

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computacionales

**DESARROLLO DE UN SISTEMA EMBEBIDO MEDIANTE EL USO DE
TÉCNICAS DE VISIÓN ARTIFICIAL PARA DETECCIÓN Y ALERTA DE
SOMNOLENCIA EN CONDUCCIÓN DIURNA EN TIEMPO REAL**

Trabajo de grado previo a la obtención del título de Ingeniero en Sistemas
Computacionales

Autor:
Josué Martín Alba Neppas

Director:
PhD. Iván Danilo García Santillán

Ibarra – Ecuador
Enero 2020



UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	172443261-0		
APELLIDOS Y NOMBRES:	ALBA NEPPAS JOSUÉ MARTÍN		
DIRECCIÓN:	GUAYLLABAMBA Y RIO AGÜARICO		
EMAIL:	jmalban@utn.edu.ec		
TELÉFONO FIJO:	2-603-991	TELÉFONO MÓVIL:	0985149455

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE UN SISTEMA EMBEBIDO MEDIANTE EL USO DE TÉCNICAS DE VISIÓN ARTIFICIAL PARA DETECCIÓN Y ALERTA DE SOMNOLENCIA EN CONDUCCIÓN DIURNA EN TIEMPO REAL
AUTOR (ES):	JOSUÉ MARTÍN ALBA NEPPAS
FECHA: DD/MM/AAAA	27/01/2020
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERÍA EN SISTEMAS COMPUTACIONALES
ASESOR / DIRECTOR:	PhD. IVÁN GARCÍA

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 27 días del mes de enero del 2020.

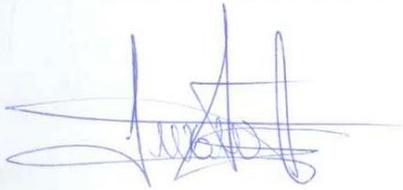
EL AUTOR:

(Firma): 
Nombre: Josué Martín Alba Neppas

Certificado tutor

En mi calidad de tutor del Trabajo de Grado presentado por el egresado **JOSUÉ MARTÍN ALBA NEPPAS** para optar por el título de Ingeniero en Sistemas Computacionales cuyo tema es: **DESARROLLO DE UN SISTEMA EMBEBIDO MEDIANTE EL USO DE TÉCNICAS DE VISIÓN ARTIFICIAL PARA DETECCIÓN Y ALERTA DE SOMNOLENCIA EN CONDUCCIÓN DIURNA EN TIEMPO REAL**. Considero que el presente trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que se designe.

En la ciudad de Ibarra, a los 23 días del mes de enero del 2020.

A handwritten signature in blue ink, appearing to be 'Iván Danilo García Santillán', written over a light blue background.

PhD. Iván Danilo García Santillán
TUTOR TRABAJO DE GRADO

Autoría

Yo, JOSUÉ MARTÍN ALBA NEPPAS, portador de la cédula de ciudadanía número 172443261-0, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, **DESARROLLO DE UN SISTEMA EMBEBIDO MEDIANTE EL USO DE TÉCNICAS DE VISIÓN ARTIFICIAL PARA DETECCIÓN Y ALERTA DE SOMNOLENCIA EN CONDUCCIÓN DIURNA EN TIEMPO REAL**, que no ha sido previamente presentado para ningún grado, ni calificación profesional, y que se han respetado las diferentes fuentes y referencias.



JOSUE MARTÍN ALBA NEPPAS

C.I: 172443261-0

Dedicatoria

Este proyecto de titulación va dedicado a mi madre Virginia Neppas Alba, quien, con su infinito amor, paciencia y arduo esfuerzo, me supo guiar por el camino del bien, me motivó a continuar preparándome y me enseñó el valor de la humildad y la constancia.

A mi padre Martín Alba, quien con su rectitud y sabiduría supo formar los cimientos de mi vida y quien desde el cielo de seguro siempre me estuvo cuidando.

Va dedicado también para Mishel Cevallos, quien ha estado a mi lado apoyándome en todo sentido y en todo momento de forma incondicional y sincera.

Agradecimientos

Agradezco primeramente a Dios mi padre celestial, por la vida, por la salud, por la protección y por la oportunidad que me ha dado de poder alcanzar tan anhelado deseo. Además, por haberme bendecido con unos padres sabios y amorosos quienes han sido los pilares fundamentales de mi vida.

A Mishel Cevallos por su apoyo incondicional en esta etapa importante de mi vida.

A mis opositores, Ing. Erick Herrera, Ing. Marco Pusdá, quienes me han dado las pautas necesarias para poder hacer un trabajo de titulación innovador, y en especial a mi tutor de tesis Dr. Iván García, quien con su paciencia y conocimientos ha hecho posible el cumplimiento de todas las metas propuestas en este proyecto y ha dejado sembrada la semilla del ímpetu por la investigación e innovación en el campo de la Inteligencia Artificial.

A todos los docentes quienes compartieron sus conocimientos a lo largo de mi carrera y más que docentes, han llegado a ser amigos y colegas de profesión.

A todos mis compañeros con quienes, en más de una ocasión, hemos podido compartir momentos de amistad y trabajo en equipo, ganando experiencia y logrando así llegar juntos a la meta.

A esta prestigiosa Universidad, generadora de conocimiento y símbolo de excelencia, la cual forma profesionales con ética y responsabilidad social, buscando siempre estar a la vanguardia.

Tabla de contenido

Certificado tutor.....	iii
Autoría	iv
Dedicatoria.....	v
Agradecimientos	vi
INTRODUCCIÓN	xv
Antecedentes	xv
Situación Actual.....	xvi
Prospectiva	xvii
Planteamiento del Problema.....	xvii
Árbol de problema causa – efecto	xviii
Objetivos	xviii
Objetivo General.....	xviii
Objetivos Específicos	xviii
Justificación.....	xix
Alcance	xix
Contexto.....	xx
CAPÍTULO 1	1
1.1. La Fatiga	1
1.1.1. Signos de fatiga en el conductor	1
1.2. Somnolencia	1
1.2.1. Riesgos que presenta la somnolencia.....	1
1.3. Métodos de detección de fatiga.....	2
1.3.1. Métodos de detección subjetiva	2
1.3.2. Métodos de detección objetiva	2
1.4. El ojo Humano.....	6
1.4.1. Parpadeo o pestañeo	6
1.4.2. Frecuencia de parpadeo.....	6
1.4.3. Tiempo de duración del parpadeo	6
1.5. Visión Artificial.....	7
1.5.1. Procesamiento digital de imágenes.....	7
1.6. Paquetes de procesamiento de imágenes.....	10
1.6.1. Librería OpenCV	10
1.7. Dispositivo embebido	10
1.8. Detección de rostros	11
1.8.1. Detección de rostros usando Haar - Cascade	12
1.9. Librería DLIB	13
1.9.1. Detector de hitos faciales de dlib.....	13
1.10. Metodología de desarrollo.....	14
1.10.1. Fases de la metodología XP	15
CAPÍTULO 2	17
2.1 Fase de planificación.....	17
2.1.1. Historias de usuario.....	17
2.2. Fase de diseño.....	21
2.2.1. Arquitectura.....	21
2.2.2. Componentes del sistema	21
2.2.3. Diagrama del detector de somnolencia	22

2.3.	Fase de desarrollo.....	23
2.3.1.	Instalación del SO para Raspberry Pi 3B+	23
2.3.2.	Instalación de librerías	24
2.3.3.	Habilitación de la cámara Pi.....	27
2.3.4.	Localización de la cara en la imagen.....	28
2.3.5.	Detección de estructuras faciales clave.....	31
2.3.6.	Ecuación para el cálculo de la relación del aspecto ocular (EAR)	33
2.3.7.	Cálculo del EAR mediante la distancia euclidiana	34
2.3.8.	Pruebas del cálculo del EAR en tiempo real.....	35
2.3.9.	Observaciones del umbral del EAR para la detección de somnolencia	37
2.3.10.	Propuesta para la configuración del umbral del EAR.....	38
2.3.11.	Medidas de tendencia central para los valores del EAR	39
2.3.12.	Media geométrica para la estimación más precisa del EAR	40
2.3.13.	Configuración dinámica del umbral del EAR.....	41
2.3.14.	Tamaño de la muestra para el cálculo del EAR.....	42
2.3.15.	Cálculo de la media geométrica, definición del umbral y detección de somnolencia en tiempo real	44
2.3.16.	Inicio automático del detector de somnolencia	46
2.3.17.	Estructura del Hardware.....	48
2.4.	Fase de pruebas	49
2.4.1.	Pruebas en ambiente controlado.....	50
2.4.2.	Pruebas de temperatura y consumo de recursos del sistema	52
2.4.3.	Pruebas de consumo de corriente.....	54
CAPÍTULO 3	57
3.1.	Ubicación de la cámara	57
3.2.	Alimentación del sistema.....	58
3.3.	Consideraciones importantes	59
3.3.1.	Iluminación.....	59
3.3.2.	Óptica de la cámara	60
3.4.	Matriz de confusión	62
3.4.1.	Ambiente controlado	64
3.4.2.	Ambiente real de conducción	67
3.5.	Resultados	70
3.5.1.	Análisis de resultados	72
Conclusiones	75
Recomendaciones	76
Glosario De Términos	77
Bibliografía	78
Anexos	83

Índice de Figuras

Fig. 1. Árbol de problema causa – efecto	xviii
Fig. 2. Estructura del prototipo	xx
Fig. 3. Métodos de detección de fatiga.....	3
Fig. 4. Cubo del espacio de color RGB	8
Fig. 5. Modelo de las técnicas de pre - procesamiento digital de imágenes	9
Fig. 6. Características de Haar.....	12
Fig. 7. Identificación de la cara por medio de Haar	13
Fig. 8. Visualización de las 68 coordenadas de las estructuras faciales en la cara	14
Fig. 9. Fase 1 Metodología XP	15
Fig. 10. Fase 2 Metodología XP	15
Fig. 11. Arquitectura del prototipo para detección de somnolencia en conducción diurna ...	21
Fig. 12. Flujograma del detector de somnolencia	23
Fig. 13. Interfaz del sistema operativo Raspbian.....	24
Fig. 14. Instalación del administrador de paquetes Python	25
Fig. 15. Instalación de OpenCV	26
Fig. 16. Activación de la cámara Pi	28
Fig. 17. Tamaño mínimo recomendado del objeto observado según la norma europea "EN 50132-7"	29
Fig. 18. Detección de rostros en imagen con baja resolución de 54 x 40 píxeles.....	30
Fig. 19. (Izquierda) imagen original en modo RGB. (Derecha) imagen convertida a escala de grises	31
Fig. 20. Detección de rostros con <i>Haar Cascade</i> y sus coordenadas (x,y) dentro de la imagen	31
Fig. 21. Visualización 68 coordenadas asignadas a las estructuras faciales de la cara.....	32
Fig. 22. Visualización de los índices de las 68 coordenadas	33
Fig. 23. Ojos abiertos y cerrados con puntos de referencia faciales.....	33
Fig. 24. Estructuras faciales de los ojos definidas por DLIB	34
Fig. 25. Representación gráfica de los valores del EAR en estado "Despierto"	36
Fig. 26. Representación gráfica de los valores del EAR en estado "Medio dormido".....	36
Fig. 27 Representación gráfica de los valores del EAR en estado "Dormido"	37
Fig. 28 Ejemplos de tamaño de ojos humanos.....	37
Fig. 29 Registro de valores del EAR, cálculo de MG y cálculo del umbral para Josué Alba	45
Fig. 30 Monitoreo diurno en estado "Despierto, medio dormido y dormido" – Josué Alba ...	46
Fig. 31. Monitoreo nocturno en estado "Despierto, medio dormido y dormido" – Josué Alba	46
Fig. 32 Creación del archivo script <i>autoiniciosomnolencia.sh</i>	46
Fig. 33 Instrucciones del script para ejecutar el detector de somnolencia	47
Fig. 34 Configuración del script como ejecutable	47
Fig. 35 Acceso al archivo <i>crontab</i>	47
Fig. 36 Instrucción para ejecución automática del script en el reinicio del sistema.....	48
Fig. 37. Pines GPIO para Raspberry Pi Modelo 3B+	49
Fig. 38 Registro de valores del EAR, cálculo de MG y cálculo del umbral para Erick Herrera	51
Fig. 39 Monitoreo diurno en estado "Despierto y dormido" – Erick Herrera	51
Fig. 40. Monitoreo diurno de una voluntaria en estado despierto y dormido.....	52
Fig. 41 Estado de CPU, RAM y temperatura sin ejecución del detector de somnolencia	52

Fig. 42 Estado de CPU, RAM y temperatura luego de 10 minutos de ejecutar el detector de somnolencia.....	53
Fig. 43 Estado de CPU, RAM y temperatura luego de 30 minutos de ejecutar el detector de somnolencia.....	53
Fig. 44 Estado de CPU, RAM y temperatura luego de 40 minutos de ejecutar el detector de somnolencia.....	54
Fig. 45. Ubicación de la cámara dentro de la cabina del auto	57
Fig. 46. Pruebas en ambiente real de conducción - Ing. Erick Herrera.....	58
Fig. 47. Cargador de coche de 5VDC 4Amp, dual micro USB.....	58
Fig. 48. Elementos usados en las pruebas de conducción.....	59
Fig. 49. Influencia de la contraluz sobre la imagen de entrada (El sistema no detecta los hitos faciales)	60
Fig. 50. Cámara web USB	61
Fig. 51. Diferencia de la distancia focal.....	61
Fig. 52. Cámara Pi (izquierda), cámara web USB (derecha)	62
Fig. 53. Imagen de entrada sin ningún tratamiento – El sistema no detectó ningún rostro ..	73
Fig. 54. Imagen de entrada convertida a escala de grises - El sistema no detectó ningún rostro	73
Fig. 55. Imagen de entrada sometida a ecualización de histograma - El sistema no detectó ningún rostro.....	74
Fig. 56. Imagen de entrada sometida a ecualización de histograma adaptativo - El sistema detectó un rostro.....	74

Índice de Tablas

TABLA 1.1 COMPARATIVA DE DISPOSITIVOS EMBEBIDOS MÁS POPULARES	11
TABLA 1.2 EVALUACIÓN METODOLOGÍAS ÁGILES	15
TABLA 2.1 EQUIPO DE TRABAJO	17
TABLA 2.2 HISTORIAS DE USUARIO	17
TABLA 2.3 ANÁLISIS DE CRITERIOS PARA LA PRIMERA ITERACIÓN	18
TABLA 2.4 ANÁLISIS DE CRITERIOS PARA LA SEGUNDA ITERACIÓN	18
TABLA 2.5 ANÁLISIS DE CRITERIOS PARA LA TERCERA ITERACIÓN	18
TABLA 2.6 HISTORIA DE USUARIO 1	18
TABLA 2.7 HISTORIA DE USUARIO 2	19
TABLA 2.8 HISTORIA DE USUARIO 3	19
TABLA 2.9 HISTORIA DE USUARIO 4	19
TABLA 2.10 HISTORIA DE USUARIO 5	19
TABLA 2.11 HISTORIA DE USUARIO 6	20
TABLA 2.12 HISTORIA DE USUARIO 7	20
TABLA 2.13 HISTORIA DE USUARIO 8	20
TABLA 2.14 HISTORIA DE USUARIO 9	20
TABLA 2.15 COMPONENTES DEL SISTEMA.....	22
TABLA 2.16 VARIABLES EN LAS QUE SE ALMACENA LA DISTANCIA EUCLIDIANA.....	35
TABLA 2.17 MEDIDAS DE TENDENCIA CENTRAL PARA LOS DATOS RECOLECTADOS.....	39
TABLA 2.18 COMPARATIVA ENTRE MEDIA ARITMÉTICA Y MEDIA GEOMÉTRICA PARA LOS VALORES RECOLECTADOS	41
TABLA 2.19 PORCENTAJE DE APERTURA DE LOS OJOS PARA EL ESTADO "DESPIERTO", "MEDIO DORMIDO" Y "DORMIDO"	41
TABLA 2.20 NIVEL DE CONFIANZA	43
TABLA 2.21 RESUMEN DE TEMPERATURA Y CONSUMO DE RECURSOS DEL SISTEMA	54
TABLA 2.22 VALORES DE CONSUMO DE CADA DISPOSITIVO.....	55
TABLA 3.1. ESTRUCTURA DE LA MATRIZ DE CONFUSIÓN	63
TABLA 3.2. MATRIZ DE CONFUSIÓN ING. MARCO PUSDÁ	64
TABLA 3.3. MATRIZ DE CONFUSIÓN BRAYAN VALENCIA.....	64
TABLA 3.4. MATRIZ DE CONFUSIÓN KARLA HERRERA	64
TABLA 3.5. MATRIZ DE CONFUSIÓN VANESSA CADENA	65
TABLA 3.6. MATRIZ DE CONFUSIÓN JOSSELIN ENRÍQUEZ	65
TABLA 3.7. MATRIZ DE CONFUSIÓN JOSTIN ROLDÁN.....	65
TABLA 3.8. MATRIZ DE CONFUSIÓN DANIELA PUENTESTAR.....	66
TABLA 3.9. MATRIZ DE CONFUSIÓN KATHERINE RODRÍGUEZ	66
TABLA 3.10. MATRIZ DE CONFUSIÓN CHARLIE DARMÍ	66
TABLA 3.11. MATRIZ DE CONFUSIÓN FRANKLIN VALLEJO.....	67
TABLA 3.12. MATRIZ DE CONFUSIÓN JOSUÉ ALBA	67
TABLA 3.13. MATRIZ DE CONFUSIÓN JESSICA CEVALLOS	67

TABLA 3.14. MATRIZ DE CONFUSIÓN ARMANDO GUAMÁN	68
TABLA 3.15. MATRIZ DE CONFUSIÓN RICHARD GUERRERO	68
TABLA 3.16. MATRIZ DE CONFUSIÓN DANIEL GUERRERO.....	68
TABLA 3.17. MATRIZ DE CONFUSIÓN DIEGO MONTENEGRO.....	69
TABLA 3.18. MATRIZ DE CONFUSIÓN FRANCISCO VINUEZA.....	69
TABLA 3.19. MATRIZ DE CONFUSIÓN FRANKLIN VALLEJO.....	69
TABLA 3.20. MATRIZ DE CONFUSIÓN FERNANDO CHAFUELÁN	70
TABLA 3.21. MATRIZ DE CONFUSIÓN MISHEL CEVALLOS	70
TABLA 3.22. RESULTADO DE LAS PRUEBAS EN AMBIENTE CONTROLADO Y AMBIENTE REAL DE CONDUCCIÓN	71

Resumen

Los accidentes de tránsito debido a la somnolencia representan una preocupante estadística en el país, la cual repercute en pérdidas económicas, materiales, lesiones graves y en el peor de los casos hasta la muerte. Es así como, en esta investigación se aborda el desarrollo de un prototipo que tiene como función detectar y alertar la presencia de somnolencia en el conductor en ambientes diurnos. Para ello, se implementó un sistema de visión artificial en un dispositivo embebido Raspberry Pi 3 B+ (mini ordenador), el cual procesa las imágenes obtenidas a través de una cámara web RGB, ubica el rostro y los ojos del conductor, y calcula el porcentaje de apertura de sus ojos (índice EAR), y si éste valor es más bajo que un umbral generado dinámicamente al arrancar el sistema, el dispositivo genera una alerta sonora mientras dura el estado de somnolencia del conductor. El uso de la información visual, la cual se considera la más fiable, permitió construir un dispositivo no invasivo, no intrusivo y económico, trabajando además en tiempo real pese a las limitaciones computacionales del mismo. Las pruebas realizadas en varios voluntarios (incluso con el uso de gorra y lentes), permitieron probar su funcionamiento ante diversos escenarios y su rendimiento se evaluó cuantitativamente por medio de la matriz de confusión, obteniendo una exactitud del 88.25% para ambientes controlados y 87% para ambientes reales de conducción diurna.

Palabras claves: sistema embebido, visión artificial, somnolencia, detección de rostros, puntos de referencia faciales, Python, OpenCV, Raspberry Pi, dlib, EAR.

Abstract

Traffic accidents by drowsiness represent a worrying statistic in the country, which has an impact on economic, material losses, serious injuries and until death in the worst case. Thus, this research addresses the development of a prototype whose function is to detect and alert the presence of drowsiness in the driver in daytime environments. For this, an artificial vision system was implemented in a Raspberry Pi 3 B+ embedded device (minicomputer), which processes the images obtained through an RGB webcam, locates the driver's face and eyes, and compute the percentage opening of your eyes (EAR index), and if this value is lower than a dynamically generated threshold when the system starts up, the device generates an audible alert while the driver's sleepiness lasts. The use of visual information, which is considered the most reliable, allowed to build a non-invasive, non-intrusive and economical device, also working in real time despite its computational limitations. The tests carried out in several volunteers (even with the use of a cap and glasses), allowed to test its operation in various scenarios and its performance was quantitatively evaluated through the confusion matrix, obtaining an accuracy of 88.25% for controlled environments and 87% for real daytime driving environments.

Keywords: embedded, artificial vision, drowsiness, face detection, facial landmarks, Python, OpenCV, Haar cascade, Raspberry Pi, dlib, EAR

INTRODUCCIÓN

Antecedentes

El informe criminológico de la Fiscalía General del Estado, da a conocer que de acuerdo a la Organización Mundial de la Salud (OMS), en su informe de seguridad vial 2013, ubica al Ecuador como el segundo país de Sudamérica con el mayor índice de muertes causadas por accidentes de tránsito, lo cual coincide con la información estadística de transporte 2012 emitido por el INEC (Fiscalía General del Estado, 2014).

Según estadísticas oficiales disponibles en la página web de la Agencia Nacional de Tránsito, desde el año 2014 hasta finales del 2018, se han suscitado mas de 3000 accidentes de tránsito por somnolencia (que implica sueño, cansancio y fatiga) y ha cobrado la vida de mas de una centena de personas (ANT, 2018).

La somnolencia se define como la sensación de pesadez y torpeza de los sentidos motivada por el sueño. Esta sensación está determinada como el estado intermedio entre el estado de vigilia y el sueño profundo. El comienzo de esta etapa no es un evento único, sino que se origina por cambios en varias funciones neurológicas, se producen cambios sensoriales, en la memoria, en la conciencia, pérdida del pensamiento lógico, latencia en la respuesta a estímulos y alteraciones en los potenciales cerebrales (Garcés, Salgado, Cruz, & Cañon, 2015).

La Inteligencia Artificial es una rama de las ciencias computacionales que ofrece una infinidad de soluciones eficientes, de alto rendimiento. Dentro de ella, hay una disciplina científica denominada visión por computador, que permiten múltiples soluciones importantes de visión artificial para el desarrollo de sistemas no invasivos.

En lo que concierne a detección de somnolencia, en Ecuador hay tesis de grado e incluso tesis doctorales que plantean sistemas de detección de somnolencia aplicando metodologías similares utilizando visión artificial que ofrecen una interesante alternativa hacia la solución de esta problemática. Sin embargo, cabe mencionar que solo se han quedado plasmadas en una tesis (biblioteca) y no se ha puesto en marcha la implementación de las mismas.

(Guevara, 2013), en su proyecto de grado planteó un interesante sistema de detección de somnolencia usando tecnología ToF (Tecnología de Tiempo de Vuelo) y el modelo CANDIDE (algoritmo), sin embargo, como lo menciona el mismo autor, busca que el proyecto solamente sirva como base para futuras investigaciones que lleven a la implementación física del dispositivo.

(Rojas, 2018), en una publicación de prensa institucional de la Universidad Nacional de San Juan en Argentina, da a conocer que en dicha universidad, se está desarrollando una tesis doctoral que busca implemetar algoritmos para un sistema de detección de somnolencia

en conductores basado en el procesamiento de señales cerebrales, y han diseñado un prototipo del dispositivo para tal fin. Afirman que está pensado para choferes de camiones, máquinas viales y ómnibus, aunque también puede llegar a ser utilizado en trenes y aviones. Su idea es pensar en la reproducción del prototipo lo mas pronto posible luego de su defensa respectiva previsto para principios del 2019, para lo cual fabricarán las placas en su país y luego las enviarán a Estados Unidos para su ensamblaje, pero a pesar de ello, aún se deberá esperar un tiempo para ver su implementación.

Seeing Machines, es una importante empresa Australiana la cual trabaja con Inteligencia Artificial y se especializa en visión computarizada, utilizando tecnología vanguardista que detecta con rapidéz y precisión la somnolencia y distracción del conductor. (Seeing Machines, 2018). Pero el costo de sus dispositivos son altos y no están acorde a la economía del país.

España es uno de los primeros países que ha impulsado la industrialización de un dispositivo denominado "MagicEye-Aviso" desarrollado por el grupo Irizar, el cual es un sistema de seguridad que da alertas al detectar fatiga. En el 2013, en solo 6 meses de su lanzamiento vendieron mas de 250 unidades, de los cuales la mayoría fueron destinados la flota de autobuses Bizkaibus. A nivel internacional ha tenido una excepcional acogida por lo que esperan grandes e importantes pedidos en breve, pues la nueva flota de buses Eireann de Irlanda ya incorporan este sistema (Irizar, 2018).

Sin embargo, Ecuador aún no tiene miras a importar dichos dispositivos.

Situación Actual

Si bien es cierto, la inteligencia artificial está dominando los nuevos enfoques tecnológicos en el mundo, cabe mencionar que en Ecuador aún no se ha explorado y explotado su potencial. Pese a que existen ya estudios sobre sistemas de detección de somnolencia, no se ha continuado con su desarrollo y menos aún se ha puesto en producción, considerando el contexto del país y también que sea de bajo costo.

Actualmente, la detección de somnolencia es una línea activa de investigación y puede ser abordado desde diferentes enfoques y/o técnicas, incluso dentro de la misma disciplina de visión por computador. Por lo que es importante proponer nuevos algoritmos y/o soluciones adaptadas a nuestro contexto que resulten ser más precisos y robustos en ambientes reales de conducción.

En la provincia de Imbabura, la Universidad Técnica del Norte es la pionera en fomentar la investigación en la búsqueda de soluciones para el bien común, es así que ya se cuenta con una tesis sobre conteo de personas basado en visión por computador, no obstante, aún no se han implementado soluciones aplicando visión artificial que se enfoquen en la disminución de los accidentes de tránsito por somnolencia.

Movidelnor EP, ha hecho inversiones importantes para la adquisición de dispositivos electrónicos que permitan fomentar una conducción responsable, así como también maquinaria que controla el estado óptimo de los vehículos, etc., pero no disponen de soluciones tecnológicas que interactúen directamente con el conductor y lo ayuden a tener una experiencia de conducción más segura.

Prospectiva

Con la presente investigación se logró desarrollar un sistema para conducción diurna que, por medio de detección facial, permitió monitorear el comportamiento del conductor analizando principalmente la apertura de sus ojos. Dicho sistema se implementó en un dispositivo embebido, no invasivo, que permitió independizarlo de un computador. Cuando el sistema detecta que los ojos se están cerrando y el área de los ojos es más bajo del umbral establecido, entonces genera una alerta sonora para que el conductor se despierte y tome las precauciones necesarias.

Planteamiento del Problema

El 16 de agosto de 2017, en el palacio de Gobierno, se firmó el Pacto Nacional por la Seguridad Vial en presencia del Presidente de la República, Lenín Moreno. (Ministerio de Transporte y Obras Públicas, 2017). Pero dentro de su Plan Operativo, no se plantea la posibilidad de integrar la academia con la industria para la colaboración en la implementación de soluciones tecnológicas que contribuyan a minimizar los accidentes de tránsito. Es inminente la falta de políticas y apoyo gubernamental que permita que la academia se involucre en la búsqueda de soluciones para el beneficio común, esto repercute en que los estudiantes tengan poca responsabilidad social universitaria.

Los accidentes de tránsito por somnolencia son un problema que se ha producido a lo largo del tiempo, causando una problemática social que afecta a la economía principalmente de los involucrados en dicho suceso, también genera impactos psicológicos, traumas permanentes, invalidez y en el peor de los casos la muerte.

En la actualidad, algunas empresas de tecnología y marcas importantes de autos han optado por desarrollar sistemas avanzados de asistencia al conductor (SAAC) que detectan la somnolencia y generan una alerta de forma automática, sin embargo, dichos dispositivos están disponibles solo en autos de alta gama y su costo es significativamente alto.

Por ello es necesario ayudarse de la tecnología y disponer de un dispositivo que detecte los síntomas de somnolencia y alerte a tiempo al conductor para evitar cualquier tragedia y que además sea de **bajo costo** y accesible para la mayoría de usuarios en nuestro país.

En Ecuador, existen estudios realizados acerca de sistemas de detección de somnolencia, pero no han pasado de un pequeño prototipo o simplemente se han quedado documentados.

Es inminente la necesidad de un dispositivo robusto y de bajo costo, por lo que se requiere la aprobación y apoyo de las autoridades pertinentes para la puesta en marcha de este proyecto que puede convertirse en un emprendimiento de base tecnológica.

Árbol de problema causa – efecto

La Fig. 1, muestra el árbol de problema causa – efecto planteado para esta investigación.

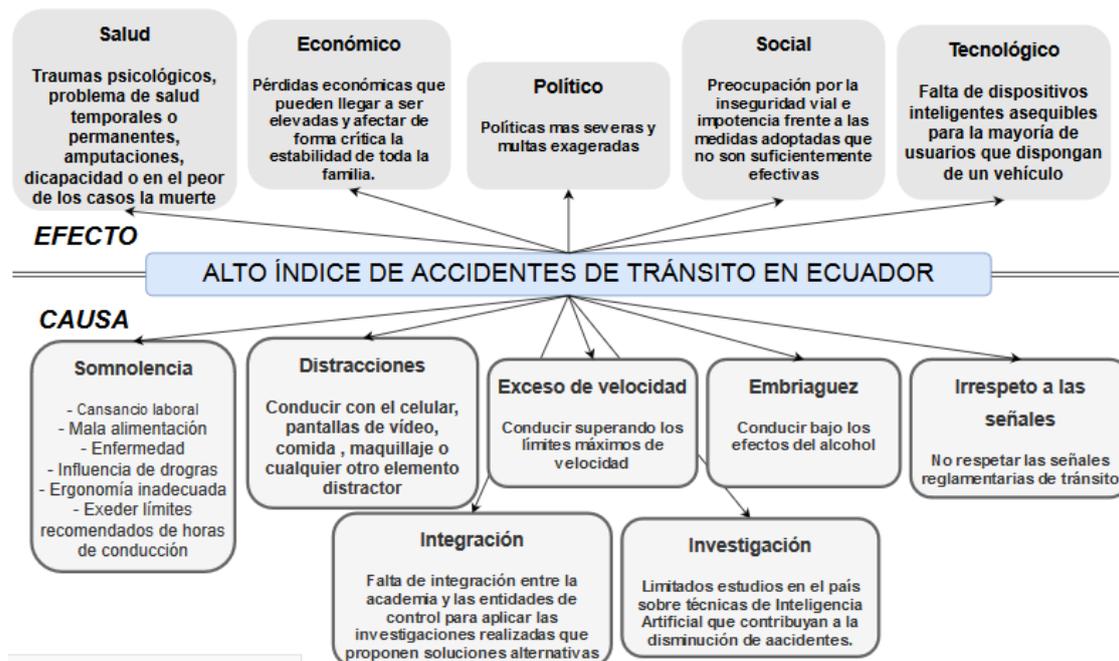


Fig. 1. Árbol de problema causa – efecto
Fuente: Propia

Objetivos

Objetivo General

Desarrollar un sistema embebido mediante el uso de técnicas de visión artificial para detección y alerta de somnolencia en conducción diurna en tiempo real.

Objetivos Específicos

- Elaborar un marco teórico que sustente las técnicas de visión artificial para reconocimiento facial y detección de somnolencia.
- Desarrollar un sistema embebido para la detección de somnolencia en conducción diurna.
- Validar y documentar los resultados obtenidos en las pruebas experimentales del prototipo del sistema, en un contexto real de conducción.

Justificación

Conceptos que hasta hace unos años eran de ciencia ficción, hoy en día son una realidad con el alcance que puede tener el estudio y aplicación de la inteligencia artificial.

Esta investigación surgió de la necesidad de disponer un dispositivo que genere alertas cuando detecte síntomas de somnolencia en un conductor y de esa manera tratar de evitar que se produzca un accidente de tránsito.

Una parte importante es que se busca impulsar el estudio de la inteligencia artificial y visión por computador dentro de la academia y proponer proyectos innovadores y relevantes que permitan dar soluciones a las distintas problemáticas existentes en el medio, pudiendo llegar a ser referentes en aplicar la Responsabilidad Social Universitaria (RSU), la cual, según (Fancois, 2014), debe basarse en la gestión de los cuatro impactos que genera siempre una institución de educación superior (IES), entre las que destacan los impactos que brotan de sus relaciones con el entorno social, sus redes, vecindarios, participaciones sociales, económicas y políticas.

La presente solución también se enfocó a contribuir con el “Decenio de Acción para la Seguridad Vial” (2011 - 2020), que es un plan de acción propuesto por la ONU, la cual exhorta a los países de todo el mundo a reducir las cifras mortales de accidentes de tránsito y buscar estrategias para la disminución de incidentes entre las cuales también se incluyen las distracciones. (Organización de las Naciones Unidas, 2010)

Se busca también generar interés y en lo posible establecer un vínculo importante entre la Academia y el resto de Instituciones Públicas, puesto que en la actualidad el Estado prefiere contratar servicios o adquirir tecnología del extranjero, sin tomar en cuenta que se podrían generar soluciones rentables y eficientes dentro de las aulas de nuestro mismo país.

Así mismo, se pretendió contribuir con la Agenda 2030 y los Objetivos de Desarrollo Sostenible, apoyando e implementando los objetivos 8, 9 y 11, los cuales en síntesis tratan de lograr niveles más elevados de productividad mediante la innovación tecnológica, fomentar la formalización de microempresas, aumentar la investigación científica, proporcionar acceso a sistemas de transporte seguros, asequibles y sostenibles para todos y mejorar la seguridad vial.

Alcance

La presente investigación tuvo como objetivo implementar un sistema que permita detectar e interpretar las expresiones oculares de una persona y generar una alarma de advertencia de forma automática en caso de detectar síntomas de somnolencia en conducción diurna. Para ello, la solución consistió en desarrollar un algoritmo utilizando las técnicas de reconocimiento facial propuestas por Viola & Jones (Viola & Jones, 2001), así como también el reconocimiento de patrones de parpadeo de Tereza Soukupová & Jan Čech (Soukupová

& Čech, 2016), mismos que fueron codificados haciendo uso principalmente de las librerías OpenCV y lenguaje de programación Python. El algoritmo fue implementado en un dispositivo embebido de hardware y software libre Raspberry pi, con lo cual se logró que sea portable y de fácil instalación en cualquier vehículo. Una cámara anclada al dispositivo embebido, capta las imágenes a color del conductor y son analizadas por el algoritmo para poder detectar su rostro y analizar la apertura de sus ojos, si el conductor empieza a cerrar los ojos y sobrepasa los umbrales de tolerancia definidos, se considera que tiene síntomas de somnolencia y de inmediato se genera una alarma sonora que ayuda al usuario a reaccionar ante tal eventualidad. El sistema propuesto se validó utilizando métricas cuantitativas de precisión en un contexto real de conducción controlado. La Fig. 2, muestra la estructura del prototipo.

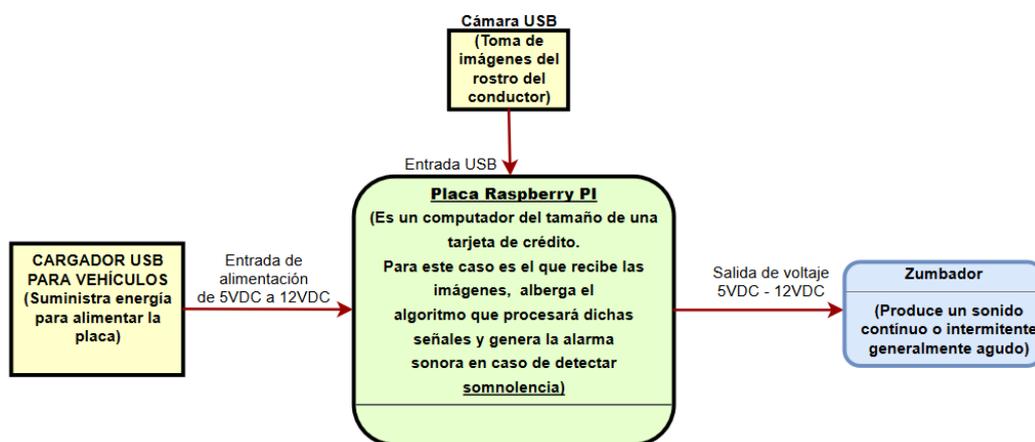


Fig. 2. Estructura del prototipo
Fuente: Propia

Contexto

En la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte, se está evidenciando la tendencia de realizar estudios basados en Inteligencia Artificial:

(Martínez, 2017), en su trabajo de titulación de la carrera de Mecatrónica, propuso un sistema de control para un brazo robótico utilizando visión artificial para los laboratorios de mecatrónica de la UTN.

(Chandi, 2017), en su trabajo de titulación de la carrera de Mecatrónica, propuso también un Sistema para conteo de personas basado en visión por computador.

(Espinosa, 2018), en su anteproyecto de tesis de la carrera de Sistemas Computacionales, propone un sistema de inteligencia artificial para detección de patrones de deserción estudiantil utilizando técnicas predictivas de clasificación y regresión de minería de datos.

Estos y otros trabajos demuestran que la Inteligencia Artificial es una disciplina interesante y muy importante, la cual se puede utilizar para plantear soluciones innovadoras a múltiples necesidades de nuestro medio.

De esta manera, el presente proyecto de investigación busca ser parte de este grupo que fomenta de manera significativa, la búsqueda de soluciones por medio del estudio de las múltiples áreas que ofrece la Inteligencia Artificial.

Así mismo, queda abierta la posibilidad de que otros estudiantes puedan mejorar y optimizar el funcionamiento de este dispositivo en el futuro.

CAPÍTULO 1

Marco Teórico

1.1. La Fatiga

La fatiga se puede describir como la falta de energía y motivación tanto física como mental. Las personas con fatiga pueden presentar tres síntomas principales; pero es variable en cada persona. Es así que, puede presentar falta de ánimo para realizar una actividad. Una vez que ha iniciado la actividad, la persona se cansa muy fácil. Además, la persona presenta fatiga mental o dificultad con la concentración y la memoria para comenzar o completar una actividad (Benjamin Wedro, 2018) .

1.1.1. Signos de fatiga en el conductor

Un estudio de la Universidad Cayetano Heredia (Perú), revela que el chofer bajo efecto de la fatiga disminuye progresivamente la atención durante el manejo y pierde capacidad de respuesta ante condiciones específicas que exigen reacciones inmediatas cuando circula por las carreteras. Pestañear y dormir durante el manejo son formas extremas de pérdida de la atención (Rey de Castro, 2003).

1.2. Somnolencia

El término somnoliento sencillamente es una tendencia a quedarse dormido. Para analizar la somnolencia del conductor, los investigadores han estudiado en su mayoría la transición de despierto a dormido, que es la fase de somnolencia. Los accidentes que se producen debido a la somnolencia del conductor tienen varias características (Sahayadhas, Sundaraj, & Murugappan, 2012):

- Ocurren a altas horas de la noche (de 0:00 a.m. a 7:00 a.m.) o durante la tarde (2:00 p.m. a 4:00 p.m.)
- Involucra a un solo vehículo que por lo general se sale de la carretera.
- Ocurren en carreteras de alta velocidad.
- A menudo, el conductor se encuentra solo.
- No se presentan marcas de frenado o deslizamiento en el área del siniestro.

1.2.1. Riesgos que presenta la somnolencia

La somnolencia perjudica funciones psicomotoras y neurocognoscitivas, como el tiempo de reacción, la concentración y la prudencia, así como el procesamiento de información, lo

cual puede conllevar a accidentes de tránsito para los conductores de vehículos o accidentes de trabajo para los operadores de maquinarias potencialmente peligrosas (De Castro, Gallo, & Loureiro, 2004).

Habitualmente, los accidentes producidos en estas circunstancias tienen altísima siniestralidad en términos de pasajeros muertos, heridos y pérdidas materiales y económicas (Rey de Castro Mujica, Rosales Mayor, & Egoavil Rojas, 2009).

1.3. Métodos de detección de fatiga

Los principales métodos de detección de fatiga de conducción pueden ser divididos en métodos de detección subjetivos y objetivos:

1.3.1. Métodos de detección subjetiva

Requieren que el conductor determine su grado de fatiga anotando su propia respuesta fisiológica en una hoja de registro de conducción. Los métodos de detección de fatiga subjetiva son simples y fáciles, pero su precisión y rendimiento en tiempo real es muy pobre porque requieren que el conductor pruebe su propio nivel de fatiga. Este enfoque se verá afectado por diferencias individuales del conductor (Shi, Tang, & Wang, 2017).

1.3.2. Métodos de detección objetiva

Utilizan varios sensores o equipo de adquisición de imágenes para recoger los parámetros fisiológicos, de comportamiento del conductor, del vehículo y otros tipos de medidas. Los parámetros recogidos se utilizan para evaluar el grado de fatiga del conductor. Los métodos de detección objetiva son más confiables y precisos porque usan un algoritmo para analizar automáticamente el grado de fatiga del conductor. Es así que, el conductor únicamente necesita centrarse en conducir el vehículo (Shi et al., 2017).

El método de detección objetiva se puede dividir en tres tipos (Fig. 3):

- a) Método de detección basada en parámetros fisiológicos del conductor.
- b) Método basado en las características faciales del conductor
- c) Método de detección basada en parámetros de conducción del vehículo.

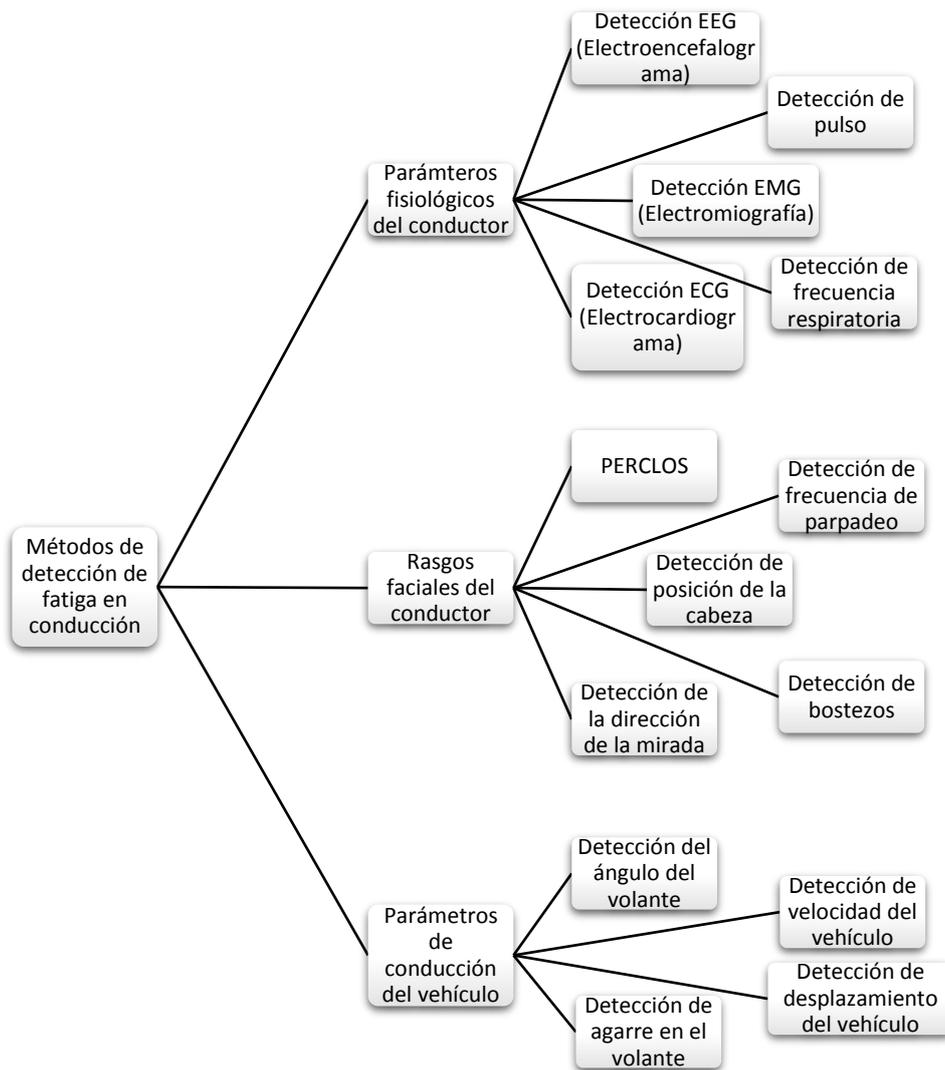


Fig. 3. Métodos de detección de fatiga

Fuente: (Shi et al., 2017)

Sin embargo (Shi et al., 2017), aduce que los sistemas actuales de detección de fatiga en conducción no han sido ampliamente utilizados porque necesitan equipo especial para la recopilación de datos, que implican un alto costo. Además, estos algoritmos actuales todavía tienen espacio para mejorar en la precisión. A pesar de que los métodos actuales tienen varios inconvenientes, tienen un gran potencial para reducir la incidencia de accidentes de tránsito.

Para este trabajo de titulación, se centró específicamente en estudiar el sistema más adecuado para detección de somnolencia utilizando técnicas de visión artificial, por ser un método no invasivo, eficaz y menos costoso. A continuación, se detallan algunas de las

técnicas de visión artificial que han sido aplicadas por investigadores para detectar la somnolencia en el conductor:

Método de detección basado en las características faciales del conductor.

Las características faciales del conductor al estar con síntomas de somnolencia son diferentes de cuando está despierto. Por lo tanto, es un método eficaz para detectar la fatiga en tiempo real al recopilar y analizar continuamente los datos faciales del conductor. Los cambios en las características de la cabeza y la cara son más notorios y se detectan con más facilidad. Los principales métodos de detección de somnolencia incluyen PERCLOS, detección de la posición de la cabeza, detección de la dirección de la mirada, detección de la frecuencia de parpadeo y detección del estado de la boca (bostezo) (Shi et al., 2017). A continuación, se detallan cada uno de ellos:

- ***Perclos***

El Centro de Investigación de Manejo de la Universidad Carnegie Mellon (Pensilvania) desarrolló el sistema PERCLOS (Grace et al., 2002) en 1998. Se basa en la teoría de que la capacidad de reflexión de la retina a la luz infrarroja de diferentes longitudes de onda es diferente en la misma iluminación. Básicamente, evalúa el grado de fatiga del conductor al comparar el porcentaje de tiempo de cierre de los párpados sobre la pupila por unidad de tiempo. Cuanto más tiempo estén cerrados los ojos, más grave será el grado de fatiga. Este método se caracteriza por ser preciso, trabaja en tiempo real y no es invasivo. Sin embargo, aún deben mejorarse ciertos aspectos, porque la luz excesiva, las gafas, etc., influyen en su funcionamiento (Shi et al., 2017).

- **Detección de posición de la cabeza**

La posición de la cabeza del conductor es diferente cuando está cansado de cuando está despierto. Por lo tanto, es factible determinar si el conductor está en un estado de fatiga detectando el ángulo de inclinación (pitch) de la cabeza del conductor. Este método detectaba la fatiga por el cambio de la aceleración de la cabeza, por medio de un sensor de aceleración de tres ejes ADXL330 ubicado en el asiento del conductor detrás de su cabeza. Las ventajas de este método son en tiempo real y de confianza. Las desventajas son alto costo, baja sensibilidad e inmunidad. Por lo ende, no es adecuado como parámetro de evaluación principal, sino como un parámetro auxiliar (Shi et al., 2017).

- **Detección de la dirección de la mirada**

La línea de visión del conductor despierto estará delante del vehículo y se moverá rápido, mientras que la línea de visión del conductor fatigado se desviará de la dirección habitual y será más lento. En el estudio de (Choi & Kim, 2014), usaron el método de análisis CDF (Cumulative distribution function) para ubicar la posición de la pupila del conductor, y seguidamente detectar la dirección de la mirada de la línea de visión del conductor y la frecuencia cambiante de la línea de visión. Estos dos indicadores se utilizaron para estimar el nivel de fatiga del conductor. El equipo de detección de la dirección de la mirada es simple y barato. Sin embargo, la fiabilidad y sensibilidad de este método es bajo. Puede tener mejores resultados al ser utilizado en combinación con otros métodos de detección (Shi et al., 2017)

- **Detección de la frecuencia de parpadeo**

La frecuencia de parpadeo hace referencia a la cantidad de parpadeos en una unidad de tiempo. Cuando el conductor está cansado, la frecuencia de parpadeo será más baja que el valor normal. En el artículo de (Sigari, Pourshahabi, Soryani, & Fathy, 2014), mencionan la utilización de tecnología de visión artificial para extraer la información de los ojos del conductor y calcular la frecuencia de parpadeo, con la cual se analizó la fatiga del conductor. El método tiene las características de precisión alta y en tiempo real. Con el desarrollo de la tecnología de visión artificial, se pueden mejorar los problemas de oclusión, iluminación y otros factores. La frecuencia de parpadeo es un importante índice de evaluación de la fatiga (Shi et al., 2017).

- **Detección de bostezos**

El bostezo es una característica facial significativa de la conducción con fatiga. Por medio del reconocimiento facial, analiza las características de la boca para determinar si está abierta o cerrada, seguidamente se evalúa el grado de fatiga del conductor según la cantidad de bostezos por unidad de tiempo y el grado de apertura de la boca cuando bosteza. Un problema que se presenta en este método es que puede equivocarse cuando la persona solo está hablando mas no bostezando. Para resolver este problema, los investigadores (Li, 2001) utilizaron el modelo oculto de Markov para analizar las características de la boca del conductor y separaron con éxito las dos características de hablar y bostezar. Este método de detección de bostezos tiene la ventaja de trabajar en tiempo real y tener alta precisión, que es adecuado para ser utilizado como un método auxiliar para la detección de fatiga (Shi et al., 2017)

1.4. El ojo Humano

El ojo humano es un órgano foto receptor, y su función consta en recibir los rayos luminosos provenientes de los objetos del mundo exterior y transformarlos en impulsos eléctricos que son conducidos al centro nervioso de la visión en el cerebro (Centro, 2018).

En la conducción, el 90% de la información de la información se la recibe a través de los ojos, por ende, el sentido de la vista es el más importante de todos. Es así que, si se produce una disminución en el rendimiento visual, puede alterar la correcta captación de los estímulos y afectar la toma de decisiones y respuesta psicomotora (López, 2012).

1.4.1. Parpadeo o pestañeo

El parpadeo se trata de una pequeña depresión del párpado que tiene como finalidad humedecer la parte externa del ojo, evitando que se evapore la película lagrimal, manteniendo íntegra la superficie ocular y la calidad óptica de la córnea. Además, permite eliminar cualquier partícula presente en la superficie ocular (López, 2012).

1.4.2. Frecuencia de parpadeo

La frecuencia de parpadeo (FP) es el tiempo de parpadeos que se producen durante un tiempo determinado. Este puede producirse de dos formas: voluntaria, de la cual somos conscientes, e involuntaria, que se produce por la influencia de varios factores como el clima, luminosidad, enfermedades oculares etc. Las tareas sencillas como escuchar o hablar aumentan la frecuencia de parpadeo. Por el contrario, las tareas que requieren procesamiento de información visual como leer o estar frente a un monitor, reducen la frecuencia de parpadeo espontáneo. En base a estas variantes, la frecuencia varía en cada individuo, pero en posición de mirada y reposo el promedio es de 15 parpadeos por minuto (López, 2012).

1.4.3. Tiempo de duración del parpadeo

Es importante mencionar que en cada parpadeo se pierden alrededor de 400ms de información visual. En la conducción, se ha estudiado el parpadeo como un mecanismo de advertencia de la pérdida de concentración del conductor. Cuando el conductor está fatigado, la velocidad de parpadeo y la apertura palpebral disminuyen, al contrario de cuando el conductor no está fatigado (López, 2012).

1.5. Visión Artificial

Con el surgimiento de las computadoras, los científicos se han venido planteando la interrogante de cómo enseñar a las computadoras realizar tareas de la inteligencia humana, tales como resolver problemas, comprender lenguajes o analizar información visual, originando con ello a una disciplina orientada a emular la inteligencia humana, denominada “inteligencia artificial” y un sin número de aplicaciones y campos de investigación científica. En los primeros años de su evolución se consiguieron importantes avances, pero con las distintas pruebas, se ha evidenciado que las tareas que para los humanos son fáciles, se tornan difíciles para las computadoras. La visión artificial es semejante al proceso de la visión humana, siendo su insumo básico las imágenes captadas por medio de una cámara (S & D, 2011).

Los avances que ha tenido la visión artificial, le ha permitido contribuir en los siguientes campos:

- Medicina: Detección de patologías, conteo de elementos en imágenes microscópicas, reconstrucción 3D, etc.
- Vigilancia: Monitorear todas las actividades o identificar personas, con el fin de incrementar la seguridad.
- Industria: Es el campo que más utiliza la visión artificial. Lo más común, es el control de calidad.
- Robótica: Creación de máquinas para ejecución de tareas peligrosas para el ser humano. Sortear obstáculos, tener flexibilidad y habilidades motrices humanas.
- Interacción Humano – Computador: Reconocimiento de señas, gestos. Uso de la mirada como un mouse visual.
- Controles biométricos: Control de acceso por reconocimiento facial o patrón del iris.
- Automotriz: Fabricación de vehículos inteligentes de conducción autónoma, asistente de reconocimiento de objetos, etc.
- Bancaria: Validación de ingreso a banca electrónica por medio de reconocimiento facial.

1.5.1. Procesamiento digital de imágenes

Antes de extraer información de las imágenes, hay que considerar los defectos que provienen de ella (S & D, 2011). Por ejemplo:

- Defectos en los dispositivos de entrada de imagen, lo cual puede provocar imágenes borrosas/desenfocadas, oscuras, con exceso de brillo, etc.
- Pérdida de información al momento de la captura.
- Alteración de la información en los medios de distribución. Ante esto, es necesario el acondicionamiento de la(s) imagen(es). Posterior al acondicionamiento se definen los criterios a resaltar, implicando técnicas de mejoramiento como redistribución de la información estadística de la imagen y la manipulación de las características del espacio de color (Intensidad, Saturación y Pureza) (S & D, 2011).

El espacio del color RGB es el modelo más conocido y utilizado, en donde cada color está constituido por tres valores de rojo (R), verde (G) y azul (B), situados a lo largo de los ejes del sistema de coordenadas cartesianas sobre un cubo (

Fig. 4). Estos valores de RGB van en un rango de 0 a 1 o de 0 a 255. Es así que, el negro por ejemplo se representa como (0, 0, 0) y el blanco como (1, 1, 1) o (255, 255, 255). El color negro y el blanco están representados por dos esquinas opuestas del cubo que se pueden definir por los ejes de R, G, B, del sistema de coordenadas cartesianas (

Fig. 4). Los otros vértices del cubo representan el rojo, el verde, el azul, el cian, el magenta y el amarillo. Los colores en escala de grises se representan con componentes idénticas de R, G, B (Cortes Osorio, Urueña, & Mendoza Vargas, 2011).

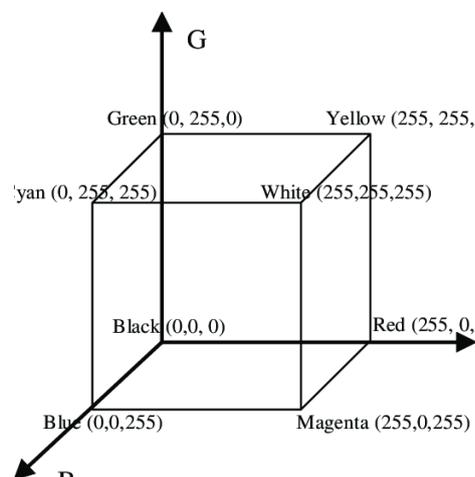


Fig. 4. Cubo del espacio de color RGB
Fuente: (Ramella & Di Baja, 2013)

El tratamiento de las imágenes es muy importante en la detección de rasgos faciales. Si se realiza una correcta conversión de la imagen a escala de grises, se facilita la extracción de las características del objeto analizado, ya que se convierte en un arreglo matricial de dos dimensiones (Ardila, Osorio, & Vargas, 2011).

Para la conversión, hay algunas técnicas que se resumen a continuación:

- El promedio (average). – Suma las componentes RGB de cada capa pixel a pixel y divide por la cantidad, que en este caso sería 3. Su fórmula es:

$$Grey(i, j) = \frac{(R(i, j) + G(i, j) + B(i, j))}{3}$$

- La Tonalidad (Midgray). – Equivalente a la saturación, luminancia o intensidad. Afecta muy significativamente el contraste de la imagen. Busca la parte media de las componentes de color máxima y mínima. Su fórmula es:

$$Grey(i, j) = \frac{(Max[R(i, j), G(i, j), B(i, j)] + Min([R(i, j), G(i, j), B(i, j)]))}{2}$$

- Luminosidad (Luminosity). - Es más avanzada que el método del promedio. Cada color presenta un valor según la percepción humana, valores que fueron establecidos por la *Rec. 601 NTSC International Telecommunication Union*, y es estándar mundial para la TV a color. Su fórmula es:

$$Grey(i, j) = 0.299 * R(i, j) + 0.587 * G(i, j) + 0.114 * B(i, j)$$

No es recomendable acudir a un solo método de conversión a escala de grises, pues los resultados van a ser diferentes. Lo más factible es experimentar con varios métodos hasta que el resultado se ajuste al problema específico.

El pre-procesamiento de las imágenes, mejora la calidad, facilita la interpretación y mejora el tratamiento de los datos (Peraza, Rondón, & Arguello Fuentes, 2011).

Las técnicas de mejoramiento de imágenes están resumidas en la Fig. 5.

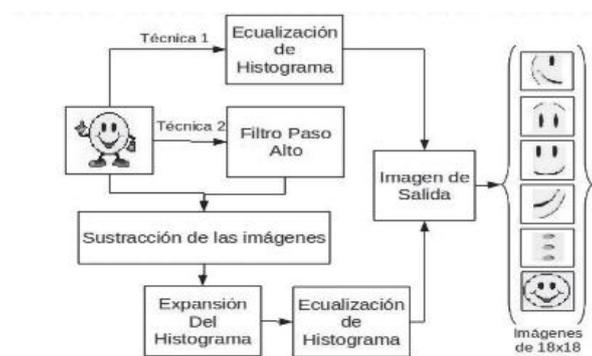


Fig. 5. Modelo de las técnicas de pre - procesamiento digital de imágenes

Fuente: (Peraza et al., 2011)

1.6. Paquetes de procesamiento de imágenes

Hay muchos paquetes de procesamiento de imágenes tanto pagados como libres. Algunos paquetes comerciales más populares son: *The Matrox Image Library (MIL)*, *Khoros*, *eVision*, *HIPS*, *Exbem*, *Aphelion*, etc., que a pesar de que son potentes, son muy poco accesibles por su alto costo, actualizaciones muy prolongadas y la mayoría son pesadas y lentas. Por otra parte, los paquetes de software libre que más destacan son: OpenCV, Gandalf, TargetJr, CVIPTools, ImageLib, LookingGlass, NeatVision(Java), Tina, etc., pero de todos esos paquetes los únicos que ofrecen un marco de trabajo completo para aplicaciones de visión artificial son OpenCV y Gandalf. Sin embargo, solo OpenCV dispone de bibliotecas de tipos de datos estáticos y dinámicos, tales como matrices, árboles, grafos, etc. Además, solo OpenCV puede trabajar con la mayoría de cámara del mercado, tiene entornos fáciles y corre sobre los dos Sistemas Operativos más utilizados en el mundo, Windows y Linux (Arévalo, González, & Ambrosio, 2002)

1.6.1. Librería OpenCV

La librería OpenCV nace tras el anuncio de la Corporación Intel en el año 2000, bajo licencia BSD (Software Libre), con el fin de ayudar a los docentes e investigadores a desarrollar nuevas formas de interactuar con los ordenadores. La librería OpenCV es una API que tiene alrededor de 300 funciones escritas en lenguaje C. Su uso es libre tanto para uso comercial como no comercial. OpenCV está orientado principalmente a la visión por computador en tiempo real. En sus áreas de aplicación destacan: interacción hombre-máquina; segmentación y reconocimiento de objetos; reconocimiento de gestos; seguimiento del movimiento; estructura del movimiento y robots móviles. Es así que OpenCV constituye una herramienta muy interesante e importante en los últimos tiempos, debido a que es software libre, es escalable, estable, y su crecimiento va en aumento, lo cual asegura su continuidad y utilidad tanto en la docencia como en la investigación a corte, mediano y largo plazo (Arévalo et al., 2002).

1.7. Dispositivo embebido

El término “sistema embebido” hace referencia a todo circuito electrónico digital que es capaz de realizar operaciones de computación, generalmente en tiempo real y que sirven para cumplir una tarea específica. Al contrario de las laptops o computadores de escritorio, sus recursos son limitados y las aplicaciones específicas que se les programa, los hacen muy útiles en áreas como el campo automotriz, teléfonos móviles, Blu-ray, refrigeradoras, alarmas de casas, lavadoras, cámaras fotográficas, equipos médicos, instrumentos musicales, etc. La

potencia de procesamiento depende de la arquitectura del microprocesador del sistema embebido, los periféricos y la máxima frecuencia de operación. El programa que ejecuta un sistema embebido es por lo general elaborado en lenguajes como el ensamblador, ANSI C, C++, Basic, Python. Algunos sistemas embebidos pueden ejecutar sistemas operativos limitados RTOS (Real Time Operating Systems), que permiten que el procesador ejecute varias tareas, asignándoles una prioridad y un orden de ejecución. Una peculiaridad importante de un sistema embebido, es el bajo consumo de energía, lo que permite el uso de baterías en muchas de sus aplicaciones (Yu et al., 2017).

A continuación, la tabla TABLA 1.1 muestra una comparativa de los dispositivos embebidos más utilizados:

TABLA 1.1.
COMPARATIVA DE DISPOSITIVOS EMBEBIDOS MÁS POPULARES

	RASPBERRY PI 3B+	ASUS TINKER BOARD S	ODROID – XU4
Procesador	Arm 64 bits Quad – Core 1.4GHz	Arm Quad-Core 1.8GHz	A15 Quead Core 2.0 GHz
Almacenamiento RAM	Tarjeta micro SD 1 Gb interna	16 Gb internas 2 Gb internas	Tarjeta microSD 2Gb internas
Conectividad	Wifi, bluetooth, hdmi, usb 2.0, micro usb, rj45	Wifi, bluetooth, hdmi, usb 2.0, micro usb	Wifi, hdmim usb 3.0, micro usb, rj45
Sistema Operativo	Linux, Windows 10 IOT	Tinker OS	Linux, Android
Medidas	8.5 x 5.9 x 1.9 pulgadas	6 x 4.3 x 1.5 pulgadas	3.2 x 2.3 x 0.9 pulgadas
Soporte comunitario	Alto	Regular	Medio
Precio en amazon sin costos de envío	\$37.56	\$91.99	\$89.95

Fuente: Elaboración propia

1.8. Detección de rostros

En la actualidad existen propuestas de métodos que ofrecen mayor precisión en la detección de objetos, pero demandan recursos computacionales aptos para poder procesar la información. Un método que ha generado buenos resultados para detectar el rostro es el algoritmo propuesto por Viola & Jones (VJ) (Viola & Jones, 2004), que combina adecuadamente la imagen integral, el algoritmo de aprendizaje AdaBoost (Schapire, 1999) y la cascada de clasificadores (OpenCV, 2013), que es un conjunto de clasificadores débiles que permiten ir descartando zonas de la imagen que no son de interés, cuyo resultado es un detector de objetos que trabaja en tiempo casi real. Así, este algoritmo genera un rectángulo denominado región de interés (ROI) que localiza la posición del rostro a través de las coordenadas de la esquina superior izquierda (x_0, y_0) y sus dimensiones de ancho y alto ($w,$

h). A pesar de las ventajas que presenta VJ, este no es absolutamente fiable debido a que no es capaz de localizar el rostro cuando no se encuentra dentro de los rangos permitidos en la etapa de entrenamiento. Esto incluye giros laterales, giros hacia arriba o hacia abajo; todos ellos habituales durante la conducción y que se producen en situaciones como mirar el retrovisor y/o los espejos laterales (Flores, Armingol M, & de la Escalera, 2011), más sin embargo, es una excelente alternativa para usar en dispositivos embebidos con limitadas prestaciones de hardware ya que es ligero, no sacrifica en gran manera la precisión y permite ganar velocidad de procesamiento.

1.8.1. Detección de rostros usando Haar - Cascade

La documentación de OpenCV, da a conocer que la detección de objetos usando clasificadores en cascada basados en características de Haar, es un método efectivo de detección de objetos propuesto por Viola & Jones, afirma que es un enfoque basado en el aprendizaje automático donde la función de cascada se entrena a partir de muchas imágenes positivas y negativas. Para trabajar con detección de rostros, inicialmente, el algoritmo necesita muchas imágenes positivas (imágenes de caras) e imágenes negativas (imágenes sin caras) para entrenar al clasificador. Entonces se necesita extraer características de él. Para esto, se utilizan las características de Haar (Fig. 6), son como el núcleo convolucional. Cada característica es un valor único obtenido al restar la suma de píxeles debajo del rectángulo blanco de la suma de píxeles debajo del rectángulo negro (OpenCV, 2013).

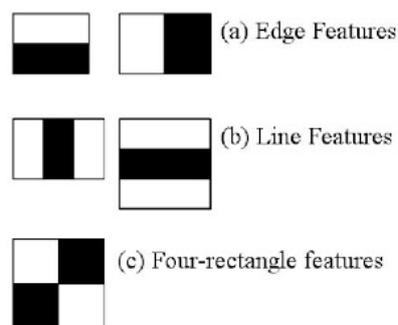


Fig. 6. Características de Haar
Fuente: (OpenCV, 2013)

El algoritmo de clasificadores de Haar detecta rápidamente cualquier objeto. La identificación del mismo objeto, pero en diferentes tamaños, requiere el mismo esfuerzo y tiempo que los objetos de tamaños comparables, ya que el escalado no demanda un esfuerzo adicional. La cascada de los clasificadores permite analizar solo las sub imágenes que tienen una gran probabilidad de poder seguir analizando todas las características de Haar que permiten reconocer un elemento. Este algoritmo de clasificadores devuelve la región de interés (ROI) del objeto encontrado en el cuadro actual (Stan, Miclea, & Centea, 2014).

Básicamente, el algoritmo en vez de aplicar todas las características en una ventana, las agrupa por etapas y las va aplicando una por una. Si una ventana falla en la primera etapa, la excluye de inmediato y no aplica las características restantes sobre dicha ventana. Caso contrario, si la ventana pasa la primera etapa, debe aplicar la segunda etapa y continuar el proceso. Si la ventana en mención pasa todas las etapas, se considera que se encontró un rostro en la región analizada (Fig. 7). He ahí los motivos para que sea tan veloz.

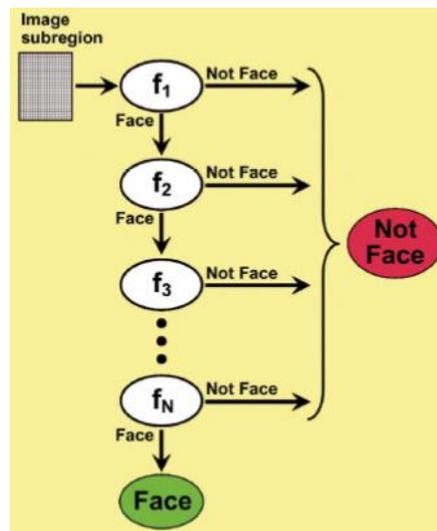


Fig. 7. Identificación de la cara por medio de Haar
Fuente: (OpenCV, 2013)

1.9. Librería DLIB

DLIB es una biblioteca de software de código abierto multiplataforma escrita en C++. Su objetivo es proporcionar un entorno rico para desarrollar software de aprendizaje automático, la visión por computadora y análisis de datos en lenguaje C++, pues intenta llenar algunos de los vacíos en el soporte de herramientas que otras bibliotecas no lo tienen. La biblioteca está publicada bajo la Licencia de Software Boost, lo que permite que se pueda incorporar en software de código abierto y comercial, además dispone de una documentación amplia y precisa. Para su uso no necesita bibliotecas adicionales ni requiere instalarse o configurarse. Se ha usado con frecuencia en entornos Windows, Linux y MacOS X, pero debería funcionar con cualquier compilador que sea compatible con ISO C++ (D. E. King, 2009).

1.9.1. Detector de hitos faciales de dlib

El detector de marcas faciales pre-entrenadas dentro de la biblioteca *dlib*, es usado para estimar la ubicación de 68 (x, y) coordenadas que se asignan a las estructuras faciales en la cara. Los índices de las 68 coordenadas se pueden visualizar en la Fig. 8. Hay que tener en

cuenta que existen más modelos de detectores de puntos de referencia faciales, incluido el modelo de 194 puntos que se puede entrenar en el conjunto de datos de HELEN que tiende a ser más preciso, pero demanda más capacidad y tiempo de procesamiento. (Adrian Rosebrock, 2017).

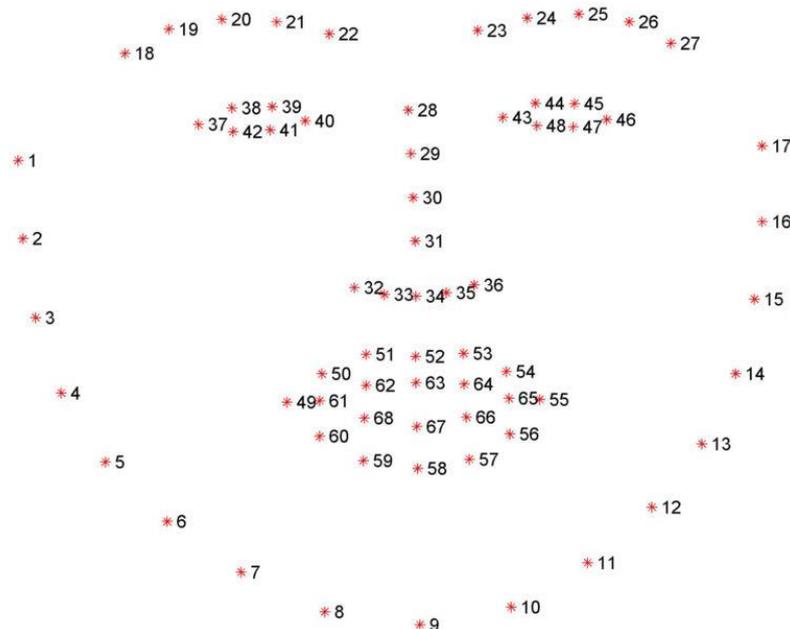


Fig. 8. Visualización de las 68 coordenadas de las estructuras faciales en la cara
Fuente: (Adrian Rosebrock, 2017)

1.10. Metodología de desarrollo

Las metodologías de desarrollo se encuentran mejorando permanentemente los procesos y métodos con el fin de mejorar la productividad en la creación del software y la calidad del producto. Las metodologías se agrupan en tradicionales y ágiles. Entre las principales características de las metodologías tradicionales se tiene que, suelen tener resistencia a los cambios, está hecha interactuar con grupos grandes, existen más roles y la arquitectura de software es esencial. Por el contrario, las metodologías ágiles están preparadas para cambios durante el desarrollo del proyecto, el contrato es más flexible, ideal para trabajar con grupos pequeños y no hay mucho énfasis en la arquitectura del software (Manzano Urquizo & Tene Escobar, 2016).

A continuación, en la TABLA 1.2 se muestra un análisis comparativo entre las principales metodologías ágiles de desarrollo, siendo 3 la puntuación más alta:

TABLA 1.2.
EVALUACIÓN METODOLOGÍAS ÁGILES

CRITERIO DE EVALUACIÓN	Ponderación	SCRUM	XP	ICONIX
Flexibilidad ante los cambios	20%	3	3	2
Gestión de tiempos y recursos	20%	3	3	3
Curva de aprendizaje	15%	2	2	2
Colaboración	15%	2	3	2
Documentación	10%	1	1	3
Flexibilidad de los roles	10%	2	3	2
Experiencia desarrolladores	15%	2	2	2

Fuente: Propia

1.10.1. Fases de la metodología XP

Según (Angamarca, 2018), las fases de la metodología XP son las siguientes:

Fase 1. Planificación de proyecto. - Se recopila los requerimientos del proyecto por medio de las iteraciones con el cliente y se hace un análisis profundo del problema que se requiere solucionar. Dentro de esta fase se tiene (Ver Fig. 9):



Fig. 9. Fase 1 Metodología XP

Fuente: (Castillo, Figueroa, & Sevilla Hector, 2019)

Fase 2. Diseño. – Muestra la estructura lógica que tendrá el sistema. Según (Castillo et al., 2019), la metodología XP sugiere diseños simples y sencillos para facilitar su comprensión y desarrollo. Dentro de esta fase se tiene (Fig. 10):



Fig. 10. Fase 2 Metodología XP

Fuente: (Castillo et al., 2019)

Fase 3. Desarrollo. – Según (Castillo et al., 2019), esta fase se trata de codificar las historias de usuario, en la cual el cliente es una parte importante ya que será quien vigile que se cumplan los requerimientos de las historias de usuarios y funcionen correctamente.

Fase 4. Pruebas. – Según (Castillo et al., 2019), en esta fase permite evaluar que las tareas de las historias de usuario cumplan con lo solicitado. Estas pruebas se hacen sobre las funcionalidades generales que debe cumplir el programa.

CAPÍTULO 2

Desarrollo

2.1 Fase de planificación

En la quipo para este trabajo de investigación. se muestran los miembros que conformaron el equipo para este trabajo de investigación.

TABLA 2.1
EQUIPO DE TRABAJO

Miembro	Grupo	Rol
Josué Alba	Tesista	Programador
PhD. Iván García	Director	Entrenador, testeador
MSc. Erick Herrera	Consultor	Testeador
MSc. Marco Pusedá	Consultor	Testeador

Fuente: Propia

2.1.1. Historias de usuario

La valoración para los criterios de “Prioridad”, “Riesgo” y “Esfuerzo” mostrados en la TABLA 2.2, están definidos con tres valores: alto, medio y bajo.

TABLA 2.2
HISTORIAS DE USUARIO

Proceso	Nro	Descripción	Prioridad	Riesgo	Esfuerzo	Iteraciones
Entorno de trabajo Raspberry Pi 3B+	H1	Preparación de la Raspberry Pi 3B+ y montaje de accesorios necesarios	Medio	Medio	Bajo	1
	H2	Instalación y configuración del SO Raspbian	Alto	Medio	Bajo	1
	H3	Instalación de OpenCV y Python	Alto	Alto	Medio	1
	H4	Instalación de librerías necesarias	Alto	Alto	Alto	1
Codificación del sistema	H5	Importación de librerías, definición de variables y funciones	Alto	Alto	Medio	2
	H6	Montaje y configuración de la Pi Cámara	Alto	Alto	Medio	2
	H7	Desarrollo del algoritmo para detección y alerta de somnolencia	Alto	Alto	Alto	2
Montaje del sistema	H8	Instalación del sistema embebido dentro de un vehículo	Alto	Alto	Medio	3
	H9	Pruebas del sistema en un contexto real de conducción	Alto	Medio	Medio	3

Fuente: Propia

Planificación de publicación

Primera Iteración. - En la TABLA 2.3 se muestran las historias de usuario que corresponden a la primera iteración.

TABLA 2.3
ANÁLISIS DE CRITERIOS PARA LA PRIMERA ITERACIÓN

Nro.	Descripción	Días
H1	Preparación de la Raspberry Pi 3B+ y montaje de accesorios necesarios	2
H2	Instalación y configuración del SO Raspbian	2
H3	Instalación de OpenCV y Python	3
H4	Instalación de librerías necesarias	4

Fuente: Propia

Segunda Iteración. - En la TABLA 2.4 se muestran las historias de usuario que corresponden a la segunda iteración.

TABLA 2.4
ANÁLISIS DE CRITERIOS PARA LA SEGUNDA ITERACIÓN

Nro.	Descripción	Días
H5	Importación de librerías, definición de variables y funciones	3
H6	Montaje y configuración de la Pi Cámara	2
H7	Desarrollo del algoritmo para detección y alerta de somnolencia	20

Fuente: Propia

Tercera Iteración. – En la TABLA 2.5 se muestran las historias de usuario que corresponden a la tercera iteración.

TABLA 2.5
ANÁLISIS DE CRITERIOS PARA LA TERCERA ITERACIÓN

Nro.	Descripción	Días
H8	Instalación del sistema embebido dentro de un vehículo	3
H9	Pruebas del sistema en un contexto real de conducción	15

Fuente: Propia

De la TABLA 2.6 a la TABLA 2.14, se presenta las historias de usuario ordenadas:

TABLA 2.6
HISTORIA DE USUARIO 1

Historia de usuario	
Número: 1	Usuario: Desarrollador
Nombre historia: Preparación de la Raspberry Pi 3B+ y montaje de accesorios necesarios	
Prioridad: Medio	Riesgo: Medio
Esfuerzo: Bajo	Iteración: 1
Programador responsable: Josué Alba	
Descripción: Revisión del manual de usuario y montaje de los accesorios necesarios para alimentación, almacenamiento y disipación del dispositivo embebido	
Observaciones:	

Fuente: Propia

TABLA 2.7
HISTORIA DE USUARIO 2

Historia de usuario	
Número: 2	Usuario: Desarrollador
Nombre historia: Instalación y configuración del SO Raspbian	
Prioridad: Alto	Riesgo: Medio
Esfuerzo: Bajo	Iteración: 1
Programador responsable: Josué Alba	
Descripción: Instalación y configuración del SO Raspbian recomendado para el dispositivo embebido	
Observaciones:	

Fuente: Propia

TABLA 2.8
HISTORIA DE USUARIO 3

Historia de usuario	
Número: 3	Usuario: Desarrollador
Nombre historia: Instalación de OpenCV y Python	
Prioridad: Alto	Riesgo: Alto
Esfuerzo: Medio	Iteración: 1
Programador responsable: Josué Alba	
Descripción: Instalación del lenguaje Python y librería OpenCV	
Observaciones:	

Fuente: Elaboración propia

TABLA 2.9
HISTORIA DE USUARIO 4

Historia de usuario	
Número: 4	Usuario: Desarrollador
Nombre historia: Instalación de librerías necesarias	
Prioridad: Alto	Riesgo: Alto
Esfuerzo: Alto	Iteración: 1
Programador responsable: Josué Alba	
Descripción: Instalación de librerías principales	
Observaciones:	

Fuente: Propia

TABLA 2.10
HISTORIA DE USUARIO 5

Historia de usuario	
Número: 5	Usuario: Desarrollador
Nombre historia: Importación de librerías, definición de variables y funciones	
Prioridad: Alto	Riesgo: Alto
Esfuerzo: Medio	Iteración: 2
Programador responsable: Josué Alba	
Descripción: Se definirán las variables locales y funciones necesarias a utilizar en la detección de somnolencia	
Observaciones:	

Fuente: Propia

TABLA 2.11
HISTORIA DE USUARIO 6

Historia de usuario	
Número: 6	Usuario: Desarrollador
Nombre historia: Montaje y configuración de la Pi Camera	
Prioridad: Alto	Riesgo: Alto
Esfuerzo: Medio	Iteración: 2
Programador responsable: Josué Alba	
Descripción: Montaje de la Pi Cámara al dispositivo embebido y su respectiva configuración para que pueda ser utilizada	
Observaciones:	

Fuente: Propia

TABLA 2.12
HISTORIA DE USUARIO 7

Historia de usuario	
Número: 7	Usuario: Desarrollador
Nombre historia: Bucle para el proceso de detección de somnolencia y generación de alarma sonora	
Prioridad: Alto	Riesgo: Alto
Esfuerzo: Alto	Iteración: 2
Programador responsable: Josué Alba	
Descripción: Desarrollo del bucle del algoritmo para el proceso de captura, procesamiento y análisis de imágenes para determinar el estado de fatiga del conductor por medio de la apertura de los ojos y generar una alarma sonora en caso de detectar somnolencia.	
Observaciones:	

Fuente: Propia

TABLA 2.13
HISTORIA DE USUARIO 8

Historia de usuario	
Número: 8	Usuario: Desarrollador
Nombre historia: Instalación del sistema embebido dentro de un vehículo	
Prioridad: Alto	Riesgo: Alto
Esfuerzo: Medio	Iteración: 3
Programador responsable: Josué Alba	
Descripción: Montaje del dispositivo en un lugar adecuado dentro de la cabina del auto para que pueda monitorear el rostro del conductor y determine la somnolencia	
Observaciones:	

Fuente: Propia

TABLA 2.14
HISTORIA DE USUARIO 9

Historia de usuario	
Número: 9	Usuario: Desarrollador
Nombre historia: Pruebas del sistema en un contexto real de conducción	
Prioridad: Alto	Riesgo: Alto
Esfuerzo: Medio	Iteración: 3
Programador responsable: Josué Alba	
Descripción: Pruebas del sistema simulando somnolencia al conducir el vehículo	
Observaciones:	

Fuente: Propia

2.2. Fase de diseño

2.2.1. Arquitectura

En la Fig. 11 se muestra la arquitectura del prototipo del sistema para detección de somnolencia, el cual básicamente consta de una cámara OV5647 que está situada frente al conductor, la misma que captura imágenes consecutivas del rostro del conductor y son enviadas a la placa Raspberry Pi 3B+ para que procese y optimice, y luego dar paso al algoritmo que detecta el rostro y sus rasgos faciales, extrae la región de los ojos y analiza la apertura de los ojos junto con el tiempo de duración del parpadeo. Si los ojos permanecen cerrados por encima del tiempo establecido, el sistema genera una señal de voltaje que permite que el *Buzzer* emita un pitido de alerta hasta que el conductor abra los ojos.

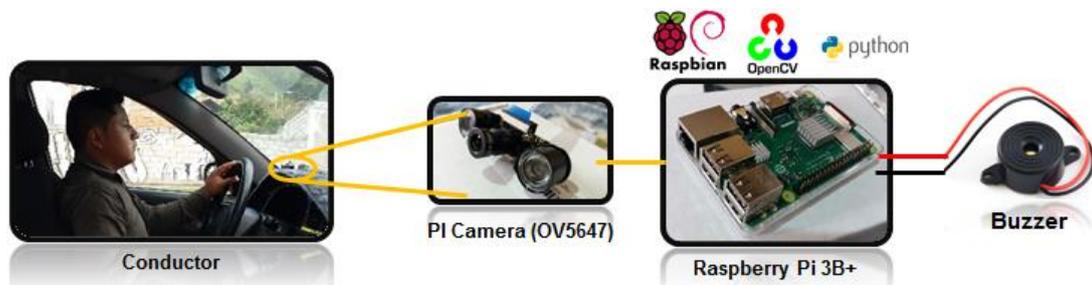


Fig. 11. Arquitectura del prototipo para detección de somnolencia en conducción diurna

Fuente: Propia

2.2.2. Componentes del sistema

A continuación, en la TABLA 2.15, se lista cada uno de los componentes de hardware que se utilizaron para la implementación del sistema de detección de somnolencia. Cabe mencionar que en base a la tabla comparativa de las SBC (Single Board Computer), se ha seleccionado como mejor candidato a la Raspberry Pi 3 B+ por sus prestaciones, precio, soporte comunitario e infinidad de dispositivos que se pueden agregar, entre los que se encuentra la cámara OV5647 que, al permitir incorporar luces infrarrojas, brinda una imagen de alta definición tanto en el día como en la noche y es especialmente desarrollada para la placa Raspberry PI.

TABLA 2.15
COMPONENTES DEL SISTEMA

Imagen	Nombre y función	Características
	Raspberry Pi 3 B+ Minicomputador que alberga la parte lógica del sistema de detección de somnolencia en conducción diurna.	Procesador Broadcom, cortex-A53 (ArmV8) 64 – bit SoC. Frecuencia 1.4 GHz. 1 Gb LPDDR2 SDRAM. Ethernet 2.4 GHz / 5 GHz. Bluetooth 4.2 GPIO 40 pines HDMI 4 x USB 2.0 CSI (Cámara Raspberry Pi) DSI (Pantalla táctil) Micro SD Micro USB (alimentación)
	Cámara OV5647 para Raspberry Pi Encargada de captar las imágenes	Cámara web 5 MP 1080p Compatible con visión nocturna Luces infrarrojas de alta potencia externas de 3W Longitud focal 3.6 mm
	Memoria MicroSd Encargada de almacenar el SO del dispositivo embebido	Capacidad 32 Gb Clase 10 Hasta 100 Mb/s
	Cargador USB para vehículos Encargado de proveer la fuente de energía a todo dispositivo	2 Puertos 24 Watt 2.0 Amps por puerto Protección contra sobrecarga o cortocircuito
	Buzzer Encargado de generar la alarma sonora	Zumbador de 90 Db
	Cable cargador USB Transporta la alimentación de la fuente hacia el dispositivo	Cable de carga con entrada USB y salida micro USB

Fuente: Propia

2.2.3. Diagrama del detector de somnolencia

La Fig. 12, representa el funcionamiento del sistema de detección de somnolencia diurna.

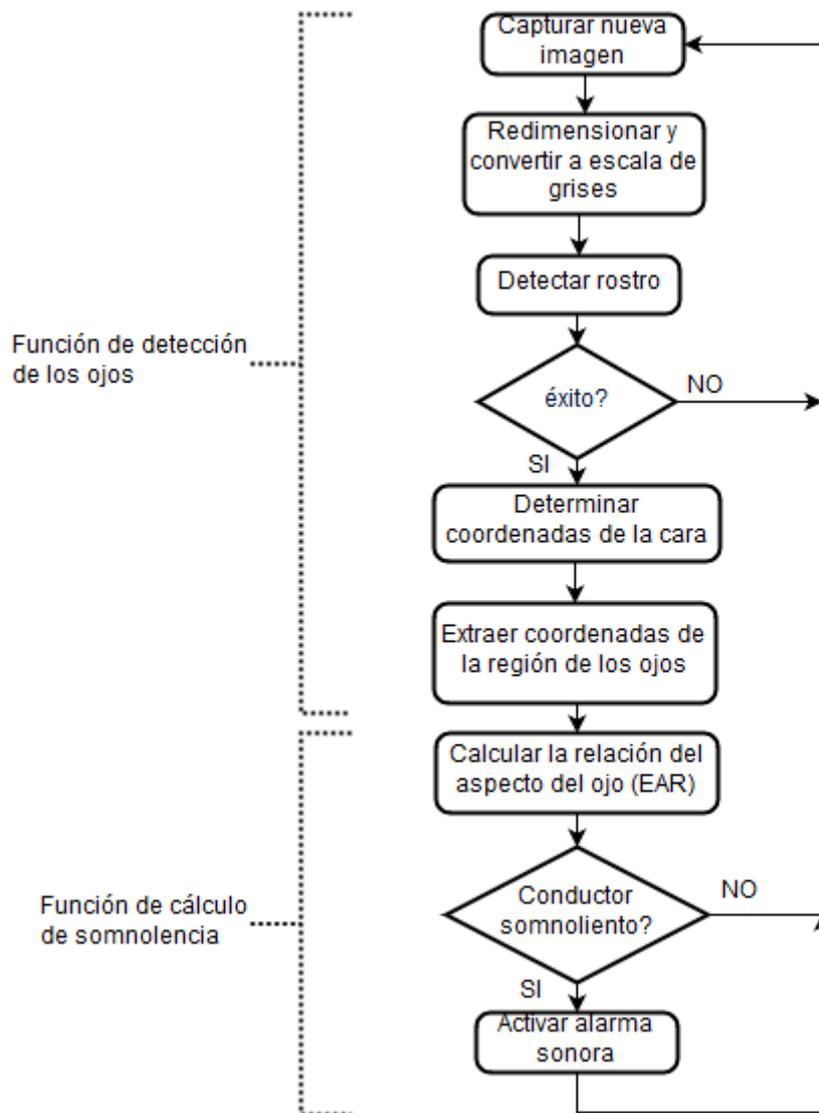


Fig. 12. Flujograma del detector de somnolencia
Fuente: Propia

2.3. Fase de desarrollo

2.3.1. Instalación del SO para Raspberry Pi 3B+

La Raspberry pi 3 B+ puede ser compatible con algunos sistemas operativos, tales como: Raspbian, Fedora, Arch Linux, Kano OS, Windows IoT Core, OSMC, OpenElec, entre otros, cada uno de los cuales tienen características especiales según la necesidad del usuario. Para este caso, se usó **Raspbian**, ya que es el sistema operativo oficial para este pequeño ordenador el cual viene preinstalado software para programación y es el más popular y, por ende, cuenta con amplia información (Santos, 2016).

Para la instalación del sistema operativo, se requiere una memoria microSD con mínimo 8 GB de capacidad y buena velocidad de transferencia ya que la Raspberry Pi no cuenta con almacenamiento interno.

El procedimiento es muy sencillo, se sigue los siguientes pasos:

- Descargar la imagen del archivo de arranque de *Noobs* con Raspbian desde la página oficial de Raspberry Pi (<https://www.raspberrypi.org/downloads/>).
- Formatear la tarjeta micro SD en formato FAT32, en cualquier computador portátil o de mesa.
- Descomprimir el archivo de arranque y copiar todo su contenido dentro de la tarjeta micro SD
- Insertar la tarjeta micro SD en la ranura correspondiente de la Raspberry Pi y seguidamente alimentar la placa con una fuente de poder del voltaje y amperaje adecuados.
- El sistema inicia el proceso de instalación de manera automática, en la cual se debe configurar aspectos básicos como el idioma, región, teclado, etc.

Después de un breve tiempo, el SO está listo y su interfaz se presenta como en la Fig. 13.

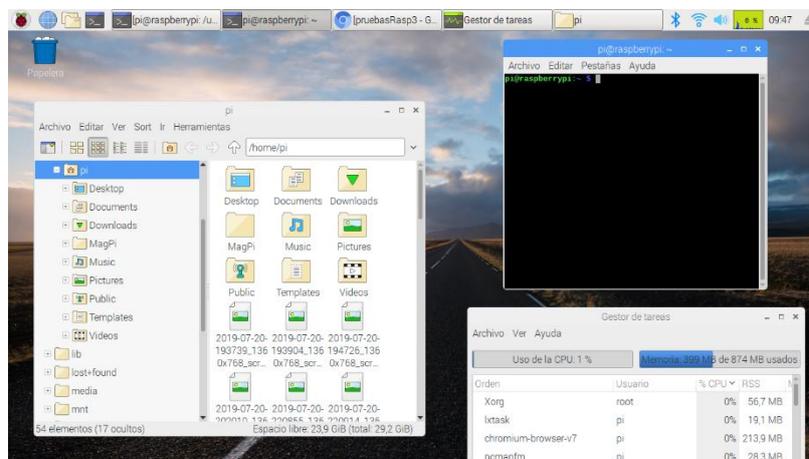


Fig. 13. Interfaz del sistema operativo Raspbian

Fuente: Propia

2.3.2. Instalación de librerías

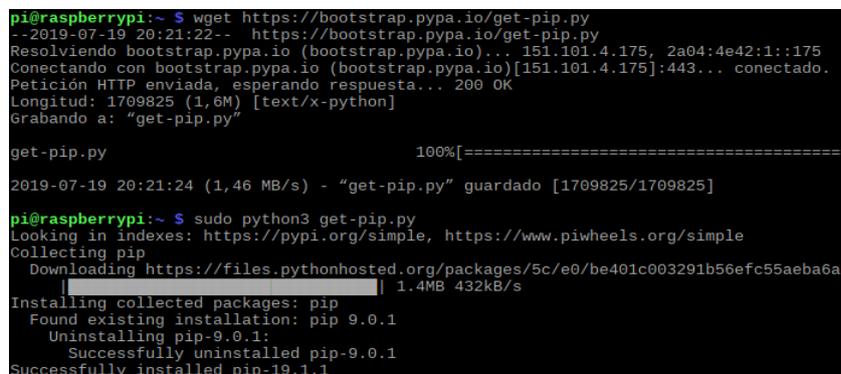
Dentro del entorno del sistema operativo de Raspberry Pi 3 B+, vienen preinstalados Python 2.7 y Python 3.5. Según recomendaciones del especialista en inteligencia artificial Albert Brugués, se debería trabajar en las versiones de Python 3.x, por el simple hecho de que las versiones de Python 2.x, solo recibirán actualizaciones hasta el año 2020 y además las librerías más populares ya están adaptadas a la versión 3 de Python (Brugués, 2019).

La versión que se encuentra configurada por defecto dentro de Raspbian, es Python 2.7, por lo que para poder instalar las librerías dentro del entorno de Python 3.5, se usó el administrador de paquetes de Python denominado *pip3*. La instalación de las librerías se las realiza directamente desde la consola de Raspbian llamado *lxterminal*. El Administrador de paquetes de Python se obtuvo mediante los siguientes comandos:

```
wget https://bootstrap.pypa.io/get-pip.py
```

```
sudo python3 get-pip.py
```

La ejecución de los comandos se muestra en la Fig. 14.



```
pi@raspberrypi:~$ wget https://bootstrap.pypa.io/get-pip.py
--2019-07-19 20:21:22-- https://bootstrap.pypa.io/get-pip.py
Resolviendo bootstrap.pypa.io (bootstrap.pypa.io)... 151.101.4.175, 2a04:4e42:1::175
Conectando con bootstrap.pypa.io (bootstrap.pypa.io)[151.101.4.175]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1709825 (1,6M) [text/x-python]
Grabando a: "get-pip.py"

get-pip.py 100%[=====]
2019-07-19 20:21:24 (1,46 MB/s) - "get-pip.py" guardado [1709825/1709825]

pi@raspberrypi:~$ sudo python3 get-pip.py
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pip
  Downloading https://files.pythonhosted.org/packages/5c/e0/be401c003291b56efc55aeba6a8
  | 1.4MB 432kB/s
Installing collected packages: pip
  Found existing installation: pip 9.0.1
  Uninstalling pip-9.0.1:
    Successfully uninstalled pip-9.0.1
  Successfully installed pip-19.1.1
```

Fig. 14. Instalación del administrador de paquetes Python

Fuente: Propia

Cabe mencionar que anteponer *sudo* a varios de los comandos para instalación de librerías, ayuda a tener privilegios de escritura, dado que se trabajó en el entorno del usuario *PIP* mas no directamente como usuario *Root*. El suprimir esta instrucción tendrá como consecuencia un error de instalación debido a privilegios insuficientes.

- **OpenCV.** – De acuerdo a (Python Package Index, 2019), hay 4 paquetes de OpenCV que pueden ser instalados desde el repositorio de PyPi:
 - opencv-python.* - Contiene solo los módulos principales.
 - opencv-contrib-python.* – Contiene todos los módulos y *contrib* (módulo de reconocimiento facial no incluido en OpenCV)
 - opencv-python-headless.* – Igual que *opencv-python* pero sin GUI
 - opencv-contrib-python-headless.* – Igual que *opencv-contrib-python* pero sin GUIEn este caso el más conveniente es *opencv-contrib-python*, su instalación se realiza desde consola (Fig. 15) digitando:

```
sudo pip install opencv-contrib-python.
```

```
pi@raspberrypi:~$ sudo pip install opencv-contrib-python
Looking in indexes: https://pypi.org/simple, https://www.piwhe
Collecting opencv-contrib-python
  Downloading https://www.piwheels.org/simple/opencv-contrib-p
    12.6MB 1.9MB/s
Requirement already satisfied: numpy>=1.12.1 in /usr/lib/pytho
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-3.4.4.19
```

Fig. 15 Instalación de OpenCV

Fuente: Propia

- **NUMPY.** - Biblioteca de Python que, entre otras cosas, facilita el trabajo con arrays y operaciones matemáticas.

Las funciones de Numpy sirvieron para calcular la distancia euclidiana, que se detalla más adelante.

Para el cálculo de distancias existen varias librerías como numpy, scipy, scikit, math, que devuelven los mismos resultados. Pero según pruebas de rendimiento realizadas por algunos investigadores de una comunidad web (StackOverFlow, 2016), sugieren usar *scipy* o *numpy* cuando sea posible, ya que están vectorizadas y son más rápidas que el código nativo de Python. Ahora, la diferencia entre ambos radica en que *scipy* primero valida los dos vectores de entrada antes de calcular la distancia, mientras que *numpy* lo hace directamente, he ahí el motivo de haber seleccionado las funciones de *numpy*, ya que, en bucles infinitos, la mínima diferencia de tiempo en cada cálculo representa una ganancia en el rendimiento general de la Raspberry Pi 3 B+ que es primordial para este tipo de computador.

La instalación de esta biblioteca se realiza de igual manera desde consola digitando: *sudo pip3 install numpy*

- **IMUTILS.** – Contiene una serie de funciones convenientes para que la funciones de procesamiento de imágenes, rotación, cambio de tamaño, esquematización, detección de bordes, y otras funciones sean más fáciles con OpenCV, Python 2.7 y Python 3 (PythonPackageIndex, 2018).

La instalación de esta librería se realizó mediante el comando: *pip3 install - -user imutils*.

En este caso, la instrucción *- -user* permite instalar el paquete solo para el usuario actual PIP y no para todos los usuarios que haya en el sistema, esto hay que hacerlo por sugerencia del mismo PIP.

- **DLIB.** – Como se revisó en el capítulo anterior, esta biblioteca es muy útil ya que cuenta con una gama de algoritmos de aprendizaje automático utilizables en plataformas móviles y de escritorio. Para la implementación de este sistema, se usó el modelo de predicción de **68 hitos faciales** que es el más adecuado para poder posteriormente realizar el cálculo del radio de aspecto ocular (EAR), el cual se encuentra detallado más adelante.

Antes de la instalación de *Dlib*, es necesario instalar *cmake*, pues es una dependencia para que pueda funcionar. Este paquete se puede instalar desde consola mediante el asistente *apt-get*, el cual primero se actualizó con el comando *sudo apt-get update* para que acceda a las versiones más recientes de los paquetes. Luego, con el comando *sudo apt-get install built-essential cmake* se consigue instalar *cmake* fácilmente.

Adicionalmente, se instaló también OpenBlas con el comando *sudo apt-get install libopenblas-dev liblapack-dev libatlas-base-dev*, el cual tiene optimizaciones de algebra lineal que permiten que las funciones de *Dlib* se ejecuten más rápido (Rosebrock, 2018). A pesar de que esto es opcional, para este sistema cualquier mejora es bienvenida.

Finalmente, la instalación del paquete *Dlib* se realizó por medio del comando *sudo pip3 install dlib*. Además, se requirió descargar el archivo con el modelo entrenado para detectar los 68 hitos faciales denominado “shape_predictor_68_face_landmark.dat” desde el repositorio de *github* de su creador (D. King, 2019). En el momento de ejecutar el sistema, la ruta de este archivo es pasado como parámetro, pues hace uso de este para poder ubicar los hitos faciales del rostro que se detecta por medio de la cámara Pi.

- **RPi.GPIO.** – Esta biblioteca facilita la programación de los pines GPIO que disponen las placas de Raspberry Pi, estos pines pueden ser programados como salidas o entradas de señal con lo cual se puede hacer infinidad de proyectos de automatización utilizando sensores o cualquier dispositivo electrónico. En este sistema, hay algunas funciones que fueron implementadas utilizando estos pines GPIO las cuales son explicadas más adelante. Para poder utilizar estos pines, se debe instalar su biblioteca digitando desde consola la instrucción *pip install RPi.GPIO*.

2.3.3. Habilidad de la cámara Pi

Para poder utilizar la cámara Pi, primero se debe habilitar desde las opciones de configuración del sistema. Para ello hay que digitar *sudo raspi-config*, y desde la ventana que se muestra, activar la cámara en la pestaña de interfaces (Fig 16).

```
pi@raspberrypi:~ $ sudo raspi-config
```

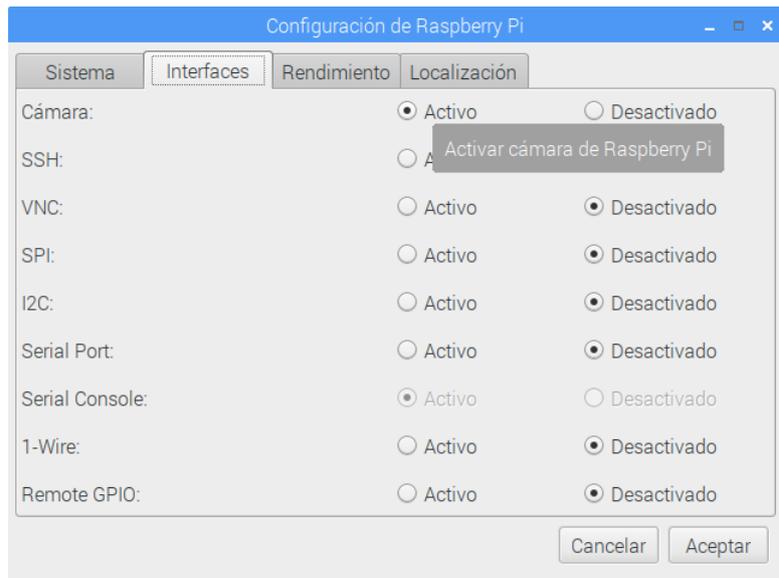


Fig 16. Activación de la cámara Pi

Fuente: Propia

Finalmente, se debe descargar el controlador respectivo digitando desde consola el comando `sudo apt-get install camera`.

A continuación, las siguientes secciones proporcionan una explicación de la lógica del funcionamiento del sistema propuesto y las mejoras que presenta frente a otros sistemas planteados.

2.3.4. Localización de la cara en la imagen

La detección de rostros se puede lograr de varias maneras y cada método puede tener diferentes costes computacionales. Lo que importa es que se necesita obtener el cuadro delimitador de caras, es decir, las coordenadas (x, y) de dicha cara en la imagen. En este caso como ya se había mencionado en el capítulo anterior, se utilizó las **Cascadas de Haar** integradas en OpenCV por su bajo peso computacional.

La detección de rostros utilizando Cascadas de Haar, es un enfoque basado en aprendizaje automático en el que una función de cascada se entrena con un conjunto de datos de entrada (imágenes), por lo que se puede entrenar para detectar casi cualquier cosa. Por otra parte, la librería de OpenCV ya contiene muchos clasificadores pre-entrenados para cara, ojos, sonrisas, etc., así que se hizo uso de uno de ellos para detectar caras. Para poder obtener dicho clasificador, hay que descargar el archivo XML del clasificador entrenado "`haarcascade_frontalface_default.xml`", el cual se encuentra disponible en el repositorio *GitHub* de OpenCV (Menon, 2019).

En primer lugar, se debe capturar la transmisión de vídeo en vivo desde la cámara Pi, pues esta va a ser la materia prima para el funcionamiento de todo el sistema de detección de somnolencia. Para ello, OpenCV dispone de un sencillo paso en donde se debe crear un objeto *VideoCapture* y pasar un índice el cual indica el número de cámara a utilizar. Para este caso, el código queda establecido de la siguiente manera: `vs=VideoStream(usePiCamera=True).start()`, seguidamente se toma el fotograma de la secuencia de vídeo con `frame = vs.read()` y desde allí, se puede hacer el tratamiento y procesamiento de las imágenes.

El fotograma (imagen) capturado, se debe redimensionar a un tamaño adecuado con el fin de que pueda hacer una detección precisa del rostro y a su vez no interfiera en el rendimiento. La finalidad del cambio de tamaño de la imagen, es producir un menor tamaño de datos, lo cual acelera el procesamiento (Hazim Barnouti, 2016).

Existen normas europeas que describen el tamaño mínimo recomendado que debe tener un objeto (Fig. 17), el cual menciona que, para poder detectar rostros, la persona debe ocupar por lo menos el 50% de la altura de la pantalla, y para poder realizar un reconocimiento facial preciso, el objeto humano debe ocupar al menos el 120% de la altura de la pantalla. Hay que tomar en cuenta que la resolución no se aplica al tamaño de la cara sino a toda la imagen. Como se puede observar en la Fig. 18, la imagen presentada tiene una resolución de 54x40 píxeles, pero el área del rostro es mucho más pequeño, el cual es de 20 a 30 píxeles, y es precisamente ésta resolución la que hay que tomar en cuenta (Marciniak, Chmielewska, Weychan, Parzych, & Dabrowski, 2015).

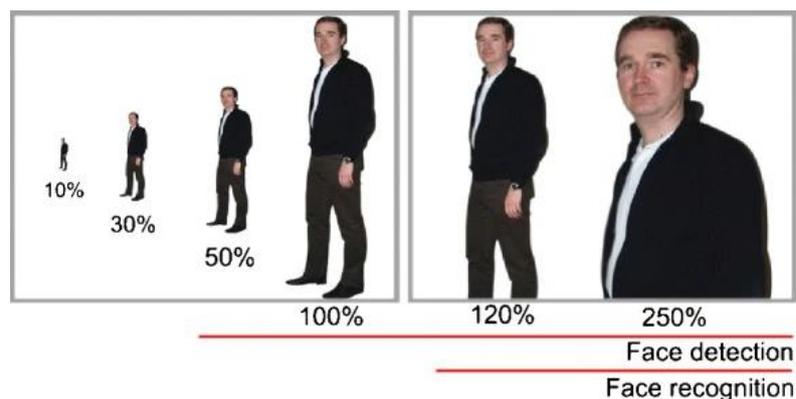


Fig. 17. Tamaño mínimo recomendado del objeto observado según la norma europea "EN 50132-7"

Fuente: (Marciniak et al., 2015)

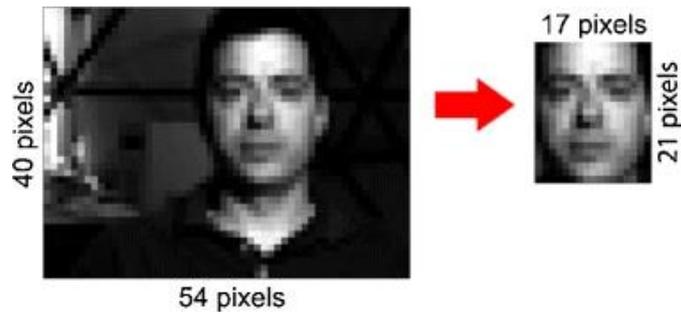


Fig. 18. Detección de rostros en imagen con baja resolución de 54 x 40 píxeles

Fuente: (Marciniak et al., 2015)

La cámara Pi y muchas de las cámaras HD, permiten tener imágenes de 1280x720 píxeles, pero ese tamaño implica un costo computacional alto para su procesamiento y eso se tornaría algo muy dificultoso para la Raspberry Pi 3 B+ y también innecesario.

El detector de rostros Haar Cascade, puede detectar correctamente rostros con una resolución de hasta 24 píxeles, pero la precisión en la detección de rostros también depende de otros factores como la cantidad de iluminación, el ángulo de donde proviene la iluminación, el color de piel, el ángulo de posición del rostro, la distancia a la que se encuentra el rostro (a más distancia menor resolución), etc. La resolución aproximada que puede brindar una detección precisa del rostro en la mayoría de ambientes, es de 250 a 350 píxeles. Si se redimensiona la imagen a 300 píxeles, teóricamente la velocidad de procesamiento se multiplica 4 veces (Marciniak et al., 2015).

De esta manera, la resolución fue establecida en 300 píxeles, puesto que esta resolución asegura un procesamiento rápido para la Raspberry Pi 3 B+, además hay la certeza de que no se va a tener imágenes de rostros que se encuentran lejanos, pues la distancia entre la cámara y el conductor no superar siquiera un metro de distancia. Además, pese a la variabilidad de la iluminación, la luz exterior brinda una iluminación suficiente y por lo general siempre entra por la parte frontal.

Hay que tener en cuenta también que la detección solo funciona en imágenes que se encuentren en escala de grises, por lo que hay que hacer la respectiva conversión de la imagen antes del proceso de detección de rostros. En la Fig. 19 se puede apreciar el aspecto de la imagen antes y después de la conversión a escala de grises por medio de la función: `cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)`.



Fig. 19. (Izquierda) imagen original en modo RGB. (Derecha) imagen convertida a escala de grises
Fuente: Propia

Al implementar el algoritmo, el detector de rostros devuelve las posiciones de las caras detectadas como una lista de rectángulos $rect(x, y, w, h)$ donde: x = coordenada del eje “x” de la esquina superior izquierda, y = coordenada del eje “y” de la esquina superior izquierda, w = ancho del rectángulo, h = altura del rectángulo (Liu, 2013).

Con estas coordenadas y haciendo uso de la función `cv2.rectangle` de OpenCV, se puede graficar un recuadro alrededor del rostro identificado, tal como se aprecia en la Fig. 20



Fig. 20. Detección de rostros con *Haar Cascade* y sus coordenadas (x,y) dentro de la imagen
Fuente: Propia

2.3.5. Detección de estructuras faciales clave

Ahora que ya se dispone de la región de la cara, se procede a detectar los puntos de referencia faciales dentro de dicha región. Hay algunos detectores para ello, pero todos en esencia intentan localizar y etiquetar las cejas derecha e izquierda, ojos derecho e izquierdo, nariz, boca y mandíbula.

Como ya se había mencionado en una sección anterior, el detector de puntos de referencia faciales pre-entrenado incluido en DLIB, sirve para determinar la ubicación de 68

coordenadas (x,y) las cuales se asignan a las estructuras faciales en la cara. Al igual que en la detección de rostros, hay que descargar un archivo denominado “*shape_predictor_68_face_landmarks.dat*” que es pasado como un parámetro obligatorio al ejecutar el programa. Dentro del algoritmo, luego de la detección de rostros, se hace uso del predictor el cual requiere dos parámetros: la imagen en escala de grises y el cuadro delimitador del rostro “*rect*”, *previamente obtenido*.

El detector devuelve un objeto que contiene las 68 coordenadas de las referencias faciales. Haciendo uso de la biblioteca *NumPy*, se puede convertir dicho objeto en una matriz, para de esa manera poder manipular mejor los datos.

Haciendo uso de la función *cv2.circle* de OpenCV, se puede dibujar un círculo en cada una de las coordenadas de los 68 hitos faciales. La Fig. 21 muestra la ubicación de cada una de las coordenadas de las estructuras faciales del rostro detectado.

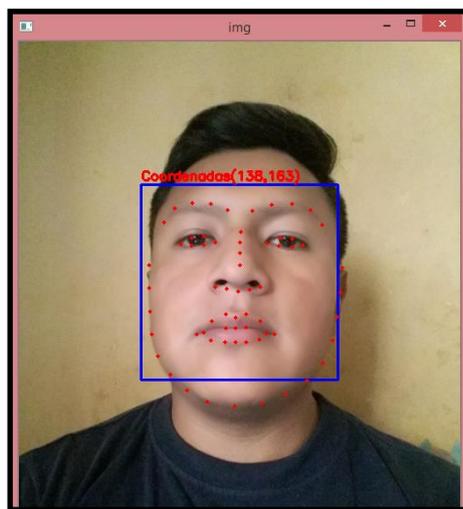


Fig. 21. Visualización 68 coordenadas asignadas a las estructuras faciales de la cara

Fuente: Propia

En la Fig. 22 se puede apreciar como DLIB asigna el orden de los índices de las 68 coordenadas.

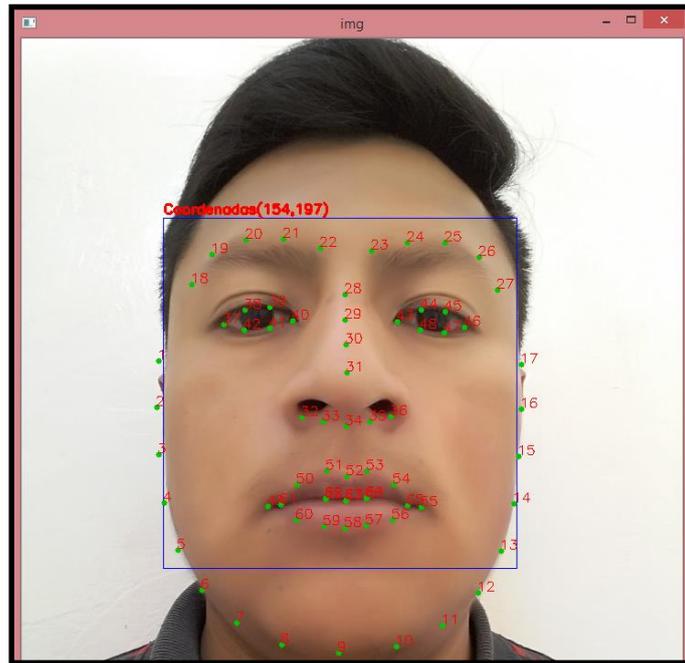


Fig. 22. Visualización de los índices de las 68 coordenadas
Fuente: Propia

2.3.6. Ecuación para el cálculo de la relación del aspecto ocular (EAR)

Con las coordenadas de los puntos de referencia faciales Fig. 23, se puede calcular una métrica denominada *relación de aspecto ocular* (EAR) que fue introducido por (Nathan & Scobell, 2012), la cual deriva en una ecuación que hace un simple cálculo de distancias entre los puntos de referencia faciales de los ojos (Ecuación 1).

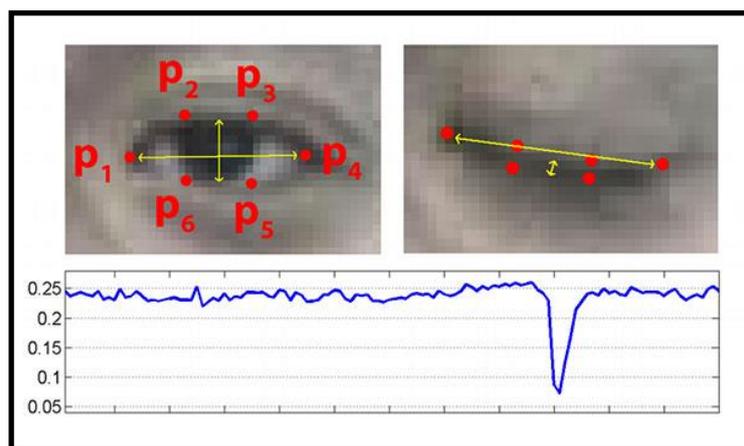


Fig. 23. Ojos abiertos y cerrados con puntos de referencia faciales
Fuente: (Nathan & Scobell, 2012)

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2||p1 - p4||}$$

Ecuación 1. Ecuación de la relación de aspecto ocular (EAR)

Fuente: (Nathan & Scobell, 2012)

Según (Nathan & Scobell, 2012), cuando el ojo está abierto, el valor del EAR es grande y casi constante, mientras que cuando el ojo parpadea, dicho valor EAR baja drásticamente cerca de cero. Esto se puede verificar en la

Fig. 23 donde se muestran los valores que toma el EAR a lo largo del tiempo en un videoclip.

2.3.7. Cálculo del EAR mediante la distancia euclidiana

Como se puede apreciar en la Fig. 24, cada ojo está representado por 6 coordenadas (x,y) las cuales se usó para determinar el estado de somnolencia del conductor. En DLIB, los puntos están indexados del 37-42 para el ojo izquierdo y 43-48 para el derecho.



Fig. 24. Estructuras faciales de los ojos definidas por DLIB

Fuente: Propia

Para el cálculo del EAR en la Ecuación 1 arriba mencionada, se requiere las distancias verticales y horizontales del ojo. Para poder conocer estas distancias, se usa la denominada *distancia euclidiana*, ésta función ya viene incorporada dentro de la clase *linalg.norm* contenida en la biblioteca de *Numpy*, la cual devuelve un número entero.

La distancia euclidiana (Ecuación 2), calcula dentro de un espacio bidimensional, la distancia entre dos puntos P1 y P2, de coordenadas (x1, y1) y (x2, y2), respectivamente (ACADEMIC, 2010).

$$d_E(P1, P2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Ecuación 2. Distancia euclidiana deducida a partir del teorema de Pitágoras

Fuente: (ACADEMIC, 2010)

Aplicando dicha función, el algoritmo almacena en unas variables locales el valor resultante de las distancias euclidianas de los puntos de cada ojo, como en la *TABLA 2.16*.

TABLA 2.16.
VARIABLES EN LAS QUE SE ALMACENA LA DISTANCIA EUCLIDIANA

Ojo derecho	Ojo izquierdo
$A = d_E(P_{38}, P_{42})$	$D = d_E(P_{44}, P_{48})$
$B = d_E(P_{39}, P_{41})$	$E = d_E(P_{45}, P_{47})$
$C = d_E(P_{37}, P_{40})$	$F = d_E(P_{43}, P_{46})$

Fuente: Propia

Luego hace un cálculo simple para determinar el valor del EAR (Ecuación 1) de cada ojo (derecho e izquierdo):

$$EAR_{derecho} = \frac{A + B}{2 * C}$$

$$EAR_{izquierdo} = \frac{D + E}{2 * F}$$

Finalmente, (Nathan & Scobell, 2012) sugieren promediar ambos valores ($EAR_{derecho}$ y $EAR_{izquierdo}$) para obtener una mejor valoración:

$$EAR = \frac{EAR_{derecho} + EAR_{izquierdo}}{2}$$

2.3.8. Pruebas del cálculo del EAR en tiempo real

En el Anexo A, se puede encontrar una tabla con los valores que tomó el EAR en estado “Despierto”, “Medio Dormido” y “Dormido” a lo largo del tiempo en las pruebas realizadas a los siguientes participantes:

- Ing. Erick Herrera (Docente)
- Ing. Marco Pusdá (Docente)
- Srta. Karla Herrera (Estudiante de Software)
- Sr. Josué Alba (Tesisista)

La Fig. 25 es la representación gráfica de los valores (Anexo A) del EAR sobre la secuencia de video captada desde la cámara cuando los participantes estaban en estado “Despierto”, donde se puede apreciar que el EAR toma valores casi constantes para cada uno de los

participantes a lo largo del tiempo. Los picos representan los parpadeos ocurridos durante las pruebas que, al ser tan rápidos, apenas se pueden percibir.

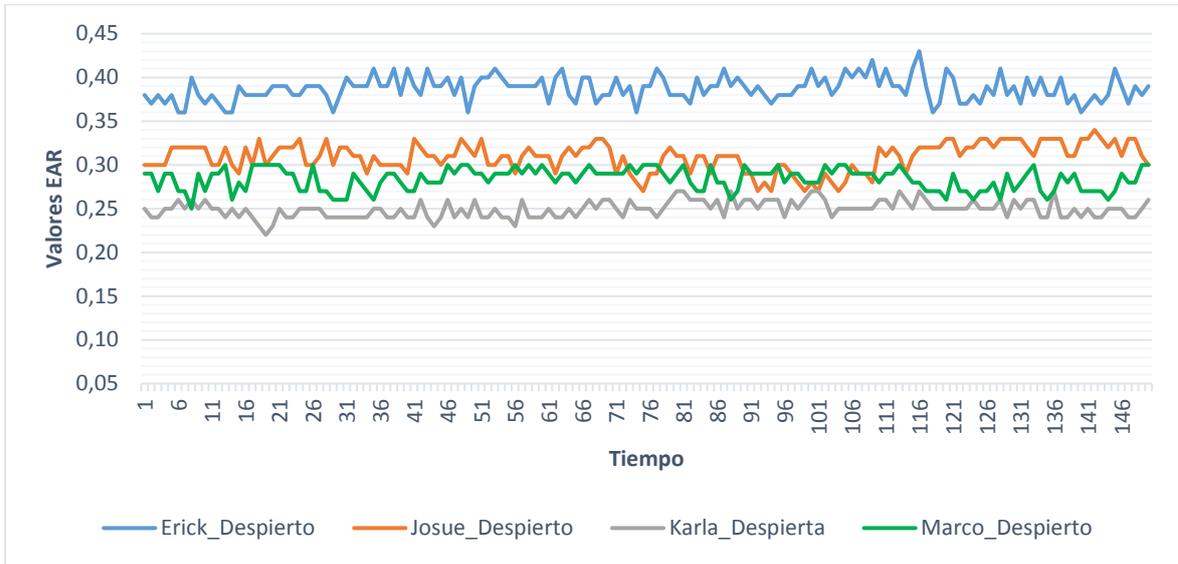


Fig. 25. Representación gráfica de los valores del EAR en estado “Despierto”
Fuente: Propia

La Fig. 26 muestra la representación gráfica de los valores (Anexo A) obtenidos del EAR de los participantes en estado “Medio dormido”

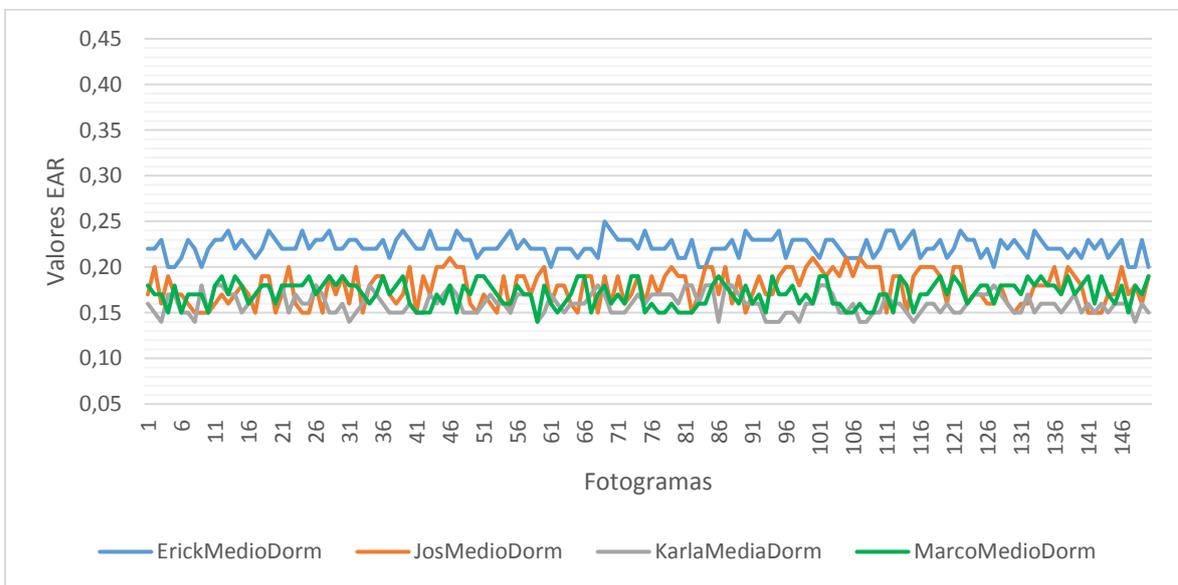


Fig. 26. Representación gráfica de los valores del EAR en estado “Medio dormido”
Fuente: Propia

La Fig. 27 muestra la representación gráfica de los valores (Anexo A) obtenidos del EAR de los participantes en estado “Dormido”

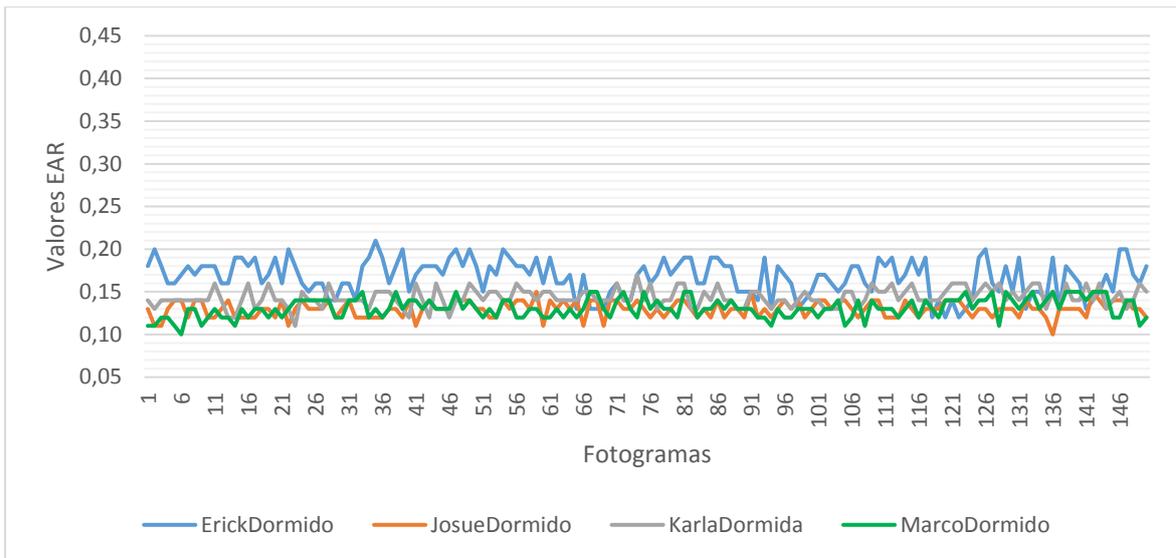


Fig. 27 Representación gráfica de los valores del EAR en estado "Dormido"

Fuente: Propia

2.3.9. Observaciones del umbral del EAR para la detección de somnolencia

En el transcurso del desarrollo del sistema y luego de algunas pruebas preliminares, el autor de este trabajo, llegó a la conclusión que establecer un valor fijo para el umbral del EAR no es muy conveniente. Las mediciones del EAR mostradas en la sección anterior, evidencian una clara variación del valor del EAR en cada persona y esto se debe a las características fisiológicas propias de cada uno.

Como es de conocimiento general, el tamaño de los ojos varía dependiendo de varios factores, los cuales pueden ser raza, género, ubicación geográfica, etc. Esta variación es más evidente aún, si se centra la atención en personas del continente asiático. En la Fig. 28 se puede apreciar la diferencia del aspecto ocular de diferentes personas.



Fig. 28 Ejemplos de tamaño de ojos humanos

Fuente: (DreamsTime, 2019)

Por lo menos, dentro de la investigación que se ha realizado para el desarrollo de este trabajo, no se ha evidenciado que otros investigadores hayan propuesto una solución a este inconveniente.

En la detección de somnolencia, este valor del umbral del EAR es el que ayuda a determinar si una persona está entrando en la etapa del sueño, ya que, si una persona está despierta, por lo general el valor del EAR es más alto y se mantiene casi constante a lo largo del tiempo, pero si dicha persona empieza a cerrar sus ojos, este valor cae inmediatamente y si dicho valor sobrepasa el umbral EAR establecido, se considera que la persona está dormitando y el sistema procede a generar una alarma que despierte al conductor.

En las pruebas realizadas por el autor de esta tesis, se pudo notar que al establecer un valor constante para el umbral del EAR, funciona bien para unas personas, pero no para otras. A modo de ejemplo, si una persona tiene ojos pequeños y el valor definido para el umbral del EAR es alto, con el solo hecho de sonreír, cerrar levemente sus ojos o en ocasiones incluso en modo normal, se producen falsas alarmas de somnolencia (falso positivo). Si, por el contrario, una persona tiene ojos un poco más grandes y el valor definido para el umbral del EAR es bajo, dicha persona tiene que cerrar bien sus ojos para que se produzca una alarma de somnolencia.

Aquí surge la hipótesis de que el valor definido para el umbral del EAR debe ser dinámico y directamente proporcional al tamaño del ojo, es decir, mientras más grande sea el ojo de una persona, el valor definido para el umbral del EAR también debe ser mayor y viceversa.

2.3.10. Propuesta para la configuración del umbral del EAR

Las razones expuestas en la sección anterior, motivaron al autor a darle un valor agregado al sistema de detección de somnolencia, el cual consiste en ejecutar un análisis rápido de los ojos del conductor al iniciar el sistema y definir de manera automática el valor del umbral del EAR que se considera ideal para dicho conductor, específicamente. Esto con el fin de reducir las falsas alarmas por la variación del aspecto ocular de cada persona y brindar un sistema más funcional y preciso.

El proceso consiste en tomar una cantidad mínima pero suficiente de fotogramas que permitan calcular el promedio del EAR del conductor, y en base a ese promedio, definir el valor más idóneo del umbral del EAR.

Para que el sistema pueda configurar el umbral adecuado del EAR, se ha establecido unos sencillos pasos: El conductor debe situarse en posición de manejo mirando normalmente de

frente a la vía, una vez que el sistema haya iniciado su funcionamiento, inmediatamente empieza a analizar el EAR de los ojos de dicho conductor hasta que recoja los valores necesarios, y seguidamente calcular el valor del umbral (dinámico) del EAR para dicha persona.

Pero antes de ello, se determinó que tan confiable puede resultar aplicar el promedio de los valores registrados para determinar el valor del EAR del conductor en cuestión. Para esto se hizo uso de los valores del Anexo A, y se calculó las medidas de tendencia central (media, mediana y moda).

2.3.11. Medidas de tendencia central para los valores del EAR

Para determinar la mejor estimación posible del conjunto de datos del Anexo A, se hizo uso de las medidas de tendencia central, que comprende la media, mediana y moda. Generalmente, la media aritmética o más conocida como promedio, sirve de elemento representativo de un conjunto de valores dados, cuya distribución es aproximadamente simétrica, pues la media resulta ser el “centro de gravedad” del espacio de valores de la población o muestra. Pero si la distribución es muy asimétrica, el valor más frecuente (Moda) o el valor central en el conjunto de datos ordenados (Mediana) podrían ser más representativo (Batanero, 2000).

La TABLA 2.17 muestra las medidas de tendencia central tomadas para el estado de “Despierto”, “Medio dormido” y “Dormido” de cada uno de los participantes a partir de los datos recolectado del Anexo A.

TABLA 2.17
MEDIDAS DE TENDENCIA CENTRAL PARA LOS DATOS RECOLECTADOS

Participate	Estado	\bar{x}	Me	Mo	Desviación estándar
Erick Herrera	Despierto	0,39	0,39	0,39	0,01
	Medio dormido	0,22	0,22	0,22	0,01
	Dormido	0,17	0,17	0,18	0,02
Josué Alba	Despierto	0,31	0,31	0,31	0,02
	Medio dormido	0,18	0,18	0,19	0,02
	Dormido	0,13	0,13	0,13	0,01
Karla Herrera	Despierto	0,25	0,25	0,25	0,01
	Medio dormido	0,16	0,16	0,16	0,01
	Dormido	0,14	0,14	0,14	0,01
Marco PUSDÁ	Despierto	0,28	0,29	0,29	0,01
	Medio dormido	0,17	0,17	0,18	0,01
	Dormido	0,13	0,13	0,13	0,01

Fuente: Propia

Gracias a la colaboración de los participantes, se pudo comprobar la variación de los valores de EAR para cada persona. Cabe recalcar que el aspecto ocular de la Srta. Karla Herrera, se asemejaba mucho al aspecto ocular que podría tener una persona asiática, por lo que los valores resultantes ayudaron mucho para las pruebas y los cálculos posteriores.

En base a estos resultados, se pudo verificar que la distribución de los valores es aproximadamente simétrica, por lo que se podría utilizar la media aritmética simple (promedio) del conjunto de valores que recolecte el sistema para definir el EAR de dicha persona y posterior a ello definir también el umbral dinámico del EAR que indica que está entrando en etapa de somnolencia.

2.3.12. Media geométrica para la estimación más precisa del EAR

Uno de los inconvenientes de la media aritmética simple, es que se ve afectada por los valores muy grandes o muy pequeños del conjunto de valores recolectados. Para ello, lo más recomendable es utilizar la media geométrica (MG), que a pesar de que su cálculo es un poco más complicado que el de la media aritmética, los valores extremos (altos o bajos) tienen menor influencia que en la media aritmética (Paz, 2007).

Si bien es cierto, la mediana es un estimador que trabaja de manera similar a la media geométrica y su cálculo es más sencillo. Sin embargo, la media geométrica podría considerarse como un índice más sabio y preciso, esto debido a que, entre otras conclusiones, el valor numérico de la mediana solo depende del valor del medio de un conjunto de datos, mientras que, en contraste, la media geométrica depende de todos los valores de dicho conjunto (Abyani, Asgarian, & Zarrin, 2019).

La fórmula de la media geométrica para datos no agrupados es:

$$MG = \sqrt[n]{(x_1)(x_2)(x_3) \dots (x_n)}$$

Donde MG es la media geométrica, n es el número total de datos recolectados y X representa cada uno de los valores entrantes (Paz, 2007).

La TABLA 2.18 muestra los resultados del cálculo de la media aritmética (\bar{x}) y media geométrica (MG) para los mismos valores del Anexo A. Ambos resultados son muy similares, esto debido a que las pruebas fueron realizadas bajo la supervisión del autor. Sin embargo, ya en un ambiente real de conducción, el conductor posiblemente no va a actuar de la misma manera que en las pruebas y los valores recolectados presentarán datos variables (altos y bajos) los cuales, si influenciarán en el cálculo del EAR, pero gracias a la media geométrica,

estas variaciones extremas estarán controladas y no presentarán mayor influencia en el promedio del EAR.

TABLA 2.18
COMPARATIVA ENTRE MEDIA ARITMÉTICA Y MEDIA GEOMÉTRICA PARA LOS VALORES RECOLECTADOS

Participante	Estado	\bar{x}	MG_EAR
Erick Herrera	Despierto	0,387	0,387
	Medio dormido	0,223	0,225
	Dormido	0,167	0,169
Josué Alba	Despierto	0,309	0,308
	Medio dormido	0,177	0,177
	Dormido	0,130	0,129
Karla Herrera	Despierto	0,249	0,249
	Medio dormido	0,160	0,160
	Dormido	0,144	0,143
Marco PUSDÁ	Despierto	0,284	0,283
	Medio dormido	0,171	0,171
	Dormido	0,131	0,130

Fuente: Propia

2.3.13. Configuración dinámica del umbral del EAR

Una vez obtenido la media geométrica del EAR del conductor, solo resta establecer el umbral del EAR. Para ello, con una regla de tres simple, se determinó el porcentaje de apertura del ojo en base al valor del EAR en los estados “Despierto”, “Medio dormido” y “Dormido”. Dicho porcentaje se calculó asumiendo que el valor del EAR en estado “Despierto” representa el 100% de la apertura del ojo. Cabe mencionar que el porcentaje de apertura del ojo en estado “Dormido” no llega al 0% porque el valor del EAR solo se aproxima a cero mas no necesariamente llega a cero al tener los ojos completamente cerrados.

TABLA 2.19
PORCENTAJE DE APERTURA DE LOS OJOS PARA EL ESTADO "DESPIERTO", "MEDIO DORMIDO" Y "DORMIDO"

Participante	Estado	MG_EAR	Porcentaje
Erick Herrera	Despierto	0,387	100%
	Medio dormido	0,225	57%
	Dormido	0,169	43%
Josué Alba	Despierto	0,308	100%
	Medio dormido	0,177	57%
	Dormido	0,129	42%
Karla Herrera	Despierto	0,249	100%
	Medio dormido	0,160	64%
	Dormido	0,143	58%
Marco PUSDÁ	Despierto	0,283	100%
	Medio dormido	0,171	60%
	Dormido	0,130	46%

Fuente: Propia

El valor del EAR cuando la persona está en estado “Medio dormido” varía un poco en cada individuo, pero se puede inferir, en base a las pruebas preliminares, que bajo el 65% ya se puede considerar que la persona está entrando en estado de somnolencia.

Se tomó como punto de análisis los valores del EAR en el estado “Medio dormido”, debido a que, a partir de ese valor, el sistema debe ya generar una alerta de somnolencia, pues el conductor está empezando a dormir.

Para validar el porcentaje establecido (65%) para el inicio del estado de somnolencia, se acudió a la experimentación del sistema con este valor definido en su algoritmo, el cual está demostrado más adelante.

Si, por ejemplo, el sistema por medio de la media geométrica calcula que el valor del EAR del conductor es de 0.30, procede a calcular el 65% de ese valor, es decir, internamente realiza el siguiente cálculo:

$$\text{umbralEar} = 0.30 * 0.65$$

$$\text{umbralEar} = 0.20$$

Esto significa que, una vez finalizada la configuración automática, y si durante el análisis de los ojos en tiempo real, el valor del EAR del conductor baja de 0.20, el sistema inmediatamente generará una alerta sonora de somnolencia, de lo contrario, si el conductor está despierto, el valor del EAR será superior a 0.20 y se mantendrá casi constante a lo largo del tiempo sin generar alertas.

2.3.14. Tamaño de la muestra para el cálculo del EAR

Como se había mencionado en las secciones anteriores, el sistema debe procesar una cantidad suficiente de fotogramas que le permitan calcular dinámicamente el valor del EAR del conductor por medio de la media geométrica.

Hay que tomar en cuenta que el tiempo de configuración no debe ser prolongado para no fastidiar al conductor, pero tampoco puede ser muy corto para no afectar la precisión del EAR.

Los valores registrados en el Anexo A, corresponden a una secuencia de 150 fotogramas, lo cual equivale a estar aproximadamente 40 segundos frente a la cámara, pero ese tiempo había que bajarlo ya que una persona no estaría en condiciones de mantenerse casi un minuto frente a la cámara hasta que se configure, y menos aún si está con apuro.

Como ya se comprobó anteriormente que los valores del EAR en estado despierto a lo largo del tiempo son aproximadamente simétricos, se utilizó la técnica del muestreo aleatorio

simple para encontrar el tamaño necesario de la muestra (fotogramas) y realizar el cálculo de la media geométrica.

La fórmula para el cálculo del muestreo aleatorio simple considerando que se desconoce la población es (Dra Bustamante Gladys, 2011):

$$n = \frac{Z^2 \times P \times Q}{E^2}$$

Donde:

n = *Tamaño requerido de la muestra*

Z = *Nivel de confianza (valores pre establecidos TABLA 2.20)*

P = *Variabilidad positiva (definido en 50% \approx 0.5 debido a que no se sabe el valor que se requiere encontrar)*

Q = *Variabilidad negativa (1 - P)*

E = *Error dispuesto a aceptar*

TABLA 2.20
NIVEL DE CONFIANZA

Nivel de confianza	Z
99,70%	3
99%	2,58
98%	2,33
96%	2,05
95%	1,96
90%	1,645
80%	1,28
50%	0,674

Fuente: (Dra Bustamante Gladys, 2011)

El nivel de confianza con el que se trabajó para este caso es del 95% ($Z=1.96$) ya que es el nivel que permite una extrapolación adecuada de resultados (Dra Bustamante Gladys, 2011).

El error que se consideró aceptar como máximo fue del 15% (0.15), esto gracias a que se colocó un filtro para descartar valores inválidos del EAR al momento de la configuración, lo cual ayuda a mejorar la precisión desde la toma de los datos. Este filtro es una condición que simplemente desecha los fotogramas en los cuales el EAR es menor o igual a 0.22, pues el conductor debe estar con los ojos abiertos al momento de la configuración del sistema, y por más pequeños que sean sus ojos, el valor del EAR estando abiertos, no va a ser más bajo que 0.22, esto se concluyó luego de las pruebas realizadas con varios voluntarios. Es así,

que se evita tomar valores cuando el conductor cierra los ojos voluntaria o involuntariamente, lo cual podría afectar en la estimación del EAR del conductor.

Reemplazando los valores en la fórmula y realizando los respectivos cálculos, se obtuvo el siguiente resultado:

$$n = \frac{(1.96)^2 \times (0.5) \times (0.5)}{(0.15)^2}$$
$$n = 42.68$$

El valor resultante de n precisó la cantidad de fotogramas válidos que el sistema debe analizar para realizar una correcta estimación del EAR del conductor en cuestión. Se precisó redondear a $n=40$ para tener un número exacto y disminuir un poco el tiempo de cálculo sin que esto represente un cambio significativo en el resultado. En base a esto, el sistema entonces toma 40 valores válidos del EAR sobre la secuencia de vídeo en tiempo real y luego calcula la media geométrica de dichos valores para finalmente en base a ese resultado, definir en umbral dinámico del EAR, tal como se explicó en las secciones anteriores.

El sistema procesa alrededor de 5 fotogramas por segundo cuando hay la presencia de un rostro, esto quiere decir que, si todos los valores analizados del EAR son válidos, el sistema toma alrededor de 8 segundos en configurar automáticamente el valor del umbral del EAR adecuado para el conductor que se encuentre en frente.

2.3.15. Cálculo de la media geométrica, definición del umbral y detección de somnolencia en tiempo real

Con los métodos propuestos en las secciones anteriores ya implementados en el algoritmo de detección de somnolencia, se procedió a probar su nivel de eficacia.

La cámara adquirida para este trabajo de tesis estaba integrada con leds de iluminación infrarroja, lo cual permitió que se pueda percibir el entorno incluso sin nada de iluminación, de esta manera, este sistema puede también funcionar en la noche.

La Fig. 29 muestra el proceso de configuración del sistema. Como se puede apreciar, primero recolectó 40 valores válidos del EAR, luego con ese conjunto de valores calculó la media geométrica ($MG_EAR_TOTAL = 0.34$) y finalmente estableció el valor del umbral del EAR (Umbral definido = 0.20) abordado en la sección 2.3.13.

```

pi@raspberrypi: ~/D
Archivo Editar Pestañas Ayuda
[INFO] loading facial landmark predictor...
[INFO] starting video stream thread...
cont: 1, EAR: 0.33422620512171064 mg_EAR_acumulado=0.33422620512171064
cont: 2, EAR: 0.3400816177614201 mg_EAR_acumulado=0.1136641885427361
cont: 3, EAR: 0.35506874590373527 mg_EAR_acumulado=0.040358608880635025
cont: 4, EAR: 0.3305167024009182 mg_EAR_acumulado=0.013339191676383972
cont: 5, EAR: 0.32968588354591497 mg_EAR_acumulado=0.004396676658282954
cont: 6, EAR: 0.3160727615094344 mg_EAR_acumulado=0.0013896695432038767
cont: 7, EAR: 0.31545496963920394 mg_EAR_acumulado=0.0004383781635599053
cont: 8, EAR: 0.3394523901514444 mg_EAR_acumulado=0.00014880851541661068
cont: 9, EAR: 0.3099980404086601 mg_EAR_acumulado=4.6130348173411204e-05
cont: 10, EAR: 0.33457001418174925 mg_EAR_acumulado=1.5433831242587217e-05
cont: 11, EAR: 0.3241861165133197 mg_EAR_acumulado=5.003433813456293e-06
cont: 12, EAR: 0.3244247772457918 mg_EAR_acumulado=1.6232379003946204e-06
cont: 13, EAR: 0.3430278028852294 mg_EAR_acumulado=5.568157385323995e-07
cont: 14, EAR: 0.3391732446356349 mg_EAR_acumulado=1.888569979888353e-07
cont: 15, EAR: 0.3388666220011873 mg_EAR_acumulado=6.399733294976165e-08
cont: 16, EAR: 0.32568235806292656 mg_EAR_acumulado=2.00428023048166e-08
cont: 17, EAR: 0.3157169946570261 mg_EAR_acumulado=6.5804269039072336e-09
cont: 18, EAR: 0.3391478677705613 mg_EAR_acumulado=2.2317377534801743e-09
cont: 19, EAR: 0.3388666220011873 mg_EAR_acumulado=7.562614337143451e-10
cont: 20, EAR: 0.3290433290290251 mg_EAR_acumulado=2.488427797656315e-10
cont: 21, EAR: 0.3538537501235025 mg_EAR_acumulado=8.805395081122554e-11
cont: 22, EAR: 0.35506874590373527 mg_EAR_acumulado=3.1265205886411045e-11
cont: 23, EAR: 0.345176066926888 mg_EAR_acumulado=1.0792000792208465e-11
cont: 24, EAR: 0.33940980511454244 mg_EAR_acumulado=3.662910885679463e-12
cont: 25, EAR: 0.330539570578632 mg_EAR_acumulado=1.210736991220286e-12
cont: 26, EAR: 0.3553753685381829 mg_EAR_acumulado=4.3026610445771985e-13
cont: 27, EAR: 0.3403202785138921 mg_EAR_acumulado=1.4642828050413863e-13
cont: 28, EAR: 0.330258324809258 mg_EAR_acumulado=4.8359158623996955e-14
cont: 29, EAR: 0.3437911205022126 mg_EAR_acumulado=1.662544932988815e-14
cont: 30, EAR: 0.3449388302971256 mg_EAR_acumulado=5.734763045015749e-15
cont: 31, EAR: 0.3259189185418341 mg_EAR_acumulado=1.8690677697252084e-15
cont: 32, EAR: 0.3244652620291292 mg_EAR_acumulado=6.064475636540899e-16
cont: 33, EAR: 0.3526366540897051 mg_EAR_acumulado=2.1385563972783173e-16
cont: 34, EAR: 0.34795266730570035 mg_EAR_acumulado=7.441164027449729e-17
cont: 35, EAR: 0.3301732446356349 mg_EAR_acumulado=2.5238437470560933e-17
cont: 36, EAR: 0.3526366540897051 mg_EAR_acumulado=8.89998144070849e-18
cont: 37, EAR: 0.353851649869938 mg_EAR_acumulado=3.149279027118856e-18
cont: 38, EAR: 0.3099980404086601 mg_EAR_acumulado=9.762703271069369e-19
cont: 39, EAR: 0.34062890114833977 mg_EAR_acumulado=3.3254393620551194e-19
cont: 40, EAR: 0.3157169946570261 mg_EAR_acumulado=1.0498977213022203e-19
MG_EAR_TOTAL: 0.34 Umbral definido: 0.2

```

Fig. 29 Registro de valores del EAR, cálculo de MG y cálculo del umbral para Josué Alba
Fuente: Propia

Luego de la configuración, el sistema comenzó el proceso de monitoreo de los ojos en tiempo real, mostrando en pantalla el valor establecido para el umbral del EAR y el valor actual del EAR. Como se puede apreciar en la primera imagen de la Fig. 30, en estado “Despierto” el valor del EAR para ese momento fue de 0.33 y por ende el sistema no generó ninguna alarma. Al entrar en estado “Medio dormido” (imagen del centro), el valor del EAR para ese instante fue de 0.16, por ende, el sistema generó una alerta sonora de somnolencia ya que se sobrepasó el umbral establecido a partir del cual se considera que el conductor está empezando a dormitar. En la tercera imagen, se puede apreciar que ya en el estado “Dormido”, el valor del EAR para ese instante fue de 0.12, por ende, el sistema también generó una alerta de somnolencia debido a que dicho valor estaba muy por debajo del umbral establecido y se consideró como somnolencia.



Fig. 30 Monitoreo diurno en estado "Despierto, medio dormido y dormido" – Josué Alba

Fuente: Propia

El mismo proceso se aplicó también para el modo nocturno, como se puede apreciar en la Fig. 31 el sistema estableció el umbral en 0.20, y cuando los valores del EAR eran más bajos que dicho umbral, se generó de inmediato una alerta de somnolencia.



Fig. 31. Monitoreo nocturno en estado "Despierto, medio dormido y dormido" – Josué Alba

Fuente: Propia

2.3.16. Inicio automático del detector de somnolencia

Algo también muy importante en el desarrollo de este sistema, fue que debía iniciar su funcionamiento de manera automática una vez que sea instalado el sistema en un vehículo, pues ya no se dispondría de ningún tipo de pantalla que permita navegar y ejecutar el detector de somnolencia de forma manual.

Para ello, primero se procedió a crear un script dentro de cualquier directorio, en este caso, como muestra la Fig. 32 se creó el script dentro de *Documents* con el nombre "autoiniciocomnolencia.sh".

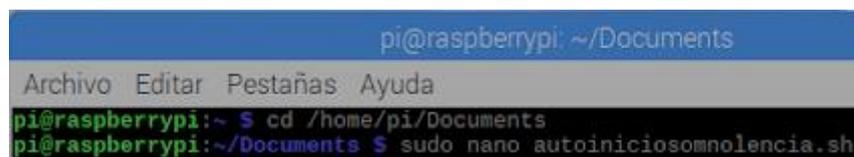
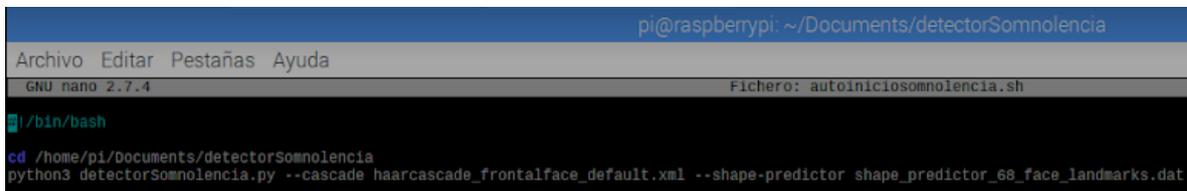


Fig. 32 Creación del archivo script *autoiniciocomnolencia.sh*

Fuente: Propia

Acto seguido, dentro del fichero que se abre, se agregó las siguientes instrucciones Fig. 33, en donde por medio de consola primero accede a la ruta donde está ubicado el archivo

del detector de somnolencia y luego ejecuta el script Python del detector de somnolencia pasando los parámetros necesarios, que en este caso viene a ser el detector de rostros pre-entrenado de OpenCV y el detector de hitos faciales de *Dlib* los cuales ya fueron previamente descargados:



```
pi@raspberrypi: ~/Documents/detectorSomnolencia
Archivo Editar Pestañas Ayuda
GNU nano 2.7.4 Fichero: autoiniciosomnolencia.sh
|/bin/bash
cd /home/pi/Documents/detectorSomnolencia
python3 detectorSomnolencia.py --cascade haarcascade_frontalface_default.xml --shape-predictor shape_predictor_68_face_landmarks.dat
```

Fig. 33 Instrucciones del script para ejecutar el detector de somnolencia

Fuente: Propia

Luego con el comando mostrado en la Fig. 34 se procedió a establecer el script “autoiniciosomnolencia.sh” como ejecutable.

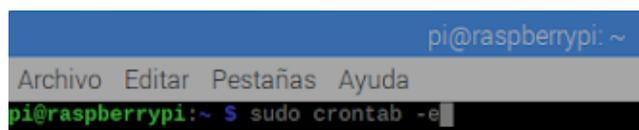


```
pi@raspberrypi:~/Documents $ chmod +x autoiniciosomnolencia.sh
```

Fig. 34 Configuración del script como ejecutable

Fuente: Propia

Hay algunos métodos que se pueden usar para ejecutar un script de forma automática al momento de iniciar el dispositivo. En este caso, se usó *crontab*, que básicamente es un servicio que se inicia en el arranque del sistema. Así que, desde la raíz del usuario, hay que digitar el siguiente comando para abrir el archivo (Fig. 35):

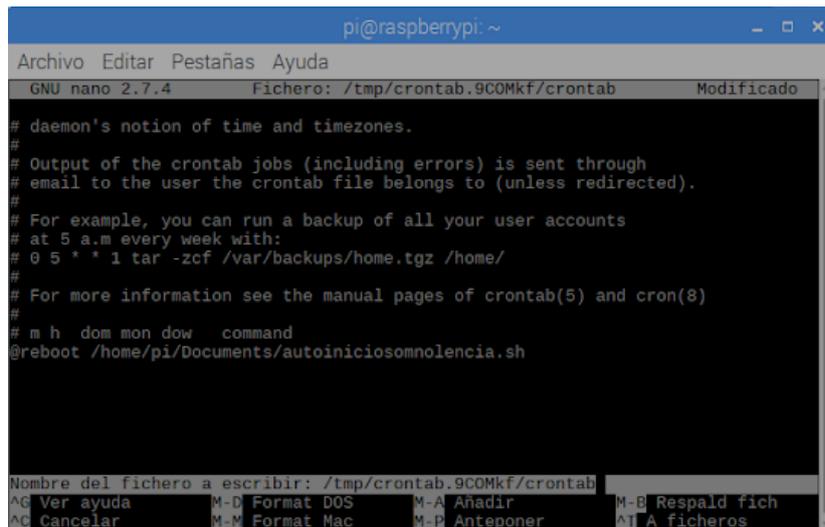


```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo crontab -e
```

Fig. 35 Acceso al archivo *crontab*

Fuente: Propia

Al final del archivo se agregó la línea que se muestra en la Fig. 36 , la cual ordena que se dirija a la ruta especificada y ejecute el script antes creado cada vez que se reinicie el dispositivo.



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 2.7.4 Fichero: /tmp/crontab.9COMkf/crontab Modificado
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
@reboot /home/pi/Documents/autoiniciosomnolencia.sh
Nombre del fichero a escribir: /tmp/crontab.9COMkf/crontab
^G Ver ayuda      M-D Format DOS   M-A Añadir       M-B Respald fich
^G Cancelar      M-M Format Mac   M-D Anteponer    M-I A ficheros
```

Fig. 36 Instrucción para ejecución automática del script en el reinicio del sistema
Fuente: Propia

Finalmente, solo basta guardar y aceptar los cambios presionando CTRL+X y el script está listo para ser iniciado de forma automática cada vez que se reinicia el sistema.

2.3.17. Estructura del Hardware

El hardware del prototipo básicamente comprende 3 partes:

- La placa principal (Raspberry Pi 3b+) responsable del procesamiento del sistema.
- La placa *protoboard* desde la cual se puede encender / apagar el sistema, monitorear el proceso de configuración y verificar que la cámara está detectando los ojos del conductor.
- La cámara OV5647, que es la responsable de capturar las imágenes, tanto en ambientes claros como en ambientes oscuros gracias a los diodos infrarrojos que tiene integrado.
- Para las funciones principales como apagado, encendido y monitoreo del sistema, se usó las salidas GPIO (General Purpose Input/Output) incorporados en la Raspberry Pi 3B+ (Fig 37), para controlar leds y pulsadores. Esto con el fin de facilitar su uso y monitoreo debido a que, una vez instalado en la cabina de un vehículo, no se dispone de una pantalla (aunque se puede agregar opcionalmente) para verificar que está sucediendo a nivel de software. Las conexiones GPIO de este sistema fueron estructuradas de la siguiente manera:
- Un pulsador conectado a una salida GND y al GPIO3, el cual, mediante programación, ha sido configurado para apagar el sistema de forma segura y

también para volver a iniciarlo. De esta manera, se puede garantizar la integridad tanto del software como del hardware.

- Un led de color **verde** conectado a una salida GND y al GPIO4, el cual, mediante programación, ha sido configurado para que se mantenga encendido cuando el algoritmo entra en funcionamiento.
- Un led de color **azul** conectado a una salida GND y al GPIO27, el cual ha sido programado para que se encienda de forma intermitente cuando está detectando los ojos del conductor. Esto a su vez sirve para calibrar manualmente la posición de la cámara en caso de que no esté detectando los ojos.
- Un led de color **rojo** conectado a una salida GND y al GPIO24, el cual se programó para que se encienda intermitentemente cuando el sistema ha detectado somnolencia.
- Un *buzzer* conectado a una salida GND y al GPIO7, el cual se programó para que active la alarma sonora cuando el sistema ha detectado somnolencia y trate de alertar al conductor.

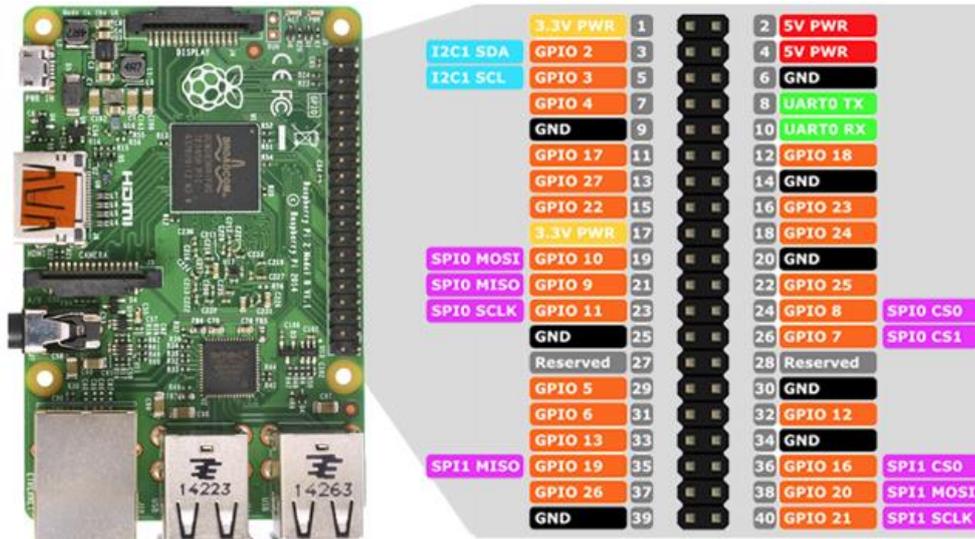


Fig 37. Pines GPIO para Raspberry Pi Modelo 3B+

Fuente: (Abellán, 2018)

2.4. Fase de pruebas

Al momento de suministrar energía al sistema, éste sigue la siguiente secuencia:

- Arranca el sistema operativo (SO) y carga las herramientas necesarias.

- Una vez listo el SO, ejecuta automáticamente un script que, por medio de consola, invoca al archivo donde se encuentra alojado el algoritmo de detección de somnolencia que se le nombró como “*detectorsomnolencia.py*”.
- El sistema en primer lugar entra en etapa de configuración, la cual consta en recolectar 40 valores válidos del EAR del conductor para posteriormente definir el umbral dinámico adecuado para dicho conductor. Este proceso puede ser revisado en detalle en las secciones **2.3.13** y **2.3.14**.
- El sistema dura aproximadamente 8 segundos en configurarse y una vez finalizado, se escuchan tres pitidos cortos seguidos de un pitido largo, que confirma que el sistema se ha configurado con éxito.
- A partir de este momento, el sistema empieza a monitorear los ojos del conductor. Si un led de color azul se prende de forma intermitente, significa que se está detectando los ojos, de lo contrario, es posible que la cámara esté mal direccionada o simplemente no haya ninguna persona en frente.
- Cuando el valor del EAR sea igual o menor que el umbral establecido, el sistema emite una alerta sonora por medio del *buzzer* y enciende a la vez un led de color rojo. Esta alerta se mantendrá activa mientras el EAR se mantenga por debajo del umbral. Si el conductor reacciona y abre los ojos, la alarma y el led rojo se apagan y se continúa monitoreando.

2.4.1. Pruebas en ambiente controlado

La Fig. 38 muestra el proceso de configuración del sistema en las pruebas realizadas por el Ing. Erick Herrera en modo **diurno**. Para este caso, el sistema luego de recolectar los 40 valores de EAR necesarios, estimó su valor medio como $MG_EAR_TOTAL=0.35$ y estableció el Umbral definido=0.19.

```

autoiniciosomnolencia.sh
Archivo Editar Pestañas Ayuda
[INFO] loading facial landmark predictor...
[INFO] starting video stream thread...
cont: 1, EAR: 0.3353168454568044 mg_EAR_acumulado=0.3353168454568044
cont: 2, EAR: 0.33515296529174843 mg_EAR_acumulado=0.11238243506712293
cont: 3, EAR: 0.3385708211198222 mg_EAR_acumulado=0.03804941332012091
cont: 4, EAR: 0.32316227264932895 mg_EAR_acumulado=0.012296134881503924
cont: 5, EAR: 0.3716117511425766 mg_EAR_acumulado=0.004569388215600992
cont: 6, EAR: 0.3451971096261452 mg_EAR_acumulado=0.0015773396047852315
cont: 7, EAR: 0.34523359839573425 mg_EAR_acumulado=0.0005445506276521108
cont: 8, EAR: 0.3594182728898624 mg_EAR_acumulado=0.00019572144609181222
cont: 9, EAR: 0.31469253363088373 mg_EAR_acumulado=6.159267775653281e-05
cont: 10, EAR: 0.34815835163610104 mg_EAR_acumulado=2.1443796265557028e-05
cont: 11, EAR: 0.3314800540144687 mg_EAR_acumulado=7.108190744382105e-06
cont: 12, EAR: 0.3869481360663183 mg_EAR_acumulado=2.750501159342511e-06
cont: 13, EAR: 0.3716117511425766 mg_EAR_acumulado=1.0221185523429578e-06
cont: 14, EAR: 0.3703435741288301 mg_EAR_acumulado=3.785350378580767e-07
cont: 15, EAR: 0.35622799501462554 mg_EAR_acumulado=1.3484477757896804e-07
cont: 16, EAR: 0.3265902299416978 mg_EAR_acumulado=4.403898691595226e-08
cont: 17, EAR: 0.3169088933031665 mg_EAR_acumulado=1.3956346605727062e-08
cont: 18, EAR: 0.3594182728898624 mg_EAR_acumulado=5.016165992882714e-09
cont: 19, EAR: 0.37338738232755925 mg_EAR_acumulado=1.872973089402999e-09
cont: 20, EAR: 0.3200452520072072 mg_EAR_acumulado=5.994361444007001e-10
cont: 21, EAR: 0.3716958480686875 mg_EAR_acumulado=2.2280792605604245e-10
cont: 22, EAR: 0.37270298378521155 mg_EAR_acumulado=8.30411788520818e-11
cont: 23, EAR: 0.3604493976237759 mg_EAR_acumulado=2.9932142895201124e-11
cont: 24, EAR: 0.33892349096964147 mg_EAR_acumulado=1.0144706362243717e-11
cont: 25, EAR: 0.31622776601683794 mg_EAR_acumulado=3.2080378298291334e-12
cont: 26, EAR: 0.36198463000973014 mg_EAR_acumulado=1.1612603868879165e-12
cont: 27, EAR: 0.37297902662862137 mg_EAR_acumulado=4.3312576876383137e-13
cont: 28, EAR: 0.37297902662862137 mg_EAR_acumulado=1.6154682764130716e-13
cont: 29, EAR: 0.36198463000973014 mg_EAR_acumulado=5.847746863298422e-14
cont: 30, EAR: 0.3269399502145375 mg_EAR_acumulado=1.911862068354004e-14
cont: 31, EAR: 0.32846063030632056 mg_EAR_acumulado=6.2797142003030185e-15
cont: 32, EAR: 0.3588206315884924 mg_EAR_acumulado=2.2532910155479536e-15
cont: 33, EAR: 0.3387496591127825 mg_EAR_acumulado=7.633015633987647e-16
cont: 34, EAR: 0.34523359839573425 mg_EAR_acumulado=2.635173453932452e-16
cont: 35, EAR: 0.34523359839573425 mg_EAR_acumulado=9.097504138980162e-17
cont: 36, EAR: 0.3232453172891314 mg_EAR_acumulado=2.940725611943829e-17
cont: 37, EAR: 0.3299813103981035 mg_EAR_acumulado=9.703844909504896e-18
cont: 38, EAR: 0.3301968754545348 mg_EAR_acumulado=3.2041792690139094e-18
cont: 39, EAR: 0.3299813103981035 mg_EAR_acumulado=1.0573192739396472e-18
cont: 40, EAR: 0.34544916345216553 mg_EAR_acumulado=3.6525005868430215e-19
MG_EAR_TOTAL: 0.35 Umbral definido: 0.19

```

Fig. 38 Registro de valores del EAR, cálculo de MG y cálculo del umbral para Erick Herrera

Fuente: Propia

La primera imagen de la Fig. 39 muestra el valor del EAR del Ing. Erick Herrera para ese instante en estado “**Despierto**” sin generar ninguna alarma. La segunda imagen muestra el valor del EAR para ese instante en estado “**Dormido**”, momento para el cual ya se generó la alerta sonora de somnolencia por haber sobrepasado el umbral establecido.



Fig. 39 Monitoreo diurno en estado "Despierto y dormido" – Erick Herrera

Fuente: Propia

La Fig. 40 muestra el funcionamiento del sistema con otra voluntaria, en la que se puede apreciar el valor del umbral definido de acuerdo con el aspecto ocular de dicha persona y la activación de la alerta al detectar síntomas de somnolencia.



Fig. 40. Monitoreo diario de una voluntaria en estado despierto y dormido

Fuente: Propia

2.4.2. Pruebas de temperatura y consumo de recursos del sistema

El administrador de tareas de Raspbian permitió visualizar de manera rápida el estado del CPU y Memoria RAM del sistema. Mientras que el comando `vcgencmd measure_temp` ayudó a visualizar la temperatura del CPU.

La Fig. 41 muestra el estado del sistema en modo normal sin ejecutar el detector de somnolencia. Como se puede apreciar, el uso del CPU es de apenas del 2%, la memoria RAM tiene 115Mb en uso y la temperatura promedio a la que estaba trabajando el sistema fue de 51.5°C.



Fig. 41 Estado de CPU, RAM y temperatura sin ejecución del detector de somnolencia

Fuente: Propia

La Fig. 42 muestra el estado de los recursos, luego de haber ejecutado el detector de somnolencia durante 10 minutos. Para ese instante, el CPU estaba siendo usado al 55%, la memoria RAM subió a 220 Mb de uso y la temperatura aumentó a 68.8°C

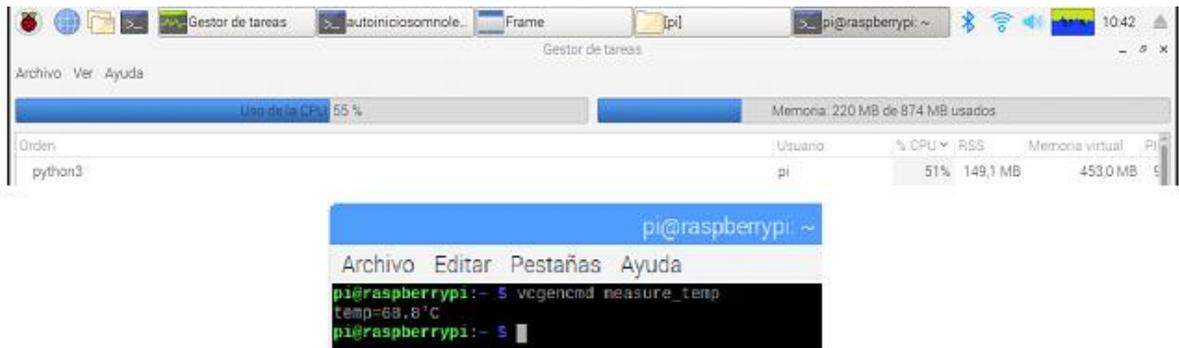


Fig. 42 Estado de CPU, RAM y temperatura luego de 10 minutos de ejecutar el detector de somnolencia

Fuente: Propia

La Fig. 43 muestra el estado de los recursos, luego de haber ejecutado el detector de somnolencia durante 30 minutos. El uso del CPU indica 55%, la memoria RAM aumentó en una unidad a 221 Mb de uso y la temperatura aumentó a 72.2°C.

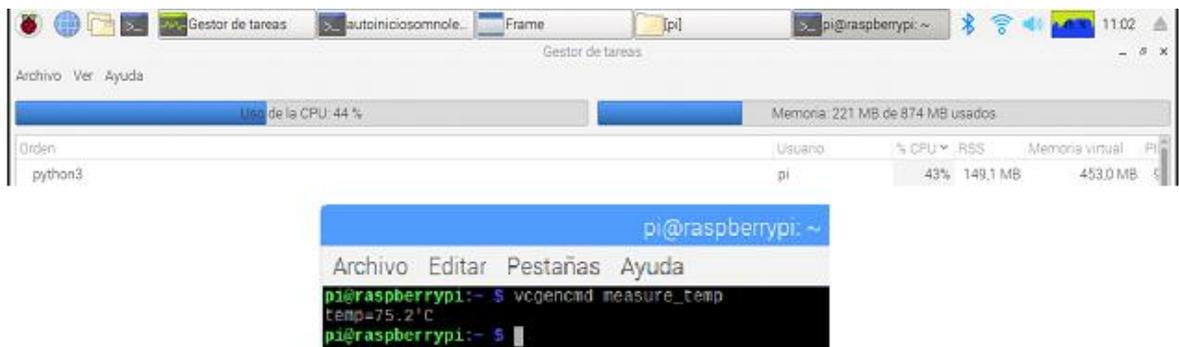


Fig. 43 Estado de CPU, RAM y temperatura luego de 30 minutos de ejecutar el detector de somnolencia

Fuente: Propia

La cuarta y última prueba mostrada en la Fig. 44 se realizó a los 40 minutos de tener funcionando el detector de somnolencia. En este punto el procesador muestra un 60% de uso, la memoria RAM mantiene 221 Mb de uso y la temperatura del dispositivo siguió en aumento hasta 75.8°C en este punto.

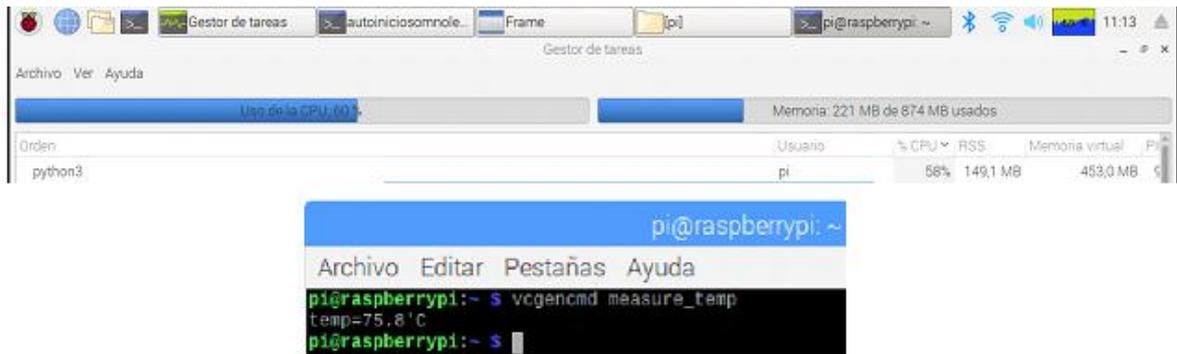


Fig. 44 Estado de CPU, RAM y temperatura luego de 40 minutos de ejecutar el detector de somnolencia
Fuente: Propia

La TABLA 2.21 muestra un resumen de los datos recolectados en estas pruebas:

TABLA 2.21
RESUMEN DE TEMPERATURA Y CONSUMO DE RECURSOS DEL SISTEMA

Tiempo	CPU	RAM	TEMPERATURA
0 min	2%	13.2%	51.5 °C
10 min	55%	25.1%	68.8 °C
30 min	44%	25.3%	75.2 °C
40 min	60%	25.3%	75.8 °C

Fuente: Propia

Los resultados obtenidos, sugirieron que el sistema debe ser provisto de un sistema de **ventilación** que permita mantener la temperatura del equipo en un valor óptimo, ya que, en tiempos prolongados de uso, el sistema puede sufrir sobrecalentamientos y en el peor de los casos puede terminarse averiando el dispositivo. Mantener una temperatura ideal, garantiza también el buen desempeño del procesador, pues a mayores temperaturas, el equipo baja su rendimiento.

2.4.3. Pruebas de consumo de corriente

Es muy importante determinar la carga de corriente que necesita todo el sistema para su correcto funcionamiento. Mientras más dispositivos se vayan agregando, más corriente necesita la placa. También es importante tomar en cuenta que los pines GPIO pueden suministrar hasta 16mA cada uno de forma segura y entre todos hasta 50mA, pues si se sobrepasa en gran magnitud ese rango de valores, puede provocar un daño inminente a la placa y sencillamente ya no funcionará.

A continuación, en la TABLA 2.22, se detalla el consumo promedio de cada dispositivo utilizado en este prototipo y el consumo total aproximado de todo el sistema cuando está en pleno funcionamiento.

TABLA 2.22
VALORES DE CONSUMO DE CADA DISPOSITIVO

	Voltaje	Amperaje
Raspberry Pi 3 B+	5V	800mA (4.0 W)
Pi Cámara	5V	250mA (1.25 W)
Led's infrarrojo	3.3V	1000mA (3W)
Led verde	1.8V	15mA (0.02W)
Led azul	3V	20mA (0.06W)
Led rojo	1.5V	15mA (0.02W)
Buzzer	5V	30mA (0.15W)
	TOTAL: 2130mA (10.65W) Aprox.	

Fuente: Propia

Como se puede observar, el valor total de consumo de corriente de todo el sistema es bastante bajo en relación a la cantidad de tareas que debe cumplir. La arquitectura y tecnología de este dispositivo embebido, permite que pueda entregar todo su potencial con un mínimo de consumo, siendo prácticamente imperceptible para la capacidad del sistema eléctrico que tiene un vehículo.

CAPÍTULO 3

Validación de resultados

3.1. Ubicación de la cámara

La ubicación de la cámara es un factor clave que influye de forma directa en la precisión de la detección del rostro y los ojos del conductor, por consecuente determina el éxito que puede tener el sistema en la detección de somnolencia.

Luego de realizar varias pruebas, se determinó que la posición más adecuada para obtener una buena toma de imágenes del rostro del conductor, era ubicarla aproximadamente a unos diez grados por debajo del nivel de los ojos y a unos 50cm de frente al rostro, evitando la vista desde arriba o de lado. Esta posición permite seguir detectando el rostro aún si hay un leve cabeceo hacia abajo. Como se puede observar en la Fig. 45, la cámara fue colocada en un soporte con ventosa adherida al parabrisas.



Fig. 45. Ubicación de la cámara dentro de la cabina del auto
Fuente: Propia

En las pruebas de conducción, se consultó a los voluntarios si percibían alguna molestia visual por la posición de la cámara, pero todos se sintieron cómodos para seguir conduciendo.

La Fig. 46 muestra las pruebas realizadas al Ing. Erick Herrera en un ambiente real de conducción, en donde luego de la configuración automática, el sistema fijó un umbral dinámica para la alerta de somnolencia, tal como se explicó en el capítulo anterior.



Fig. 46. Pruebas en ambiente real de conducción - Ing. Erick Herrera

Fuente: Propia

3.2. Alimentación del sistema

Para la alimentación del sistema, como se había mencionado en la sección 2.2.2, se utilizó un cargador USB que va conectado al encendedor del vehículo.

El cargador inicialmente adquirido (Véase TABLA 2.15), tuvo dificultades al energizar el prototipo con todos sus componentes, pues el consumo de corriente de los dispositivos (Véase TABLA 2.22), superaba la corriente entregada por este cargador, por lo que se optó por adquirir un cargador de mayor amperaje (Fig. 47). Este factor influye en el rendimiento del dispositivo embebido, pues se pudo notar que, al no suministrar la cantidad suficiente de corriente, el sistema trabaja muy lento, y su tiempo de reacción aumentó considerablemente.



Fig. 47. Cargador de coche de 5VDC 4Amp, dual micro USB

Fuente: (Amazon, 2019)

Así mismo, se hizo uso de una mini pantalla para poder manipular el sistema operativo desde el vehículo y realizar las grabaciones de video para las pruebas de conducción y su posterior análisis.

La Fig. 48, muestra los elementos utilizados para las pruebas realizadas en el vehículo.



Fig. 48. Elementos usados en las pruebas de conducción
Fuente: Propia

3.3. Consideraciones importantes

Las pruebas fueron realizadas con la participación de algunos voluntarios (se especifica más adelante), los cuales, luego de tomar las precauciones necesarias, simulaban quedarse dormidos. Todos los acontecimientos fueron registrados en grabaciones de vídeo desde el inicio de la configuración del sistema hasta el fin de las pruebas.

En esta sección, se dará a conocer los inconvenientes suscitados en la fase de pruebas en un ambiente real de conducción, puesto que la luz del día y las características de la cámara utilizada, influyeron en la precisión del dispositivo.

3.3.1. Iluminación

En las pruebas realizadas en ambientes controlados, es decir, con iluminación artificial y sin mayores obstáculos o movimientos, el dispositivo presentó una buena precisión en su funcionamiento, sin embargo, ya en ambientes reales de conducción donde la iluminación es variable y hay mayor movimiento del vehículo, el dispositivo presentó falsos positivos y falsos negativos que eran mayores o menores dependiendo de la persona. Es así como se pudo deducir que había algo en las imágenes de entrada, que no permitía que el algoritmo haga un monitoreo preciso de los ojos del conductor.

De esta manera, se realizó un análisis de las imágenes captadas por la cámara en ambientes iluminados por la luz natural del sol para verificar que es lo que estaba pasando.

La Fig. 49 muestra el efecto contraluz que provoca el resplandor de la luz del sol que ingresa por la parte posterior del conductor, como se puede observar, esto deriva en que

algunas veces el sistema no pueda detectar el rostro, pues resalta la silueta de la persona y no se puede distinguir de forma clara los hitos faciales que es lo que interesa.



Fig. 49. Influencia de la contraluz sobre la imagen de entrada (El sistema no detecta los hitos faciales)

Fuente: Propia

Una de las áreas que comúnmente se descuida en aplicaciones de visión artificial, es precisamente la iluminación, y en ocasiones se busca implementar software y hardware complejo y costoso para solucionar estos problemas de visión, cuando se podría resolver poniendo atención simplemente a la iluminación y la óptica de la cámara (Novini, 1993).

Para este caso, la iluminación es un factor que no es muy controlable en el vehículo, puesto que está sujeto a las variaciones naturales de la luz del sol, o también puede verse influenciado por el encandilamiento provocado en la noche por las luces de los vehículos que vienen detrás. Sin embargo, una solución viable para evitar la saturación que provoca el efecto contraluz, podría ser la instalación de polarizados que minimicen la entrada de la luz natural o artificial. Esto aplicaría únicamente en vehículos que tengan vidrio en la parte posterior del conductor por donde puede ingresar la luz del sol. Si se tratase de un bus interprovincial, casi en la mayoría de casos, toda la cabina está aislada y el único lugar por donde ingresa la luz del sol es por el parabrisas, lo cual se torna en un ambiente propicio para una buena toma de imágenes del rostro del conductor.

3.3.2. Óptica de la cámara

En lo que sí se puede trabajar y mejorar la calidad de la imagen, es en la óptica. Para esto, se adquirió una cámara web USB (Fig. 50) que tenía un par de características particulares, pues podía hacer un balance de blancos automático y corrección automática de color, mostrando así una imagen más nítida. En segundo lugar, ésta cámara tenía una distancia focal de aproximadamente 85mm que es más larga que la distancia focal de la cámara PI de 36mm.



Fig. 50. Cámara web USB

Fuente: <http://www.chinazrh.com/>

Los expertos en cámaras de la marca NIKON, dan a conocer que mientras más larga sea la distancia focal de una cámara (Fig. 51), el ángulo de visión es más estrecho y tiene un mayor acercamiento y viceversa (Nikon, 2019), con lo que los objetos de interés obtienen una mejor resolución espacial para su posterior procesamiento.



Fig. 51. Diferencia de la distancia focal

Fuente: (Nikon, 2019)

En base a esto, se pudo determinar que, para este aplicativo, era mejor disponer de una cámara con una distancia focal más larga, pues la única región de interés es el rostro del conductor y es ahí donde se requiere tener un mayor acercamiento para obtener una imagen más detallada con la cual el sistema pueda monitorear al conductor sin mucha dificultad.

La Fig. 52 muestra la diferencia del tamaño y calidad imagen de entrada entre la cámara Pi utilizada inicialmente y la cámara web USB implementada posteriormente.



Fig. 52. Cámara Pi (izquierda), cámara web USB (derecha)

Fuente: Propia

Como se puede apreciar, una cámara con un ángulo de visión más estrecho, permite hacer un mejor acercamiento a la zona de interés y descarta el resto de áreas que no tienen relevancia, lo cual para esta aplicación resultó ser propicio, pues lo único que interesa es el rostro del conductor y nada más. Por consiguiente, la cámara web USB fue la mejor opción.

3.4. Matriz de confusión

Las pruebas de funcionamiento fueron efectuadas en dos tipos de ambientes para comparar la influencia de las condiciones del medio en la precisión del prototipo. Es así como se pidió la colaboración de 10 voluntarios para realizar las pruebas en un ambiente controlado de iluminación y otros 10 voluntarios para las pruebas en un ambiente real de conducción. Estos eventos fueron efectuados en diferentes horarios del día y en personas de distinta edad, género y raza, esto con el fin de evaluar el rendimiento del prototipo ante las distintas condiciones que se pueden presentar en el medio real de conducción.

Para determinar la precisión del sistema embebido, se aplicó la matriz de confusión, la cual es una herramienta usada en el campo de la inteligencia artificial para visualizar el desempeño de un algoritmo en aprendizaje supervisado. Las columnas representan las predicciones del algoritmo, mientras que las filas representan los valores reales de lo que se está evaluando (Wikipedia, 2019).

La TABLA 3.1 muestra la estructuración de la matriz de confusión para la respectiva evaluación del rendimiento de este sistema detector de somnolencia, donde:

TP: True-Positive es la cantidad de casos en los que el sistema predijo *Somnolencia* y efectivamente si hubo somnolencia.

FN: False-Negative son los casos en donde el sistema predijo *No Somnolencia*, cuando en realidad si hubo somnolencia.

FP: False-Positive son los casos en donde el sistema predijo *Somnolencia*, pero en realidad no hubo somnolencia.

TN: True-Negative número de casos en los que el sistema predijo *No Somnolencia* y efectivamente no había somnolencia.

TABLA 3.1.
ESTRUCTURA DE LA MATRIZ DE CONFUSIÓN

Matriz de confusión		PREDICHO	
		Somnolencia	No Somnolencia
REAL	Somnolencia	TP	FN
	No Somnolencia	FP	TN

Fuente: Propia

Cada una de las pruebas realizadas tuvo una duración aproximada de 20 minutos, las cuales fueron registradas en vídeo para poder hacer su posterior análisis.

Por medio del muestreo aleatorio simple aplicado en la sección 2.3.14, se procedió a calcular la cantidad de fotogramas necesarios para analizar el rendimiento del sistema. El nivel de confianza con el que se trabajó para este caso es del 95% ($Z=1.96$) y con un error máximo del 5% (0.05). Reemplazando estos valores en la fórmula y calculando, se obtuvo:

$$n = \frac{(1.96)^2 x (0.5) x (0.5)}{(0.05)^2}$$

$$n = 384.16$$

En base a este resultado, se procedió a redondear y hacer un análisis de 400 fotogramas aleatorios en cada vídeo, de las cuales 200 debían pertenecer al grupo de *Somnolencia* y 200 al grupo de *No Somnolencia* con el fin de mantener balanceados los grupos y no provocar sesgos en los resultados.

A continuación, se presenta la matriz de confusión de cada una de las personas con las cuales se realizaron las pruebas, tanto en ambiente controlado como en ambiente real de conducción.

3.4.1. Ambiente controlado

La TABLA 3.2 muestra los resultados de las pruebas para el Ing. Marco PUSDÁ en ambiente controlado.

TABLA 3.2.
MATRIZ DE CONFUSIÓN ING. MARCO PUSDÁ

	Marco PUSDÁ		PREDICTED VALUES	
	EAR: 0.28			
	Umbral: 0.20		Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia	170	30	
	No somnolencia	10	190	

Fuente: Propia

La TABLA 3.3 muestra los resultados de las pruebas para Brayan Valencia en ambiente controlado.

TABLA 3.3.
MATRIZ DE CONFUSIÓN BRAYAN VALENCIA

	Brayan Valencia		PREDICTED VALUES	
	EAR: 0.32			
	Umbral: 0.22		Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia	200	0	
	No somnolencia	10	190	

Fuente: Propia

La TABLA 3.4 muestra los resultados de las pruebas para Karla Herrera en ambiente controlado.

TABLA 3.4.
MATRIZ DE CONFUSIÓN KARLA HERRERA

	Karla Herrera		PREDICTED VALUES	
	EAR: 0.26			
	Umbral: 0.18		Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia	170	30	
	No somnolencia	10	190	

Fuente: Propia

La TABLA 3.5 muestra los resultados de las pruebas para Vanessa Cadena en ambiente controlado y haciendo uso de lentes.

TABLA 3.5.
MATRIZ DE CONFUSIÓN VANESSA CADENA

	Vanessa Cadena EAR: 0.26 Umbral: 0.18 Lentes: Sí	PREDICTED VALUES	
		Somnolencia	No somnolencia
	ACTUAL VALUES Somnolencia	160	40
	No somnolencia	40	160

Fuente: Propia

La TABLA 3.6 muestra los resultados de las pruebas para Josselin Enríquez en ambiente controlado.

TABLA 3.6.
MATRIZ DE CONFUSIÓN JOSSELIN ENRÍQUEZ

	Josselin Enríquez EAR: 0.30 Umbral: 0.21	PREDICTED VALUES	
		Somnolencia	No somnolencia
	ACTUAL VALUES Somnolencia	180	20
	No somnolencia	20	180

Fuente: Propia

La TABLA 3.7 muestra los resultados de las pruebas para Jostin Roldán en ambiente controlado y haciendo uso de lentes.

TABLA 3.7.
MATRIZ DE CONFUSIÓN JOSTIN ROLDÁN

	Jostin Roldán EAR: 0.29 Umbral: 0.20 Lentes: Sí	PREDICTED VALUES	
		Somnolencia	No somnolencia
	ACTUAL VALUES Somnolencia	170	30
	No somnolencia	10	190

Fuente: Propia

La TABLA 3.8 muestra los resultados de las pruebas para Daniela Puentestar en ambiente controlado y haciendo uso de lentes.

TABLA 3.8.
MATRIZ DE CONFUSIÓN DANIELA PUENTESTAR

	Daniela Puentestar EAR: 0.27 Umbral: 0.19 Lentes: Sí	PREDICTED VALUES	
		Somnolencia	No somnolencia
	ACTUAL VALUES Somnolencia	140	60
	No somnolencia	50	150

Fuente: Propia

La TABLA 3.9 muestra los resultados de las pruebas para Katherine Rodríguez en ambiente controlado.

TABLA 3.9.
MATRIZ DE CONFUSIÓN KATHERINE RODRÍGUEZ

	Katherine Rodríguez EAR: 0.26 Umbral: 0.18	PREDICTED VALUES	
		Somnolencia	No somnolencia
	ACTUAL VALUES Somnolencia	190	10
	No somnolencia	0	200

Fuente: Propia

La TABLA 3.10 muestra los resultados de las pruebas para Charlie Darmi en ambiente controlado.

TABLA 3.10.
MATRIZ DE CONFUSIÓN CHARLIE DARMI

	Charlie Darmi EAR: 0.30 Umbral: 0.21	PREDICTED VALUES	
		Somnolencia	No somnolencia
	ACTUAL VALUES Somnolencia	150	50
	No somnolencia	10	190

Fuente: Propia

La TABLA 3.11 muestra los resultados de las pruebas para Franklin Vallejo en ambiente controlado.

TABLA 3.11.
MATRIZ DE CONFUSIÓN FRANKLIN VALLEJO

	Franklin Vallejo		PREDICTED VALUES	
	EAR: 0.34		Somnolencia	No somnolencia
	Umbral: 0.24			
ACTUAL VALUES	Somnolencia	160	40	
	No somnolencia	0	200	

3.4.2. Ambiente real de conducción

La TABLA 3.12 muestra los resultados de las pruebas para Josué Alba en ambiente real de conducción.

TABLA 3.12.
MATRIZ DE CONFUSIÓN JOSUÉ ALBA

	Josué Alba		PREDICTED VALUES	
	EAR: 0.29		Somnolencia	No somnolencia
	Umbral: 0.20			
ACTUAL VALUES	Somnolencia	160	40	
	No somnolencia	0	200	

Fuente: Propia

La TABLA 3.13 muestra los resultados de las pruebas para Jessica Cevallos en ambiente real de conducción.

TABLA 3.13.
MATRIZ DE CONFUSIÓN JESSICA CEVALLOS

	Jessica Cevallos		PREDICTED VALUES	
	EAR: 0.29		Somnolencia	No somnolencia
	Umbral: 0.20			
ACTUAL VALUES	Somnolencia	160	40	
	No somnolencia	50	150	

Fuente: Propia

La TABLA 3.14 muestra los resultados de las pruebas para Armando Guamán en ambiente real de conducción, haciendo uso de una gorra.

TABLA 3.14.
MATRIZ DE CONFUSIÓN ARMANDO GUAMÁN

	Armando Guamán		PREDICTED VALUES	
	EAR: 0.26			
	Umbral: 0.18			
	Gorra: Sí		Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia		150	50
	No somnolencia		60	140

Fuente: Propia

La TABLA 3.15 muestra los resultados de las pruebas para Richard Guerrero en ambiente real de conducción.

TABLA 3.15.
MATRIZ DE CONFUSIÓN RICHARD GUERRERO

	Richard Guerrero		PREDICTED VALUES	
	EAR: 0.26			
	Umbral: 0.18			
			Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia		180	20
	No somnolencia		20	180

Fuente: Propia

La TABLA 3.16 muestra los resultados de las pruebas para Daniel Guerrero en ambiente real de conducción.

TABLA 3.16.
MATRIZ DE CONFUSIÓN DANIEL GUERRERO

	Daniel Guerrero		PREDICTED VALUES	
	EAR: 0.26			
	Umbral: 0.18			
			Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia		170	30
	No somnolencia		30	170

Fuente: Propia

La TABLA 3.17 muestra los resultados de las pruebas para Diego Montenegro en ambiente real de conducción.

TABLA 3.17.
MATRIZ DE CONFUSIÓN DIEGO MONTENEGRO

	Diego Montenegro		PREDICTED VALUES	
	EAR: 0.24			
	Umbral: 0.17		Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia	190	10	
	No somnolencia	50	150	

Fuente: Propia

La TABLA 3.18 muestra los resultados de las pruebas para Francisco Vinueza en ambiente real de conducción.

TABLA 3.18.
MATRIZ DE CONFUSIÓN FRANCISCO VINUEZA

	Francisco Vinueza		PREDICTED VALUES	
	EAR: 0.29			
	Umbral: 0.20		Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia	190	10	
	No somnolencia	30	170	

Fuente: Propia

La TABLA 3.19 muestra los resultados de las pruebas para Franklin Vallejo en ambiente real de conducción.

TABLA 3.19.
MATRIZ DE CONFUSIÓN FRANKLIN VALLEJO

	Franklin Vallejo		PREDICTED VALUES	
	EAR: 0.27			
	Umbral: 0.19		Somnolencia	No somnolencia
ACTUAL VALUES	Somnolencia	190	10	
	No somnolencia	20	180	

Fuente: Propia

La TABLA 3.20 muestra los resultados de las pruebas para Fernando Chafuelán en ambiente real de conducción, haciendo uso de una gorra.

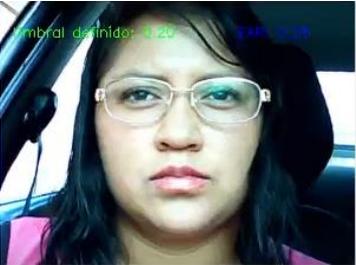
TABLA 3.20.
MATRIZ DE CONFUSIÓN FERNANDO CHAFUELÁN

	Fernando Chafuelán		PREDICTED VALUES	
	EAR: 0.33		Somnolencia	No somnolencia
	Umbral: 0.23			
Gorra: Sí				
ACTUAL VALUES	Somnolencia	200	0	
	No somnolencia	30	170	

Fuente: Propia

La TABLA 3.21 muestra los resultados de las pruebas para Mishel Cevallos en ambiente real de conducción, haciendo uso de lentes.

TABLA 3.21.
MATRIZ DE CONFUSIÓN MISHEL CEVALLOS

	Mishel Cevallos		PREDICTED VALUES	
	EAR: 0.29		Somnolencia	No somnolencia
	Umbral: 0.20			
Lentes: Sí				
ACTUAL VALUES	Somnolencia	200	10	
	No somnolencia	10	200	

Fuente: Propia

3.5. Resultados

A partir de las matrices de confusión, podemos calcular una serie de métricas muy útiles. Según (Izco, 2018), las principales son:

- Exactitud o “Accuracy”, indica la proporción de predicciones correctas. Determinada por la fórmula:

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

- Precisión, indica la proporción de predicciones positivas correctas. Determinada por la fórmula:

$$precisión = \frac{TP}{TP + FP}$$

- Sensibilidad o “Recall”, indica la proporción de positivos reales predichos correctamente. Determinada por la fórmula:

$$recall = \frac{TP}{TP + FN}$$

- F1-score, indica la precisión de las pruebas, considerando a la precisión y al Recall. Su fórmula es:

$$F1 - score = 2 * \frac{precisión * recall}{precisión + recall}$$

TABLA 3.22.
RESULTADO DE LAS PRUEBAS EN AMBIENTE CONTROLADO Y AMBIENTE REAL DE CONDUCCIÓN

Pruebas en Ambiente Controlado											
Voluntario	Umbral	Lente	TP	FN	FP	TN	Tot.	Exact.	Recall	Precisión	F1
Marco PUSDÁ	0,20	N	170	30	10	190	400	90,00%	85,00%	85,00%	85,00%
Brayan Valencia	0,22	N	200	0	10	190	400	97,50%	100,00%	100,00%	100,00%
Karla Herrera	0,18	N	170	30	10	190	400	90,00%	85,00%	85,00%	85,00%
Vanessa Cadena	0,18	S	160	40	40	160	400	80,00%	80,00%	80,00%	80,00%
Josselin Enriquez	0,21	N	180	20	20	180	400	90,00%	90,00%	90,00%	90,00%
Jostin Roldan	0,20	S	170	30	10	190	400	90,00%	85,00%	85,00%	85,00%
Daniela Puentestar	0,19	S	140	60	50	150	400	72,50%	70,00%	70,00%	70,00%
Katherine Rodríguez	0,18	N	190	10	0	200	400	97,50%	95,00%	95,00%	95,00%
Charlie Darmi	0,21	N	150	50	10	190	400	85,00%	75,00%	75,00%	75,00%
Franklin Vallejo	0,24	N	160	40	0	200	400	90,00%	80,00%	80,00%	80,00%
TOTAL								88,25%	84,50%	85%	84,50%
Pruebas en Ambiente No Controlado - Conducción real											
Voluntario	Umbral	Lente	TP	FN	FP	TN	Tot.	Exact.	Recall	Precisión	F1
Josué Alba	0,20	N	160	40	0	200	400	90,00%	80,00%	80,00%	80,00%
Jessica Cevallos	0,20	N	160	40	50	150	400	77,50%	80,00%	80,00%	80,00%
Armando Guamán	0,18	N	150	50	60	140	400	72,50%	75,00%	75,00%	75,00%
Richard Guerrero	0,18	N	180	20	20	180	400	90,00%	90,00%	90,00%	90,00%
Daniel Guerrero	0,18	N	170	30	30	170	400	85,00%	85,00%	85,00%	85,00%
Diego Montenegro	0,17	N	190	10	50	150	400	85,00%	95,00%	95,00%	95,00%
Francisco Vinuesa	0,20	N	190	10	30	170	400	90,00%	95,00%	95,00%	95,00%
Franklin Vallejo	0,19	N	190	10	20	180	400	92,50%	95,00%	95,00%	95,00%
Fernando Chafuelán	0,23	N	200	0	30	170	400	92,50%	100,00%	100,00%	100,00%
Mishel Cevallos	0,20	S	190	10	10	190	400	95,00%	95,00%	95,00%	95,00%
TOTAL								87,00%	89,00%	89,00%	89,00%

Fuente: Propia

3.5.1. Análisis de resultados

Los resultados muestran la divergencia de exactitud y precisión del dispositivo al ser utilizado en ambientes controlados y en ambientes reales de conducción. La inestabilidad de la iluminación y el movimiento al cual están expuestos los conductores, incidieron un poco en la precisión de la detección de sus rostros y ojos, sin embargo, los resultados obtenidos son buenos para este fin.

Aún en ambientes controlados, el sistema tuvo inconvenientes en algunos casos. Para ser más específicos, el sistema tiene dificultad al detectar los ojos cuando una persona está usando lentes, esto debido al brillo generado por los mismos lo cual impide distinguir claramente los ojos. El inconveniente con los lentes se podría mitigar si los conductores usaran lentes antirreflejos.

Los lentes antirreflejos tienen un recubrimiento que permite eliminar el reflejo de las superficies anterior y posterior de los lentes, optimiza la precisión visual y los hace casi invisibles, esto ayuda a tener un mejor contacto visual con los demás. Incluso podrían usarlas como gafas, ya que elimina el resplandor del sol (Heiting, 2019).

También se pudo notar que al tener cabello cubriendo parte del rostro, dificulta al sistema poder detectar el rostro y los hitos faciales. Es necesario que el conductor mantenga su rostro libre de obstáculos. El uso de gorras, no afecta la detección del rostro, siempre y cuando se encuentre como máximo hasta el nivel de la frente.

Se pudo notar también que hay una ligera disminución de la precisión del algoritmo al detectar los hitos faciales de personas con piel oscura. Mientras más clara sea la piel, el sistema responde de forma más precisa.

Una de las razones para que tenga ciertos desaciertos al detectar distintos tonos de piel, es que el clasificador *Haar Cascade* de OpenCV de por sí no incorpora un analizador que permita distinguir si se trata del color de piel humana o no, por lo cual hay ocasiones en las que no detecta el rostro o, al contrario, detecta rostros donde no los hay. Una solución viable para este pequeño problema sería implementar un método que permita analizar el color y distinguir si se trata de piel humana o no, con el fin de reducir las detecciones erróneas (MAH, 2016).

Conforme se iban realizando las pruebas, cada vez era más evidente que la inestabilidad de iluminación también provoca que la imagen no tenga un contraste adecuado. Para ello, con el afán de mejorar el contraste de las imágenes de entrada y por ende la precisión del

algoritmo, se procedió a indagar la manera de estabilizar la imagen y lograr una mejoría significativa.

Es así como se puso a prueba al algoritmo, para que intente detectar el rostro dentro de una imagen con mucha saturación. La Fig. 53 muestra la imagen de entrada normal sin ningún tipo de tratamiento. Como se puede apreciar, el sistema no pudo detectar ningún rostro.



Fig. 53. Imagen de entrada sin ningún tratamiento – El sistema no detectó ningún rostro

Fuente: Propia

En la Fig. 54 se puede apreciar la misma imagen convertida a escala de grises, que es la imagen que necesita recibir el algoritmo para ser procesado. Como se puede apreciar, el sistema tampoco pudo detectar ningún rostro.



Fig. 54. Imagen de entrada convertida a escala de grises - El sistema no detectó ningún rostro

Fuente: Propia

La Fig. 55 representa a la misma imagen sometida a un proceso previo de ecualización de histogramas, que es una función disponible en OpenCV. Como se puede observar, ciertas zonas ganaron nitidez, sin embargo, el sistema no encontró ningún rostro. Esto sucede porque este método no funciona bien donde hay grandes variaciones de intensidad, es decir, hay la presencia de píxeles muy brillantes y oscuros (Mordvintsev & Abid, 2013).



Fig. 55. Imagen de entrada sometida a ecualización de histograma - El sistema no detectó ningún rostro
Fuente: Propia

Finalmente, como se puede observar en la Fig. 56 , luego de someter a la misma imagen a un proceso denominado Ecualización de histograma adaptativo (CLAHE), disponible en OpenCV, el resultado del tratamiento de la imagen es mucho mejor que los anteriores métodos, mostrando un contraste equilibrado y permitiendo finalmente al sistema poder detectar el rostro presente en la imagen. Este método básicamente aplica la limitación del contraste, distribuyendo uniformemente los píxeles con un contraste alto a otras regiones con contraste bajo, antes de aplicar la ecualización del histograma (Mordvintsev & Abid, 2013).

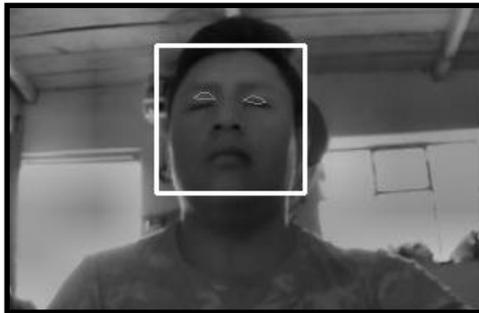


Fig. 56. Imagen de entrada sometida a ecualización de histograma adaptativo - El sistema detectó un rostro
Fuente: Propia

Este procedimiento mejoró considerablemente la precisión del sistema, pues equilibró el contraste de la imagen y permitió distinguir de mejor manera las características faciales de la persona.

Conclusiones

- La importancia de la visión artificial, ha motivado a que muchos investigadores experimenten y compartan sus experiencias en esta disciplina, por lo cual se ha podido recabar valiosa información que ha permitido desarrollar un sistema de detección de somnolencia basado en el cálculo de índice EAR que logra trabajar sin dificultad en tiempo real en conducción diurna, tomando en cuenta las limitaciones computacionales de la Raspberry Pi 3 B+.
- Luego de las pruebas y ajustes necesarios, se analizó el rendimiento del prototipo en ambiente controlado y en ambiente real de conducción aplicando la matriz de confusión (Sección 3.4), teniendo como resultado una exactitud del 88.25% y una precisión del 85% para ambientes controlados y una exactitud del 87% y una precisión del 89% para ambientes reales de conducción (Sección 3.5). Estos resultados, satisfacen los requisitos de valores de precisión comúnmente aceptados que deben ser iguales o superiores al 85% (Foody, 2002).
- El prototipo desarrollado, integra como valor agregado una fase de configuración que se ejecuta al iniciar el dispositivo, su objetivo es ajustar el umbral dinámico según el aspecto ocular de cada conductor (Sección 2.3.9). Esta función, hasta donde se conoce, no había sido aplicada con anterioridad en trabajos similares, y permitió mejorar la precisión del dispositivo notablemente. Este dispositivo, inicia automáticamente su funcionamiento al momento de suministrarle energía y se han agregado diodos led que permiten verificar el funcionamiento del sistema y la detección correcta de los ojos, lo cual permite ajustar la posición de la cámara. También dispone de un pulsador que permite iniciar o apagar el sistema de manera segura (Sección 2.3.17). Estos elementos y mejoras permitieron obtener un dispositivo prototipo funcional y ligero, capaz de detectar y alertar la presencia de somnolencia (incluso con el uso de gorra y lentes) y trabajar tanto en ambientes diurnos como nocturnos (Sección 2.3.15).

Recomendaciones

La visión artificial ha tenido una gran acogida a nivel mundial y sus aplicaciones son muy importantes e innumerables para resolver problemas del mundo real, es así que se motiva al lector a indagar sobre esta disciplina y crear soluciones innovadoras a problemas presentes en la sociedad.

El prototipo desarrollado ha demostrado ser bastante funcional, obteniendo buenos resultados, sin embargo, aún hay detalles que podrían mejorar aún más la funcionalidad y precisión de este dispositivo, los cuales puede ser abordados como trabajos a futuro:

- La implementación de un módulo que permita empezar a monitorear al conductor luego de que el vehículo supere una cierta velocidad (> 20 km/h por ejemplo), puesto que no tendría sentido que siga monitoreando y dando alertas cuando el vehículo esté completamente detenido, pues esto se tornaría innecesario y molesto.
- Un estudio más profundizado sobre el tratamiento de imágenes por medio de la ecualización de histograma adaptativo (CLAHE) y la comparación de los resultados y mejoras que esto representa en el aumento de la calidad de las imágenes, sobre todo cuando la iluminación del ambiente es muy variable como el caso del ambiente exterior en donde múltiples factores pueden provocar que la imagen se encuentre saturada e impida distinguir con claridad las características faciales del conductor.
- Uno de los inconvenientes encontrados en esta investigación, es la manera de superar la oclusión de los ojos causada por el uso de las gafas de sol. Por ende, se requiere un estudio específicamente para esta problemática, en donde se exponga las posibles soluciones haciendo uso de elementos electrónicos y/o software, así como el uso de la banda infrarroja (NIR) que permitan superar esta limitante.

Glosario De Términos

EAR: Representa las siglas en inglés de *Eye Aspect Ratio*. Es una cantidad escalar que caracteriza la apertura del ojo en cada frame.

Sistema Embebido: Es un dispositivo de computación que está diseñado para realizar una o pocas funciones específicas. Se trata de una placa electrónica reducida en la cual se encuentran incluidos la mayoría de los componentes como; tarjeta de vídeo, audio, wifi, bluetooth, usb, etc. Poseen un procesador y memoria pequeños.

Algoritmo: Es un proceso o conjunto de procesos a seguir para resolver problemas por medio de una computadora

Visión artificial o visión por computador: Campo científico que busca que las computadoras tengan una comprensión avanzada del mundo real a través la interpretación de imágenes o vídeos y complejos algoritmos.

Histograma: Representación gráfica de una variable en forma de barras. Permite tener un panorama de la distribución de una muestra respecto a una característica cuantitativa y continua.

Fotograma: Cada una de las imágenes que en conjunto forman una secuencia de vídeo. Cuando esta secuencia es visualizada a una determinada frecuencia de imágenes por segundo, se genera la sensación de movimiento en el espectador.

CLAHE: Siglas en inglés de “Contrast Limited Adaptive Histogram Equalization”. Procedimiento de tratamiento de imágenes que permite realizar a la imagen una Ecuación de Histograma Adaptativo Limitado por Contraste. Equilibra la saturación de ciertos sectores de la imagen con el fin de obtener una imagen más clara en la que se puedan distinguir todas o la mayoría de características.

Bibliografía

- Abellán, M. Á. (2018). Led intermitente - Control de GPIO con Python en Raspberry Pi. Retrieved July 22, 2019, from <https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/intermitente>
- Abyani, M., Asgarian, B., & Zarrin, M. (2019). Sample geometric mean versus sample median in closed form framework of seismic reliability evaluation: a case study comparison. *Earthquake Engineering and Engineering Vibration*, 18(1), 187–201. <https://doi.org/10.1007/s11803-019-0498-5>
- ACADEMIC. (2010). Distancia Euclidiana. Retrieved from <https://esacademic.com/dic.nsf/eswiki/356722>
- Adrian Rosebrock. (2017). Puntos de referencia faciales con dlib, OpenCV y Python - PyImageSearch. Retrieved April 30, 2019, from <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>
- Amazon. (2019). Amazon.com: PWR + Cargador de Coche para Tablet ASUS Transformer Book T100, t100ta-b1-gr, T100TA-C1-GR 10.1 inch Convertible 2-en-1 Touchscreen Laptop Power Supply Cord DC Adapter: Computers & Accessories. Retrieved December 12, 2019, from https://www.amazon.com/-/es/Transformer-t100ta-b1-gr-T100TA-C1-GR-Convertible-2-en-1-Touchscreen/dp/B00JK15VA4/ref=sr_1_4?__mk_es_US=ÅMÅŽÕÑ&keywords=usb+car+charger+double+pwr&qid=1576176528&sr=8-4
- Angamarca, V. del R. (2018). *Implementación de un sistema de gestión para los procesos de capacitación y educación cristiana de la parroquia santuario “Nuestra señora del cisne de la provincia de Imbabura.”* Universidad Técnica del Norte. Retrieved from <http://repositorio.utn.edu.ec/bitstream/123456789/8038/1/04 ISC 456 TESIS.pdf>
- Ardila, W., Osorio, J. A. C., & Vargas, J. A. M. (2011). Scientia et technica. *Scientia et Technica*, 1(47), 207–212. Retrieved from <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/533/263>
- Arévalo, V. M., González, J., & Ambrosio, G. (2002). *La librería de visión artificial opencv aplicación a la docencia e investigación 1*. Retrieved from www.sourceforge.net
- Batanero, C. (2000). Significado y comprensión de las medidas de posición central. *Uno: Revista de Didáctica de Las Matemáticas*, 25, 45–58. Retrieved from <http://www.ugr.es/~batanero/pages/ARTICULOS/isboa.pdf>
- Benjamin Wedro. (2018). Síntomas, causas, tratamiento y pruebas de fatiga. Retrieved February 20, 2019, from https://www.medicinenet.com/fatigue/article.htm#what_is_fatigue_what_does_it_feel_like
- Brugués, A. (2019). Diferencias entre Python 2 y 3: ¿Qué versión debería aprender? | Programa en Python. Retrieved July 25, 2019, from <https://www.programaenpython.com/miscelanea/diferencias-entre-python-2-y-3/>
- Castillo, O., Figueroa, D., & Sevilla Hector. (2019). ProgramacionExtrema. Retrieved July 10, 2019, from <http://programacionextrema.tripod.com/index.htm>

- Centro, C. V. del. (2018). Información general acerca del Ojo humano. Retrieved February 21, 2019, from <http://www.clinicavisualdelcentro.com.mx/salud-visual/acerca-del-ojo/>
- Choi, I. H., & Kim, Y. G. (2014). Head pose and gaze direction tracking for detecting a drowsy driver. In *2014 International Conference on Big Data and Smart Computing, BIGCOMP 2014* (pp. 241–244). IEEE. <https://doi.org/10.1109/BIGCOMP.2014.6741444>
- Cortes Osorio, J., Urueña, W., & Mendoza Vargas, J. (2011). Técnicas alternativas para la conversión de imágenes a color a escala de grises en el tratamiento digital de imágenes. *Scientia et Technica*, *1*(47), 207–212. <https://doi.org/10.22517/23447214.533>
- De Castro, J. R., Gallo, J., & Loureiro, H. (2004). Tiredness and sleepiness in bus drivers and road accidents in Peru: A quantitative study. *Revista Panamericana de Salud Publica/Pan American Journal of Public Health*, *16*(1), 11–18. <https://doi.org/10.1590/S1020-49892004000700002>
- Dra Bustamante Gladys, M. C. (2011). Revista de Actualización Clínica. Retrieved from http://www.revistasbolivianas.org.bo/pdf/raci/v10/v10_a06.pdf
- DreamsTime. (2019). Collage de ojos humanos imagen de archivo. Imagen de asiático - 45826287. Retrieved July 29, 2019, from <https://es.dreamstime.com/foto-de-archivo-collage-de-ojos-humanos-image45826287>
- Flores, M. J., Armingol M, J. M., & de la Escalera, A. (2011). Sistema avanzado de asistencia a la conducción para la detección de la somnolencia. *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, *8*(3), 216–228. <https://doi.org/10.1016/j.riai.2011.06.009>
- Foody, G. M. (2002). Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, *80*(1), 185–201. [https://doi.org/10.1016/S0034-4257\(01\)00295-4](https://doi.org/10.1016/S0034-4257(01)00295-4)
- Garcés, M. A., Salgado, J. D. J., Cruz, J. A., & Cañon, W. H. (2015). Sistemas de detección de somnolencia en conductores: inicio, desarrollo y futuro. *Ingeniería y Región*, *13*(1), 159. <https://doi.org/10.25054/22161325.717>
- Grace, R., Byrne, V. E., Bierman, D. M., Legrand, J.-M., Gricourt, D., Davis, B. K., ... Carnahan, B. (2002). A drowsy driver detection system for heavy vehicles. In *17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No.98CH36267)* (Vol. 2, p. I36/1-I36/8). IEEE. <https://doi.org/10.1109/dasc.1998.739878>
- Hazim Barnouti, N. (2016). Improve Face Recognition Rate Using Different Image Pre-Processing Techniques. *American Journal of Engineering Research (AJER)*, *(5)*, 46–53. Retrieved from www.ajer.org
- Heiting, G. (2019). Beneficios de las gafas con lentes antirreflejos | All About Vision. Retrieved November 29, 2019, from <https://www.allaboutvision.com/es/gafas/lentes-antirreflejos.htm>
- Izco, F. (2018). Base de datos corporativa de personas. Retrieved November 27, 2019, from https://bookdown.org/f_izco/BDC-POC/
- King, D. (2019). GitHub - davisking / dlib-models: archivos de modelo entrenados para programas de ejemplo dlib. Retrieved July 27, 2019, from https://github.com/davisking/dlib-models?source=post_page-----

- King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, 10(Jul), 1755–1758. Retrieved from <http://www.jmlr.org/papers/v10/king09a.html>
- Li, H. (2001). Computer recognition of human emotions. In *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, ISIMP 2001* (pp. 490–493). IEEE. <https://doi.org/10.1109/isimp.2001.925440>
- Liu, J. (2013). `rect(x, y, width, height, radius)` | Programación de computadoras | Khan Academy. Retrieved July 17, 2019, from <https://es.khanacademy.org/computer-programming/rectx-y-width-height-radius/839496660>
- López, N. (2012). *Estudio del parpadeo durante la conducción de vehículos (Aspectos cognitivos y de flujo de información)*. Catalunya. Retrieved from https://upcommons.upc.edu/bitstream/handle/2099.1/15598/TFM_Noa_Quevedo.pdf?sequence=1&isAllowed=y
- MAH, A. (2016). Improvement of Haar Feature Based Face Detection in OpenCV Incorporating Human Skin Color Characteristic. *Journal of Computer Science Applications and Information Technology*, 1(1), 1–8. <https://doi.org/10.15226/2474-9257/1/1/00106>
- Manzano Urquiza, C. E., & Tene Escobar, E. P. (2016). *Desarrollo e Implantación de un Sistema Móvil para evaluación de daños y análisis de necesidades en organizaciones de socorro y atención de desastres*. Retrieved from <https://bibdigital.epn.edu.ec/handle/15000/15235>
- Marciniak, T., Chmielewska, A., Weychan, R., Parzych, M., & Dabrowski, A. (2015). Influence of low resolution of images on reliability of face detection and recognition. *Multimedia Tools and Applications*, 74(12), 4329–4349. <https://doi.org/10.1007/s11042-013-1568-8>
- Menon, A. (2019). Detección de rostros en 2 minutos usando OpenCV y Python - Hacia la ciencia de datos. Retrieved July 16, 2019, from <https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81>
- Mordvintsev, A., & Abid, K. (2013). Histograms - 2: Histogram Equalization — OpenCV-Python Tutorials 1 documentation. Retrieved November 28, 2019, from https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_histograms/py_histogram_equalization/py_histogram_equalization.html
- Nathan, A. J., & Scobell, A. (2012). How China sees America. *Foreign Affairs*, 91(5), 1–8. <https://doi.org/10.1017/CBO9781107415324.004>
- Nikon, I. (2019). Distancia focal | Entendiendo qué es el zoom de la cámara y la distancia focal del lente | Nikon de Nikon. Retrieved November 5, 2019, from <https://www.nikon.com.mx/learn-and-explore/a/tips-and-techniques/entendiendo-la-distancia-focal.html>
- Novini, A. (1993). Fundamentals of machine vision lighting. In *Proceedings of WESCON 1993 Conference Record* (pp. 44–52). <https://doi.org/10.1109/WESCON.1993.488407>
- OpenCV, T. D. (2013). Face Detection using Haar Cascades. Retrieved February 21, 2019, from http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html

- Paz, K. (2007). Media Aritmética Simple. *Facultad de Ingeniería*, (07), 1–13. Retrieved from http://www.tec.url.edu.gt/boletin/URL_07_BAS01.pdf
- Peraza, B., Rondón, P., & Arguello Fuentes, H. (2011). Sistema de reconocimiento facial basado en imágenes con color. *Revista UIS Ingenierías*, 10(2), 113–122. Retrieved from <https://www.redalyc.org/pdf/5537/553756874004.pdf>
- Python Package Index. (2019). opencv-contrib-python · PyPI. Retrieved July 26, 2019, from <https://pypi.org/project/opencv-contrib-python/>
- PythonPackageIndex. (2018). imutils · PyPI. Retrieved July 27, 2019, from <https://pypi.org/project/imutils/>
- Ramella, G., & Di Baja, G. S. (2013). A new technique for color quantization based on histogram analysis and clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 27(3). <https://doi.org/10.1142/S0218001413600069>
- Rey de Castro, J. (2003). *Revista médica herediana : órgano oficial de la Facultad de Medicina "Alberto Hurtado"*, Universidad Peruana Cayetano Heredia, Lima, Peru. *Revista Medica Herediana* (Vol. 6). Universidad Peruana Cayetano Heredia. Retrieved from http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1018-130X1995000200005
- Rey de Castro Mujica, J., Rosales Mayor, E., & Egoavil Rojas, M. (2009). Somnolencia y cansancio durante la conducción: accidentes de tránsito en las carreteras del Perú. *Acta Médica Peruana*, 26(1), 48–54. Retrieved from http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1728-59172009000100011
- Rojas, F. (2018). El dispositivo para detectar somnolencia al conducir ya puso primera. Retrieved January 28, 2020, from <http://www.suplementos.unsj.edu.ar/octubre73/?p=1463>
- Rosebrock, A. (2018). Install dlib (the easy, complete guide) - PyImageSearch. Retrieved July 28, 2019, from <https://www.pyimagesearch.com/2018/01/22/install-dlib-easy-complete-guide/>
- S, J. J. S., & D, J. F. A. (2011). Detección y análisis de movimiento usando visión artificial. *Scientia Et Technica*, 16(49), 180–188. <https://doi.org/10.22517/23447214.1513>
- Sahayadhas, A., Sundaraj, K., & Murugappan, M. (2012). Detecting driver drowsiness based on sensors: A review. *Sensors (Switzerland)*, 12(12), 16937–16953. <https://doi.org/10.3390/s121216937>
- Santos, E. (2016). 9 sistemas operativos y gestores de contenido que puedes instalar en una Raspberry Pi. Retrieved May 29, 2019, from <https://www.genbeta.com/herramientas/9-sistemas-operativos-y-gestores-de-contenido-que-puedes-instalar-en-una-raspberry-pi>
- Schapire, R. E. (1999). *A brief introduction to boosting. IJCAI International Joint Conference on Artificial Intelligence* (Vol. 2). Retrieved from www.research.att.com/fyoav,
- Shi, S.-Y., Tang, W.-Z., & Wang, Y.-Y. (2017). A Review on Fatigue Driving Detection. *ITM Web of Conferences*, 12, 01019. <https://doi.org/10.1051/itmconf/20171201019>
- Sigari, M.-H., Pourshahabi, M.-R., Soryani, M., & Fathy, M. (2014). A Review on Driver Face Monitoring Systems for Fatigue and Distraction Detection. *International Journal of*

Advanced Science and Technology, 64(1), 73–100.
<https://doi.org/10.14257/ijast.2014.64.07>

StackOverFlow. (2016). python - Cálculo eficiente y preciso de la distancia euclidiana - Desbordamiento de pila. Retrieved July 25, 2019, from <https://stackoverflow.com/questions/37794849/efficient-and-precise-calculation-of-the-euclidean-distance/37795190#37795190>

Stan, O., Miclea, L., & Centea, A. (2014). Eye-Gaze Tracking Method Driven by Raspberry PI Applicable in Automotive Traffic Safety. *Proceedings - 2nd International Conference on Artificial Intelligence, Modelling, and Simulation, AIMS 2014*, 126–130. <https://doi.org/10.1109/AIMS.2014.45>

Viola, P., & Jones, M. J. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2), 137–154. <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>

Wikipedia. (2019). Matriz de confusión - Wikipedia, la enciclopedia libre. Retrieved November 26, 2019, from https://es.wikipedia.org/wiki/Matriz_de_confusi3n

Yu, Z., Ye, X., Xu, Q., Xie, X., Dong, H., & Su, W. (2017). A Fully Continuous-Flow Process for the Synthesis of *p*-Cresol: Impurity Analysis and Process Optimization. *Organic Process Research and Development* (Vol. 21). <https://doi.org/10.1021/acs.oprd.7b00250>

Anexos

Anexo A: Tabla de los valores que tomó el EAR a lo largo del tiempo en una secuencia de vídeo para cada uno de los participantes en distintos estados: “Despierto”, “Medio dormido” y “Dormido”.

Significado de las siglas:

EH = Erick Herrera

JA = Josué Alba

KH = Karla Herrera

MP = Marco Pusedá

DE = Despierto/a

MD = Medio dormido/a

DO = Dormido/a

EH_DE	JA_DE	KH_DE	MP_DE	EH_MD	JA_MD	KH_MD	MP_MD	EH_DO	JA_DO	KH_DO	MP_DO
0,38	0,30	0,25	0,29	0,22	0,17	0,16	0,18	0,18	0,13	0,14	0,11
0,37	0,30	0,24	0,29	0,22	0,20	0,15	0,17	0,20	0,11	0,13	0,11
0,38	0,30	0,24	0,27	0,23	0,16	0,14	0,17	0,18	0,11	0,14	0,12
0,37	0,30	0,25	0,29	0,20	0,19	0,17	0,15	0,16	0,13	0,14	0,12
0,38	0,32	0,25	0,29	0,20	0,17	0,17	0,18	0,16	0,14	0,14	0,11
0,36	0,32	0,26	0,27	0,21	0,17	0,15	0,15	0,17	0,14	0,14	0,10
0,36	0,32	0,25	0,27	0,23	0,16	0,15	0,17	0,18	0,12	0,14	0,13
0,40	0,32	0,26	0,25	0,22	0,15	0,14	0,17	0,17	0,14	0,14	0,13
0,38	0,32	0,25	0,29	0,20	0,15	0,18	0,17	0,18	0,14	0,14	0,11
0,37	0,32	0,26	0,27	0,22	0,15	0,15	0,15	0,18	0,12	0,14	0,12
0,38	0,30	0,25	0,29	0,23	0,16	0,18	0,18	0,18	0,12	0,16	0,13
0,37	0,30	0,25	0,29	0,23	0,17	0,18	0,19	0,16	0,13	0,14	0,12
0,36	0,32	0,24	0,30	0,24	0,16	0,17	0,17	0,16	0,14	0,12	0,12
0,36	0,30	0,25	0,26	0,22	0,17	0,17	0,19	0,19	0,12	0,12	0,11
0,39	0,29	0,24	0,28	0,23	0,18	0,15	0,18	0,19	0,12	0,14	0,13
0,38	0,32	0,25	0,27	0,22	0,17	0,16	0,16	0,18	0,12	0,16	0,12
0,38	0,30	0,24	0,30	0,21	0,15	0,17	0,17	0,19	0,12	0,13	0,13
0,38	0,33	0,23	0,30	0,22	0,19	0,18	0,18	0,16	0,13	0,14	0,13
0,38	0,30	0,22	0,30	0,24	0,19	0,18	0,18	0,17	0,13	0,16	0,12
0,39	0,31	0,23	0,30	0,23	0,15	0,16	0,16	0,19	0,12	0,14	0,13
0,39	0,32	0,25	0,30	0,22	0,17	0,18	0,18	0,16	0,14	0,14	0,12
0,39	0,32	0,24	0,29	0,22	0,20	0,15	0,18	0,20	0,11	0,13	0,13
0,38	0,32	0,24	0,29	0,22	0,16	0,17	0,18	0,18	0,13	0,11	0,14
0,38	0,33	0,25	0,27	0,24	0,15	0,16	0,18	0,16	0,14	0,15	0,14
0,39	0,30	0,25	0,27	0,22	0,15	0,16	0,19	0,15	0,13	0,14	0,14
0,39	0,30	0,25	0,30	0,23	0,18	0,18	0,17	0,16	0,13	0,14	0,14
0,39	0,31	0,25	0,27	0,23	0,15	0,17	0,18	0,16	0,13	0,13	0,14
0,38	0,33	0,24	0,27	0,24	0,19	0,15	0,19	0,14	0,14	0,16	0,14
0,36	0,30	0,24	0,26	0,22	0,17	0,15	0,18	0,14	0,12	0,14	0,12
0,38	0,32	0,24	0,26	0,22	0,19	0,16	0,19	0,16	0,13	0,14	0,12

0,40	0,32	0,24	0,26	0,23	0,16	0,14	0,18	0,16	0,14	0,14	0,14
0,39	0,31	0,24	0,29	0,23	0,20	0,15	0,18	0,14	0,12	0,14	0,14
0,39	0,31	0,24	0,28	0,22	0,15	0,16	0,17	0,18	0,12	0,14	0,15
0,39	0,29	0,24	0,27	0,22	0,18	0,18	0,16	0,19	0,12	0,13	0,12
0,41	0,31	0,25	0,26	0,22	0,19	0,17	0,17	0,21	0,12	0,15	0,13
0,39	0,30	0,25	0,28	0,23	0,19	0,16	0,19	0,19	0,12	0,15	0,12
0,39	0,30	0,24	0,29	0,21	0,17	0,15	0,17	0,16	0,13	0,15	0,13
0,41	0,30	0,24	0,29	0,23	0,16	0,15	0,18	0,18	0,13	0,14	0,15
0,38	0,30	0,25	0,28	0,24	0,17	0,15	0,19	0,20	0,12	0,13	0,13
0,41	0,29	0,24	0,27	0,23	0,20	0,16	0,16	0,15	0,14	0,12	0,14
0,39	0,33	0,24	0,27	0,22	0,15	0,15	0,15	0,17	0,11	0,16	0,14
0,38	0,32	0,26	0,29	0,22	0,19	0,15	0,15	0,18	0,13	0,14	0,13
0,41	0,31	0,24	0,28	0,24	0,17	0,17	0,15	0,18	0,14	0,12	0,14
0,39	0,31	0,23	0,28	0,22	0,20	0,16	0,17	0,18	0,13	0,16	0,13
0,39	0,30	0,24	0,28	0,22	0,20	0,17	0,16	0,17	0,13	0,14	0,13
0,40	0,31	0,26	0,30	0,22	0,21	0,18	0,18	0,19	0,13	0,12	0,13
0,38	0,31	0,24	0,29	0,24	0,20	0,17	0,15	0,20	0,14	0,14	0,15
0,40	0,33	0,25	0,30	0,23	0,20	0,15	0,18	0,18	0,14	0,14	0,13
0,36	0,32	0,24	0,30	0,23	0,16	0,15	0,17	0,20	0,14	0,16	0,14
0,39	0,31	0,26	0,29	0,21	0,15	0,15	0,19	0,18	0,13	0,15	0,13
0,40	0,33	0,24	0,29	0,22	0,17	0,16	0,19	0,15	0,13	0,14	0,12
0,40	0,30	0,24	0,28	0,22	0,16	0,17	0,18	0,18	0,12	0,15	0,13
0,41	0,30	0,25	0,29	0,22	0,15	0,16	0,17	0,17	0,12	0,15	0,12
0,40	0,31	0,24	0,29	0,23	0,19	0,16	0,16	0,20	0,14	0,14	0,14
0,39	0,31	0,24	0,29	0,24	0,15	0,15	0,16	0,19	0,13	0,14	0,14
0,39	0,29	0,23	0,30	0,22	0,19	0,17	0,18	0,18	0,14	0,16	0,12
0,39	0,31	0,26	0,29	0,23	0,19	0,17	0,17	0,18	0,14	0,15	0,12
0,39	0,32	0,24	0,30	0,22	0,17	0,17	0,17	0,17	0,13	0,15	0,13
0,39	0,31	0,24	0,29	0,22	0,19	0,14	0,14	0,19	0,15	0,14	0,13
0,40	0,31	0,24	0,30	0,22	0,20	0,15	0,18	0,16	0,11	0,15	0,12
0,37	0,31	0,25	0,29	0,20	0,16	0,17	0,16	0,19	0,14	0,15	0,12
0,40	0,29	0,24	0,28	0,22	0,18	0,16	0,15	0,16	0,13	0,14	0,13
0,41	0,31	0,24	0,29	0,22	0,18	0,15	0,16	0,16	0,14	0,14	0,12
0,38	0,32	0,25	0,29	0,22	0,16	0,16	0,17	0,17	0,13	0,14	0,13
0,37	0,31	0,24	0,28	0,21	0,15	0,16	0,19	0,13	0,14	0,14	0,12
0,40	0,32	0,25	0,29	0,22	0,19	0,16	0,19	0,17	0,11	0,15	0,13
0,40	0,32	0,26	0,30	0,22	0,19	0,17	0,15	0,13	0,14	0,15	0,15
0,37	0,33	0,25	0,29	0,21	0,15	0,18	0,17	0,13	0,14	0,14	0,15
0,38	0,33	0,26	0,29	0,25	0,19	0,17	0,18	0,13	0,11	0,14	0,13
0,38	0,32	0,26	0,29	0,24	0,16	0,15	0,16	0,15	0,14	0,14	0,12
0,40	0,29	0,25	0,29	0,23	0,19	0,15	0,17	0,16	0,14	0,16	0,14
0,38	0,31	0,24	0,29	0,23	0,16	0,15	0,16	0,14	0,13	0,14	0,15
0,39	0,29	0,26	0,30	0,23	0,17	0,16	0,19	0,14	0,13	0,14	0,13
0,36	0,28	0,25	0,29	0,22	0,19	0,17	0,19	0,17	0,14	0,17	0,12
0,39	0,27	0,25	0,30	0,24	0,16	0,16	0,15	0,18	0,13	0,14	0,15
0,39	0,29	0,25	0,30	0,22	0,19	0,17	0,16	0,16	0,12	0,16	0,13

0,41	0,29	0,24	0,30	0,22	0,17	0,17	0,15	0,17	0,13	0,13	0,14
0,40	0,31	0,25	0,29	0,22	0,19	0,17	0,15	0,19	0,12	0,14	0,13
0,38	0,32	0,26	0,28	0,23	0,20	0,17	0,16	0,17	0,13	0,14	0,13
0,38	0,31	0,27	0,29	0,21	0,19	0,16	0,15	0,18	0,14	0,16	0,12
0,38	0,31	0,27	0,30	0,21	0,19	0,18	0,15	0,19	0,14	0,16	0,15
0,37	0,29	0,26	0,28	0,23	0,15	0,18	0,15	0,19	0,13	0,13	0,15
0,40	0,31	0,26	0,27	0,20	0,17	0,16	0,16	0,16	0,12	0,13	0,12
0,38	0,31	0,26	0,27	0,20	0,20	0,18	0,16	0,16	0,13	0,15	0,13
0,39	0,29	0,25	0,30	0,22	0,20	0,18	0,18	0,19	0,12	0,14	0,13
0,39	0,31	0,26	0,28	0,22	0,17	0,14	0,19	0,19	0,14	0,16	0,14
0,41	0,31	0,24	0,28	0,22	0,20	0,18	0,18	0,18	0,12	0,14	0,13
0,39	0,31	0,27	0,26	0,23	0,16	0,18	0,17	0,18	0,13	0,14	0,14
0,40	0,31	0,25	0,27	0,21	0,19	0,17	0,16	0,15	0,13	0,13	0,13
0,39	0,29	0,26	0,30	0,24	0,15	0,16	0,18	0,15	0,12	0,13	0,13
0,38	0,29	0,26	0,29	0,23	0,17	0,16	0,16	0,15	0,15	0,15	0,13
0,39	0,27	0,25	0,29	0,23	0,19	0,16	0,17	0,14	0,12	0,15	0,12
0,38	0,28	0,26	0,29	0,23	0,17	0,14	0,15	0,19	0,13	0,14	0,12
0,37	0,27	0,26	0,29	0,23	0,17	0,14	0,19	0,13	0,12	0,13	0,11
0,38	0,30	0,26	0,30	0,24	0,19	0,14	0,17	0,18	0,13	0,14	0,13
0,38	0,30	0,24	0,28	0,21	0,20	0,15	0,17	0,17	0,14	0,14	0,12
0,38	0,29	0,26	0,29	0,23	0,20	0,15	0,18	0,16	0,13	0,13	0,12
0,39	0,28	0,25	0,29	0,23	0,18	0,14	0,16	0,13	0,14	0,14	0,13
0,39	0,27	0,26	0,28	0,23	0,20	0,16	0,17	0,14	0,12	0,15	0,13
0,41	0,28	0,27	0,28	0,22	0,21	0,16	0,16	0,15	0,13	0,14	0,13
0,39	0,27	0,27	0,28	0,21	0,20	0,18	0,19	0,17	0,14	0,14	0,12
0,40	0,29	0,26	0,30	0,23	0,19	0,18	0,19	0,17	0,14	0,13	0,13
0,38	0,28	0,24	0,29	0,23	0,20	0,17	0,16	0,16	0,13	0,13	0,13
0,39	0,27	0,25	0,30	0,22	0,19	0,15	0,16	0,15	0,14	0,13	0,14
0,41	0,28	0,25	0,30	0,21	0,21	0,15	0,15	0,16	0,14	0,15	0,11
0,40	0,30	0,25	0,29	0,21	0,19	0,16	0,15	0,18	0,13	0,15	0,12
0,41	0,29	0,25	0,29	0,21	0,21	0,14	0,16	0,18	0,12	0,13	0,14
0,40	0,29	0,25	0,29	0,23	0,20	0,14	0,15	0,16	0,13	0,14	0,11
0,42	0,28	0,25	0,29	0,21	0,20	0,15	0,15	0,15	0,14	0,16	0,14
0,39	0,32	0,26	0,28	0,22	0,20	0,15	0,17	0,19	0,14	0,15	0,13
0,41	0,31	0,26	0,29	0,24	0,15	0,17	0,17	0,18	0,12	0,15	0,13
0,39	0,32	0,25	0,29	0,24	0,19	0,16	0,15	0,19	0,12	0,16	0,13
0,39	0,31	0,27	0,30	0,22	0,19	0,16	0,19	0,16	0,12	0,14	0,12
0,38	0,29	0,26	0,29	0,23	0,15	0,15	0,18	0,17	0,14	0,15	0,13
0,41	0,31	0,25	0,28	0,24	0,19	0,14	0,15	0,19	0,13	0,16	0,14
0,43	0,32	0,27	0,28	0,21	0,20	0,15	0,17	0,17	0,12	0,14	0,12
0,39	0,32	0,26	0,27	0,22	0,20	0,16	0,17	0,19	0,13	0,14	0,14
0,36	0,32	0,25	0,27	0,22	0,20	0,16	0,18	0,12	0,13	0,14	0,13
0,37	0,32	0,25	0,27	0,23	0,19	0,15	0,19	0,14	0,13	0,14	0,12
0,41	0,33	0,25	0,26	0,21	0,16	0,16	0,17	0,12	0,14	0,15	0,14
0,40	0,33	0,25	0,29	0,22	0,20	0,15	0,19	0,14	0,14	0,16	0,14
0,37	0,31	0,25	0,27	0,24	0,20	0,15	0,18	0,12	0,14	0,16	0,14

0,37	0,32	0,25	0,27	0,23	0,16	0,16	0,16	0,13	0,13	0,16	0,15
0,38	0,32	0,26	0,26	0,23	0,17	0,17	0,17	0,14	0,12	0,14	0,13
0,37	0,33	0,25	0,27	0,21	0,17	0,17	0,18	0,19	0,13	0,15	0,14
0,39	0,33	0,25	0,27	0,22	0,16	0,17	0,18	0,20	0,13	0,16	0,14
0,38	0,32	0,25	0,28	0,20	0,16	0,18	0,16	0,16	0,12	0,15	0,15
0,41	0,33	0,26	0,26	0,23	0,18	0,17	0,18	0,15	0,13	0,16	0,11
0,38	0,33	0,24	0,29	0,22	0,16	0,16	0,18	0,18	0,13	0,14	0,15
0,39	0,33	0,26	0,27	0,23	0,15	0,15	0,18	0,15	0,13	0,15	0,14
0,37	0,33	0,25	0,28	0,22	0,16	0,15	0,17	0,19	0,12	0,14	0,13
0,40	0,32	0,26	0,29	0,21	0,16	0,17	0,19	0,13	0,14	0,15	0,14
0,38	0,31	0,26	0,30	0,24	0,18	0,15	0,18	0,15	0,13	0,16	0,15
0,40	0,33	0,24	0,27	0,23	0,18	0,16	0,19	0,15	0,13	0,16	0,13
0,38	0,33	0,24	0,26	0,22	0,18	0,16	0,18	0,14	0,12	0,13	0,14
0,38	0,33	0,27	0,27	0,22	0,20	0,16	0,18	0,19	0,10	0,15	0,15
0,40	0,33	0,24	0,29	0,22	0,17	0,15	0,17	0,13	0,13	0,13	0,13
0,37	0,31	0,24	0,28	0,21	0,20	0,16	0,19	0,18	0,13	0,16	0,15
0,38	0,31	0,25	0,29	0,22	0,19	0,17	0,17	0,17	0,13	0,14	0,15
0,36	0,33	0,24	0,27	0,21	0,18	0,15	0,18	0,16	0,13	0,14	0,15
0,37	0,33	0,25	0,27	0,23	0,15	0,16	0,19	0,13	0,12	0,16	0,14
0,38	0,34	0,24	0,27	0,22	0,15	0,15	0,16	0,14	0,15	0,14	0,15
0,37	0,33	0,24	0,27	0,23	0,15	0,16	0,19	0,15	0,14	0,16	0,15
0,38	0,32	0,25	0,26	0,21	0,17	0,15	0,17	0,17	0,13	0,13	0,15
0,41	0,33	0,25	0,27	0,22	0,17	0,16	0,16	0,15	0,14	0,14	0,12
0,39	0,31	0,25	0,29	0,23	0,20	0,16	0,18	0,20	0,14	0,15	0,12
0,37	0,33	0,24	0,28	0,20	0,17	0,16	0,15	0,20	0,14	0,13	0,14
0,39	0,33	0,24	0,28	0,20	0,18	0,14	0,18	0,17	0,13	0,14	0,14
0,38	0,31	0,25	0,30	0,23	0,16	0,16	0,17	0,16	0,13	0,16	0,11
0,39	0,30	0,26	0,30	0,20	0,19	0,15	0,19	0,18	0,12	0,15	0,12