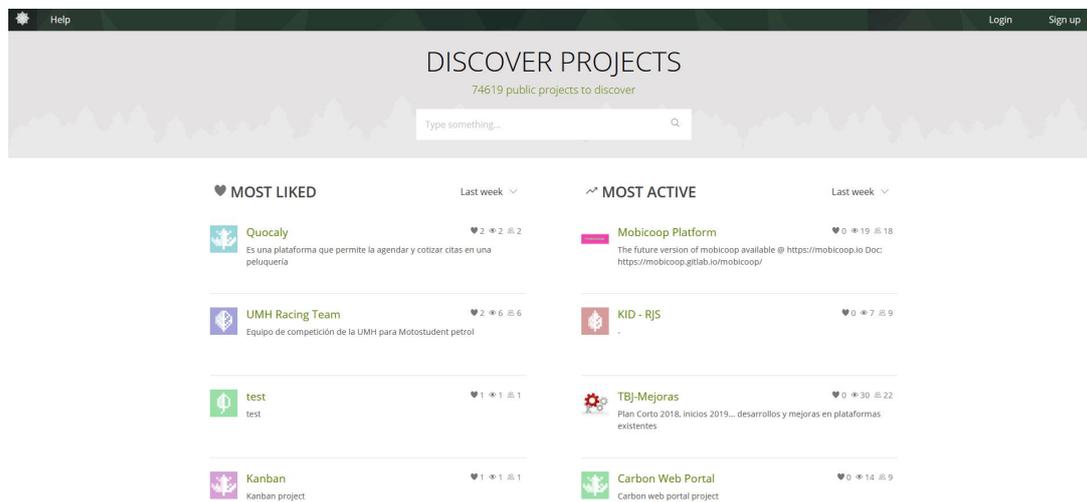


Figura 3.37: Captura de la aplicación

Nos permitirá registrarnos, y una vez que administremos nuestro proyecto veremos lo siguiente:

¹⁴Enlace a la aplicación: <https://tree.taiga.io>

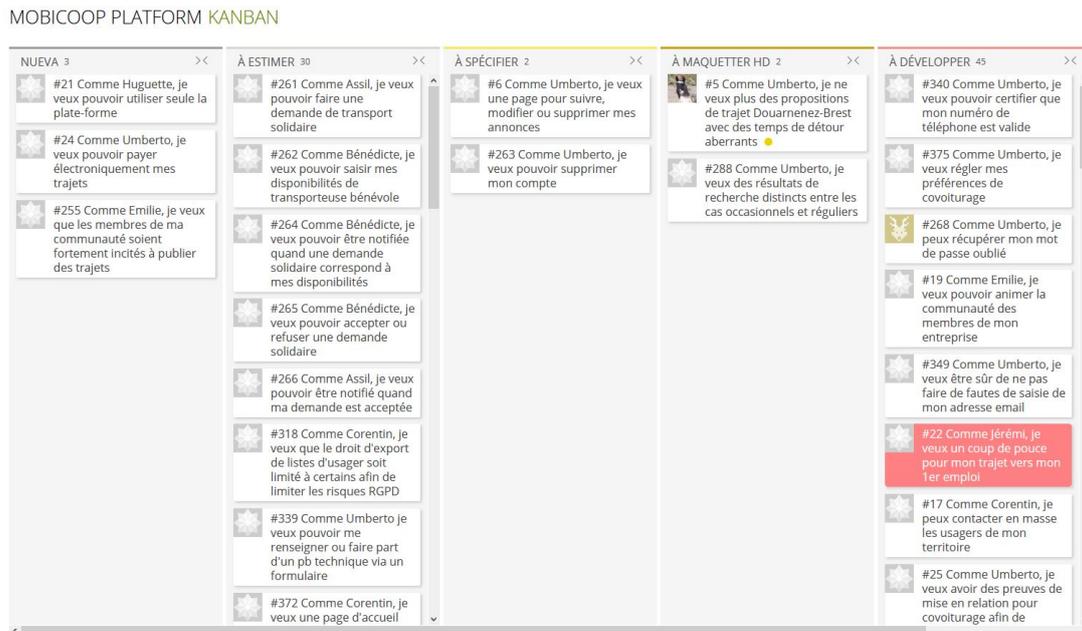


Figura 3.38: Captura del trabajo

Nos permitirá organizar el trabajo en columnas, que son las siguientes:

- **Nueva:** Para empezar a colocar nuestras tareas.
- **Preparada:** Tareas preparadas.
- **En curso:** Tareas que se están llevando a cabo.
- **Listas para testear:** Tareas terminadas, que se tienen que comprobar.
- **Hecha:** Tareas ya comprobadas.

3.3.2. Latex-Overleaf

L^AT_EX es un formato de texto de software libre en texto plano que nos permite programar nuestros textos.

Por otro lado, tenemos Overleaf¹⁵, que es la plataforma online donde podemos utilizar L^AT_EX. Una vez nos registremos en la plataforma veremos lo siguiente:

¹⁵Dirección de Overleaf: <https://es.overleaf.com>

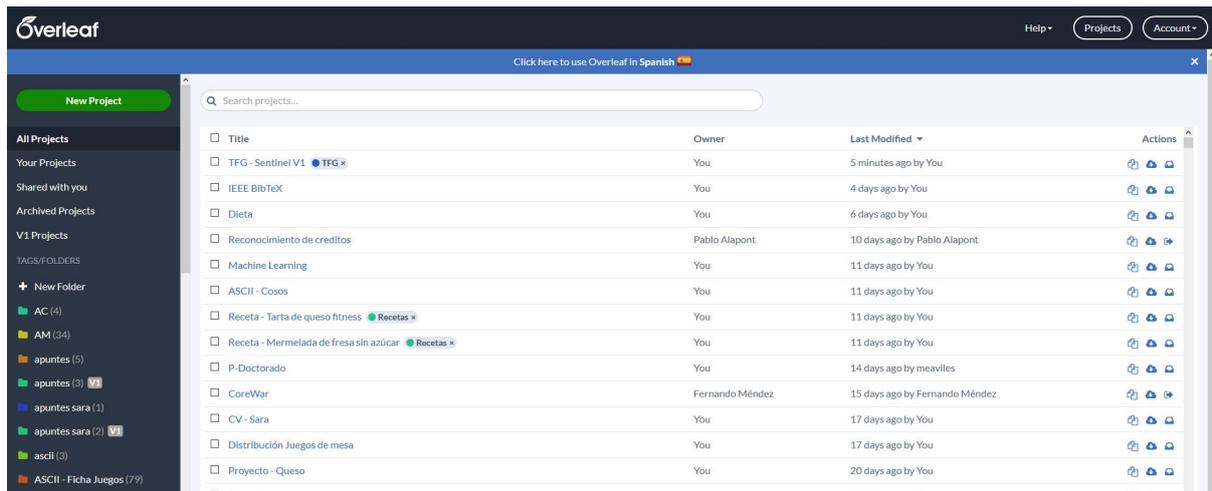


Figura 3.39: Captura de Overleaf

Nos permitirá editar y crear nuevos proyectos todo en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

3.3.3. Python

Python¹⁶ es un lenguaje de programación, administrado por Python Software Foundation License y que es capaz de manejar múltiples lenguajes.

Se ha utilizado para la programación del texto en donde se realiza el informe.

3.3.4. Arduino Studio

Arduino Studio¹⁷ es una aplicación de software libre cuyo entorno es escrito en Java y se basa en el procesamiento de Arduino por software libre.

¹⁶Dirección de Python:<https://www.python.org>

¹⁷<https://www.arduino.cc/en/Main/Software>



Figura 3.40: Captura de pantalla de Arduino Studio

3.3.5. AWS

AWS¹⁸ es una plataforma online de Amazon que ofrece servicios de inteligencia artificial o el internet de las cosas.

3.3.6. Ultimaker Cura

Ultimaker Cura¹⁹ es un software de impresión 3D, gratuito, que ofrece una impresión personalizada de las distintas plantillas utilizadas para este proyecto.

3.3.7. ThingSpeak

ThingSpeak²⁰ es una aplicación en la nube del internet de las cosas, que nos permitió guardar los datos del proyecto.

3.3.8. CircuitLab

CircuitLab²¹ es una herramienta online de previo registro, que nos ayudó a construir y a simular los circuitos.

¹⁸<https://aws.amazon.com/es/what-is-aws/>

¹⁹<https://ultimaker.com/es/software/ultimaker-cura>

²⁰<https://thingspeak.com/>

²¹<https://www.circuitlab.com/>

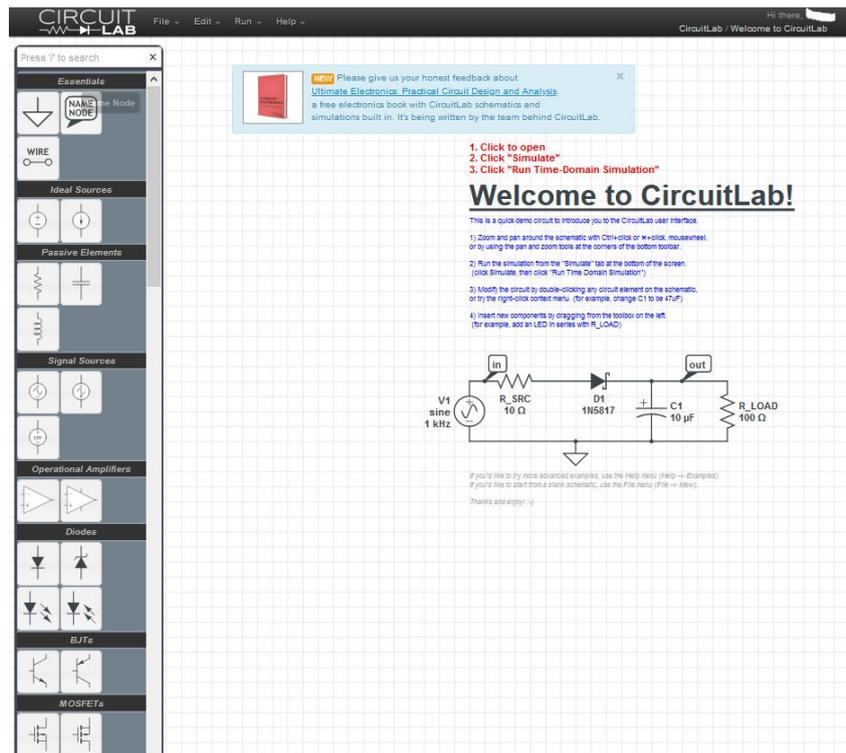


Figura 3.41: Captura CircuitLab

3.3.9. EasyEDA

EasyEDA²² es una plataforma online, que nos permite simular los circuitos con Arduino y otros sistemas.

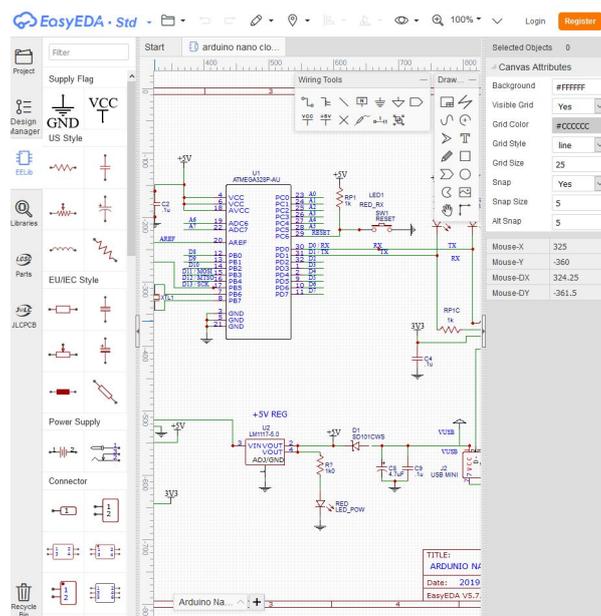


Figura 3.42: Captura de EasyEDA

²²<https://easyeda.com/es>

Parte B

Diseño y arquitectura

Capítulo 4

Diseño de la solución

—*El entrenamiento no es nada. ¡La voluntad lo es todo! La voluntad de actuar*
Ra's al Ghul

4.1. La importancia del parpadeo

Los párpados²³ son membranas que cubren el exterior de los ojos y tienen el efecto de proteger los ojos. Las pestañas también ayudan a proteger los ojos. La estructura del párpado incluye los siguientes elementos.:

- Piel, es la parte más fina.
- El borde libre es una zona de transición cutáneo-mucosa donde se encuentra un margen anterior cutáneo y una cara plana con dos zonas, una hidrófoba y una húmeda, y un margen posterior que elimina todas las irregularidades en el parpadeo.
- Conjuntiva palpebral.

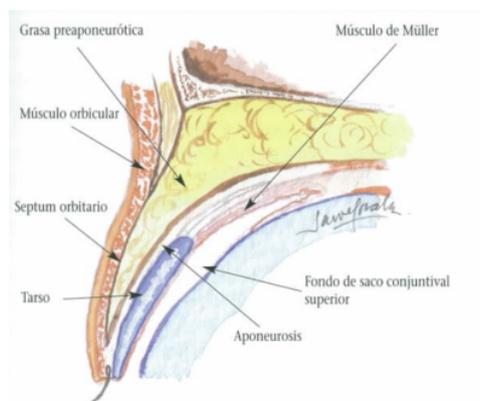


Figura 4.43: Estructura párpados

²³Información obtenida de un artículo sobre el parpadeo: <https://www.opticaayora.es/la-importancia-del-parpadeo>

Mediante el parpadeo se consigue la difusión de la lágrima y su distribución homogénea sobre la superficie ocular. Normalmente la tasa de parpadeo es de entre 5 y 15 veces por minuto pero esta frecuencia puede verse afectada por acciones o factores ambientales. La disminución o aumento de la misma nos debe avisar de alguna patología ocular. Las funciones del parpadeo son las siguientes:

- **Función defensiva:** Los anticuerpos de la lágrima atacan a las bacterias y son arrastrados como desechos en el parpadeo.
- **Función metabólica:** Mediante el parpadeo se le suministra oxigenación a la córnea. En usuarios de lentes de contacto es menor.
- **Función óptica:** La lágrima es una superficie homogénea en la cara anterior que en el caso que no haya suficiente parpadeo empeora la calidad visual.
- **Función humectante:** Impide que se seque la córnea.
- **Función protectora:** Absorbe parte de los rayos UV de la luz solar.
- **Función refleja:** Impide que lleguen cuerpos extraños al globo ocular.

Parpadear es un acto semi involuntario y es necesario un parpadeo completo en el que el párpado superior sea capaz de subir y bajar completamente, y el párpado inferior debe realizar un movimiento oblicuo para llevar los desechos hacia el punto lagrimal.

Realizar un parpadeo incompleto puede producir sequedad ocular y mayor tasa de infecciones oculares.

4.2. Patologías a detectar

Con nuestra aplicación queremos prever patologías relacionadas con la frecuencia del parpadeo y elementos ajenos al ojo, como la iluminación o la temperatura.

4.2.1. Luminosidad y calidad visual

Podemos encontrar cuatro tipos de deslumbramientos²⁴:

- Iluminación de semáforos o luces de tráfico durante la noche: Esta iluminación provoca fatiga y desorientación.
- Cambios de iluminación: Causa fatiga ocular.
- Iluminación incapacitante: Como su propio nombre indica no permite ver bajo esta iluminación.
- Iluminación originada por la luz de Sol directa: Este tipo de iluminación según el tiempo de exposición puede causar ceguera. En caso de mirar directamente un eclipse solar, la ceguera es inmediata ya que, se queman los fotorreceptores en retina.

²⁴Información obtenida de una base de datos de la UCM:<http://0-search.proquest.com.cisne.sim.ucm.es.bucm.idm.oclc.org/docview/308109129?accountid=14514>

Estos tipos de deslumbramientos afectan en general a toda la población, pero la población que más los pueden sufrir son los diabéticos y los hipertensos.

Para prevenir patologías asociadas a la iluminación, la población debería utilizar gafas de sol con filtros que protejan del UV y adquirirlas en establecimientos en los que se atestigüe su eficacia.

Uso de cristales polarizados



Figura 4.44: Ejemplo de lentes polarizadas

Una alternativa para evitar estos deslumbramientos es el uso de cristales polarizados, con ellos sólo pasa la luz útil permitiendo una visión sin reflejos.

Este tipo de cristales es recomendable utilizarlos mientras se conduce porque reduce el cansancio, mejora la capacidad de concentración y disminuye el tiempo de reacción.

Pero este tipo de cristales tienen sus limitaciones que son las siguientes²⁵:

- No es recomendable ponerlas en superficies horizontales.
- Impiden ver algunas pantallas tipo LCD.
- Están limitadas en graduación y colores.
- No es recomendable para esquiadores ni pilotos de avión.

4.2.2. Temperatura y sequedad ocular

En todo nuestro cuerpo²⁶ tenemos receptores térmicos del frío, especialmente a nivel ocular. En algunos roedores existen “detectores de humedad”.

²⁵Ventajas y limitaciones de lentes polarizadas:<https://tienda.linazasoro-optika.eus/lentes-polarizadas-ventajas-y-limitaciones/>

²⁶Información obtenida sobre un estudio de la influencia de la temperatura en la superficie ocular:<https://nuevaalcarria.com/articulos/la-sensacion-de-frio-clave-en-la-sequedad-ocular>

Los receptores del frío en la córnea siguen el canal TRPV1 y TRPA1. Cuando estos canales son expuestos a ambientes cálidos la tasa de lagrimeo basal baja.

4.2.3. Ojo seco

Ojo seco²⁷ se define como una enfermedad de la superficie ocular multifactorial caracterizada por la pérdida de la homeostasis de la película lagrimal y la hiperosmolaridad de la superficie ocular según Dry EyeWorkShop (DEWS II) 2017.

Clasificación

Se dividen en dos tipos:

- **Ojo seco acuodeficiente:** Disminución de la lágrima y aumento de la osmolaridad. Se asocia a tracoma, sarcoidosis, queratitis neutrófica y diabetes. Puede estar o no asociado al síndrome de Sjögren.
- **Ojo seco evaporativo:** Pérdida excesiva de lagrime en pacientes con secreción normal. Asociado a alteraciones en las glándulas de Meibomio, trastornos en el parpadeo e hipertiroidismo.

Etiología

La hiperosmolaridad provoca un daño epitelial y una respuesta inflamatoria que reduce la producción de mucina, éste déficit produce inestabilidad de la película lagrimal que se evapora con más facilidad.

Otro factor a tener en cuenta son las alteraciones neurosensoriales que afectan a la superficie ocular y a las glándulas lagrimales.

Como factores de riesgo podemos destacar a los operados de cirugía corneal (PRK y LASIK), el uso excesivo de pantallas de ordenador, el sexo femenino, edad avanzada y la exposición a ambientes muy secos o ventosos.

Síntomas

Suelen aumentar a lo largo del día y son los siguientes:

- Picor y escozor.
- Sensación de cuerpo extraño.
- Dificultad para abrir los ojos al despertar.
- Visión borrosa que puede mejorar con el parpadeo.
- Exceso de lágrima.
- Fotofobia si cursa con queratitis.

²⁷Definición de ojo seco según la Dry EyeWorkShop:<http://www.osanet.euskadi.eus/cevime/es>

Signos

- Queratitis punctata: Localizada en el tercio inferior de la córnea.
- Edema corneal.
- Hiperemia conjuntival.
- Blefaritis con irregularidad en el borde y enrojecimiento.

4.3. Explicación de términos ópticos/optométricos

- **Conjuntiva:** Esta mucosa cubre la parte posterior del párpado y la parte posterior de la córnea.
- **Patología:** Una parte de la medicina utilizada para estudiar los trastornos anatómicos y fisiológicos de los tejidos y órganos enfermos, así como los síntomas y signos de enfermedades y sus causas.
- **Deslumbramiento:** La luz intensa o el deslumbramiento causan pérdida de visión instantánea.
- **Fatiga ocular:** Debido a un trabajo excesivo se realizan cambios de acomodación constantes o por cambios de luz.
- **Luz polarizada**²⁸: Rayos de luz que vibran en un solo plano. La luz que generalmente se emite es una mezcla de ondas de luz que vibran en todas las direcciones. Esta luz se puede polarizar por reflexión, birrefringencia, absorción selectiva o difusión. La polarización permite distinguir cambios en la estructura y composición del material que no se pueden discernir bajo la luz ordinaria. Los cambios de apariencia experimentados por la muestra cuando se ve con luz polarizada se pueden usar como identificación.
- **Receptores térmicos:** Las terminaciones nerviosas ubicadas en la piel pueden enviar información al sistema nervioso central.
- **Ojo seco**²⁹: Es una enfermedad multifactorial, compleja y crónica que afecta la superficie del ojo y causa molestias, problemas visuales y, en algunos casos, daño corneal y conjuntival. Por lo tanto, aunque su nombre puede causar malentendidos, su significado va más allá de “sin lágrimas”.
- **Osmolaridad:** La concentración de partículas osmóticas activas en la solución, expresada como la presión osmótica o milimolar por litro de disolvente..
- **Tracoma:** Conjuntivitis granular causada por virus específicos, prevalente en ciertos países o regiones populares.
- **Sarcoidosis:** Es una enfermedad caracterizada por la acumulación de una pequeña cantidad de células inflamatorias (granulomas) en cualquier parte del cuerpo (más

²⁸Definición de luz polarizada: <https://glosarios.servidor-alicante.com/optica/luz-polarizada>

²⁹Definición de ojo seco según la IMO:<https://www.imo.es/es/ojo-seco>

comúnmente en los pulmones y los ganglios linfáticos). Pero también afecta los ojos, la piel, el corazón y otros órganos.

- **Queratitis neutrófica:** Es una enfermedad degenerativa corneal rara causada por la sensibilidad corneal disminuida o ausente.
- **Diabetes:** Es una patología causada por altos niveles de glucosa en la sangre.
- **Síndrome de Sjögren:** Es una enfermedad autoinmune. Esto significa que el sistema inmunitario atacará por error ciertas partes de su cuerpo. En el síndrome de Sjögren, ataca las glándulas que producen lágrimas y saliva. Esto puede causar sequedad en la boca y ojos secos.
- **Hipertiroidismo:** Hiperfunción tiroidea.
- **Fotofobia:** Es la intolerancia anormal a la luz. Esto es muy común en personas con albinismo, y también puede deberse a enfermedades relacionadas con los ojos o el sistema nervioso.
- **Queratitis:** Es la inflamación de la córnea, la estructura más frontal y transparente del globo ocular, que puede ser causada por una variedad de razones. Si afecta solo la parte externa más común (epitelio), se llama queratitis superficial y generalmente es curable sin secuelas. Por el contrario, si afecta las capas más profundas, es una queratitis ulcerosa que, aunque es poco frecuente, puede ser muy grave. Cuando deja una cicatriz en la córnea (leucocitoma), la visión se dañará gravemente.
- **Queratitis punctata:** Es una enfermedad ocular causada por la muerte de un pequeño grupo de células en la superficie de la córnea (el iris y la capa transparente frente a la pupila).
- **Edema corneal:** Causado por defectos endoteliales secundarios a diversas patologías (inflamación, trauma, cirugía, hipoxia). Aparece de la matriz y puede ser focal o difusa, aguda o crónica, y puede revertirse después de que la causa haya desaparecido.
- **Hiperemia conjuntival:** Puede estar relacionado con alergias, daños mecánicos en la córnea o infecciones bacterianas o virales. Las personas con hiperemia conjuntival pueden experimentar ardor, picazón y aumento de la presión intraocular.
- **Blefaritis:** Inflamación párpados.

4.4. Cómo escanear la cara con el sistema Kinect

Para realizar el escaner de la cara hemos utilizado el sistema Kinect. Para ello hemos requerido:

- Un ordenador con la tarjeta gráfica actualizada.
- Sensor Kinect de la Xbox 360.
- Adaptador USB para conectar el Kinect al PC.
- Software de captura: Skanect[52].

4.5. Construcción de la herramienta

El montaje de la impresora se encuentra adjuntada en el anexo, en el capítulo 1.2 del anexo.

4.5.1. Escáner 3D con kinect

Para realizar el proceso de escaneo de la parte superior para imprimirla en nuestra impresora necesitamos:

- Kinect X360.
- Fuente de alimentación para el Kinect.
- Adaptador Kinect al ordenador.
- Ordenador con gran capacidad para procesar las imágenes del Kinect.

Para el software necesitamos:

- Kinect SDK 1.8³⁰.
- Kscan3D³¹.
- Meshmixer³².

Y el procedimiento será el siguiente:

- a). Conectar el Kinect a la corriente y al ordenador.
- b). Esperar hasta que se ilumine para verificar que el Kinect está conectado al ordenador.
- c). Conectarlo vía usb al ordenador.
- d). Esperar hasta que Kscan3D reconozca el Kinect.
- e). Es importante disponer de buena iluminación para realizar el scanner.
- f). Realizar las capturas despacio y sin moverse.
- g). Poner el kinect a unos 70 cm-1 metro.
- h). El programa permite ajustar el rango para quitar el fondo.
- i). En el momento que le damos a escanear, movemos al modelo girando poco a poco para ir capturando cada una de las imágenes. Es importante mantener la postura.
- j). Inspeccionamos las capturas para eliminar los pixeles erroneos.
- k). Una vez depuramos todas las imágenes, le damos el check y empieza el reconocimiento.

³⁰Dirección para descargarlo:<https://www.microsoft.com/en-us/download/details.aspx?id=40278>

³¹Dirección para descargarlo:<https://kscan3d.software.informer.com/1.2/>

³²Dirección para descargarlo:<http://www.meshmixer.com/download.html>

- 1). Finalmente, con build construirá la imagen. El programa suavizará las partes que han quedado extrañas. Combinaremos las imágenes con combine y finalizaremos para empezar a crear el proyecto.

Por otro lado, con Meshmixer podremos darle un acabado para imprimirlo:

- a). Analizaremos e inspeccionaremos para rellenar las partes vacías de la figura.
- b). La guardamos y la importamos e imprimimos.

El esquema que tiene la conexión al Kinect con el ordenador es la siguiente:

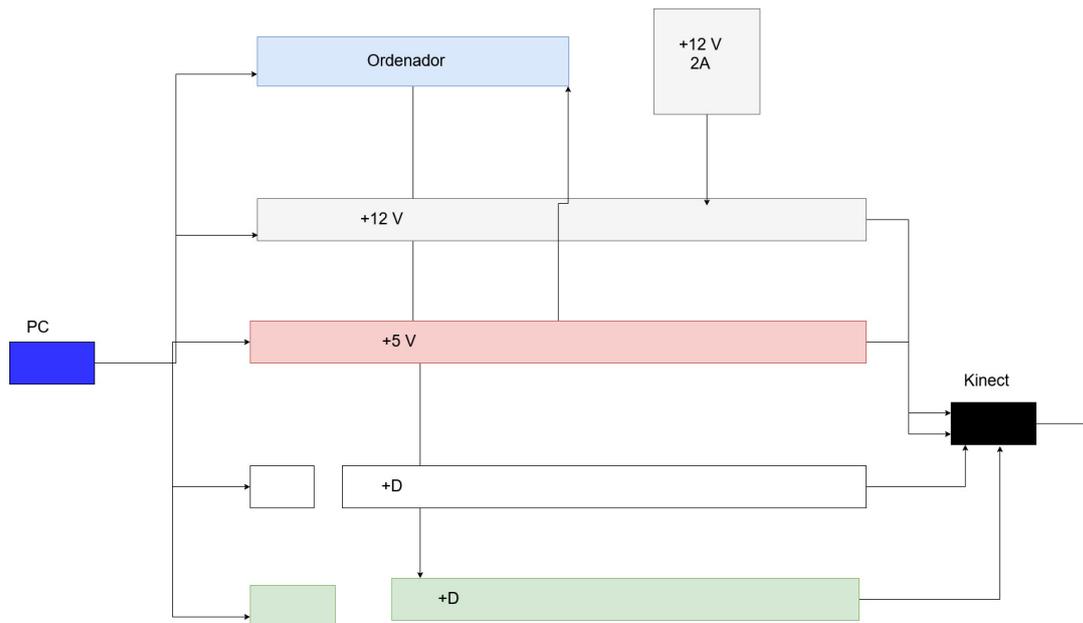


Figura 4.45: Conexión Kinect al ordenador

Seguidamente, procesará las imágenes de la siguiente forma:

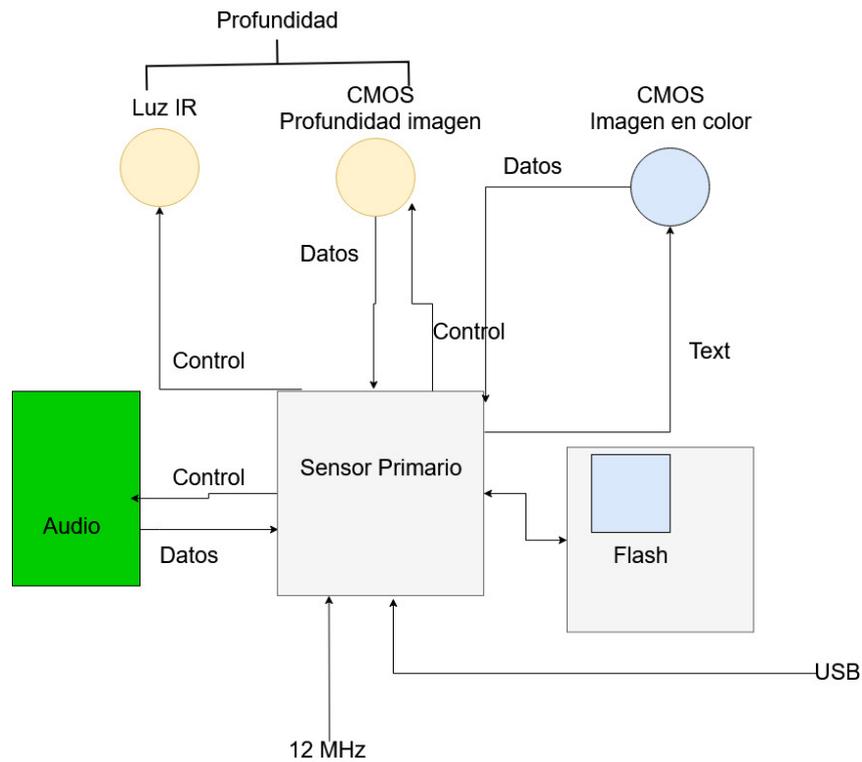


Figura 4.46: Detección imágenes

Y dentro del Kinect:

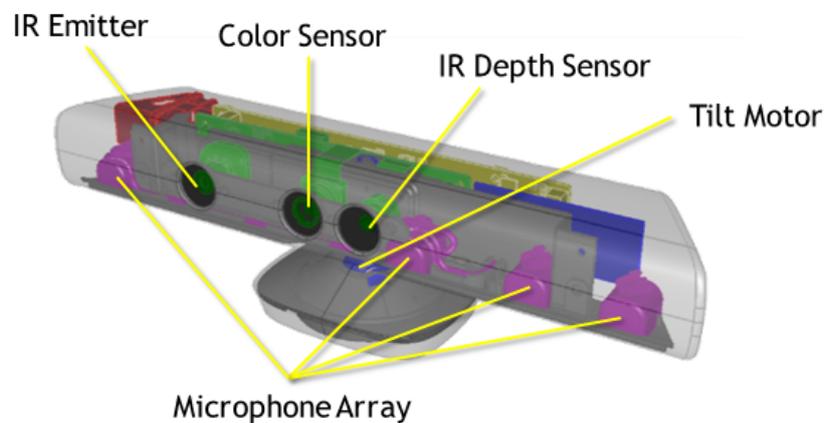


Figura 4.47: Estructura dentro del Kinect

- **Emisor IR:** Para que detecte las distintas figuras.
- **Sensor del color:** Para detectar las variaciones en el calor mediante distintos colores.
- **Sensor de profundidad infrarrojo:** Para detectar la profundidad entre los objetos.
- **Motor tilt:** Para que funcione independiente de la consola.
- **Array de micrófonos:** Para que detecte las distintas voces del entorno.

4.6. Hardware diseñado y estudio

4.6.1. Efecto Doppler

Descripción

El efecto Doppler³³ fue descrito por primera vez por Andreas Doppler en 1842 al darse cuenta que el sonido de la locomotora de un tren se hacía más agudo al acercarse y más grave al alejarse. Posteriormente, este fenómeno fue descrito por Armand Hippolyte Louis Fizeau en 1848, quien utilizó este mismo efecto para describir las ondas electromagnéticas. Por lo que este efecto se pasó a denominar efecto Doppler-Fizeau.

Vamos a describir el efecto Doppler y su aplicación en el trabajo.

Su principio general es el siguiente:

- Si la fuente se acerca al observador, la frecuencia que se mide es mayor que la medida desde el lado del observador. Este fenómeno se denomina “corrimiento hacia el azul”.
- Si la fuente se aleja del observador, su frecuencia es menor que la del emisor. Este fenómeno se denomina “corrimiento hacia el rojo”.

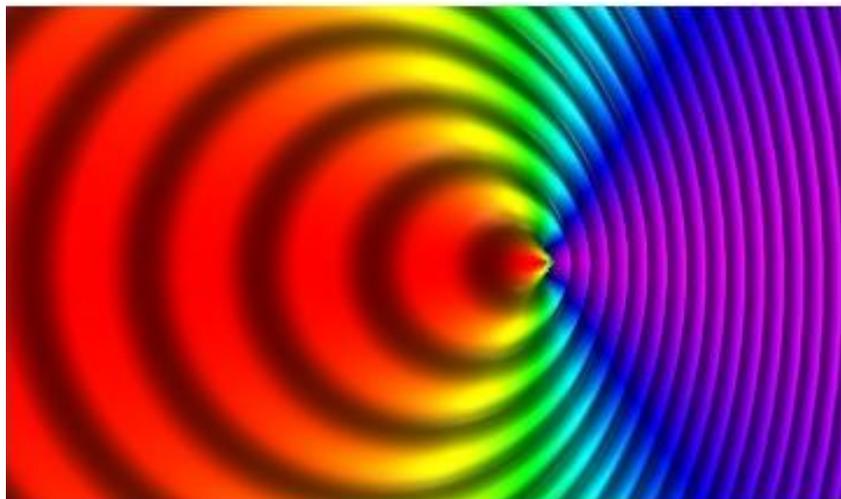


Figura 4.48: Vista efecto Doppler³⁴

Posteriormente, vamos a explicar cómo influye el movimiento del foco emisor respecto al receptor para el efecto Doppler:

Fuente y observador en reposo: No se va a producir efecto Doppler porque los frentes de ondas están separados unos de otros.

Fuente en movimiento y observador en reposo: En este caso vamos a suponer que la fuente está en movimiento a una velocidad v , por lo que los frentes de onda que se generarán tendrán una longitud de onda más pequeña, por lo que encontraremos un efecto Doppler según nos acercamos a la fuente.

$$f' = f \cdot \frac{v}{v \pm v_F} \quad (4.1)$$

³³http://laplace.us.es/wiki/index.php/Efecto_Doppler

- f, f' : Frecuencia emitida por la fuente y frecuencia percibida por el emisor. Su unidad son los Hz.
- v : Velocidad de propagación de la onda en el medio.
- v_F : Velocidad de la fuente.
- \pm : Si es negativo se acerca a la fuente. Si es positivo se aleja de la fuente.

Fuente en reposo y observador en movimiento: Ahora es el observador el que se dirige a una fuente con una velocidad v . Por lo que los frentes de onda variarán respecto al observador se acerque o aleje de la fuente.

$$\lambda' = \lambda - d_{F_0F_T} \rightarrow \lambda' = \lambda - v_F \cdot T \quad (4.2)$$

$$\lambda' = \lambda - v_F \cdot T \rightarrow \frac{v}{f'} = \frac{v}{f} - v_F \cdot \frac{1}{f} \rightarrow f \cdot v = f' \cdot (v - v_F) \rightarrow f' = f \cdot \frac{v}{v - v_F} \quad (4.3)$$

$$f' = f \cdot \frac{v}{v + v_F} \rightarrow f' < f \quad (4.4)$$

$$f' = f \cdot \frac{v \pm v_R}{v} \quad (4.5)$$

- f, f' : Frecuencia emitida por la fuente y frecuencia percibida por el emisor. Su unidad son los Hz.
- v : Velocidad de propagación de la onda en el medio.
- v_F : Velocidad de la fuente.
- \pm : Si es negativo se acerca a la fuente. Si es positivo se aleja de la fuente.

Aplicaciones

Este efecto ha sido aprovechado en muchos ámbitos, por ejemplo, en medicina, optometría, radares, astrofísica o el mismo trabajo.

- **Medicina:** Para medir la obstrucción en las válvulas cardíacas. Así como ecoradiografías o ecografías.
- **Astrofísica:** Se utiliza para ver si una estrella se aleja o se acerca. Se relaciona el color de la estrella según la distancia.
- **Radares:** Se utilizan para medir la velocidad a la que se aleja o se acerca el coche del radar.
- **Optometría:** Se utiliza para realizar ecografías a nivel ocular y otras técnicas como la OCT.

Aplicación en el trabajo

Nos vamos a basar como interacciona el frente de ondas del emisor con el ojo y cómo nos devuelve el frente de ondas reflejado el ojo al sensor.

Para ello, nos vamos a basar en el principio de la OCT. La OCT presenta las siguientes características:

- Alta resolución.
- Se puede usar en tejidos dispersivos.
- Altos tiempos de adquisición.
- Técnicas no invasivas.

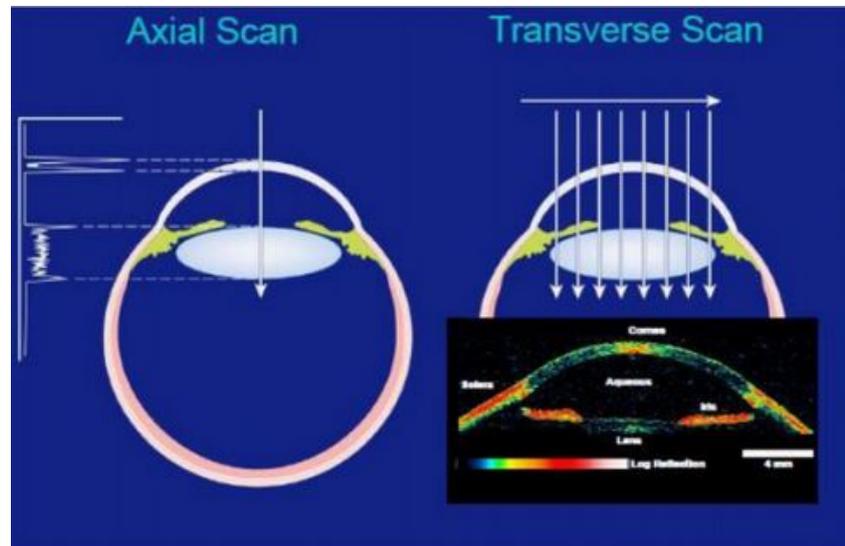


Figura 4.49: Escaneo OCT

Con este principio, vamos a detectar cuando el ojo se cierra y se abre. Pero le añadiremos un sensor infrarrojo para la reflexión del infrarrojo en el ojo nos ayudará a detectar cuando el ojo parpadea.

4.6.2. Implementación del efecto Doppler en el trabajo

En este apartado se va a explicar cómo es el sensor infrarrojo para detectar el parpadeo.

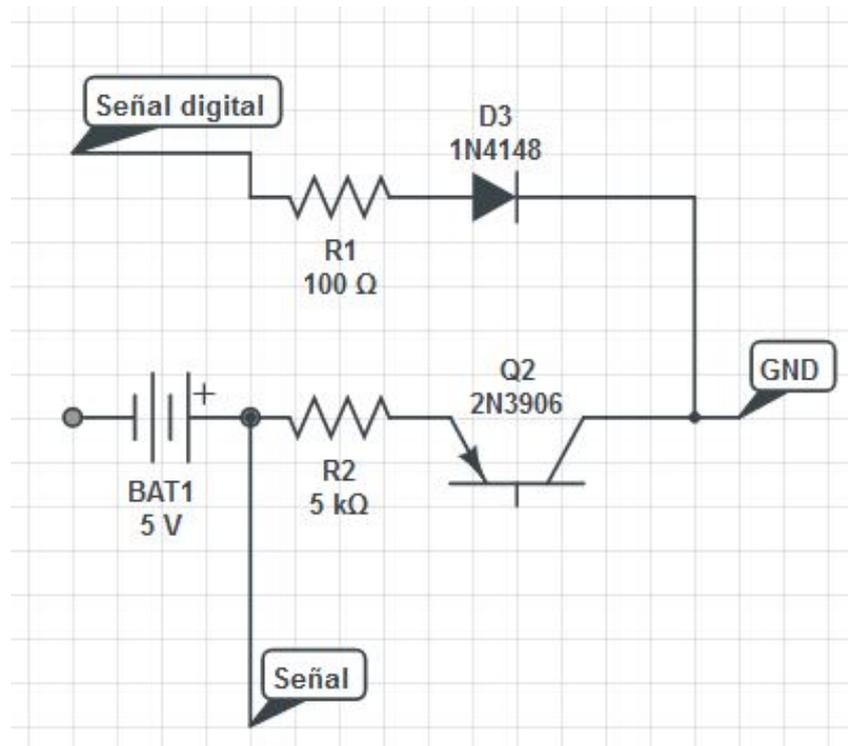


Figura 4.50: Sensor infrarrojo

En el sensor infrarrojo encontramos los siguientes componentes:

- Dos resistencias.
- Un fotodiodo.
- Un BTJ.
- Un GND.

Con este sensor infrarrojo haremos llegar una señal al ojo y esta señal se nos será devuelta en el detector. Con ello, se podrá detectar cuando el ojo parpadea. Finalmente, veremos cómo se monta el sensor junto con el detector.

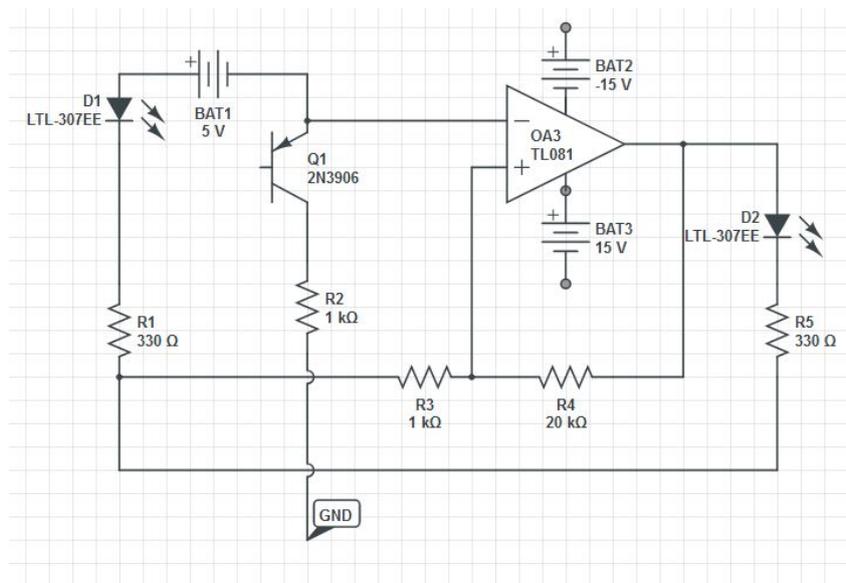


Figura 4.51: Detector y sensor IR

La señal que realiza el detector le llegará y la reflejará el ojo de la siguiente forma:

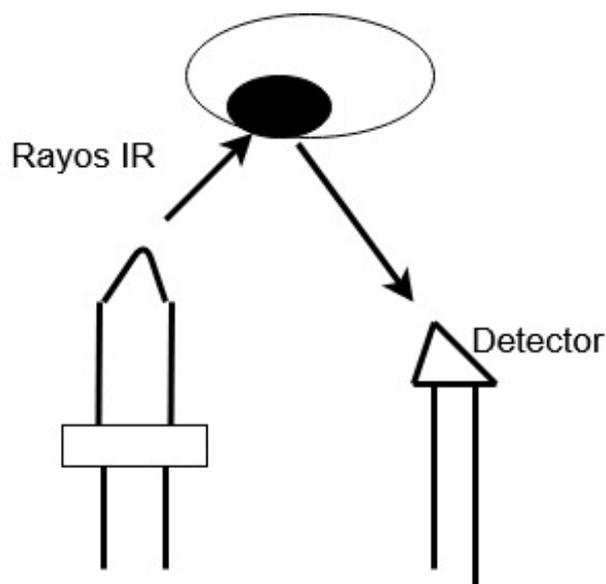


Figura 4.52: Señal detector-ojo

4.7. Fases del proyecto

4.7.1. Investigación y estudio

Investigación y estudio

En esta parte se han investigado estudios similares basados en el conteo del parpadeo como lo hemos visto en el capítulo 1.3 y se han estudiado diseños de distintos instrumentos como en el capítulo 3.3.9.

Con los estudios del capítulo 1.3 se ha decidido cómo colocar los sensores, qué tipo de montura utilizar y para qué campo se puede utilizar. Mientras que en el capítulo 3.3.9 hemos visto tecnología que se basa en interferencia de ondas para determinar cómo se colocan los sensores.

Y se dividió el proyecto en las siguientes fases:



Figura 4.53: Fases del proyecto

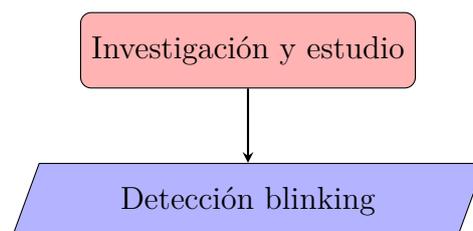
En la primera fase se buscó documentación acorde con la temática escogida y se empezó a programar el Arduino para valorar el entorno.

Seguidamente, en la segunda fase se valoró el primer prototipo, se calculó la frecuencia del parpadeo asociada a patologías y se realizó una prueba de 72 horas para comprobar su utilidad en casos de frecuencia de parpadeo baja.

Posteriormente, en la tercera fase se construyó el primer prototipo funcional y se empezó a subir los datos extraídos vía online a las distintas bases de datos.

Finalmente, en la última fase se construyó el Centinela con sus mejoras y se tradujo y adaptó toda la documentación.

4.7.2. Prototipo - Alpha: HAN SOLO



En este apartado se va a centrar en cómo detectar el parpadeo. Para ello, se estudió los distintos sistemas utilizados en el capítulo 1.3.

La primera Alpha que se propuso fue con una montura sin cristales y con un sensor pegado. El prototipo fue el siguiente:



Figura 4.54: Prototipo alpha

En este primer prototipo vemos que el sensor está pegado a la montura, es muy incómodo de llevar y no detecta de todo bien el parpadeo. Una de las principales desventajas de este sistema es que no terminaba de detectar bien el parpadeo porque el sensor se movía constantemente y no detectaba el parpadeo en su totalidad.

Este prototipo contenía el siguiente Arduino:

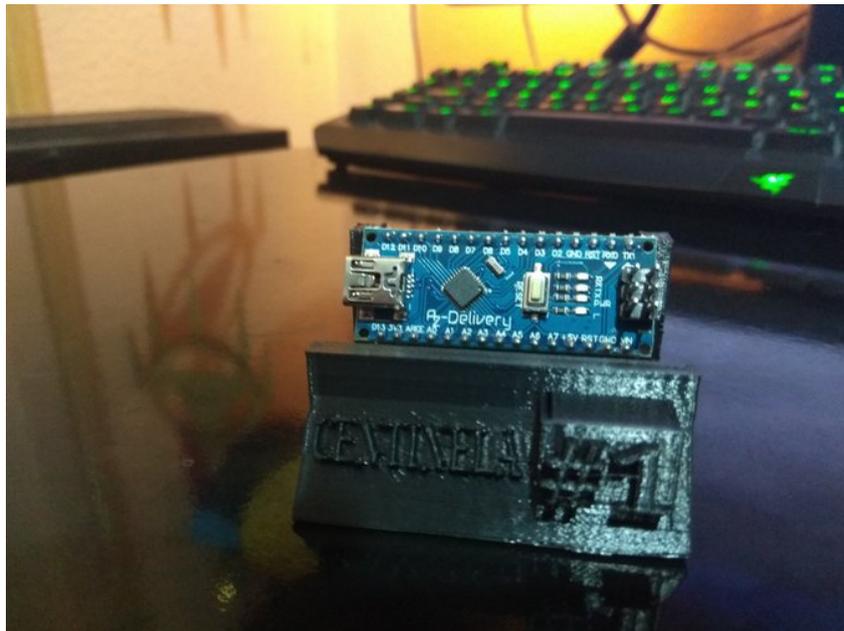
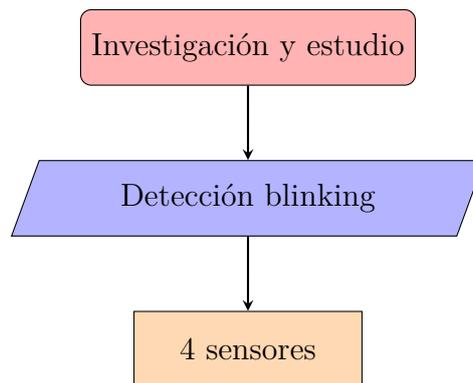


Figura 4.55: Arduino prototipo

4.7.3. Prototipo - 4 sensores



En esta fase del proyecto se propuso colocar cuatro sensores a la gafa para detectar mejor los párpados y su movimiento. La montura en la impresora 3D quedaba de la siguiente forma:

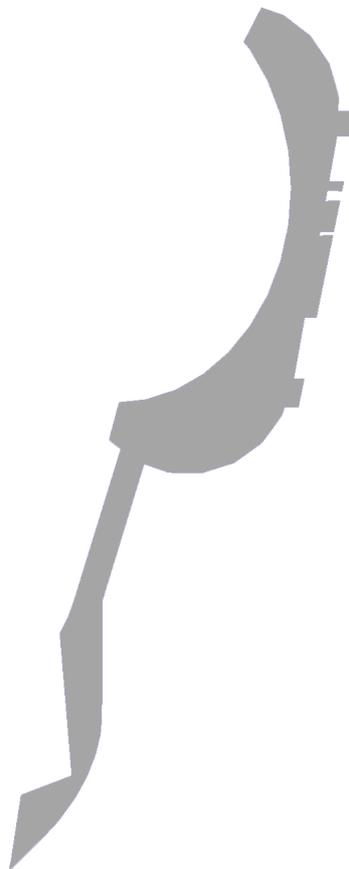


Figura 4.56: Impresión montura 4 sensores

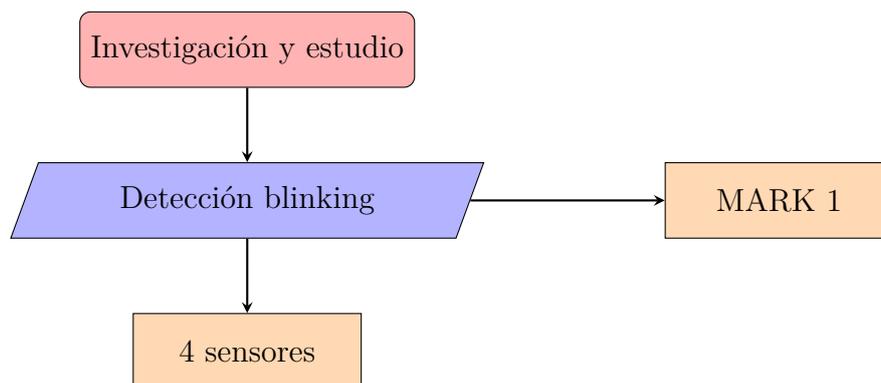
Con estos sensores la montura se nos quedaba muy pesada y era difícil de manejar. Y se empezó a realizar el registro de los parpadeos y a guardarlos en una tarjeta de memoria. El registro del parpadeo es el siguiente:

```
15:55:34.284 -> Parpado cerrado
15:55:34.555 -> Parpado cerrado
15:55:34.793 -> Parpado cerrado
15:55:35.030 -> Parpado abierto
15:55:35.302 -> Parpado cerrado
15:55:35.540 -> Parpado cerrado
15:55:35.777 -> Parpado abierto
15:55:36.046 -> Parpado abierto
```

Figura 4.57: Primera detección

Estos cuatro sensores se descartaron por el peso de la montura y la dificultad en la detección del parpadeo.

4.7.4. Prototipo - MARK1



Por otro lado, comenzamos el prototipo MARK 1. Seguía siendo offline y el tiempo entre la detección de parpadeo y este prototipo fue de 2 meses.

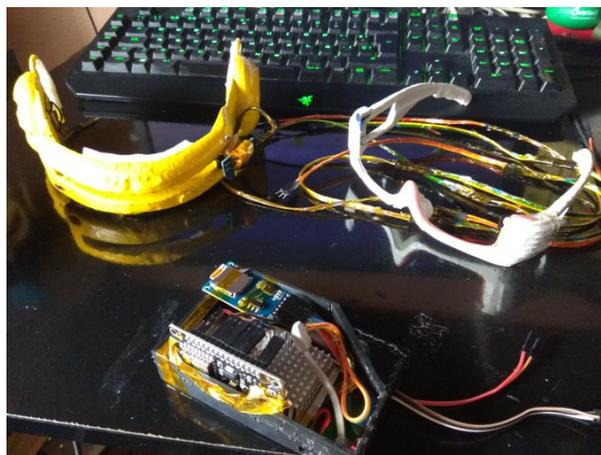
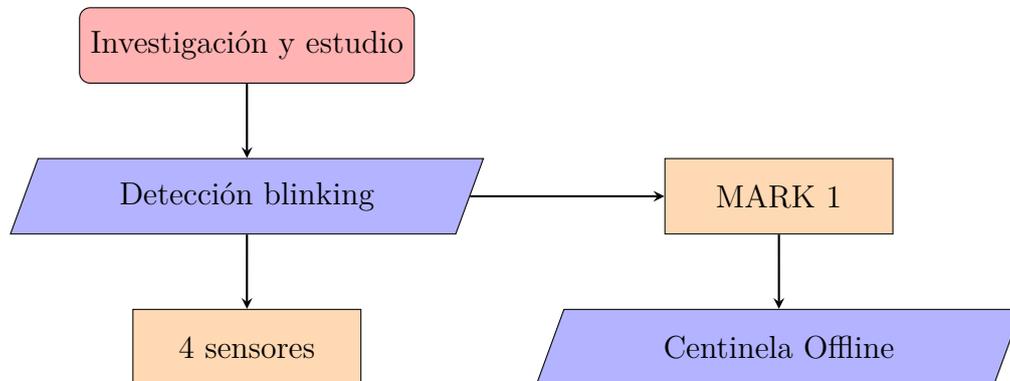


Figura 4.58: MARK 1

La montura blanca fue utilizada durante el proceso de MARK 1. Con ella se volvió a

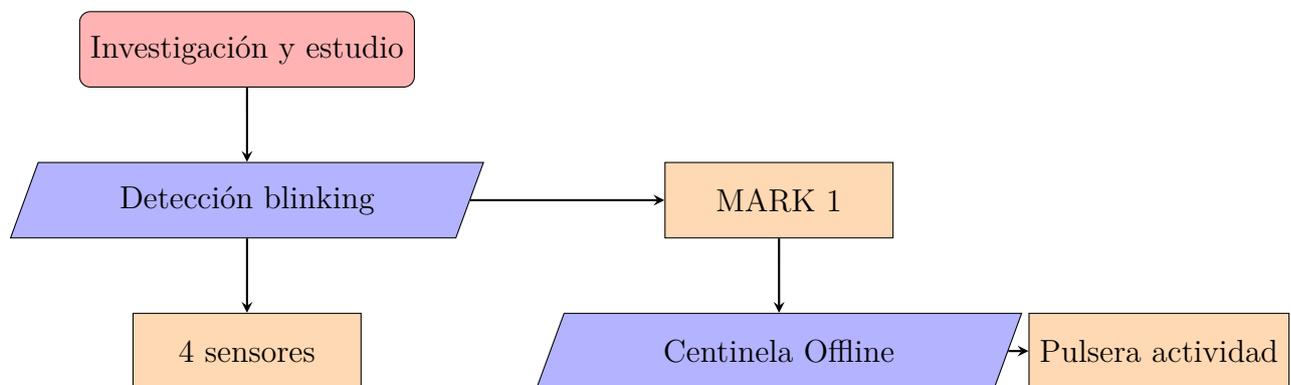
recolectar datos offline sobre el parpadeo, y se mejoró este MARK 1 para el proceso final de Centinel.

4.7.5. Centinela Offline



En esta parte se decidió mejorar el almacenamiento de la información, para ello se creó un Merovingio para guardar los distintos componentes electrónicos y se enganchó a la gafa para que empezara a guardar información. La implementación del Merovingio se implementará en capítulos posteriores. Su creación fue realizada mediante materiales no plásticos, utilizando fibra de maíz para su realización.

4.7.6. Integración con pulsera de actividad



En esta parte se decidió integrar la pulsera de actividad para ver si las horas de sueño tenían influencia en el parpadeo. Para ello, se hizo una prueba con un sujeto indicándole que no durmiera en dos días y se le realizó una prueba de parpadeo para ver si éste decaía. Se descartó esta idea debido a que la pulsera de actividad no es un artículo asequible y que no era asumible dejar a un sujeto sin dormir para tomar los datos de parpadeo.

Para esto se desarrolló una aplicación y un preset vía arduino que detectaba arritmias cardiacas en tiempo real y recogía los datos para posterior análisis offline. Se recogía los datos de los ojos y de la pulsación cardiaca en formato ISO. En el supuesto caso de que el sujeto estuviera en peligro, se mandaba un aviso vía twitter para que cualquier persona pudier auxiliarse.

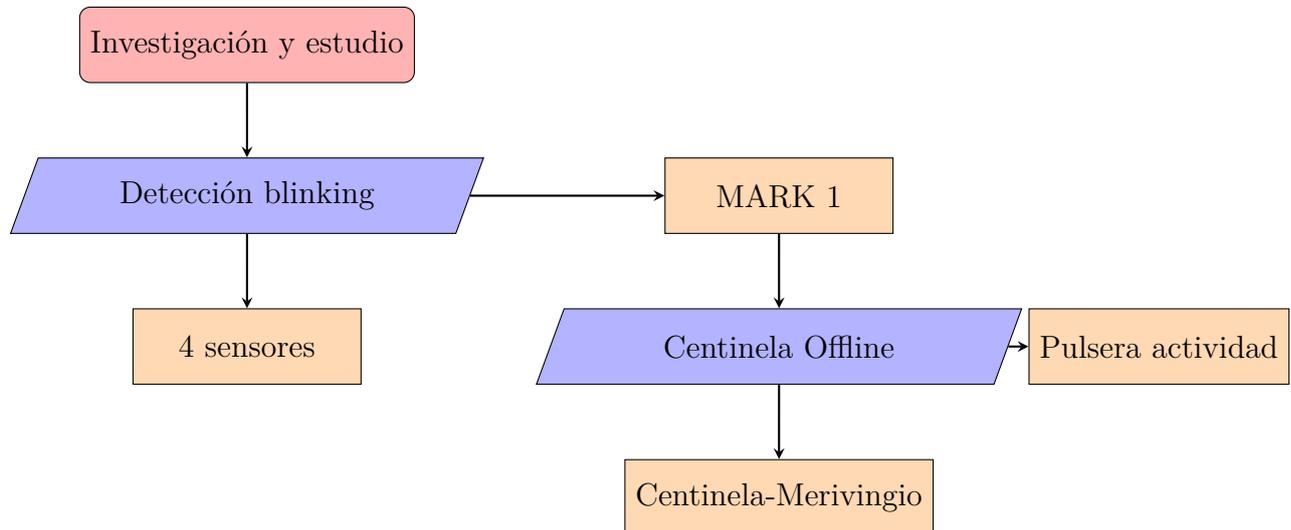


Figura 4.59: Evaluación online de la pulsera con el dispositivo

Esta vía se descartó porque no había relación entre la enfermedad de arritmia y la visión

del sujeto y porque nadie acudió en ayuda de la persona enferma cuando tenía arritmias. Pero se denotó una clara relación entre la falta del sueño del sujeto de pruebas y la fatiga ocular.

4.7.7. Centinela - Merovingio



En esta parte vamos a ver los resultados finales de este trabajo. Primero, tenemos el prototipo final de Centinela, que consiste en una montura hecha de fibra de maíz y que al contrario que las demás monturas, tiene una pequeña mirilla donde situamos el sensor IR para detectar mejor los párpados y su movimiento. Y por otro lado, tenemos el Merovingio donde tendremos todo el circuito y la antena que envía todos los datos recogidos a las distintas bases como lo hemos visto en el apartado 5.2. Y además, podremos personalizar las monturas mediante el sistema Kinect, ya que, permite escanear toda la cara y personalizar la misma como lo hemos visto en el apartado 4.5.1.

Primero se va a mostrar el escáner realizado por Kinect para determinar la cara del usuario y poder realizar la montura personalizada.



Figura 4.60: Escáner Kinect

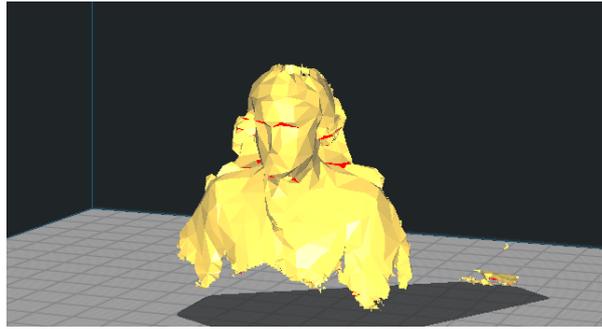


Figura 4.61: Escáner Kinect

Como se puede ver en la figura 4.61, se puede restar la parte de la cara que corresponde a los ojos para poder crear esta montura personalizada. A continuación se va a indicar el montaje de la misma.

Seguidamente se va a mostrar el resultado final de la montura de Centinela:

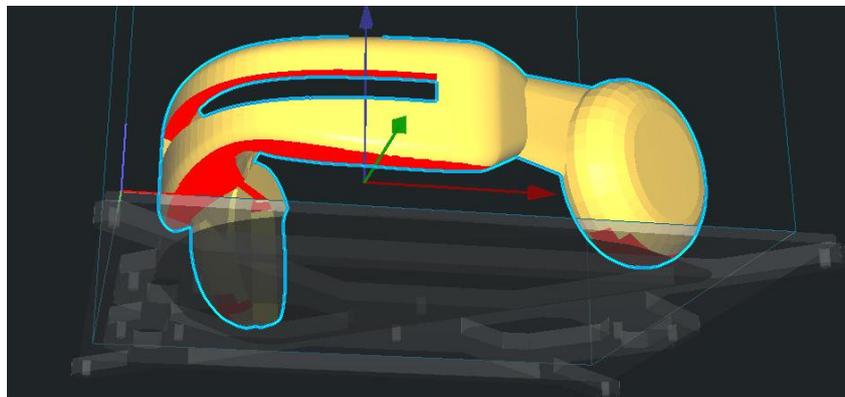


Figura 4.62: Montura Centinela

Como podemos ver, la distancia entre ambos párpados es menor por lo que el sensor IR podrá medir mejor el cierre y la apertura del párpado. Y su preparación fue de la siguiente forma:

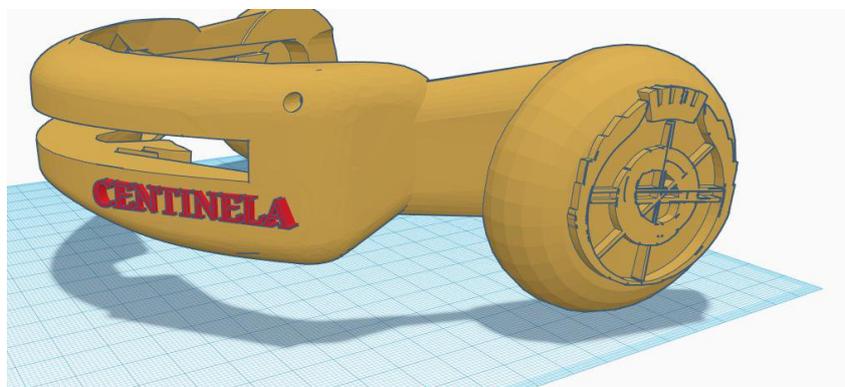


Figura 4.63: Boceto para impresora

Al ver que la montura ocupaba mucho se decidió dividir todos sus componentes para que

se pudieran imprimir bien y que no hubiera fallos en su desarrollo. Y se dispuso de la siguiente forma:

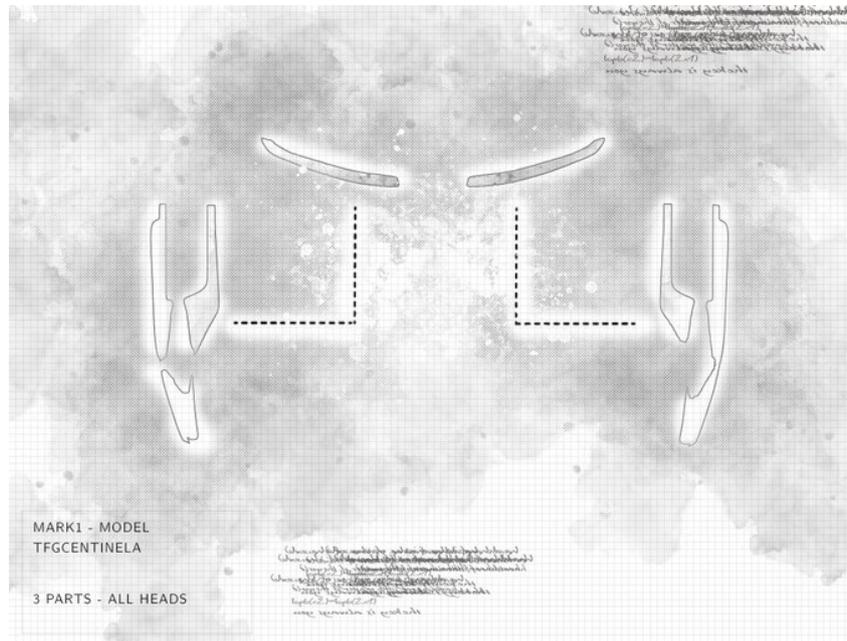


Figura 4.64: División montura boceto

Y los archivos preparados para imprimir quedaron de la siguiente forma:

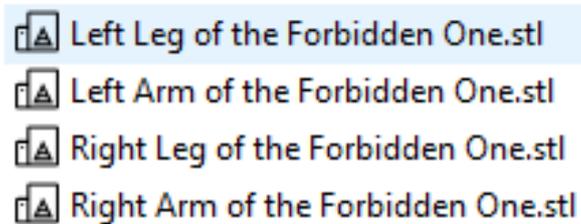


Figura 4.65: Archivos para imprimir

Después de realizar la impresión de la montura y la limpieza de escoria con materiales ecológicos, la montura se quedó de la siguiente forma:



Figura 4.66: Resultado final de la montura

Por otro lado, tenemos el Merovingio, donde va a tener todos los componentes esenciales de Centinela. Se planteó que tuviera la siguiente forma:

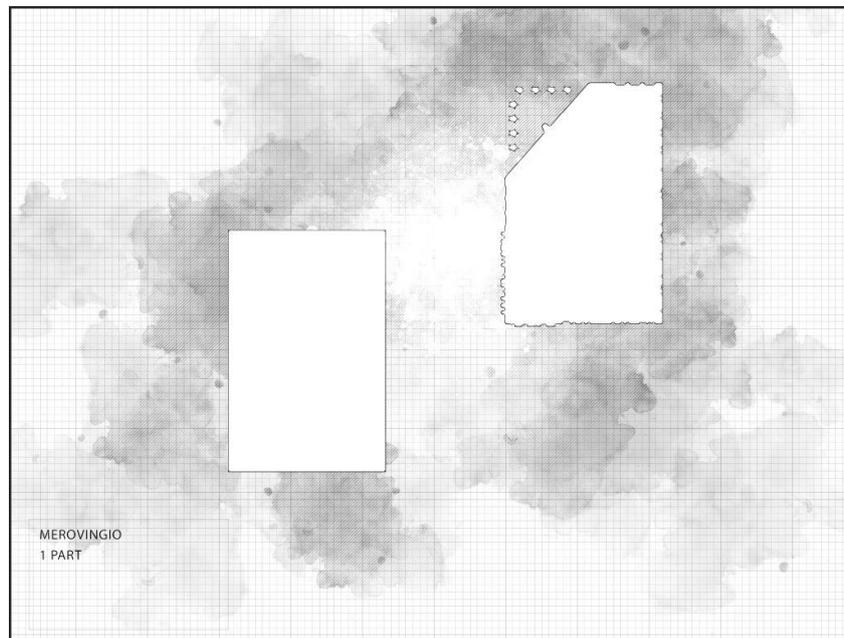


Figura 4.67: Boceto Merovingio

Con este boceto lo comenzamos a diseñar para la impresora 3D, quedando el diseño de la siguiente forma:

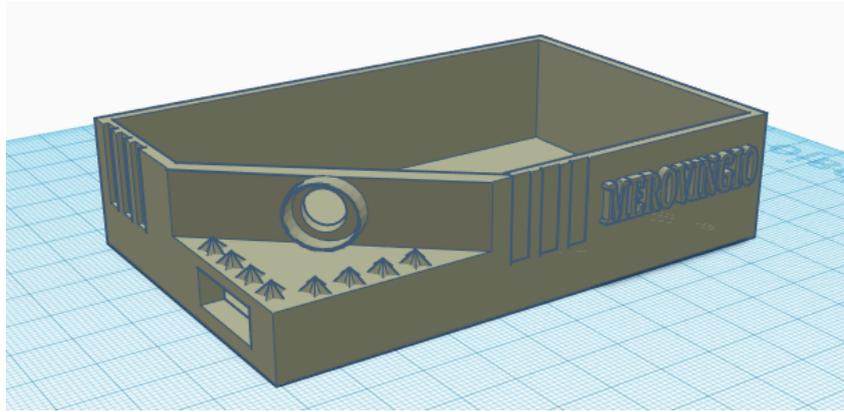


Figura 4.68: Merovingio 3D

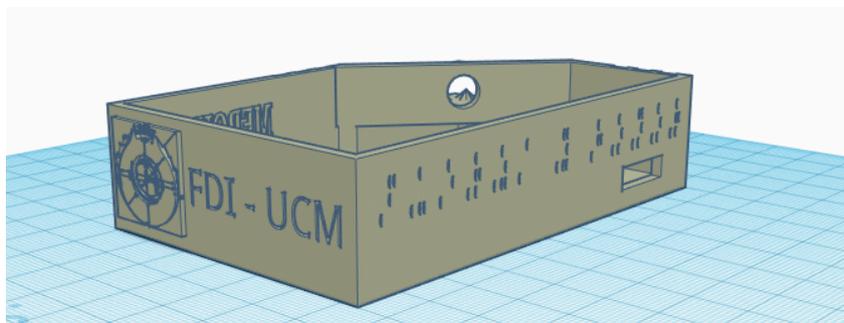


Figura 4.69: Merovingio 3D

Se diseñó parte de su texto en Braille para estar perfectamente adaptado. Una vez impreso y con todo los componentes en su interior el resultado es el siguiente:



Figura 4.70: Merovingio



Figura 4.71: Merovingio

Y finalmente, se va a mostrar un ejemplo de medida en tiempo real, el cual, es el siguiente:



Figura 4.72: Medida

4.7.8. Gestión de riesgos activa

En el desarrollo de este proyecto se han podido detectar varios errores, no del proyecto en sí sino de otros factores. Se va a proceder a describir cada uno de los riesgos y ver si se puede plantear una solución o se tienen que asumir.

4.7.9. Riesgos del planteamiento del proyecto

- **Riesgos de montaje de la montura:** Al haber más de un prototipo de montura, no se pudo estructurar bien.

- **Conocimientos en el campo de la optometría y medicina:** Se procedió a preguntar a optometras para conocer el funcionamiento de la bibliografía y ampliar el contenido del proyecto.
- **Gestión de los datos:** Al trabajar con datos online, se cuidaron de que no se pudieran obtener por fuentes externas.

4.7.10. Riesgos de la creación

- **Manejo de datos:** Al depender de datos externos se dependía de las mismas para saber si los datos se pasaban o no.

4.7.11. Riesgos con el proyecto finalizado

- **Riesgo del estándar:** Para convertirse en un estándar en el prediagnóstico se necesita que sea verificada por profesionales antes y una vez verificada, requiere una publicidad para que se dé uso.
- **Riesgo por caída del servidor:** Al subirse los datos a una plataforma online, si alguna de estas se cayeran. Habría un riesgo de perder datos.

Capítulo 5

Arquitectura e implementación

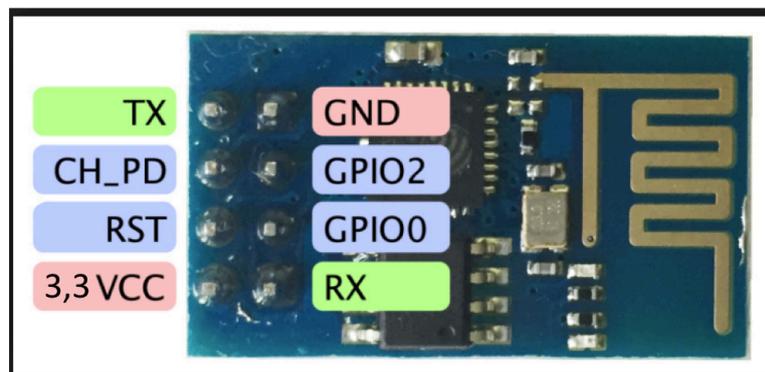
Quédate donde debes, en mis recuerdos
Cloud Strife

5.1. Esquemas de implementación Centinela - Final

5.1.1. Modulo ESP8266

En este apartado vamos a ver cómo conectar el módulo ESP8266 al módulo wi-fi y al arduino.

Primero vamos a ver cómo está formado el módulo wi-fi:



Nota: No permite Voltaje diferente a 3,3Vcc
al conectarlo a 5V se Quema

Figura 5.73: Módulo wi-fi

En el GND está la conexión a tierra, arriba a la derecha está la transmisión, abajo está un pin para resetearlo y en la parte inferior se encuentra el voltaje, en este caso 3.3 V. Es importante indicar que sólo funciona con un Arduino de 3.3 V, si se conecta un Arduino de 5 V se quemará. Por otro lado, los puertos GP hacen de microcontrolador y se puede

programar como un Arduino o una Raspberry Pi.

Primero, para conectarlo al ordenador hay que conectarlo mediante un conversor de USB a TTL o serial, con una salida de 3.3 V o conectarlo directamente a un Arduino, ya que, tiene este conector dentro.



Figura 5.74: Arduino nano utilizado

Para conectarlo al ordenador vamos a conectar el módulo al Arduino de la siguiente forma:

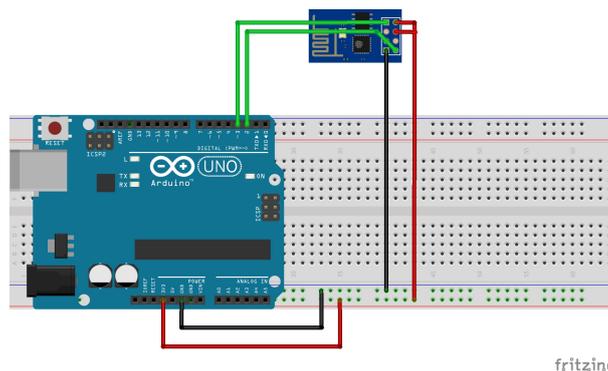


Figura 5.75: Conectando el módulo al Arduino

La recepción va a en color verde y la transmisión va en el mismo color. Es importante realizar la transmisión por arriba. Por otro lado, conectamos la fuente externa al Arduino. Una vez conectemos el Arduino, empezará a parpadear la bombilla en color verde y el módulo wifi empezará a emitir una luz roja y verde. Si el Arduino parpadea, se deberá conectar una fuente externa al adaptador de pared del Arduino.

Seguidamente, abrimos el ID del Arduino sin darle a cargar, ya que no tenemos ningún controlador para controlar el programa, y vamos a la parte derecha donde pone el serial del monitor. Y en esa pestaña, buscamos donde ponga **Ambos NL & CR** y cambiar los bandios hasta que funcione. Se le quita de la fuente de alimentación y se vuelve a conectar para comprobar que no parpadea. Ahora pasamos a escribir los comandos AT:

Listing 5.1: Comandos AT

```
1 AT
2 OK
3 AT+CIOBAUD=9600
4 OK
5 AT+GMR
6 OK
7 AT+CWMODE?
8 AT+CWMODE=3
9 AT+CWLAP
10 AT+CWJAP="","" ," "
11 AT+CIPMUX=1
12 AT+CIPSERVER=1.80
13 AT+CIFSR
```

Donde los comandos significan:

- **AT**: Ver si está conectado el Arduino. Si está conectado devolverá el OK.
- **AT+CIOBAUD=9600**: Para indicar que recibe 9600 baudios.
- **AT+GMR**: Para ver en qué versión está.
- **AT+CWMODE?**: Para ver en qué modo está.
- **AT+CWMODE=3**: Modo estación y AP.
- **AT+CWLAP**: Detección redes wifi.
- **AT+CWJAP="",""**: Analizar red wifi y conectarse a ella. Hay que saber qué nivel de seguridad tiene y en qué puerto conectarnos para no saturar el puerto. Entre el segundo comillado se debe conectar la clave de la red wifi que queremos conectarnos.
- **AT+CIMUX=1**: Para que sea capaz de soportar múltiples conexiones.
- **AT+CIPSERVER=1.80**: Para que funcione como un servidor.
- **AT+CIFSR**: Ver dirección IP asignada.

5.2. Escenarios con AWS

5.2.1. Amazon Web Service

Es un conjunto de herramientas y servicios de cloud computing de Amazon. Entre las herramientas que ofrece podemos encontrar las siguientes:

- **Cloud computing**: Computación en la nube.
- **Bases de datos**: Se puede elegir entre MySQL, PostgreSQL, Oracle, SQL Server y Amazon Aurora.

- **Creación de redes virtuales:** Permite la creación de redes virtuales a través del servicio VPC.
- **Aplicaciones empresariales:** Mediante Amazon WorkMail se puede acceder a un correo interno y a un almacenamiento interno.
- **Almacenamiento y gestores de contenido:** Ofrece Amazon S3 como servicio principal para almacenamiento regular.
- **Inteligencia de negocios:** Un sistema de análisis de datos empresariales a gran escala.
- **Gestión de aplicaciones móviles:** Mediante Amazon Mobile Hub se gestiona, crea y testea la aplicación móvil en la nube.
- **Internet de las cosas:** Analizar dispositivos conectados a internet.
- **Herramientas para desarrolladores:** Almacena código y lo implementa.
- **Seguridad y control de acceso:** Establece una conexión en varios pasos.

5.2.2. Conectar el módulo ESP8266 a AWS

Para conectar el módulo utilizaremos MQTT y el WebSocket. Para ello, utilizaremos el siguiente repositorio de contenido Github³⁵ para orientarnos. Y utilizaremos el siguiente código de ejemplo:

Listing 5.2: Conexión Arduino a AWS

```
1 #include <ESP8266WiFi.h>
2 #include <ArduinoJson.h>
3 #include <ESP8266AWSIoTMQTTWS.h>
4
5
6 const char *ssid = "WIFI_SSID";
7 const char *password = "WIFI_PASSWORD";
8
9 // See 'src/aws_iot_config.h' for formatting
10 char *region = (char *) "us-east-1";
11 char *endpoint = (char *) "xxxxxxxxxxxx";
12 char *mqttHost = (char *) "xxxxxxxxxxxx.iot.us-east-1.
    amazonaws.com";
13 int mqttPort = 443;
14 char *iamKeyId = (char *) "yyyyyyyyyyyyyyyyyyyy";
15 char *iamSecretKey = (char *) "
    YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY";
16 const char* aws_topic = "$aws/things/ZZZZZZZZZZZZ/shadow/
    update";
```

³⁵<https://github.com/debsahu/esp8266-arduino-aws-iot-ws>

```
18 #if !(ARDUINOJSON_VERSION_MAJOR == 6 and
    ARDUINOJSON_VERSION_MINOR == 5)
19 #error "Install ArduinoJson v6.5.0-beta"
20 #endif
21
22 ESP8266DateTimeProvider dtp;
23 AwsIotSigv4 sigv4(&ntp, region, endpoint, mqttHost, mqttPort,
    iamKeyId, iamSecretKey);
24 AWSConnectionParams cp(sigv4);
25 AWSWebSocketClientAdapter adapter(cp);
26 AWSMqttClient client(adapter, cp);
27
28 void setup() {
29
30     Serial.begin(115200);
31     while(!Serial) {
32         yield();
33     }
34
35     WiFi.begin(ssid, password);
36     while (WiFi.status() != WL_CONNECTED) {
37         delay(500);
38         Serial.print(".");
39     }
40
41     int res = client.connect();
42     Serial.printf("mqtt connect=%d\n", res);
43
44     if (res == 0) {
45         client.subscribe(aws_topic, 1,
46             [](const char* topic, const char* msg)
47             { Serial.printf("Got msg '%s' on topic %s\n", msg,
48                 topic); }
49         );
50     }
51
52 void loop() {
53     if (client.isConnected()) {
54         DynamicJsonDocument jsonBuffer;
55         JsonObject root = jsonBuffer.to<JsonObject>();
56         JsonObject state = root.createNestedObject("state");
57         JsonObject state_reported = state.createNestedObject("
            reported");
58         state_reported["value"] = random(100);
59         serializeJson(root, Serial);
60         Serial.println();
```

```
61     char shadow [measureJson(root) + 1];
62     serializeJson(root, shadow, sizeof(shadow));
63
64     client.publish(aws_topic, shadow, 0, false);
65     client.yield();
66 } else {
67     Serial.println("Not connected...");
68     delay(2000);
69 }
70
71 delay(30000);
72 }
```

Lo primero que tendremos que hacer será crearnos una cuenta en Amazon Web Service. Una vez nos registremos visitaremos el [IoT Core](#) y crearemos un objeto con un nombre con su tipo. Seguidamente, crearemos el certificado, los descargamos todos y los guardamos. También es importante activarlos. Es importante crear una política con una acción y con un recurso activo. Con esta política, daremos acceso al IoT. Por otro lado, en la pestaña de interacción copiaremos el https para que el Arduino pueda acceder a la base de datos y la copiamos en el código mostrado arriba. Posteriormente, seleccionaremos [IAM](#), y crearemos un nuevo usuario con sus accesos. Finalmente, una vez tengamos el usuario creado nos dará una clave que también pegaremos en el código más arriba para que almacene los datos en un usuario. Si queremos añadir más políticas podemos crearlas tanto en la creación, en la lectura y al escribir los datos.

5.2.3. Coste AWS

En este apartado se va a planear la instancia “t3.nano”.

Se dispone de todo el almacenamiento y bases de datos que se quiera y se paga por lo que realmente se utiliza.

Con lo anterior dicho, vamos a calcular los costes:

- Si se oferta bajo demanda el coste en un año sería de 45.52 **USD/año**.
- t3.nano utilizado durante 365 días en la promoción de un año sería de 26.28 **USD/año**.
- En la promoción de 3 años, el precio de un año sería de 17.52 **USD/año**.

5.3. Escenarios con ThingSpeak

5.3.1. ThingSpeak

ThingSpeak es una plataforma de análisis IoT que permite analizar y visualizar los contenidos en la nube. Esta plataforma nos permite:

- Configurar los dispositivos usando protocolos IoT.
- Visualizar los sensores en tiempo real.
- Agregar datos sacados de terceras fuentes.
- Manejar algoritmos de Matlab mediante datos IoT.
- Analizar IoT basados en horarios o eventos.
- Construir prototipos o sistemas IoT.
- Publicar los datos automáticamente utilizando redes sociales como Twitter.

Lo que destaca de esta plataforma respecto a la anterior es que es de software libre.

5.3.2. Conectar el módulo ESP8266 a ThingSpeak

El módulo ESP8266 se conectará al Arduino de la siguiente forma:

- $T_x \rightarrow D_2$.
- $R_x \rightarrow D_3$ (3.3 V).
- VCC \rightarrow 3.3 V.
- GND \rightarrow GND.

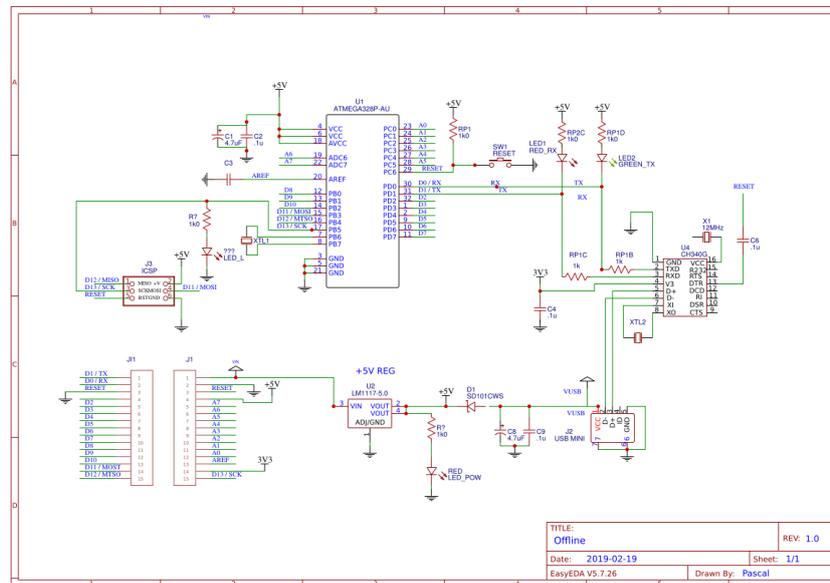
Una vez lo tengamos todo conectado y repitamos los pasos mostrados en el apartado anterior iremos a ThingSpeak. Donde nos conectaremos y rellenaremos todos los datos. Con el código anterior nos conectaremos al canal creado anteriormente como lo explicado en el capítulo anterior.

5.3.3. Coste ThingSpeak

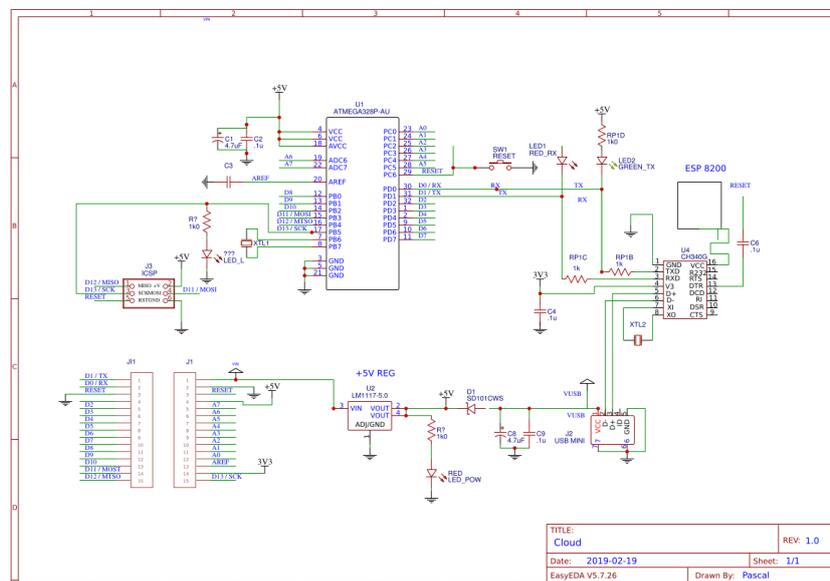
En este caso, se va a calcular el precio de los distintos paquetes al año:

- Precio del paquete estándar: 650 USD/año.
- Precio académico: 79 USD/año.
- Precio de estudiante al año: 250 USD/año.

5.6. Diagrama de conexiones offline



5.7. Diagrama de conexiones en Cloud



Capítulo 6

Mediciones y resultados

*La justicia está sujeta a disputa; la fuerza es
fácilmente reconocida y no entra en disputa.
Así que no le podemos dar fuerza a la justicia*
Pascal

6.1. Pruebas en sujetos con patologías

Se ha realizado la prueba en un sujeto con parpadeo incompleto en donde se podía ver claramente la disminución del parpadeo en el apartado del parpadeo más largo.

Con esto se quiere probar que el dispositivo centinela detecta el parpadeo incompleto. Además, puede detectar otras patologías relacionadas con el parpadeo porque mide los siguientes parámetros:

- Parpadeo más largo.
- Máxima entre parpadeos.
- Minuto máximo.
- Minuto mínimo.
- Detección predictiva.

1. Tiempo del dispositivo centinela

Tiempo que ha estado activo el dispositivo sobre el paciente.

0h 50m

15,815999999999804s

Se recomienda usar «Centinela» durante 3 horas para cálculos más precisos.

2. Cálculos sobre el parpadeo

Cálculos base sobre los parpadeos analizados.

2.1. Parpadeo más largo

Cálculo del parpadeo más largo analizando toda la secuencia.

11,886s

2.2. Máxima entre parpadeos

Dato máximo de la medición entre parpadeos.

32,949s

2.3. Muestras analizadas

Contador de parpadeos completos.

1816 veces

2.4. Minuto máximo

Se ha detectado una máxima de parpadeos en este minuto.

Minuto 4 número de parpadeos: 74

2.5. Minuto mínimo

Se ha detectado una mínima de parpadeos en este minuto.

Minuto 38 número de parpadeos: 7

3. Detección predictiva

Según los datos de exposición de luz y la frecuencia de parpadeo se ha detectado:

Vista sin altos estímulos

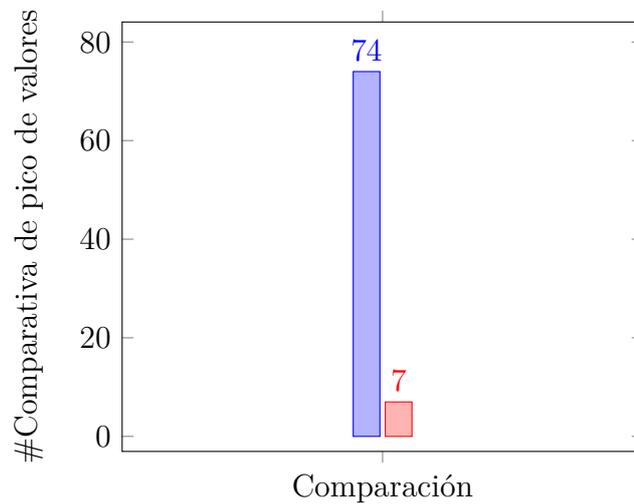
3.1. Cálculos para 12 horas

Basándonos en la frecuencia, realizarás estos parpadeos si no se añade ningún estímulo externo extra.

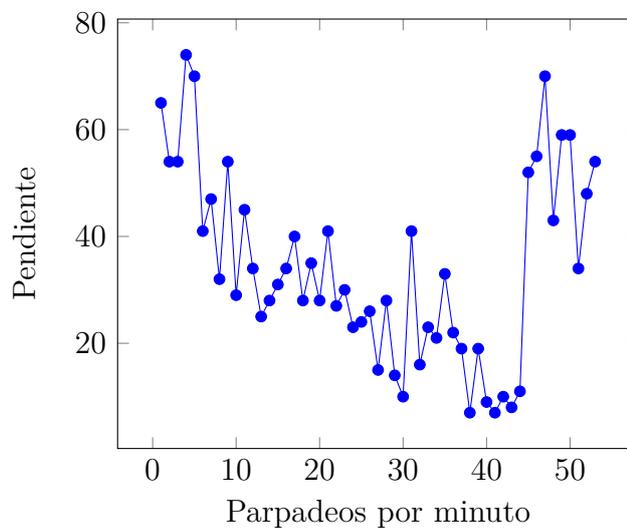
26150 veces

4. Gráficas de evolución

4.1. Gráfica comparativa de dato máximo y mínimo



4.2. Cálculo pendiente



5. Resultado final

Se ha detectado principio de cansancio

En este caso se realizó un escaneo de 50 minutos, en donde se midieron los parpadeos máximos, mínimos y se muestra las gráficas del desarrollo del parpadeo.

6.2. Prueba de 72 horas

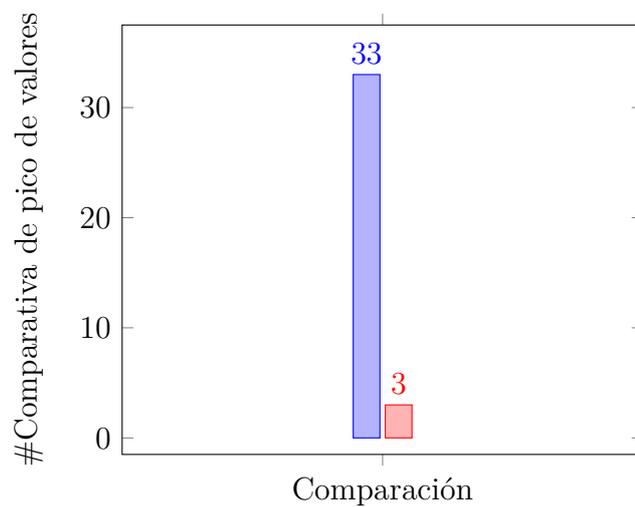
Se realizó la prueba en un servidor. En la que estuvo 72 horas sin dormir para ver si influía en alguno de los siguientes parámetros:

- Gráfica de evolución.
- Cálculo de la pendiente.
- Resultado final.

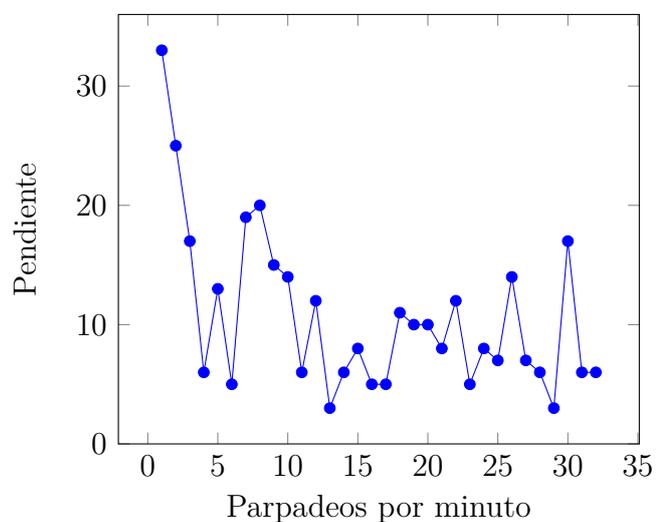
En la gráfica de evolución se muestra la diferencia entre los parpadeos máximos y mínimos, el cálculo de la pendiente se realiza midiendo los distintos parpadeos durante un periodo de tiempo y el resultado final muestra el resultado del compendio con todas las medidas e indica si hay o no fatiga ocular.

4. Gráficas de evolución

4.1. Gráfica comparativa de dato máximo y mínimo



4.2. Cálculo pendiente



5. Resultado final

Tienes fatiga ocular, para un poco

Se realizó una prueba de 72 horas para detectar si hay alguna irregularidad en alguna medida.

1. Tiempo del dispositivo centinela

Tiempo que ha estado activo el dispositivo sobre el paciente.

0h 29m

1.017000000000528s

Se recomienda usar «Centinela» durante 3 horas para cálculos más precisos.

2. Cálculos sobre el parpadeo

Cálculos base sobre los parpadeos analizados.

2.1. Parpadeo más largo

Cálculo del parpadeo más largo analizando toda la secuencia.

5.31s

2.2. Máxima entre parpadeos

Dato máximo de la medición entre parpadeos.

12.599s

2.3. Muestras analizadas

Contador de parpadeos completos.

633 veces

2.4. Minuto máximo

Se ha detectado una máxima de parpadeos en este minuto.

Minuto 1 número de parpadeos: 37

2.5. Minuto mínimo

Se ha detectado una mínima de parpadeos en este minuto.

Minuto 24 número de parpadeos: 9

3. Detección predictiva

Según los datos de exposición de luz y la frecuencia de parpadeo se ha detectado:

Momentos de lectura

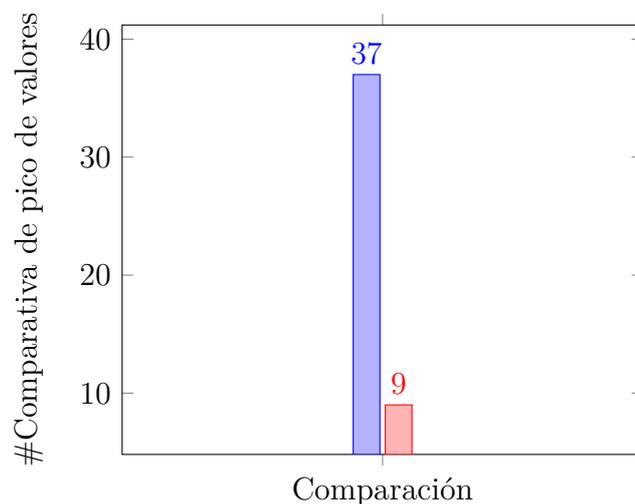
3.1. Cálculos para 12 horas

Basándonos en la frecuencia, realizarás estos parpadeos si no se añade ningún estímulo externo extra.

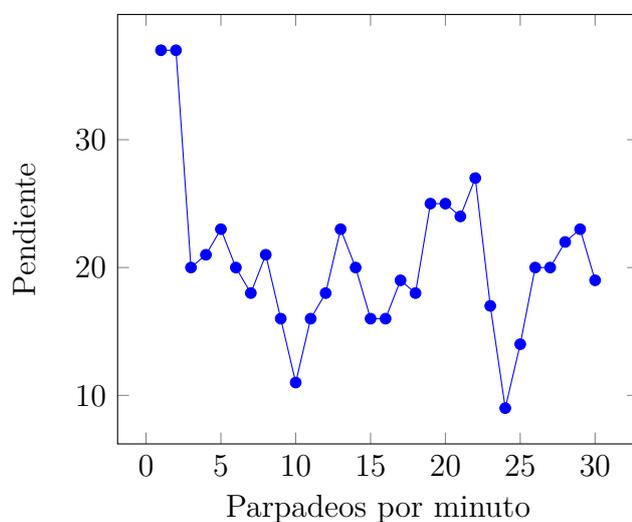
15715 veces

4. Gráficas de evolución

4.1. Gráfica comparativa de dato máximo y mínimo



4.2. Cálculo pendiente



5. Resultado final

Tienes la vista en condiciones adecuadas

6.3. Pruebas públicas

Se realizaron una serie de pruebas públicas para ver si usuarios al azar presentaban fatiga ocular. Para ello se presentaron voluntarios de las siguientes facultades:

- Facultad de Informática de la Universidad Complutense de Madrid.
- Facultad de Óptica y Optometría de la Universidad Complutense de Madrid.

Con estas pruebas públicas se quiere demostrar la eficacia independientemente del paciente o sujeto. Se intentó que estuvieran el máximo tiempo posible para que detectara bien los párpados y pudiera medir bien el parpadeo.

1. Tiempo del dispositivo centinela

Tiempo que ha estado activo el dispositivo sobre el paciente.

0h 4m

21.154999999999973s

Se recomienda usar «Centinela» durante 3 horas para cálculos más precisos.

2. Cálculos sobre el parpadeo

Cálculos base sobre los parpadeos analizados.

2.1. Parpadeo más largo

Cálculo del parpadeo más largo analizando toda la secuencia.

23.553s

2.2. Máxima entre parpadeos

Dato máximo de la medición entre parpadeos.

24.169s

2.3. Muestras analizadas

Contador de parpadeos completos.

187 veces

2.4. Minuto máximo

Se ha detectado una máxima de parpadeos en este minuto.

Minuto 1 número de parpadeos: 55

2.5. Minuto mínimo

Se ha detectado una mínima de parpadeos en este minuto.

Minuto 4 número de parpadeos: 36

3. Detección predictiva

Según los datos de exposición de luz y la frecuencia de parpadeo se ha detectado:

Vista sin altos estímulos

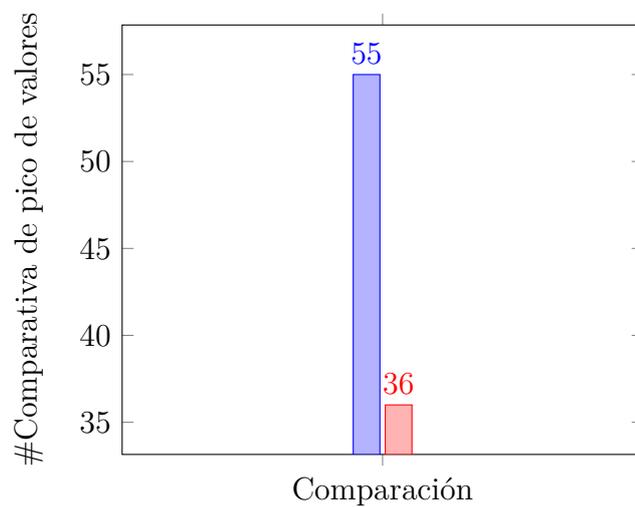
3.1. Cálculos para 12 horas

Basándonos en la frecuencia, realizarás estos parpadeos si no se añade ningún estímulo externo extra.

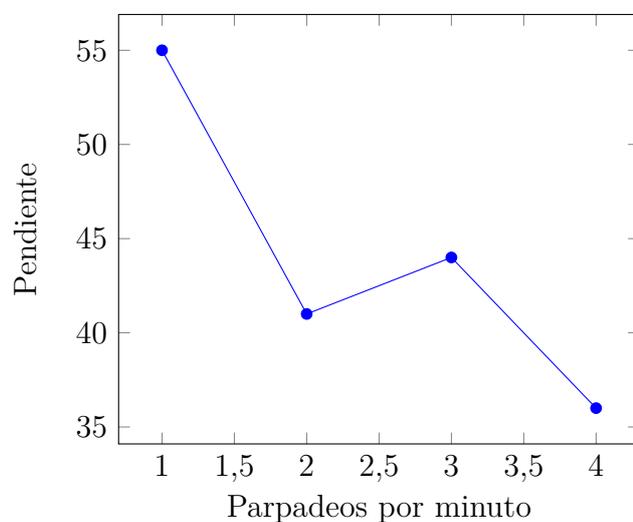
33660 veces

4. Gráficas de evolución

4.1. Gráfica comparativa de dato máximo y mínimo



4.2. Cálculo pendiente



5. Resultado final

Tienes la vista en condiciones adecuadas

1. Tiempo del dispositivo centinela

Tiempo que ha estado activo el dispositivo sobre el paciente.

0h 50m

15.815999999999804s

Se recomienda usar «Centinela» durante 3 horas para cálculos más precisos.

2. Cálculos sobre el parpadeo

Cálculos base sobre los parpadeos analizados.

2.1. Parpadeo más largo

Cálculo del parpadeo más largo analizando toda la secuencia.

11.886s

2.2. Máxima entre parpadeos

Dato máximo de la medición entre parpadeos.

32.949s

2.3. Muestras analizadas

Contador de parpadeos completos.

1816 veces

2.4. Minuto máximo

Se ha detectado una máxima de parpadeos en este minuto.

Minuto 4 número de parpadeos: 74

2.5. Minuto mínimo

Se ha detectado una mínima de parpadeos en este minuto.

Minuto 38 número de parpadeos: 7

3. Detección predictiva

Según los datos de exposición de luz y la frecuencia de parpadeo se ha detectado:

Vista sin altos estímulos

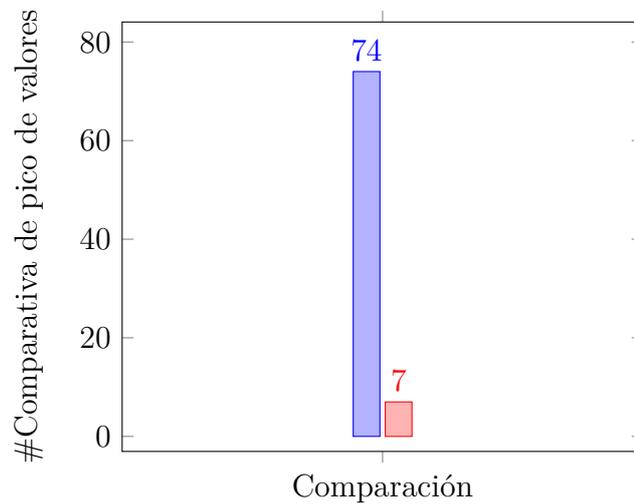
3.1. Cálculos para 12 horas

Basándonos en la frecuencia, realizarás estos parpadeos si no se añade ningún estímulo externo extra.

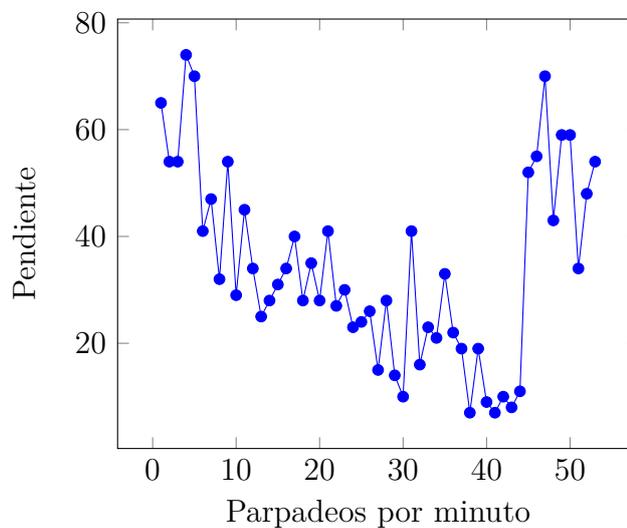
26150 veces

4. Gráficas de evolución

4.1. Gráfica comparativa de dato máximo y mínimo



4.2. Cálculo pendiente



5. Resultado final

Se ha detectado principio de cansancio

En esta parte se van a mostrar las pruebas públicas realizadas del proyecto.

6.4. Documento médico generado

En el documento médico generado encontramos los siguientes elementos:

- Parpadeo máximo: Detecta la frecuencia máxima de parpadeo.
- Parpadeo mínimo: Detecta la frecuencia mínima de parpadeo.
- Muestras analizadas: Detecta todas las veces que ha parpadeado el ojo.
- Minuto máximo y mínimo: Parpadeos máximos y mínimos en minutos.
- Detección predictiva: Analiza los datos y da un prediagnóstico.
- Realiza un prediagnóstico a las 12 horas: A partir de los datos realiza un cálculo aproximado para 12 horas.
- Gráfica comparativa de máximo y mínimo: Compara el parpadeo máximo y mínimo en una gráfica.
- Cálculo pendiente: A partir del parpadeo por minuto se crea una gráfica con pendiente.

Con el cálculo del parpadeo se puede detectar las siguientes patologías oculares:

- Ojo seco.
- Conjuntivitis.
- Patologías relacionadas con la baja frecuencia del parpadeo.

1. Tiempo del dispositivo centinela

Tiempo que ha estado activo el dispositivo sobre el paciente.

0h 21m
27,378999999999905s

Se recomienda usar «Centinela» durante 3 horas para cálculos más precisos.

2. Cálculos sobre el parpadeo

Cálculos base sobre los parpadeos analizados.

2.1. Parpadeo más largo

Cálculo del parpadeo más largo analizando toda la secuencia.

18,542s

2.2. Máxima entre parpadeos

Dato máximo de la medición entre parpadeos.

196,298s

2.3. Muestras analizadas

Contador de parpadeos completos.

634 veces

2.4. Minuto máximo

Se ha detectado una máxima de parpadeos en este minuto.

Minuto 2 número de parpadeos: 261

2.5. Minuto mínimo

Se ha detectado una mínima de parpadeos en este minuto.

Minuto 9 número de parpadeos: 2

3. Detección predictiva

Según los datos de exposición de luz y la frecuencia de parpadeo se ha detectado:

Vista sin altos estímulos

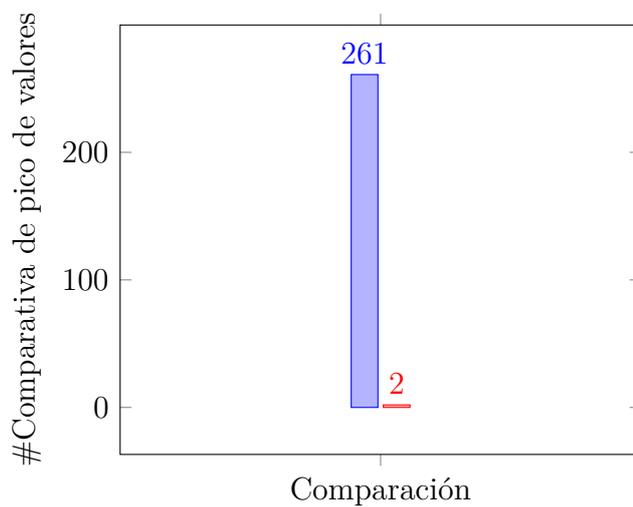
3.1. Cálculos para 12 horas

Basándonos en la frecuencia, realizarás estos parpadeos si no se añade ningún estímulo externo extra.

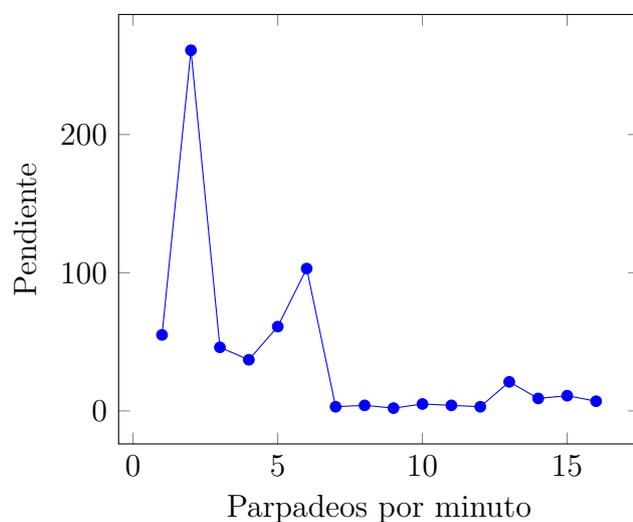
21737 veces

4. Gráficas de evolución

4.1. Gráfica comparativa de dato máximo y mínimo



4.2. Cálculo pendiente



5. Resultado final

Se ha detectado principio de cansancio

Capítulo 7

Conclusiones y futuras aplicaciones del proyecto

*For the doom of Gunmar, Eclipse is mine to
command*
Jim Lake Jr.

7.1. Conclusiones

Este proyecto tiene como objetivo detectar la fatiga ocular. Para ello, se ha basado en proyectos similares como los vistos en el estado del arte para crear un hardware y un software que sean cómodos y rápidos para su uso en el ámbito de la medicina. La metodología elegida para llevarlo a cabo es la Kanban y el plan de proyecto fue realizado en la aplicación Taiga.io. Por otro lado, para desarrollar los circuitos internos del Arduino se utilizaron las aplicaciones CircuitLab y EasyEDA, y para la realización de las medidas del parpadeo se utilizaron AWS como aplicación privada y ThingSpeak como aplicación libre. Así, se le da la opción al profesional de elegir entre las dos aplicaciones. Una vez se comenzó a realizar el proyecto, se escaneó la cara mediante Kinect para implementar el Centinela con la forma de la cara del usuario, se realizaron una serie de prototipos de manera online y offline hasta que se obtuvo el diseño definitivo. Seguidamente, una vez se obtuvo el diseño se realizaron una serie de pruebas en sujetos con patologías, una prueba de 72 horas previa y una serie de pruebas públicas para demostrar su eficacia. Y por último, se presentó el proyecto en distintos medios y facultades para demostrar su uso y eficacia.

El trabajo resultó correcto y se presentó ante investigadores y personal del ámbito médico-óptico como método para pre-diagnóstico. Las mediciones y la calibración fueron correctas y el informe médico de pre-diagnóstico tiene verificada la utilidad.

7.2. Futuras aplicaciones del proyecto

7.2.1. Aplicaciones ópticas

Se puede utilizar en un futuro en el campo de la óptica, ya que, el sistema es capaz de ver si hay exceso de iluminación y lo notifica. Se podría aplicar en un futuro para que los radares detecten si hay exceso de iluminación de las luces de los coches en situaciones que no sean indicadas y avisar a la autoridad oportuna.

7.2.2. Aplicaciones optometristas

Se puede utilizar como herramienta complementaria en la consulta optometrista para pacientes no colaborativos, en los que no se pueda medir el BUT ni el NIBUT, acompañándole de los tests oportunos y de la ficha que se adjunta pueden orientar al optometrista en el prediagnóstico de patologías relacionadas con el parpadeo, como ojo seco y fatiga ocular.

7.2.3. Otros futuros estudios

Finalmente, además puede servir de utilidad para determinar la frecuencia de parpadeo en futuros estudios sobre ojo seco y otras patologías relacionadas con el mismo de una manera más sencilla.

7.3. Futuros prototipos

7.3.1. Ojo sólido

La intención es fabricar un ojo sin personalizar que se adapte a todos los tamaños de ojo y profundidades. Este prototipo está en marcha y constaría de un ojo sólido³⁶ que se probaría en oscuridad, dando otro enfoque al proyecto entero y reduciendo los costes en más de un 370%

- Costes investigación Centinela: 1870 euros
- Costes investigación Ojo Sólido: 200 euros

Este coste sería para investigación, el coste de producción no llegaría a 0,87 euros y el tiempo de fabricación es de 2 horas. Mientras que el tiempo de fabricación de un dispositivo centinela es de 2 días y tiene un coste de aprox 8-10 euros.

³⁶<https://www.pinterest.co.uk/pin/760193612076990666/>



Figura 7.76: Presentación del prototipo de ojo sólido

Este prototipo fue presentado ante las cámaras de telemadrid como parte de la repercusión real del proyecto.

7.4. Futuras aplicaciones en Cloud

La intención es que el entorno se adapte de forma automática mediante el cálculo en la nube, si detecta cansancio, fatiga u ojo seco se podría menguar la luz del entorno para evitar este tipo de patologías y ayudar a corregirlas. También se está trabajando en un modelo capaz de predecir cuántas horas de visión tiene una persona en el instante que se pone las gafas gracias al cálculo predictivo y al machine learning.

Capítulo 7

Conclusions and future applications of the project

*For the doom of Gunmar, Eclipse is mine to
command*
Jim Lake Jr.

7.1. Conclusions

This project aims to detect eye fatigue. To do this, it has relied on similar projects such as those seen in the state of the art to create hardware and software that are comfortable and fast for use in the field of medicine. The methodology chosen to carry it out is the Kanban and the project plan was carried out in the application Taiga.io. On the other hand, the CircuitLab and EasyEDA applications were used to develop the Arduino's internal circuits, and for the realization of flicker measures AWS were used as a private application and ThingSpeak as a free application. Thus, the professional is given the option to choose between the two applications. Once the project was started, the face was scanned using Kinect to implement the Sentinel in the shape of the user's face, a series of prototypes were made online and offline until the final design was obtained. Then, once the design was obtained, a series of tests were carried out on subjects with pathologies, a previous 72-hour test and a series of public tests to demonstrate its effectiveness. And finally, the project was presented in different means and faculties to demonstrate its use and effectiveness.

The work was correct and presented to researchers and staff in the medical-optical field as a method for pre-diagnosis. Measurements and calibration were correct, and the pre-diagnosis medical report has verified utility.

7.2. Future project applications

7.2.1. Optical applications

It can be used in the future in the field of optics, since, the system is capable of see if there's too much illumination and it notifies you. It could be applied in the future so that radars detect if there is excess illumination of car lights in situations unaddressed and notify the appropriate authority.

7.2.2. Optometrist applications

It can be used as a complementary tool in the optometrist consultation to non-collaborative patients, where BUT and NIBUT cannot be measured, accompanied by the appropriate tests and attached tab can guide the optometrist pre-diagnosis of flicker-related pathologies, such as dry eye and fatigue eye.

7.2.3. Other future studies

Finally, it can also be useful in determining the frequency of blinking in future studies on dry eye and other pathologies related to it in a simpler way.

7.3. Future prototypes

7.3.1. Solid eye

The intention is to manufacture an uncustomized eye that fits all sizes of eye and depths. This prototype is underway and would consist of a solid eye³⁷ that is would test in darkness, giving another approach to the whole project and reducing costs in more than 370 %.

- Sentinel research costs: 1870 euros.
- Eye solid research costs: 200 euros.

This cost would be for research, the cost of production would not reach 0.87 euros and manufacturing time is 2 hours. While the manufacturing time of a Sentinel device is 2 days and costs approx. 8-10 euros.

³⁷<https://www.pinterest.co.uk/pin/760193612076990666/>



Figura 7.77: Presentation of the eye solid prototype

This prototype was presented to telemadrid cameras as part of the impact actual lyfore of the project.

7.3.2. Future Cloud Applications

The intention is that the environment adapts automatically by calculating cloud, if you detect tiredness, fatigue or dry eye could diminish the light of the environment to avoid these types of pathologies and help correct them. It is also working on a model capable of predicting how many hours of vision a person has at the moment puts on glasses thanks to predictive calculation and machine learning.

Capítulo 8

Repercusión

(See you) Next Illusion!
Kaito Kuroba como Kaito Kid

Nuestro proyecto tendrá una repercusión en el prediagnóstico de patologías relacionadas con el parpadeo. Con ello, queremos avanzar y ayudar a los profesionales sanitarios. Esto ha conseguido despertar el interés de varios centros, medios de comunicación y expertos.

8.1. Cadena Ser

Figura 8.78: Promoción Cadena Ser

Con motivo del TFG se nos invitó a la Cadena Ser a exponer el trabajo donde todos los de la cadena pudieron tener a su disposición el aparato para poder probarlo.

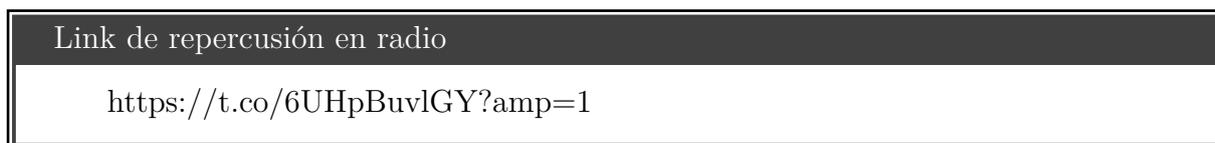


Figura 8.79: Foto oficial en Cadena Ser

Generando un gran impacto mediático:



Hoy por Hoy Madrid
@HoyxHoyMadrid

Nuestro compañero [@diegoserradilla](#) nos trae el proyecto de [@semi_ingenieros](#), un estudiante de ingeniería informática de la [@unicomplutense](#) que ha creado, con una impresora 3D y el desarrollo de su propio hardwer, unas gafas que detectan patologías oculares.



1:00 p. m. · 30 oct. 2019 · [Twitter Web App](#)

22 Retweets **47** Me gusta

Figura 8.80: Promoción en twitter e impacto

8.2. Telemadrid

De la misma forma, reporteros de telemadrid se pusieron en contacto e hicieron un reportaje amplio sobre todo el funcionamiento y vistas al futuro del aparato:



Figura 8.81: Entrevista Telemadrid



Figura 8.82: Entrevista Telemadrid II



Figura 8.83: Entrevista Telemadrid III



Figura 8.84: Entrevista Telemadrid IV

Acabaron encantados con la facultad y con el trato recibido.

8.3. Concurso Universitario de Software Libre

Aquí se ha presentado el proyecto a concurso con todas las licencias libres que se tienen, siendo algunas de las más libres del mundo (CC-0)



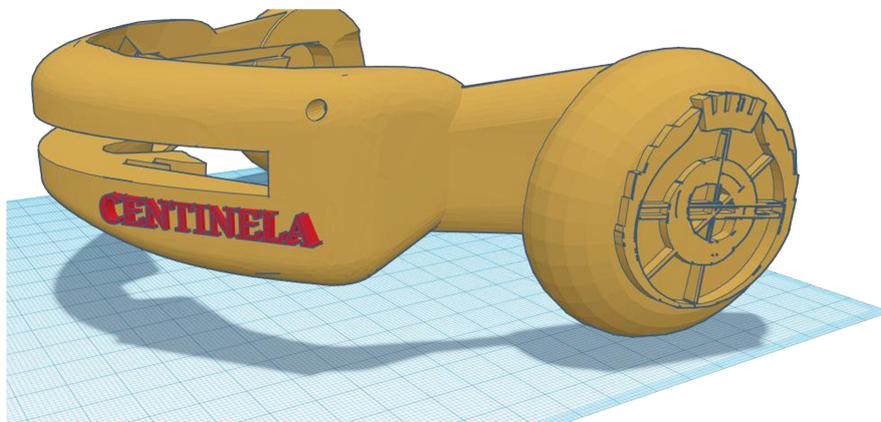
Figura 8.85: Enlace al Concurso Universitario de Software Libre

8.4. Presentación del TFG ante Óptica y Optometría

La presentación del TFG se realizará el miércoles 18 de Diciembre a las doce y media. La difusión de la misma ha sido llevada por el equipo de Gobierno de la facultad. Se mostró interés por este trabajo por su relación en el campo de la optometría y sus posibles valores en el prediagnóstico de patologías visuales asociadas a la fatiga ocular.

Presentación TFG Centinela

David Pacios Izquierdo - Facultad de Informática
18 diciembre 2019 - 12:30
Salón de actos - Facultad de Óptica y Optometría



Descubre las gafas que ayudan a detectar y diagnosticar posibles patologías oculares.



Figura 8.86: Cartel de promoción

8.4.1. Entrevista de la Facultad de Óptica y Optometría

Por iniciativa de la facultad de Óptica y Optometría se realizó una entrevista en el despacho de José Luis Vázquez Poletti el día 4 de Diciembre a las once menos veinte.



Figura 8.87: Inicio de la entrevista

En esa entrevista se habló un poco de los cursos de L^AT_EX, de mi trayectoria en la facultad, del TFG y de OTEA.



Figura 8.88: Pregunta Centinela



Figura 8.89: Pregunta sobre OTEA

La versión corta de esta misma entrevista será expuesta en las pantallas de la Facultad de Óptica y Optometría para promocionar tanto los futuros cursos como la conferencia del TFG que ya ha sido comentada en la sección 8.4.

Índice de figuras

2.1. Funcionamiento de la aplicación OpenFace	9
2.2. Facefirst en aeropuertos	10
2.3. Captura de Snapchat	10
2.4. Imágenes del proyecto	11
2.5. Traducción de las letras por órdenes	12
2.6. Uniones Arduino	13
2.7. Gafas 3D utilizadas en este proyecto	13
2.8. Configuración de la gafa smart	14
2.9. Detección del ojo cerrado	15
2.10. Esquema sensor utilizado	16
2.11. Diagrama EBCM	17
2.12. Detección ojos	17
2.13. Esquema límite	18
2.14. Esquema detección parpadeo	18
2.15. Esquema algoritmo EBCM	19
2.16. Electrooculograma de los distintos tipos de parpadeo	21
2.17. Detección parpadeo IR	21
2.18. Diagrama del proyecto	22
2.19. Cámara utilizada	22
2.20. Módulo bluetooth	23
2.21. Técnica utilizada	24
2.22. Detección cara	24
2.23. Distintas gráficas del proceso para la detección del parpadeo	25
2.24. Anchura picos de parpadeo	26
2.25. FSM para la duración estimada del parpadeo	26
2.26. Esquema duración parpadeo	27
2.27. Amplitud de parpadeo medida	28
2.28. Detección del parpadeo mediante la diferenciación de imágenes: a) imagen del ojo abierto, b) imagen del ojo cerrado, c) diferencia de imágenes, d) limite de la diferencia de imágenes	29
2.29. Esquema propuesto para detectar el parpadeo	30
2.30. Cambio en el valor de la correlación del parpadeo	30
2.31. Posible distribución del detector del parpadeo	31
2.32. Sistema detección	32
2.33. Diferenciación parpadeo detectado	32
2.34. Captura de pantalla del software EyeTribe	33
3.35. Tabla de ejemplo	36
3.36. Captura del proyecto	38

3.37. Captura de la aplicación	39
3.38. Captura del trabajo	40
3.39. Captura de Overleaf	41
3.40. Captura de pantalla de Arduino Studio	42
3.41. Captura CircuitLab	43
3.42. Captura de EasyEDA	43
4.43. Estructura párpados	47
4.44. Ejemplo de lentes polarizadas	49
4.45. Conexión Kinect al ordenador	54
4.46. Detección imágenes	55
4.47. Estructura dentro del Kinect	55
4.48. Vista efecto Doppler	56
4.49. Escaneo OCT	58
4.50. Sensor infrarrojo	59
4.51. Detector y sensor IR	60
4.52. Señal detector-ojo	60
4.53. Fases del proyecto	61
4.54. Prototipo alpha	62
4.55. Arduino prototipo	62
4.56. Impresión montura 4 sensores	63
4.57. Primera detección	64
4.58. MARK 1	64
4.59. Evaluación online de la pulsera con el dispositivo	66
4.60. Escáner Kinect	67
4.61. Escáner Kinect	68
4.62. Montura Centinela	68
4.63. Boceto para impresora	68
4.64. División montura boceto	69
4.65. Archivos para imprimir	69
4.66. Resultado final de la montura	70
4.67. Boceto Merovingio	70
4.68. Merovingio 3D	71
4.69. Merovingio 3D	71
4.70. Merovingio	71
4.71. Merovingio	72
4.72. Medida	72
5.73. Módulo wi-fi	75
5.74. Arduino nano utilizado	76
5.75. Conectando el módulo al Arduino	76
7.76. Presentación del prototipo de ojo sólido	103
7.77. Presentation of the eye solid prototype	107
8.78. Promoción Cadena Ser	109
8.79. Foto oficial en Cadena Ser	109
8.80. Promoción en twitter e impacto	110
8.81. Entrevista Telemadrid	111
8.82. Entrevista Telemadrid II	111
8.83. Entrevista Telemadrid III	112
8.84. Entrevista Telemadrid IV	112

8.85. Enlace al Concurso Universitario de Software Libre	113
8.86. Cartel de promoción	114
8.87. Inicio de la entrevista	115
8.88. Pregunta Centinela	115
8.89. Pregunta sobre OTEA	116
1.90. Nombre piezas impresora	130
1.91. Resultado de la unión de las piezas	131
1.92. Resultado de la unión de las piezas	132
1.93. Resultado de la unión de las piezas	132
1.94. Rodamientos LM8UU	132
1.95. Rodamientos de la impresora	133
1.96. Resultado de la unión de las piezas	133
1.97. Resultado de la unión de las piezas	134
1.98. Resultado de la unión de las piezas	134
1.99. Resultado de la unión de las piezas	134
1.100 Resultado de la unión de las piezas	135
1.101 Resultado de la unión de las piezas	135
1.102 Resultado de la unión de las piezas	136
1.103 Resultado unión de las piezas	136
1.104 Resultado de la unión de las piezas	137
1.105 Resultado de la unión de las piezas	137
1.106 Resultado de la unión de las piezas	137
1.107 Resultado de la unión de las piezas	138
1.108 Resultado de la unión de las piezas	138
1.109 Resultado de la unión de las piezas	139
1.110 Resultado de la unión de las piezas	139
1.111 Resultado de la unión de las piezas	140
1.112 Finales de carrera	140
1.113 Resultado de la unión de las piezas	141
1.114 Resultado de la unión de las piezas	141
1.115 Resultado de la unión de las piezas	142
1.116 Resultado de la unión de las piezas	142
1.117 Resultado de la unión de las piezas	142
1.118 Resultado de la unión de las piezas	143
1.119 Resultado de la unión de las piezas	143
1.120 Resultado de la unión de las piezas	143
1.121 Resultado de la unión de las piezas	144
1.122 Resultado	144
1.123 Acoplamiento de la placa	144
1.124 Acoplamiento ventilador	145
1.125 Montaje pantalla LCD	145
1.126 Montaje pantalla LCD	146
1.127 Conexiones	146
1.128 Conexiones	146
1.129 Circuito Mosfet	147

Índice de cuadros

2.1.	Resultados del estudio	19
2.2.	Resultados obtenidos	27
2.3.	Estadísticas del proyecto donde GT (verdaderos parpadeos), DB (parpadeos detectados), RGT (ratio de parpadeo), RD (ratio de parpadeo detectado) y duración media	28
2.4.	Medidas realizadas con buena iluminación	31
2.5.	Medidas hechas con poca iluminación	31
3.6.	Comparación de Kanban y Scrum	38

Bibliografía

- [1] R. Iribarren, G. Iribarren, and A. Fornaciari, “Estudio de la función visual en el trabajo con computadoras,” *MEDICINA (Buenos Aires)*, vol. 62, no. 2, pp. 141–144, 2002.
- [2] A. Prado Montes, Á. Morales Caballero, M. Cassia, and J. Navor, “Síndrome de fatiga ocular y su relación con el medio laboral,” *Medicina y Seguridad del Trabajo*, vol. 63, no. 249, pp. 345–361, 2017.
- [3] W. P. Bloun, “Studies of the movements of the eyelids of animals: Blinking,” *Experimental Physiology*, vol. 18, pp. 111–125, 1927.
- [4] J. G. K. Grauman, M. Betke and G. Bradski, “Communication via eye blinks - detection and duration analysis in real time,” in *IEEE CVPR*, pp. 1010–1017, 2001.
- [5] I. M. Laura C. Trutoiu, Elizabeth J. Carter and J. sica K. Hodgins, *Modeling and animating eye blinks*, vol. 8. ACM Trans. Appl. Percep, 2011.
- [6] M. Turk and A. Pentland, “Face recognition using eigen- faces,” in *IEEE CVPR*, vol. 29, pp. 586–591, 1991.
- [7] I. Jolliffe, *Principal component analysis*. Wiley Online Li- brary, 2005.
- [8] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of Statistics*, vol. 29, pp. 1189–1232, 2000.
- [9] A. Hyv`arinen and E. Oja vol. 13, no. 4, 2000.
- [10] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol*, vol. 2, no. 3, pp. 27:1–27, 2011.
- [11] S. A.A.Mohammed, “Efficient eye blink detection method for disabled-helping domain,” (*IJACSA*) *International Journal of Advanced Computer Science and Appli- cations*, vol. 5, no. 5, 2014.
- [12] B. S. S.Naveed and M. Khyal, *Eye Tracking System with Blink Detection*, vol. 4 of *Issue 3*. Journal of Computing, 2012.
- [13] P. Viola and M. Jones, *Rapid Object Detection using a Boosted Cascade of Simple Features*, vol. 1. Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR), 2001.
- [14] J. B. M. G. J. Grauman, K. and G. R. Bradski, *Communication via Eye Blinks – Detection and Duration Analysis in Real Time*, vol. 1. Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), 2001.

-
- [15] L. G. N. T. M. Lalonde, D. Byrns and D. Laurendeau, *Real-time eye blink detection with GPU based SIFT tracking*, vol. 1. Proc. of the 4th Canadian Conf. on Computer and Robot Vision (CRV), 2007.
- [16] C. Devahasdin and T. Srinark, *A Method for Real time Eye Blink detection And Its Application*. The 6th International Joint Conference on Computer Science and Software Engineering, 2009.
- [17] T. W. E. Miluzzo and T. Campell, *EyePhone: Activating Mobile Phones With Your Eyes*. MobiHeld 2010, 2010.
- [18] K. L. et al, “Image capturing using blink detection,” *International Journal of Computer Science and Information Technologies*, vol. 6, pp. 4965–4968, 2015.
- [19] Y. W. Y.-T. W. Masaki Nakanishi, Yasue Mitsukura, *Online voluntary eye blink detection using Electrooculogram*. International Symposium on Nonlinear Theory and its Applications, 2012.
- [20] P. Królak, A.; Strumiłło, “Eye-blink detection system for human–computer interaction,” vol. 11, pp. 409–419, 2012.
- [21] M. K. A. Rahman, A.; Sirshar, “A. real time drowsiness detection using eye blink monitoring,” *In Proceedings of the 2015 National Software Engineering Conference (NSEC)*, pp. 1–7, 2015.
- [22] G. Wilson, “An analysis of mental workload in pilots during flight using multiple psychophysiological measures,” *Int. J. Aviat. Psychol*, vol. 12, pp. 3–18, 2002.
- [23] M. Viola, P.; Jones, “Rapid object detection using a boosted cascade of simple features,” *In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, p. I, 2001.
- [24] B. Malik, K.; Smolka vol. 14, 2014.
- [25] B. Radlak, K.; Smolka vol. 27, 2013.
- [26] S. O. J. Y.-S. Mohanakrishnan, J.; Nakashima vol. 15, 2013.
- [27] W. Fogelton, A.; Benesova, “Eye blink detection based on motion vectors analysis,” *Comput. Vis. Image Underst*, vol. 148, pp. 23–33, 2016.
- [28] P. Thoumie, J. Charlier, M. Alecki, D. d’Erceville, A. Heurtin, J. Mathe, G. Nadeau, and L. Wiart, “Clinical and functional evaluation of a gaze controlled system for the severely handicapped,” *Spinal Cord*, vol. 36, no. 2, p. 104, 1998.
- [29] D. P. C. F. O. Gips, J., “Using eagleeyes-an electrodes based device for controlling the computer with your eyes-to help people with special needs,” *Spinal Cord*, vol. 1, pp. 77–83, 1996.
- [30] B. A. W. J. Kocejko, T., “Eye mouse for disabled,” *Spinal Cord*, pp. 199–202, 2008.
- [31] W.-B. Horng, C.-Y. Chen, Y. Chang, and C.-H. Fan, “Driver fatigue detection based on eye tracking and dynamic template matching,” *in IEEE International Conference on Networking, Sensing and Control, 2004*, vol. 1, pp. 7–12, IEEE, 2004.

-
- [32] A. Królak and P. Strumiłło, “Fatigue monitoring by means of eye blink analysis in image sequences,” *ICSES*, vol. 1, pp. 219–222, 2006.
- [33] Z.-H. Zhou and X. Geng, “Projection functions for eye detection,” *Pattern recognition*, vol. 37, no. 5, pp. 1049–1056, 2004.
- [34] T. Kawaguchi, M. Rizon, and D. Hidaka, “Detection of eyes from human faces by hough transform and separability filter,” *Electronics and Communications in Japan (Part II: Electronics)*, vol. 88, no. 5, pp. 29–39, 2005.
- [35] J.-W. Li, “Eye blink detection based on multiple gabor response waves,” in *2008 International Conference on Machine Learning and Cybernetics*, vol. 5, pp. 2852–2856, IEEE, 2008.
- [36] G. Bradski, A. Kaehler, and V. Pisarevsky, “Learning-based computer vision with intel’s open source computer vision library.,” *Intel technology journal*, vol. 9, no. 2, 2005.
- [37] G. Bradski, A. Kaehler, and V. Pisarevsky, “Learning-based computer vision with intel’s open source computer vision library.,” *Intel technology journal*, vol. 9, no. 2, 2005.
- [38] J. J. Magee, M. R. Scott, B. N. Waber, and M. Betke, “Eyekeys: A real-time vision interface based on gaze detection from a low-grade video camera,” in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 159–159, IEEE, 2004.
- [39] G. Yang and T. S. Huang, “Human face detection in a complex background,” *Pattern recognition*, vol. 27, no. 1, pp. 53–63, 1994.
- [40] K. C. Yow and R. Cipolla, “Feature-based human face detection,” *Image and vision computing*, vol. 15, no. 9, pp. 713–735, 1997.
- [41] A. Lanitis, C. J. Taylor, and T. F. Cootes, “Automatic face identification system using flexible appearance models,” *Image and vision computing*, vol. 13, no. 5, pp. 393–401, 1995.
- [42] H. A. Rowley, S. Baluja, and T. Kanade, “Rotation invariant neural network-based face detection,” 1998.
- [43] P. Viola, M. Jones, *et al.*, “Rapid object detection using a boosted cascade of simple features,” *CVPR (1)*, vol. 1, no. 511-518, p. 3, 2001.
- [44] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *Proceedings. international conference on image processing*, vol. 1, pp. I–I, IEEE, 2002.
- [45] Y. Freund and R. Schapire, “A short introduction to boosting: Journal of japanese society for artificial intelligence, v. 14,” 1999.
- [46] J. M. Association *et al.*, *KANBAN: Y Just-in-time en Toyota*. Routledge, 2018.
- [47] M. D. Arango Serna, L. F. Campuzano Zapata, and J. A. Zapata Cortes, “Mejoramiento de procesos de manufactura utilizando kanban,” *Revista Ingenierías Universidad de Medellín*, vol. 14, no. 27, pp. 221–233, 2015.

- [48] S. Gstettner and H. Kuhn, “Analysis of production control systems kanban and conwip,” *International journal of production research*, vol. 34, no. 11, pp. 3253–3273, 1996.
- [49] O. J. P. Ortega, “Sistemas de producción tipo kanban: Descripción, componentes, diseño del sistema, y bibliografía relacionada,” *Panorama*, vol. 2, no. 6, 2013.
- [50] I. López Durbá, “Proyecto de implantación de un sistema de gestión de flujo kanban en el proyecto de fabricación de forros de embrague en valeo materiales de fricción,” 1998.
- [51] H. Kniberg, M. Skarin, and D. Anderson, “Kanban y scrum—obteniendo lo mejor de ambos,” *Prólogo de Mary Poppendieck & David Anderson. ESTADOS UNIDOS DE AMÉRICA: C4Media Inc*, 2010.
- [52] “<https://skanect.occipital.com/>,”
- [53] “http://laplace.us.es/wiki/index.php/Efecto_Doppler,”
- [54] “<https://aws.amazon.com/es/what-is-aws/>,”

Parte C

Anexos

Capítulo A

Montaje de la impresora 3D

1.1. Planos y montaje de la PRUSA i3

En este anexo se va a explicar cómo se ha realizado el montaje de la impresora, para su replicación en cualquier entorno.

Primero vamos a mostrar cómo se nombran cada pieza según el manual oficial para no perderse a la hora de realizar el montaje de la misma y luego veremos las modificaciones realizadas.

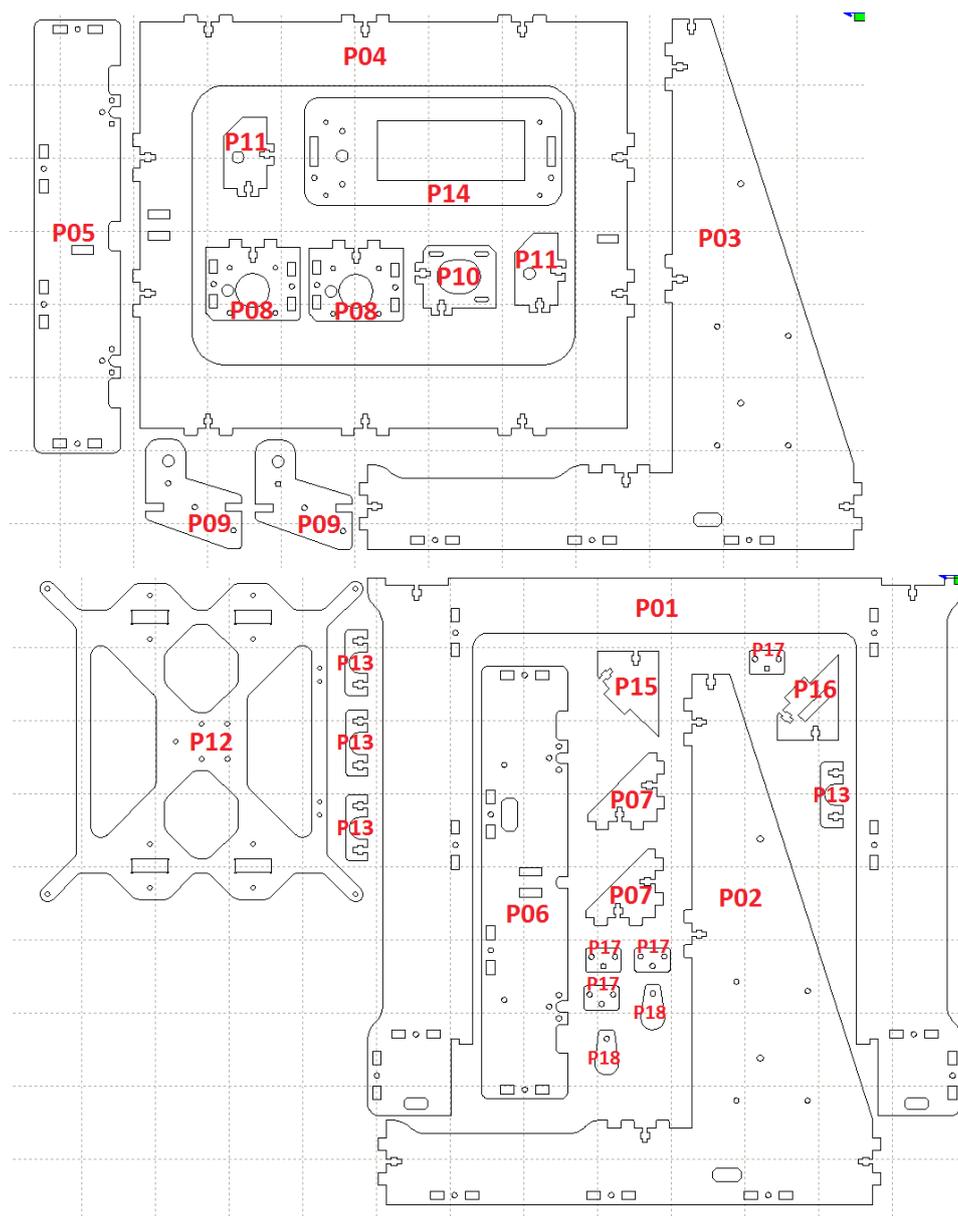


Figura 1.90: Nombre piezas impresora

Algunos de estos componentes se usarán para instalar la estructura, otros se usarán para instalar el eje x , otros se usarán para instalar el eje y , otros se usarán para instalar el eje z y el último se usará para instalar la cama calentada. Se especificará el resultado final de cada pieza.

1.1.1. Montaje de estructura

En este capítulo vamos a explicar cómo unir las piezas de la estructura principal de la impresora.

Primero, para unir la estructura principal se van a utilizar tuercas M3 y tornillos M3×12, procediendo con mucho cuidado para no romper la estructura principal. Seguidamente, para empezar con la parte delantera y trasera de la impresora vamos a utilizar las si-

güentes piezas:

- P01 para la parte delantera.
- P02 para la parte delantera.
- P03 para la parte delantera.
- P04 para la parte trasera.
- Dos piezas P07 para las uniones.
- Dos piezas P08 para las uniones.
- P06 para la parte delantera.
- Dos piezas P011 para la parte delantera.
- P05 para la parte trasera.
- P10 para la parte trasera.

Posteriormente, vamos a explicar en qué orden se realizan cada una de las uniones:

- a). Unir P01 con P02, P03 y por la parte trasera, P04.
- b). Colocar piezas P05 y P06 en la parte trasera.



Figura 1.91: Resultado de la unión de las piezas

- c). Unir pieza P07 con dos piezas P08, orientando los agujeros 90 grados, quedando perpendiculares ambas piezas.
- d). Poner la pieza P09 en la parte superior de la estructura resultante anterior.



Figura 1.92: Resultado de la unión de las piezas

- e). Unir la pieza P06 con dos piezas P11 para la parte delantera mediante cuatro tornillos M3×12 y cuatro tuercas M4.
- f). Unir la pieza P05 con una pieza P10 para la parte trasera mediante dos tornillos M3×12 y dos tuercas M3.

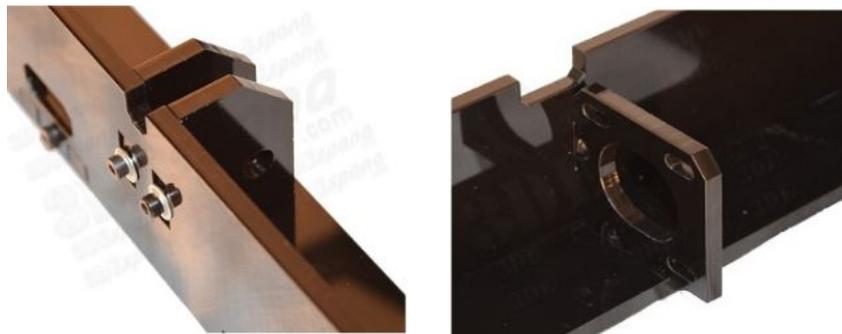


Figura 1.93: Resultado de la unión de las piezas

Eje *y*

Para el eje *y* utilizaremos los siguientes rodamientos:



Figura 1.94: Rodamientos LM8UU

Los cuales uniremos a la bandeja P12 mediante cuatro piezas P13. Quedando como resultado lo siguiente:



Figura 1.95: Rodamientos de la impresora

Después de realizar el montaje de los rodamientos necesitaremos las siguientes piezas:

- Varillas Y340 para unir la bandeja con la parte superior.
- Cuatro piezas P17 para asegurar la parte inferior.
- Rodamiento 608ZZ en la parte derecha.
- Una pieza de sujeción para la bandeja.

Seguidamente, vamos a explicar paso a paso como realizar el montaje del eje *y* completo:

- a). Unimos los rodamientos de la bandeja P12 a las piezas superiores P05/P06 mediante dos varillas Y340 con cuidado de no partir ninguna de las varillas.



Figura 1.96: Resultado de la unión de las piezas

- b). Atornillar las cuatro piezas P17 a la parte inferior mediante tornillos M3×12 a la pieza P06.



Figura 1.97: Resultado de la unión de las piezas

- c). Colocar rodamiento 608ZZ entre la pieza P11 mediante dos arandelas M8, sujetas por un tornillo M8×25 y una tuerca M8 autoblocante.
- d). Acoplar una polea GT2 a la pieza P10 mediante tres tornillos M3×10 y acoplarle un motor Nema 17.

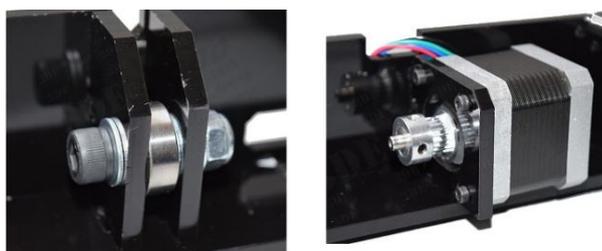


Figura 1.98: Resultado de la unión de las piezas

- e). Asegurar la parte central mediante una pieza Y-sujección correa, cuatro tornillos M3×25 y tuercas M3.
- f). Pasar la correa GT2 acoplada junto con el motor y los rodamientos a cada una de las correas. Este paso hay que realizarlo con mucho cuidado y tensando bien las correas.



Figura 1.99: Resultado de la unión de las piezas

Eje x

Primera parte

Para el montaje en el eje x necesitaremos las siguientes piezas:

- Dos piezas de moto tipo X-Motor y X-Idler.
- Dos rodamientos LM8UU.
- Una pieza tensora X-Tensor.
- Dos varillas X400.
- Una pieza X-Carriage.

Seguidamente, pasaremos a explicar cómo realizar el montaje en el eje x :

- a). Para montar los motores que varían con el eje x necesitaremos los rodamientos LM8UU y las piezas de los motores en X. Para ello, uniremos los rodamientos LM8UU a las piezas X-Motor y X-Idler mediante la tuerca M5.



Figura 1.100: Resultado de la unión de las piezas

- b). Unir la pieza X-Tensor a la polea mediante un tornillo M3×20 y una tuerca M3. La tuerca debe ir en la parte superior de la pieza y se recomienda para evitar que se salga se debe aplicar un poco de pegamento a la tuerca.



Figura 1.101: Resultado de la unión de las piezas

- c). Colocar el X-Tensor junto con el X-Idler en la posición adecuada mediante un tornillo M3×30, una arandela M3 tras la cabeza del tornillo y la tuerca M3 colocada en X-Tensor.



Figura 1.102: Resultado de la unión de las piezas

- d). Introducir las varillas X400 a través de los huecos del X-Motor y los cuatro rodamientos LM8UU a través del X-Carriage.
- e). Terminamos uniendo los rodamientos a través de las varillas a la pieza X-Idler. Es importante comprobar que todos los elementos estén alineados.



Figura 1.103: Resultado unión de las piezas

Segunda parte

En esta segunda parte del eje x vamos a utilizar las siguientes piezas:

- Motor Nema 17.
- Polea GT2.
- X-Motor.
- X-Carriage.
- X-Tensor.
- Correa GT2.

A continuación vamos a terminar de realizar el montaje de la eje x :

- a). Acoplamos el motor Nema 17 a la polea GT2 y lo atornillaremos a la pieza X-Motor mediante tres tornillos M3×8.

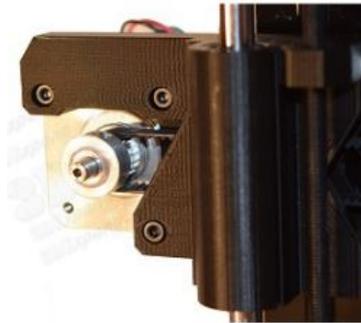


Figura 1.104: Resultado de la unión de las piezas

- b). Unir la correa a uno de los huecos del X-Carriage mediante el extremo de la brida.
- c). Pasar la correa por el rodamiento del tensor de la pieza X-Idler y por la polea GT2. Fijar el extremo en el segundo hueco de X-Carriage con la máxima tensión posible.

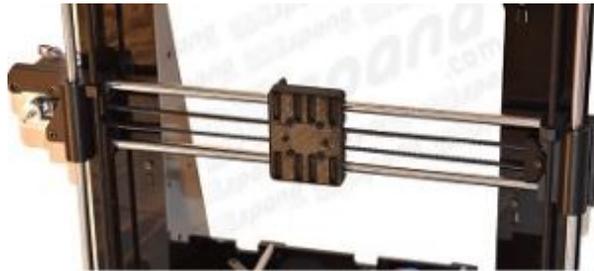


Figura 1.105: Resultado de la unión de las piezas

- d). Tensar la correa GT2 atornillando el tornillo X-Idler que tira de X-Tensor.



Figura 1.106: Resultado de la unión de las piezas

Eje z

Para montar el eje z vamos a utilizar las siguientes piezas:

- Dos motores Nema 17.
- Piezas P08.
- Varillas Z320.
- Piezas P09.
- Piezas P18.
- Varillas roscadas.
- Acopladores.

A continuación vamos a explicar cómo montar el eje z:

- a). Acoplamos los dos motores Nema 17 bajo las piezas P08 mediante tornillos M3×8.

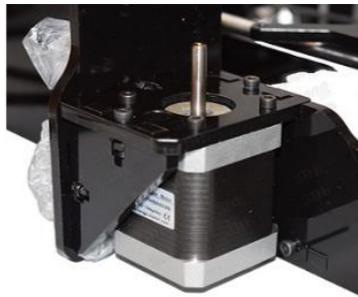


Figura 1.107: Resultado de la unión de las piezas

- b). Introducir las varillas Z320 en las piezas P09. Además, introducir las varillas en los rodamientos.
- c). Colocar las dos piezas P18 en las varillas con los tornillos M3×12 y tuercas M3.



Figura 1.108: Resultado de la unión de las piezas

- d). Introducir las dos varillas roscadas M5×300 con las tuercas M5 en las piezas X-Motor y X-Idler.
- e). Unir las varillas roscadas a los ejes de los motores mediante los acopladores.



Figura 1.109: Resultado de la unión de las piezas

Cama caliente

Para la cama caliente necesitaremos las siguientes piezas:

- Fuente de 12V.
- Dos cables de hilos.
- Cinta Kapton.
- Muelles M3.

Y vamos a mostrar cómo se monta la cama caliente:

- a). Soldar el cable negativo en los espacios 2 y 3 de la fuente.
- b). Soldar el cable positivo en el espacio 1 de la fuente.

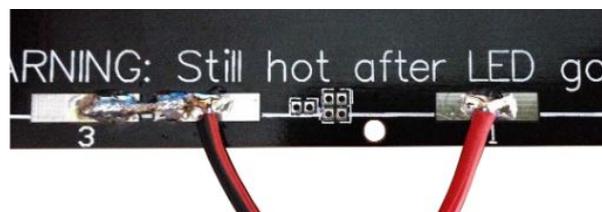


Figura 1.110: Resultado de la unión de las piezas

- c). Colocar el termistor en la parte inferior de la cama y sujetarla por la parte superior con la cinta Kapton.
- d). Unir la cama caliente a la bandeja mediante los tornillos M3×35, cuatro muelles M3, cuatro arandelas y cuatro tuercas M3. Dejamos los cables orientados hacia atrás.

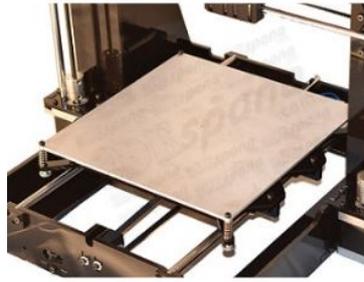


Figura 1.111: Resultado de la unión de las piezas

Finales de carrera

Hay que poner un final de carrera en cada eje, en la posición 0.



Figura 1.112: Finales de carrera

Sobre los finales de carrera a los cables colocaremos estaño siguiendo el siguiente esquema:

Número	Final de carrera	Ramps 1.4
1	C o COM	S
2	NC	+(VACIO)
3	NO	-

Si tenemos solo dos pines conectaremos los cables en las posiciones 1 y 3.

En la parte del final de cada carrera colocaremos un soporte de plástico. Para el eje x utilizaremos la pieza Final de Carrera-X. Colocaremos esta pieza sobre la varilla izquierda debajo del eje. Hay que tener cuidado a acoplarlo de no mover mucho la varilla para que

no se rompa. También añadir la pieza Final de Carrera-Z sobre la varilla lisa del lado izquierda del eje z .



Figura 1.113: Resultado de la unión de las piezas

Finalmente, para el Final de Carrera-Y colocaremos la varilla en la parte trasera de la impresora. La fijaremos con una brida o con un poco de pegamento.



Figura 1.114: Resultado de la unión de las piezas

Extrusor

Para montar el extrusor necesitaremos las siguientes piezas:

- Dos rodamientos 608zz.
- Cuerpo del Extrusor.
- Polea Extrusor.
- Motor Nema 17.
- Polea Principal Extrusor.
- Pieza X-Carriage.

A continuación vamos a ver cómo realizar el montaje del extrusor:

- a). Introducir en el extrusor los dos rodamientos 608zz mediante un tornillo $M3 \times 30$ y una tuerca $M3$.



Figura 1.115: Resultado de la unión de las piezas

- b). Unir la Polea Extrusor a un motor Nema y atornillarla al cuerpo de extrusor mediante tres tornillos M3×12.



Figura 1.116: Resultado de la unión de las piezas

- c). Pasar el tornillo del extrusor por la pieza Polea Principal Extrusor y afianzarlas con seis arandelas M8.
d). Es importante que las poleas deberán coincidir con la posición del motor.



Figura 1.117: Resultado de la unión de las piezas

- e). Introducir la varilla lisa a través de un rodamiento 608zz e introducirla en el Cierre Extrusor.
f). Para conseguir que la pieza haga presión utilizaremos dos tornillos M3×40, dos muelles, cuatro arandelas y dos tuercas M3.



Figura 1.118: Resultado de la unión de las piezas

- g). Acoplar el ventilador en la parte superior del extrusor sobre el disipador y atornillarlo al ventilador mediante dos tornillos M3×16.



Figura 1.119: Resultado de la unión de las piezas

- h). Meter a presión el HotEnd dentro del hueco inferior del hueco. Utilizar dos tornillos M3×30 y dos tuercas M3.



Figura 1.120: Resultado de la unión de las piezas

- i). Añadir el extrusor a la pieza X-Carriage mediante cuatro tornillos M3×25 y dos tuercas M3.

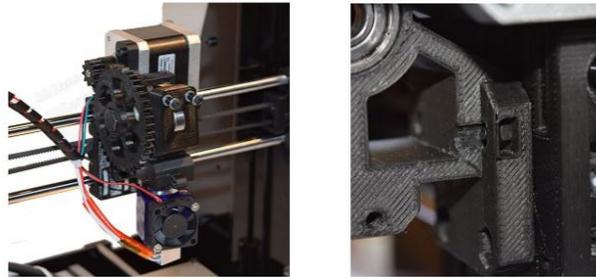


Figura 1.121: Resultado de la unión de las piezas

Placa electrónica

Acoplar el Arduino Mega la Ramps 1.4 con lo siguiente:

- Jumpers.
- Stepper Drivers DRV8825.
- Disipadores.

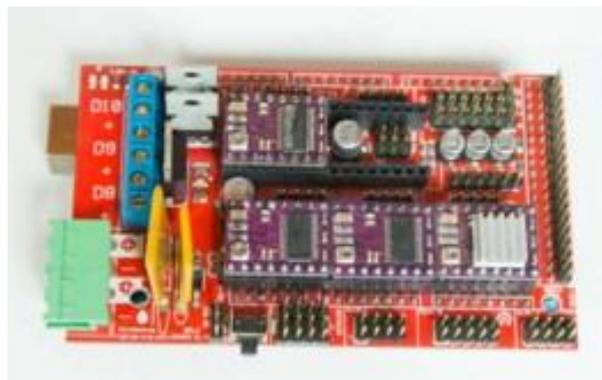


Figura 1.122: Resultado

Con el potenciómetro regulamos a 1100 Mv el extrusor y los ejes a 850 Mv. Atornillar las placas a las piezas P02 o P03 mediante tres tornillos M3×35 y tres tuercas M3. Colocar entre la placa y la pieza lateral los separadores de plástico.

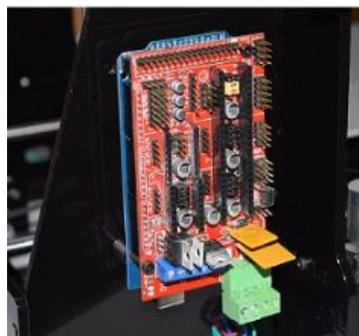


Figura 1.123: Acoplamiento de la placa

Colocar en la placa el ventilador mediante cuatro tornillos M3×16 y cuatro tuercas M3.

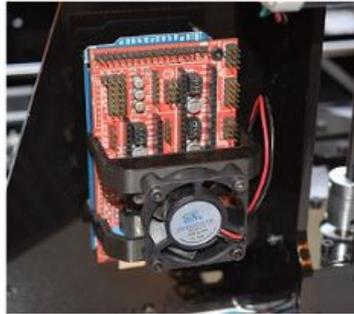


Figura 1.124: Acoplamiento ventilador

Pantalla LCD

Para la pantalla LCD utilizaremos las siguientes piezas:

- Piezas P15.
- Piezas P16.
- Piezas P06.
- Pantalla LCD.

Para el montaje de la pantalla realizaremos los siguientes pasos:

- a). Unir las piezas P15 y P16 a la pieza P14 mediante dos tornillos M3×12 y dos tuercas M3.
- b). Atornillar la pantalla a esta pieza mediante cuatro tornillos M3×18 y cuatro tuercas M3.

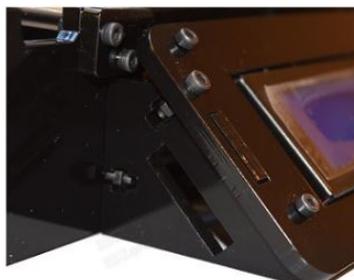


Figura 1.125: Montaje pantalla LCD

- c). Atornillar esta estructura a la pieza P06 mediante dos tornillos M3×12 y dos tuercas M3.
- d). Conectar la pantalla LCD a través de los pines situados al final de la placa.



Figura 1.126: Montaje pantalla LCD

Conexiones y soldaduras

Una vez realizado el montaje conectaremos los cables a la placa de la siguiente forma:

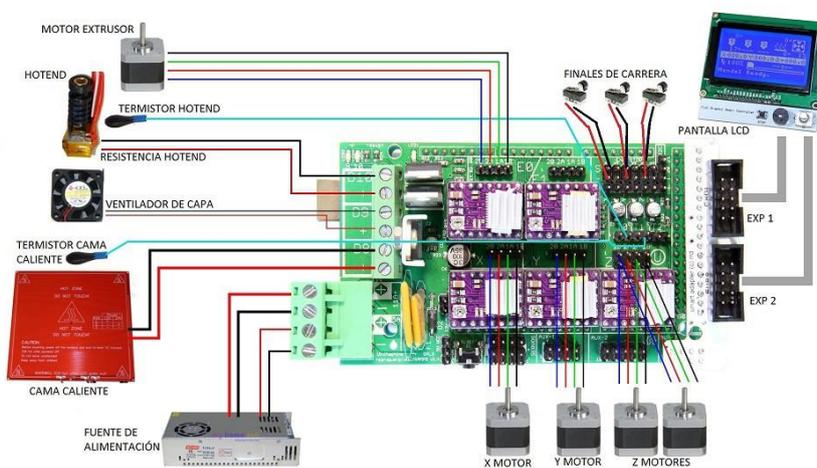


Figura 1.127: Conexiones

Es muy importante invertir el conector del motor en la Ramps de la siguiente forma:

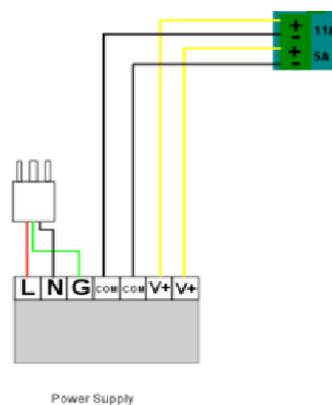


Figura 1.128: Conexiones

La fuente de alimentación va conectada al Ramps mediante dos cables dobles, uno positivo y negativo que va al 11A y uno negativo y positivo que va al 5A. Para enchufar la fuente necesitaremos uno de línea y otro neutro. Los ventiladores van a los huecos 5A y 11A.

1.2. Mosfet externo para impresora 3D

El mosfet se coloca en la impresora para calentar la cama, para dispersar la potencia de forma adecuada. Tiene una capacidad de 25 Amperios, con un disipador puede manejar aún más corriente. Los componentes del mosfet son los siguientes:

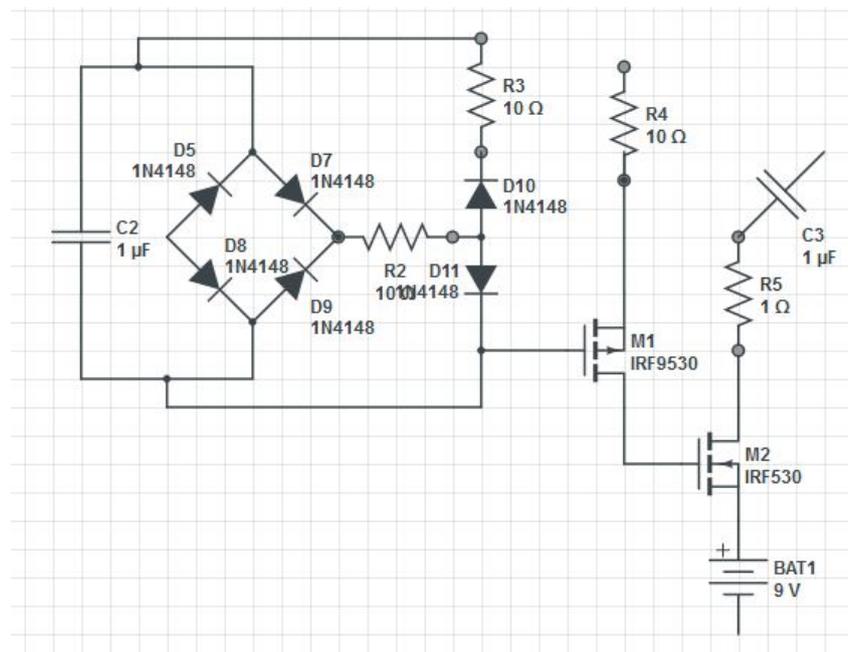


Figura 1.129: Circuito Mosfet

Está formado por una serie de barreras que impiden la entrada de voltaje y son conectadas a la fuente de alimentación, también tiene una serie de módulos que van conectados a la cama. Los cables de control no tienen orden, no importa cuál es el positivo o el negativo. El conector está conectado a los diodos, llevan a la corriente en un solo sentido. Tampoco importa el orden del positivo al negativo para conectarlo a la cama. Si no se calienta la cama, puede ser por el mosfet y se necesitará un mosfet adicional para calentarla. Para calentar la cama se necesita un cable de alto calibre, de calibre 12, para que no se queme el cable. Para emitir la luz que nos dice que está caliente la cama utilizaremos un LED.

Capítulo B

Software usado y final

1.1. Software y resultados

Todo el software y resultados: <https://drive.google.com/drive/u/3/folders/1h2AuzVlau75Qbe78tsIQnLV1HCsSKHGi>

1.2. Creaciones propias y códigos fuente

Contiene:

- Código HanSolo
- Código milis1
- Código milis2
- Código de simulación para Twitter
- Código inicial de Centinela
- Código final Centinela en Python-LaTeX

<https://drive.google.com/drive/folders/1j8Wm5y-EbQsRivFCJsHX66Q7Vj29MSj5?usp=sharing>

Si puedes imaginarlo, puedes crearlo
Magic Kaito

David Pacios Izquierdo

2019–2020

Ult. actualización 24 de junio de 2020

L^AT_EX lic. LPPL & powered by **TEFLON** CC-ZERO

Este documento esta realizado bajo licencia Creative Commons “CC0 1.0 Universal”.

