

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE TECNOLOGÍA
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES



DISEÑO DE UN DISPOSITIVO DE MONITOREO, ALERTA DE
SOMNOLENCIA Y FATIGA DEL CONDUCTOR

Proyecto de grado presentado para obtener el Grado de Licenciatura

POR: PAMELA PACHECO CABEZAS
MAURICIO ALEJANDRO CALLA MARIN

TUTOR: ING. LUIS RAMIRO VELARDE CHÁVEZ

LA PAZ – BOLIVIA
Agosto, 2018

DEDICATORIA

Mi proyecto de grado lo dedico con todo amor y cariño a mí querida madre Plácida Cabezas, por su sacrificio, esfuerzo, paciencia, insistencia en yo tenga una carrera profesional, para que pueda ser libre de soñar y de elegir mi propio destino.

Mami, es gracias a todo tu amor, comprensión y ejemplo que no me he rendido.

Pamela Pacheco Cabezas

AGRADECIMIENTOS

Quiero agradecer de la manera más atenta a las personas que me brindaron todo su apoyo para culminar con este primer logro profesional:

A mis padres Miguel Pacheco y Plácida Cabezas, por haberme apoyado desde siempre. Este, mi primer logro sin lugar a dudas es gracias a ustedes, no sé dónde me encontraría si no fuera por todo su amor, cariño y paciencia.

A mi hermano Nelson Pacheco, por todo su cariño y complicidad en todo momento.

A mi novio Alberto Inch, por aparecer en un momento difícil de mi vida, devolverme la sonrisa, la alegría y apoyarme con tanto amor en todas mis decisiones y acciones.

A mi compañero de proyecto Mauricio Calla, por todo su esfuerzo, amistad, paciencia y entrega con este trabajo. Sin su participación este proyecto no hubiera sido posible.

A nuestro Docente tutor el Ing. Ramiro Velarde, por la orientación y predisposición de manera desinteresada para concluir este proyecto, por todos sus consejos para el mejoramiento del mismo, por su tiempo y dedicación.

A nuestros tribunales, Msc Edwin Alave, Ing. José Arturo Marin y Franklin Rada Telleria, por su tiempo y por su valiosa colaboración en la revisión y mejoramiento de este proyecto.

A Msc. Richard Marquez, Melania Andrade, todo el plantel docente y administrativo de la carrera de Electrónica y Telecomunicaciones de la Facultad de Tecnología, por todo su apoyo recibido durante mis años en aulas.

Pamela Pacheco Cabezas

DEDICATORIA

A mi familia que es el pilar fundamental de mi vida, porque ellos están en los momentos felices y sobre todo no se separaron de mí en los momentos más difíciles.

A todas esas personas que significan y significaron enseñanzas en mi vida porque de cada experiencia que viví aprendí, volviéndome más fuerte para poder caminar en la vida.

Solo pido que la fuerza y la determinación nunca me abandonen para poder seguir creciendo como persona, como profesional. Pudiendo transitar esta senda hacia el futuro sin olvidarme del presente y sin encadenarme al pasado.

Mauricio Alejandro Calla Marin

AGRADECIMIENTOS

A mis Padres Lazaro Calla e Ines Marin por Acompañarme en esta etapa de mi vida que cierro y que sin su apoyo no hubiera sido posible lograrlo.

A mis hermanos Cindy Calla, Manuel Calla y Carmen Calla por esa mezcla de emociones y apoyo que me brindaron.

A mi compañera Pamela Pacheco con quien realice este trabajo, porque sin su iniciativa, esmeró, fortaleza, comprensión y apoyo nunca hubiéramos podido concluir con esta meta que nos propusimos.

A mi tutor el Ing. Ramiro Velarde por la orientación brindada para realizar este trabajo.

Y finalmente a todas las personas que forman parte de mi vida por todo el apoyo brindado.

Por todo muchas gracias.

Mauricio Alejandro Calla Marin

ÍNDICE DE CONTENIDO

RESUMEN	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
CAPÍTULO I.....	4
MARCO REFERENCIAL.....	4
1.1. ANTECEDENTES	4
1.1.1. Mundial.....	4
1.1.2. Sud América (chile).....	5
1.1.3. Bolivia.....	5
1.2. DEFINICIÓN DEL PROBLEMA	6
1.3. OBJETIVOS	8
1.3.1. Objetivo general.....	8
1.3.2. Objetivos específicos	8
1.4. LÍMITES Y ALCANCES.....	9
1.4.1. Límites	9
1.4.2. Alcances.....	9
1.5. JUSTIFICACIÓN	10
CAPÍTULO II.....	12
MARCO TEÓRICO	12
2.1. CAUSAS DE ACCIDENTES AUTOMOVILÍSTICOS EN BOLIVIA.....	12
2.2. ELEMENTOS DE LA CONDUCCIÓN.....	18
2.2.1. El conductor	18
2.2.1.1. Estado físico y mental del conductor	19
2.2.1.3. La somnolencia	20
2.2.1.4. Trastornos del sueño	21
2.2.1.4.1. Insomnio	22
2.2.1.4.2. Hipersomnia.....	23
2.2.1.4.3. Narcolepsia	24
2.2.1.4.4. Parasomnias	25
2.2.1.4.5. Apnea del sueño.....	25
2.2.2. El vehículo	26

2.2.2.1.	Sistemas de seguridad existentes en el vehículo	27
2.2.2.1.1.	Seguridad activa.....	27
2.2.2.1.2.	Seguridad pasiva	28
2.2.2.1.3.	Nuevos sistemas de seguridad en el vehículo	29
2.2.2.	Las vías	31
2.2.3.	El entorno.....	32
2.3.	HERRAMIENTAS APLICADAS AL DESARROLLO DEL PROYECTO	32
2.3.1.	Open source	32
2.3.2.	Software open source.....	32
2.3.3.	Raspbian.....	33
2.3.4.	Python	33
2.3.5.	Open CV	35
2.3.6.	Hardware open source.....	39
2.3.7.	Raspberry pi 3 modelo B	39
2.3.8.	Cámara de visión nocturna compatible con Raspberry Pi	42
2.3.9.	Modem E303.....	43
2.3.10.	Amplificador operacional TDA2822M.....	44
2.3.11.	Visión computacional	46
2.3.12.	Algoritmo viola y jones	46
2.3.13.	Haar-like	47
2.3.14.	Imagen integral	51
2.3.15.	Adaboost	52
2.3.16.	Tecnología GSM.....	54
CAPÍTULO III.....		61
DESARROLLO DEL PROYECTO		61
3.1.	ANÁLISIS PRELIMINAR.....	61
3.2.	INGENIERÍA DEL PROYECTO.....	63
3.2.1.	Diseño del hardware	66
3.2.1.1.	Puertos y ranuras de conexión	66
3.2.1.2.	Amplificador de audio para el módulo de alerta.....	70
3.2.1.3.	Ensamblaje del hardware	73
3.2.1.4.	Conexión del hardware	77
3.2.2.	Diseño software	78
3.2.2.1.	Instalación del sistema operativo raspbian.....	78

3.2.2.2.	Instalación de la librería open CV.....	80
3.2.2.3.	Instalación de python	82
3.2.2.4.	Instalación modem E303.....	83
3.2.2.5.	Desarrollo del programa	83
3.2.2.5.1.	Módulo de detección del iris del conductor	84
3.2.2.5.2.	Reconocimiento de los ojos en función a la variación de la luz	93
3.2.2.5.3.	Módulo de tiempo	95
3.2.2.5.4.	Módulo de registro.....	96
3.2.2.5.5.	Módulo SMS.....	97
3.2.2.5.6.	Módulo de sonido	98
3.2.3.	Presupuesto	99
3.2.4.	Cronograma.....	101
CAPITULO IV		102
PRUEBAS Y RESULTADOS.....		102
4.1.	PRUEBA DE RECONOCIMIENTO FACIAL PARA DIFERENTES PERSONAS	102
4.2.	PRUEBA DE DETECCIÓN DEL IRIS PARA DIFERENTES PERSONAS.....	105
4.3.	PRUEBA DE DETECCIÓN DEL IRIS PARA DIFERENTES DISTANCIA	106
4.4.	PRUEBA DE DETECCIÓN DEL IRIS PARA VARIACIONES DE LUZ.....	108
CAPITULO V.....		113
MARCO CONCLUSIVO		113
5.1.	CONCLUSIONES	113
5.2.	RECOMENDACIONES.....	114
BIBLIOGRAFÍA		115
ANEXOS		117
CÓDIGO MÓDULO IRIS		119
CÓDIGO MÓDULO TIEMPO.....		124
CÓDIGO MÓDULO REGISTRO.....		125
CÓDIGO MÓDULO SMS		127
CÓDIGO MÓDULO SONIDO		128

ÍNDICE DE TABLA

Tabla 1. Porcentaje de accidentes automovilísticos 2014 Bolivia datos INE y Policía Boliviana	13
Tabla 2. Accidentes de tránsito registrados según clase, 2014 Bolivia datos INE y Policía Boliviana.	14
Tabla 3. Accidentes de tránsito, según tipo de vehículo y área 2014, Bolivia datos INE y Policía Boliviana.	14
Tabla 4. Distribución de accidentes de tránsito por departamento, datos INE y Policía Boliviana.	15
Tabla 5. Artículos del Código de tránsito que hacen mención al relevamiento del conductor y problemas percibidos. Elaboración propia.....	16
Tabla 6. Distancias entre la ciudad de La Paz y los diferentes departamentos.	17
Tabla 7. Características Raspbian.	33
Tabla 8. Características de Python.	34
Tabla 9. Módulos a utilizar de Python.	35
Tabla 10. Características de Open CV.	37
Tabla 11. Funciones de Open CV.	38
Tabla 12. Librerías a utilizar de <i>Open CV</i>	39
Tabla 13. Características técnicas Raspberry PI 3 B.....	41
Tabla 14. Características técnicas Cámara de visión nocturna.	42
Tabla 15. Características del HUAWEI E303.....	43
Tabla 16. Descripción de los Haar de Viola y Jones.	48
Tabla 17. Descripción de los Haar de Lienhar y Maydt.	49
Tabla 18. Redes celulares GSM.....	56
Tabla 19. Componentes para el amplificador de audio.....	71
Tabla 20. Características del App.	94
Tabla 21. Tabla comparativa de cantidad de Lux.	95
Tabla 22. Presupuestó del proyecto de grado.....	100
Tabla 23. Cronograma.....	101
Tabla 24. Resultados de aciertos al detectar el pestañeo.	108
Tabla 25. Detección del rostro en distintos cambios de luz.....	112

ÍNDICE DE FIGURAS

Figura 1. Accidente en la carretera de los Yungas.....	19
Figura 2. Ingreso de Vehículos a Bolivia.....	26
Figura 3. Sistema de reconocimiento facial.....	31
Figura 4. Accidente en la carretera de los Yungas.....	31
Figura 5. Logo de Raspbian.....	33
Figura 6. Python lenguaje Multiuso y Multiplataforma.....	35
Figura 7. Funcionamiento de Open CV.....	36
Figura 8. Categorías de Open CV.....	38
Figura 9. Raspberry PI 3 Modelo B.....	40
Figura 10. Cámara visión nocturna.....	43
Figura 11. Modem HUAWEI E303.....	44
Figura 12. Diagrama esquemático del Amplificador Operacional TDA2822M.....	45
Figura 13. Configuración de los pin del TDA2822M.....	45
Figura 14. Encapsulado del TDA2822M.....	46
Figura 15. Característica Haar.....	47
Figura 16. A) Características de dos B) Características de tres C) Características de Cuatro.....	49
Figura 17. A) <i>Features</i> de dos inclinado B) <i>Features</i> de tres inclinado C) <i>Features</i> de dos centro-contornos.....	50
Figura 18. Funcionamiento y detección de Haar cascade.....	50
Figura 19. Calculo de una región a partir de una imagen integral.....	52
Figura 20. Diagrama de flujo que explica el funcionamiento de AdaBoost.....	53
Figura 21. Arquitectura de red GSM.....	59
Figura 22. Diagrama funcional del dispositivo.....	62
Figura 23. Diagrama conexión del hardware.....	63
Figura 24. Diagrama del funcionamiento detector de somnolencia.....	64
Figura 25 Diagrama flujo del funcionamiento del monitoreo de horas delante el volante.....	65
Figura 26. Conexiones y características utilizadas en el Raspebrry pi 3 B.....	67
Figura 27. Conexiones y características utilizadas en el modem Huawei e303.....	68
Figura 28. Conexiones y características utilizadas Cámara de visión nocturna.....	69
Figura 29. Amplificador de Audio proporcionado por el fabricante.....	70
Figura 30. Circuito modificado para cubrir las necesidades del dispositivo.....	71
Figura 31. Diseño de la placa en PCB Wizard.....	72
Figura 32. Negativo para realizar el quemado de la placa PBC.....	72
Figura 33. Vista en 3D del Amplificador de Audio.....	73

Figura 34. Componentes utilizados en el dispositivo (<i>Raspberry PI 3B</i> y Amplificador de audio).....	73
Figura 35. Esqueleto y tapas metálicas del dispositivo.....	74
Figura 36. Secuencia de fotos de ensamblaje del dispositivo.....	75
Figura 37. Ensamblado final del dispositivo.....	76
Figura 38. Conexiones generales del hardware.	77
Figura 39. Win32DiskImager.	79
Figura 40. Raspbian.	79
Figura 41. Variables de la cara (x, y, w, h).	87
Figura 42. Áreas de detección de la cara y los ojos.	88
Figura 43. Partes del ojo.	89
Figura 44. Logo App Luxómetro.	93
Figura 45. Entorno grafico utilizando por App.....	94
Figura 46. Algoritmo de funcionamiento del timer.	96
Figura 47. Detección de distintas facciones.....	104
Figura 48. Detección del iris en distintos rostros.....	106
Figura 49. Pruebas de Distancias para la detectar el iris.....	107

RESUMEN

El propósito de este proyecto es diseñar un dispositivo electrónico que permita monitorear la somnolencia del conductor durante la conducción, alertando de forma inmediata con un aviso sonoro y envío de mensajes de texto a familiares, si el conductor se duerme o presenta síntomas de fatiga frente al volante.

El funcionamiento del dispositivo se basa en el reconocimiento del iris utilizando *hardware* y *software open source*, donde la base del proyecto es una tarjeta de desarrollo Raspberry PI 3B, encargada del procesamiento de las imágenes enviadas por la cámara de visión nocturna conectada a la tarjeta como periférico, mediante las librerías de visión artificial de *Open CV* y un lenguaje de alto nivel. Para ayudar a la prevención de accidentes de tránsito por somnolencia y fatiga del conductor.

ABSTRACT

The purpose of this project is to design an electronic prototype that allows to monitor the drowsiness of the driver while driving, alerting immediately with a sound warning and sending text messages to family members, if the driver falls asleep or has fatigue symptoms in front of the steering wheel.

The operation of the prototype is based on the Iris detection using hardware and "open source" software, where the basis of the project is a Raspberry PI 3B development card, responsible for the processing of the images sent by the night vision camera connected to the card as a peripheral, through the Open CV artificial vision libraries and a high level language. To help prevention of traffic accidents due to drowsiness and fatigue of the driver.

INTRODUCCIÓN

El alto índice de accidentes automovilísticos causados por la imprudencia del conductor es un factor de alerta, que debe preocupar a la sociedad en general, porque los siniestros automovilísticos además de generar pérdidas materiales y humanas son considerados como el segundo causante de muertes en el mundo.

El cansancio y la somnolencia del conductor son los factores más preocupantes dentro de este índice de accidentes de tránsito, pues salen del control del chófer y de las mismas autoridades. Lamentablemente en Bolivia todavía no existe ninguna normativa o sistema que regule o controle de alguna manera el sueño y la fatiga laboral de los conductores.

En el caso de algún control de tránsito solo se observa que el conductor no se encuentre en estado de ebriedad, no exceda los límites de velocidad y que cuente con licencia de conducir, pero no existe ningún mecanismo que verifique la cantidad de horas que hubiese conducido el conductor, así también que haya cumplido las horas de descanso correspondientes para evitar problemas frente al volante.

Considerando todo lo mencionado, el presente proyecto tiene como propósito el diseño de un dispositivo con módulos de hardware *open source*, que permita monitorear la cantidad de horas que lleva el conductor frente al volante, alertarlo, recomendarle una parada de descanso en caso de presentar conductas de somnolencia y fatiga durante la conducción, que colabore con la prevención de accidentes automovilísticos ocasionados por la somnolencia y la fatiga de los conductores.

CAPÍTULO I.

MARCO REFERENCIAL

1.1.ANTECEDENTES

El cansancio y la somnolencia del conductor frente al volante, son factores importantes dentro de las causas de siniestros automovilísticos. Durante los últimos años se han estado planteando distintos tipos de soluciones para su prevención.

1.1.1. Mundial

En el año 2004, Estados Unidos hizo la modificación de su normativa para limitar las horas de conducción, alargando los períodos de descanso obligatorio. Estimando una reducción considerable de accidentes automovilísticos debidos a fatiga frente al volante.

Con las modificaciones legislativas los conductores pueden conducir en la jornada diaria hasta 11 horas en lugar de 10, pero el período de descanso obligatorio se amplía de 8 a 10 horas. El límite total, incluido el tiempo para la comida y la recarga de combustible, se reduce de 15 a 14 horas. Dicho tiempo no podrá superarse a no ser que se disponga de un sitio adecuado para dormir, y deberá ir seguido del descanso obligado de 10 horas.

En agosto del año 2012, en España, el “Instituto de Biomecánica de Valencia” (IBV) con financiación de la comisión Europea ha desarrollado un dispositivo capaz de medir la actividad cardíaca y la respiración del conductor para prevenir la somnolencia y la fatiga frente al volante. Este es un sistema de sensores no invasivo integrado en la cubierta del asiento y en el cinturón de seguridad del vehículo.

El año 2013, en México, estudiantes de la “Escuela Superior de Computo” (ESCOM) crearon el sistema denominado “Anti-sleep”, el cual se inicia con un software que mediante preguntas determina el estado del conductor como ser, cantidad de horas que durmió, si sufre algún padecimiento o si ha ingerido algún fármaco.

En septiembre de 2015, la asamblea general de las naciones unidas y jefes de estado, adoptaron la histórica agenda 2030 para el desarrollo sostenible, donde una de las nuevas metas es reducir a la mitad el número mundial de muertes y traumatismos por accidente de tránsito hasta el 2020.

Esto demuestra que existe un reconocimiento de los accidentes de tránsito como una de las causas más importantes de muerte y se exige adecuación de normas de seguridad vial en diferentes países para reducir la mortalidad por siniestros automovilísticos.

1.1.2. Sud América (chile)

En el año 2012 en Calama, Chile. La empresa minera CODELCO implementó un sistema de monitoreo de somnolencia para reducir la cantidad de accidentes ocasionados por el cansancio de los operadores de maquinaria pesada.

Al implementar la empresa su sistema de prevención, se lograron mitigar de gran manera los siniestros, pero muchos de los choferes se sintieron invadidos en su privacidad.

1.1.3. Bolivia

En Septiembre de 2011, en la ciudad de La Paz, el Comité de Industria, Comercio, Transporte y Turismo, en coordinación con el Instituto de Investigación de Seguridad Vial de la Universidad Mayor de San Andrés (UMSA) organizaron el seminario “Incidencia del Sueño en los accidentes de tránsito (sueño blanco), elementos para una normativa”, con el fin de conocer los aspectos fundamentales como tercera causa más frecuente de los accidentes en carreteras.

El diputado Javier Santibáñez, secretario del Comité, explicó que el objetivo del evento era dilucidar la temática en la perspectiva de evitar todo tipo de accidentes en carreteras, que en primera instancia se deben al consumo de alcohol, que actualmente es sancionado por la Ley de Transportes; en segundo lugar por la invasión de carril; y en tercer lugar por el cansancio de los conductores.

El desarrollo del evento contó con destacados profesionales del área médica que explicaron temáticas como la Fisiología del Sueño y la Fatiga, como el Dr. José Luis San Miguel, del Instituto Boliviano de Biología de la Altura (IBBA).

El Sueño Blanco y los hechos de Tránsito en las Carreteras, fue expuesto por Eynar Padilla, director interino de la Patrulla Caminera; la Señalización y Medidas de Seguridad en Carreteras”, estuvo a cargo del Ing. Freddy Chui y Marcelo Castro, en representación de la Administradora Boliviana de Carreteras (ABC).

La temática de Contingencias, Riesgos y Consecuencias del Sueño Blanco en el Transporte Terrestre, estuvo a cargo del Dr. Jorge Ocampo Castelú, ex rector de la Universidad Mayor de San Andrés (UMSA).

El mencionado evento se realizó con el objetivo de realizar una normativa acerca de la somnolencia y la fatiga de los conductores adecuada al contexto boliviano y apoyo en la prevención de muertes y heridos por accidentes de tránsito.

El 13 de agosto de 2014, se lanza el Decreto Supremo 2079, que aprueba el “Plan Nacional de Seguridad Vial 2014 - 2018”, para reducir la cantidad de mortalidad y traumatismos por siniestros automovilísticos.

1.2. DEFINICIÓN DEL PROBLEMA

En Bolivia, según el Instituto Nacional de Estadística (INE), el 53,5 % de los accidentes automovilísticos son ocasionados por la imprudencia del conductor.

La somnolencia sería una de las causas importantes de estos siniestros según un estudio realizado por estudiantes de la Universidad Mayor de San Andrés y el docente Jorge Ocampo Castelú, pues de 6800 accidentes reportados en carreteras durante el año 2010, más de 4000 casos, casi un 83% son atribuidos al sueño blanco (agotamiento y falta de descanso) de los conductores.

La fatiga laboral por la cantidad de horas frente al volante, también es otro factor importante dentro de la causas de siniestros automovilísticos, debido a que por necesidades económicas, algunos chóferes se ven obligados a no respetar horarios de relevo, trabajando incluso más de 24 de horas y sosteniéndose con el consumo de coca remojada en alcohol.

Los trastornos del sueño son otro tipo de factores de alto, por lo dificultoso que resulta el poder diagnosticarlos. Se sabe que muchas personas que los padecen pueden convivir con estos trastornos sin ser conscientes de ello.

No existe ninguna normativa o sistema en el país que regule o controle de alguna manera el sueño y la fatiga laboral de los conductores. En el caso de algún control de tránsito solo se observa que el conductor no se encuentre en estado de ebriedad, no exceda los límites de velocidad y que cuente con licencia de conducir, pero no existe ningún mecanismo que verifique la cantidad de horas que hubiese conducido el conductor, así también que haya cumplido las horas de descanso correspondientes para evitar problemas frente al volante.

Si bien la Policía Boliviana hace un control del estado de los conductores en las terminales de buses interdepartamentales, esta no cubre el desempeño del chofer en carreteras y no se toma en cuenta a los vehículos particulares.

Otro problema alarmante es la falta de auxilio inmediato a los heridos en caso de siniestros automovilísticos.

1.3. OBJETIVOS

1.3.1. Objetivo general

Diseño de un dispositivo electrónico de monitoreo, alerta de somnolencia y fatiga del conductor en un vehículo para evitar posibles accidentes de tránsito.

1.3.2. Objetivos específicos

- Monitorear la somnolencia y fatiga del conductor para poder alertar en caso de quedarse dormido frente al volante; mediante un dispositivo electrónico basado en una tarjeta de desarrollo *Raspberry PI 3B* y una cámara.
- Desarrollar un programa basado en un lenguaje de programación, para el análisis del comportamiento de los ojos del conductor, que permita alertar en caso de somnolencia.
- Incluir dentro del hardware una placa de circuito modular de comunicación que permita el envío de mensajes de texto (sms), para alertar a algún familiar del conductor, en caso de activación recurrente de la alarma de somnolencia.
- Controlar la cantidad de horas del conductor frente al volante, durante la conducción.
- Realizar pruebas de funcionamiento y obtener resultados para su posterior análisis.

1.4. LÍMITES Y ALCANCES

1.4.1. Límites

No existe documentación e información sobre estudios realizados acerca de la somnolencia, fatiga, sueño blanco y trastornos del sueño en el territorio boliviano, lo que dificulta tener referencias adecuadas al contexto nacional.

El hardware empleado para el diseño de este proyecto es material de investigación y desarrollo, esto por el costo que significarían algunos componentes dedicados.

Los conductores bolivianos no tienen hábitos de seguridad vial. Las autoridades de tránsito indican que los conductores se rehúsan a usar el cinturón de seguridad por comodidad o simplemente no saben cómo ponérselo. Por este motivo el uso de sensores de tipo “invasivo” que requieren ser conectados en el cuerpo de los conductores no se tomaron en cuenta en el diseño del dispositivo. Si bien un “Biosensor” puede coleccionar datos más precisos, no se garantiza que los conductores lo utilicen todo el tiempo, dejando sin utilidad al proyecto.

1.4.2. Alcances

Diseño de un dispositivo electrónico de monitoreo, alerta de la somnolencia y fatiga del conductor basado en software *open source* y tarjeta de desarrollo, que apoye a la prevención de accidentes de tránsito ocasionados por el sueño de los conductores.

El dispositivo deberá funcionar en cualquier momento del día, entonces la intensidad de la luz no debe ser un factor preponderante, pues el monitoreo debe ser en tiempo real.

El proyecto servirá de base para la elaboración de futuras soluciones tecnológicas a esta problemática.

1.5. JUSTIFICACIÓN

El cansancio y la somnolencia del conductor son causas importantes de accidentes automovilísticos. A pesar de algunas iniciativas realizadas por la carrera de Medicina de la Universidad Mayor de San Andrés, para concientizar a las autoridades sobre sus riesgos, continúa siendo un factor poco conocido y por lo tanto no controlado.

En Bolivia no existe la legislación adecuada para controlar esta problemática, pero el Código de Tránsito, específicamente el Decreto Supremo 2079 que aprueba el “Plan Nacional de Seguridad Vial 2014 - 2018”, tiene como objetivo reducir la cantidad de mortalidad, traumatismos por siniestros automovilísticos y ofrecer auxilio inmediato a los heridos en caso de accidente.

Se propone entonces una solución desde el área de la electrónica, con el diseño de un dispositivo que colabore con la prevención de accidentes automovilísticos ocasionados por la somnolencia y fatiga de los conductores.

Este dispositivo permitiría monitorear la cantidad de horas que lleva el conductor frente al volante, sobre todo durante viajes o tiempos extensos de conducción.

Este dispositivo también alertará y recomendará al conductor que debe hacer una parada de descanso, al momento de presentar conductas de somnolencia y fatiga durante la conducción, donde factores como el clima y capacidad del conductor, salen de los controles de la policía.

Se enviará un mensaje de texto a algún familiar del conductor, en caso de activación recurrente de la alarma de somnolencia, para que esté pueda estar al tanto de la seguridad del conductor, brindando auxilio inmediato a heridos en caso de posible siniestro.

Se requieren de elementos electrónicos que puedan brindar estabilidad y la capacidad necesaria para el procesamiento de imágenes en tiempo real, que puedan cubrir las funcionalidades requeridas en el proyecto.

Un computador puede realizar el procesamiento de datos en tiempo real, pero el dispositivo debe ser portátil, pequeño, que permita la comodidad del conductor y sobretodo que no sea costoso.

Entonces se eligió como base del proyecto el uso de un ordenador de placa reducida *Raspberry PI 3B*, por el precio accesible y la licencia libre con la que cuenta esta tarjeta de desarrollo.

Para apoyar con la reducción de consecuencias trágicas, sociales y económicas que traen los accidentes de tránsito.



CAPÍTULO II

MARCO TEÓRICO

El presente capítulo está dedicado a mostrar los factores de análisis y las tecnologías empleadas en el diseño de este proyecto.

El dispositivo utiliza una cámara de video (transductor fotoeléctrico) con visión nocturna, que con la ayuda de la librería de visión por computador de código abierto *OpenCV*, cumplen la función de sensor “no invasivo”, para la toma de datos.

La visión artificial es detecta el iris de los ojos del conductor y con el procesamiento digital de señales en tiempo real que realiza la tarjeta de desarrollo *Raspberry PI 3B*, puede monitorearse el comportamiento del conductor y alertarlo de forma oportuna en caso de somnolencia mediante un aviso sonoro proporcionado por un amplificador de audio de baja potencia.

La activación recurrente puede ser considerada situación de peligro, el dispositivo cuenta con un módulo de comunicación GSM que puede enviar mensajes de texto a algún familiar del conductor, lo que permitiría brindar el apoyo oportuno de personas que se encuentren fuera del vehículo.

2.1. CAUSAS DE ACCIDENTES AUTOMOVILÍSTICOS EN BOLIVIA

Los accidentes de tránsito pueden ocasionarse por fallas mecánicas, mal estado de las carreteras, factores climáticos y por imprudencia del conductor.

Según el “Informe sobre la situación mundial de la seguridad vial 2015” de la Organización mundial de la salud (OMS), todos los años casi 1,3 millones de personas

pierden la vida a consecuencia de los traumatismos causados por los accidentes de tránsito y sería la principal causa de muerte entre personas de edades comprendidas entre los 15 y los 29 años.

En Bolivia, según un estudio realizado por el Instituto Nacional de Estadística (INE) en el año 2015 del procesamiento de los registros administrativos de la Policía Nacional en la gestión 2014, la cantidad de accidentes de tránsito ocasionados por la imprudencia del conductor es del 53,3 % del total de siniestros, solo el 2,0% a fallas mecánicas y un 2,1 % al mal estado del camino y otro porcentaje similar en cuanto a la mala señalización.

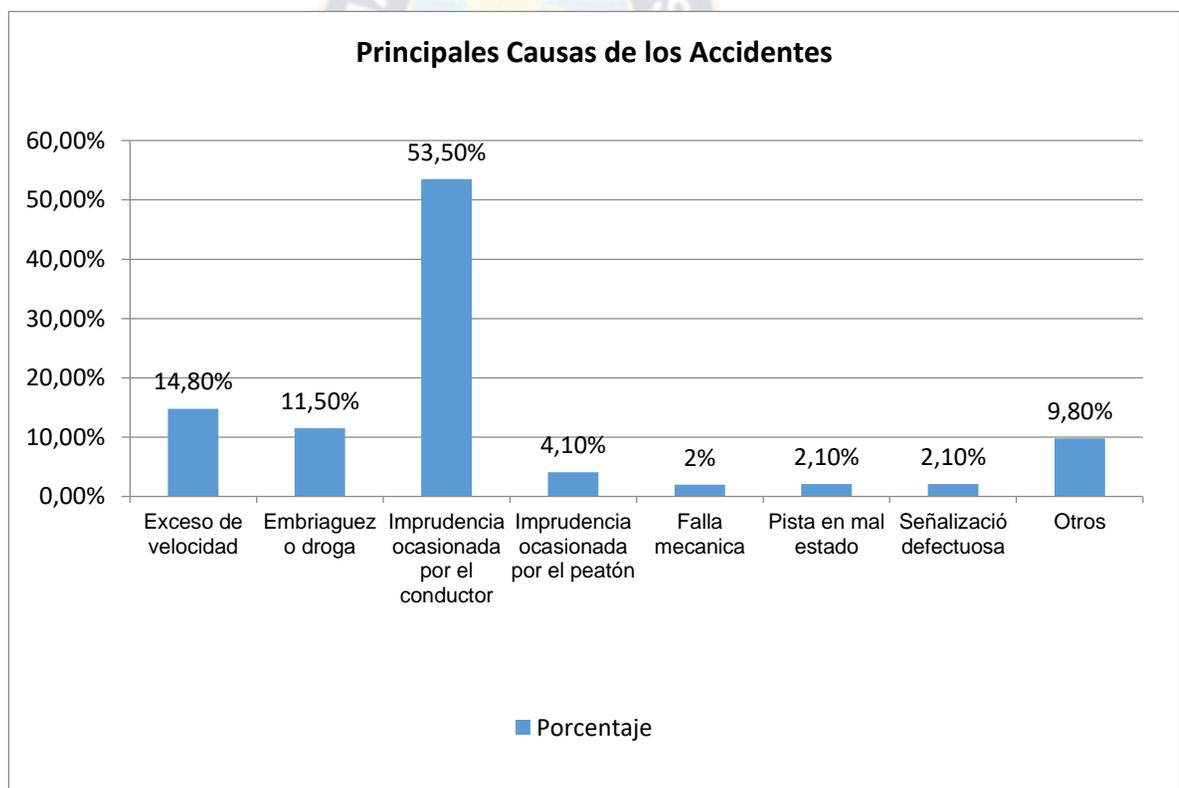


Tabla 1. Porcentaje de accidentes automovilísticos 2014 Bolivia datos INE y Policía Boliviana
Recuperado de www.ine.gob.bo.

Otro porcentaje que sería considerado segunda causa dentro de este estudio, es el exceso de velocidad con un 14, 8 %, seguido de un 11, 5% por embriaguez o consumo de estupefacientes, un 4,1 acerca de la imprudencia del peatón y un 9,8% ocasionado por otros factores como el exceso de carga.

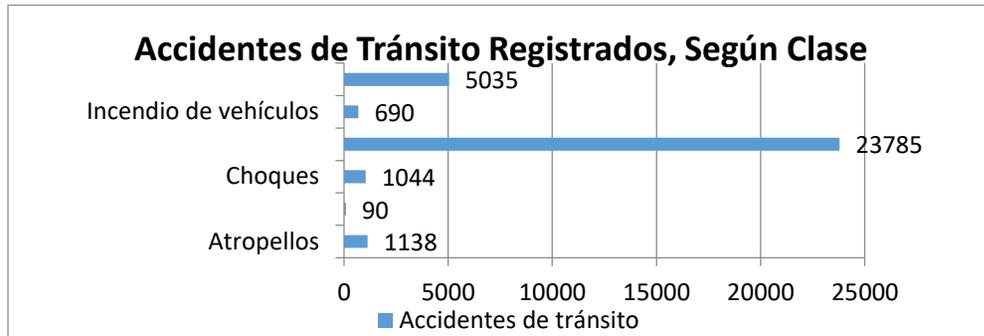


Tabla 2. Accidentes de tránsito registrados según clase, 2014 Bolivia datos INE y Policía Boliviana. Recuperado de ine.gob.bo

Otro dato preocupante dentro de estas estadísticas es que del total (31.782) de accidentes de tránsito registrados el año 2014, 23.785 fueron choques y 5.035 atropellamientos.

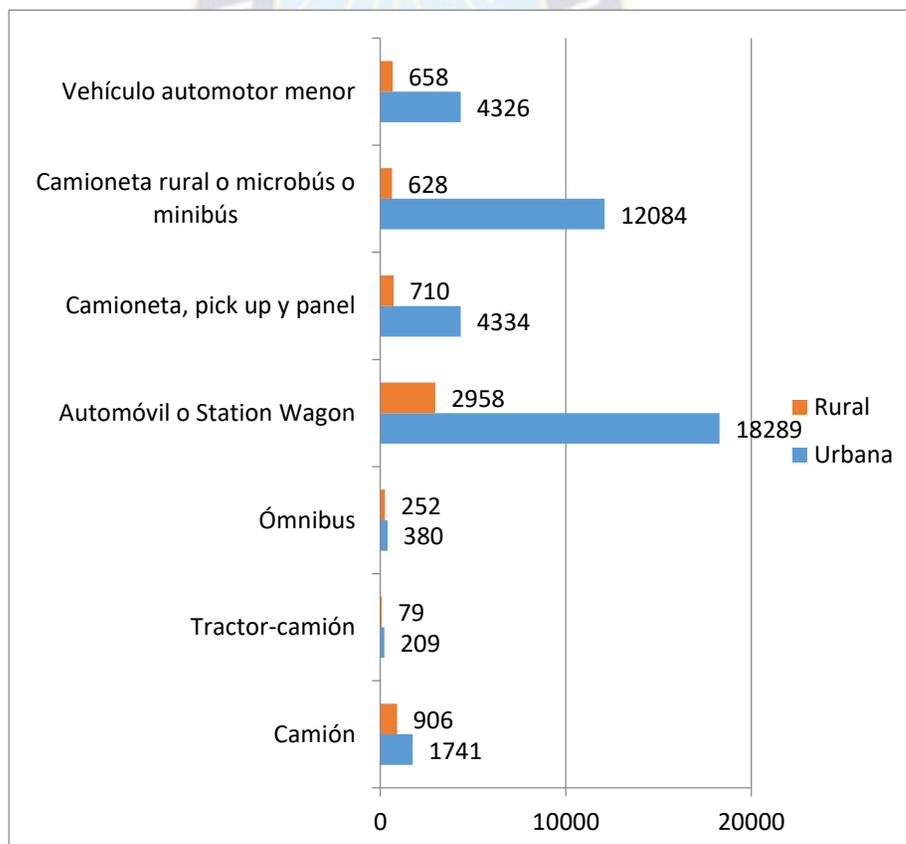


Tabla 3. Accidentes de tránsito, según tipo de vehículo y área 2014, Bolivia datos INE y Policía Boliviana. Recuperado de www.ine.gob.bo

De esos resultados 47.557 accidentes de tránsito fueron ocasionados en su mayoría por vehículos, de ese total 41.363 eventos ocurrieron en el área urbana y 6.191, en el área rural.

Las furgonetas son protagonistas de 18.289 casos en el área urbana, con 2958 en el área rural, y los microbuses o minibuses, el segundo número mayor de siniestros, con 12.084 accidentes en el área urbana y 628 en el área rural.

El departamento de La Paz, sería la población con más registros de casos con 34,1%, seguido de Santa Cruz con 31,4%.

DISTRIBUCIÓN DE ACCIDENTES DE TRÁNSITO POR DEPARTAMENTO					
DEPARTAMENTO	2010	2011	2012	2013	2014
Chuquisaca	5,1%	5,8%	6,0%	5,9%	5,8%
La Paz	40,4%	44,9%	37,7%	36,3%	34,1%
Cochabamba	14,0%	9,4%	10,3%	8,7%	8,0%
Oruro	5,7%	4,0%	5,3%	5,2%	5,3%
Potosí	3,3%	3,6%	4,1%	4,9%	4,3%
Tarija	8,1%	7,8%	7,6%	7,5%	6,8%
Santa Cruz	17,7%	19,8%	24,6%	25,8%	31,4%
Beni	4,1%	3,2%	3,6%	4,5%	3,4%
Pando	1,6%	1,5%	0,9%	1,1%	0,9%

Tabla 4. Distribución de accidentes de tránsito por departamento, datos INE y Policía Boliviana.
Recuperado de www.ine.gob.bo

A pesar de los datos extendidos por el INE, elaborados con los registros de la policía Boliviana, notamos que todavía no se considera a la fatiga y la somnolencia como causante directo de la mayoría de los siniestros.

Tampoco se tiene una estadística específica de la cantidad de casos con relación a esta problemática, y se desconoce el tema de los trastornos del sueño como ocasionantes de siniestros automovilísticos.

El nuevo código de tránsito, resalta la importancia del relevamiento de chofer en tramos mayores a los 300 km, lo que a una velocidad de 50km/hora, daría un aproximado de 3

horas al volante, pero no se hace mención del estado físico mental del chofer, ni tampoco la cantidad de horas de descanso que este debe cumplir para desempeñarse de manera adecuada frente al volante. En el siguiente cuadro se mencionan algunos de los artículos del nuevo código de tránsito, que mencionan el relevamiento y el estado del conductor.

ARTÍCULOS QUE HACEN MENCIÓN AL RELEVAMIENTO DEL CONDUCTOR	
CÓDIGO DE TRANSITO¹	PROBLEMAS PERCIBIDOS
<p>Art. 20 Cap. 1 inc. g) está prohibido circular describiendo “eses”</p> <p>Art. 25 El conductor debe disponer del suficiente espacio y comodidad en la cabina de modo que pueda maniobrar el vehículo con seguridad.</p> <p>Art. 89 Cap. 2 inc. b) Un requisito indispensable para ser conductor es contar con buena salud y no estar impedido por defecto físico.</p> <p>Art 96 Cap. 3 Conducir con atención y los cuidados que requiera la seguridad de tránsito.</p> <p>Art 97. Está terminantemente prohibido conducir vehículos bajo el efecto de drogas u otros intoxicantes, en estado de embriaguez o cuando las condiciones de salud físico-mentales no permitan la normal y segura conducción.</p> <p>Art. 163 Cap. 4 Los propietarios o empresas de transportes son responsables directos de los daños y perjuicios.</p> <p>Inc. d) Si obligan al conductor a trabajar excediendo su capacidad física o cuando éste no se encuentre en condiciones normales de salud.</p> <p>Art. 282 y 286 (conductor de relevo) en recorridos mayores a 300 km se debe contar obligatoriamente con un chofer de relevo y está completamente prohibido que un chofer exceda su jornada laboral en esta actividad. Ya sea para viajes interprovinciales y también interdepartamentales.</p>	<p>No se toma en cuenta la somnolencia de los conductores, como causa de accidentes de tránsito, por ese motivo no existen estadísticas sobre este tema.</p> <p>Si bien el código de transito exige la relevancia de chofer en tramos mayores a 300km.</p> <p>No se contemplan trastornos del sueño dentro de los exámenes de salud para la extensión de las licencias de conducir. Tampoco se hace un control del estado físico del conductor en relación a la fatiga laboral, tampoco no se le preguntan cuántas horas hubiese descansado. Tampoco existe algún método de verificación en estos casos.</p> <p>Y a pesar del control de relevos que se hace en los tramos interdepartamentales, en muchas ocasiones estos suelen ser burlados por algunos choferes y empresas de transporte.</p> <p>Existe un control en las terminales de buses Interdepartamentales. Acerca de los relevos y estado de los conductores.</p> <p>Pero no existe un control en rutas interprovinciales, por diversos factores que salen del control policial. Tampoco existe dicho control en cuanto a vehículos particulares.</p>

Tabla 5. Artículos del Código de transito que hacen mención al relevamiento del conductor y problemas percibidos. Elaboración propia

¹ Decreto Ley No 10135

ELEVADA A RANGO DE LEY EL DECRETO 10135, DE 16 DE FEBRERO DE 1973, CODIGO DE TRANSITO

Ley No 3988 del 18 de diciembre de 2008

APRUEBA MECANISMOS DE CONTROL, FISCALIZACIÓN Y SEGURIDAD VIAL, EN EL SECTOR DE TRANSPORTE AUTOMOTOR PÚBLICO TERRESTRE DE PASAJEROS, PARA DISMINUIR EL RIESGO DE ACCIDENTES DE TRÁNSITO EN LAS CARRETERAS Y CAMINOS DEL ESTADO PLURINACIONAL DE BOLIVIA, ESTABLECIENDO LAS INFRACCIONES Y SANCIONES ANTE EL INCUMPLIMIENTO DE LOS MISMOS.

D.S. No 0420

Del 13 de febrero de 2010

Habiendo reportado un vacío legal, un grupo de estudiantes de la carrera de medicina y el docente Jorge Ocampo Castelú de Universidad Mayor de San Andrés, realizaron un estudio sobre esta problemática, brindando datos alarmantes que debían llamar a la reflexión de las autoridades.

De 6800 accidentes reportados en carreteras durante el año 2010, más de 4000 casos, casi un 83% son atribuidos al sueño blanco (Agotamiento y falta de descanso) de los conductores.

Los controles que realiza la Policía Boliviana, acerca del estado del chofer se hacen en las terminales interdepartamentales, precisamente por ser tramos mayores a los 300km, donde se le exige relevo en los viajes de partida, descanso en las llegadas y no estar en estado de embriaguez.

DISTANCIAS ENTRE LA CIUDAD DE LA PAZ Y LOS DIFERENTES DEPARTAMENTOS	
RUTAS DESDE LA PAZ	DISTANCIA (Km)
Oruro	227
Cochabamba	382
Santa cruz	857
Potosí	553
Sucre	714
Tarija	919
Villazón	917

Tabla 6. Distancias entre la ciudad de La Paz y los diferentes departamentos.
Elaboración propia

Según el Cnel. Rómulo Cáceres en un artículo elaborado por Alvaro Luksic en octubre del 2011, muchos choferes que realizan el viaje a tramos largos optan por “venticuatrear” (trabajar más de 24 horas), sosteniéndose con el consumo de coca remojada en alcohol.

Dichos controles no están enfocados al transporte interprovincial y mucho menos al transporte particular, algo que revela un gran vacío tanto legal como de seguridad.

El pestañear y adormitar durante la conducción expresan un nivel extremo de deuda de sueño, un conductor somnoliento disminuye progresivamente su capacidad de atención, concentración y pierde capacidad de respuesta oportuna ante condiciones que exigen reacciones inmediatas.

2.2. ELEMENTOS DE LA CONDUCCIÓN

Los elementos importantes, para el desempeño de la conducción o en su defecto accidentes automovilísticos son los siguientes:

- El conductor.
- El vehículo.
- Las vías.
- El entorno.

2.2.1. El conductor

Es el factor más importante porque todos los anteriores factores convergen a él, además es el causante del mayor número de accidentes automovilísticos.

Las causas conocidas por este factor son la imprudencia al conducir, la falta de hábitos de seguridad vial, el exceso de velocidad, la conducción en estado de ebriedad, fatiga o somnolencia.



Figura 1. Accidente en la carretera de los Yungas.
Recuperado de www.la-razon.com

2.2.1.1. Estado físico y mental del conductor

La persona que conduce debe encontrarse en condiciones físicas y psicológicas estables, además de contar con los conocimientos teóricos y prácticos sobre la conducción y que permiten adquirir la licencia de conducir.

La persona que conduce debe poder actuar de forma oportuna y adecuada ante las diferentes circunstancias que conlleva la conducción.

2.2.1.2. El sueño

Dormir es un proceso necesario para restablecer las funciones cerebrales y el mismo organismo para afrontar la posterior etapa de vigilia.

Durante el día, mientras una persona se encuentre despierta, sus neuronas establecen conexiones entre ellas como resultado de sus experiencias. Si la persona ha tenido un pequeño accidente, como la quemadura con una sartén caliente, su cerebro asociará el concepto de sartén con el dolor de la quemadura, creando un recuerdo y una respuesta que evite en lo posible que vuelva a ocurrir, este proceso ocurre millones de veces al día.

Durante el sueño, el cerebro se dedica a podar estas conexiones, eliminando aquellas que no son importantes, y reforzando las que sí lo son.

Los sueños parecen ser parte del proceso, como una especie de simulador donde se ensayan distintas respuestas, especialmente emocionales, a las experiencias adquiridas.

Nuevos descubrimientos indican que además se produce un proceso de "lavado" durante la noche en el que el cerebro elimina subproductos neurotóxicos que se acumulan durante las horas de vigilia.

Según la Organización Mundial de la Salud (OMS), un buen descanso implica ocho horas de sueño ininterrumpidos, sin embargo una persona común en promedio completa tan solo cinco.

2.2.1.3. La somnolencia

La somnolencia es la necesidad de quedarse dormido y es considerada una necesidad fisiológica básica, tan importante como el hambre o la sed, que definen la supervivencia de la persona.

Las personas que sufren somnolencia diurna pueden quedarse dormidas en situaciones o momentos inapropiados, siendo la consecuencia de no haber dormido el tiempo suficiente.

Causas de la somnolencia

Las causas más frecuente en las personas por las que sufren somnolencia, son las siguientes:

- Presión arterial baja. La hipotensión severa puede causar somnolencia y fuerte pérdida de energía.
- Calor excesivo: Baja la tensión arterial y el tono vital produciendo sueño.

- Digestión pesada: La digestión demasiado lenta hace que el organismo necesite invertir mucha energía en ello y produce una bajada de la actividad y somnolencia para facilitarla.
- Anemia: El hierro bajo produce síntomas como desánimo, astenia y somnolencia.
- Efectos secundarios de medicamentos: Algunos antidepresivos, ansiolíticos, relajantes nerviosos o musculares, antiestamínicos, pueden causar somnolencia.
- Interrupciones del sueño: Las interrupciones del sueño aún cortas o un descanso poco profundo causan somnolencia.
- Trastornos del sueño: Muchas personas pueden sufrir diferentes trastornos del sueño y no ser conscientes de ello, lo que ocasionaría serios problemas de desempeño en las horas de vigilia.

Las causas de somnolencia en conductores de vehículos son las siguientes:

- Privación aguda del sueño.
- Conducción durante la noche.
- Conducción en horario vespertino.
- Conducir durante jornadas prolongadas sin el descanso necesario.
- Conducción luego de una resaca.

2.2.1.4. Trastornos del sueño

Los trastornos del sueño no son considerados como un factor a la hora de conducir, pero se ha comprobado que tienen grandes consecuencias en la vida diaria, sus características son agotamiento físico, bajo rendimiento, sueño diurno, etc.

Muchas personas pueden vivir con trastornos del sueño sin saberlo, por lo inaccesibles que serían los estudios para diagnosticarlos.

Estos trastornos son:

- Insomnio
- Hipersomnia
- Narcolepsia
- Parasomnias
- Apnea del sueño

2.2.1.4.1. Insomnio

Es la incapacidad para conciliar el sueño o permanecer dormido. En los pacientes que lo sufren, la duración y la calidad del sueño son bajas e interfiere en su vida cotidiana.

Tipos de insomnio

Insomnio transitorio

Dura menos de tres semanas y en su origen intervienen numerosos factores que generalmente pueden ser modificados, como los ambientales y relacionados con el estilo de vida, ciertas enfermedades y los fármacos con los que se tratan.

Insomnio Crónico

Supera las tres semanas de duración. Puede ser percibido como una patología que interfiere en la actividad diaria del enfermo con graves consecuencias físicas y psíquicas.

Causas

Cambios fisiológicos

El envejecimiento produce cambios en el patrón del sueño. En las personas mayores es frecuente la reducción de las horas y la calidad del sueño y un aumento de la somnolencia diurna.

Estilo de vida

Los cambios constantes de horario, bien por cuestiones laborales o por los viajes provocan trastornos del ritmo circadiano.

Fármacos

Todos los medicamentos y sustancias que pueden alterar el sueño.

2.2.1.4.2. Hipersomnia

Es la somnolencia excesiva durante el día o excesivamente larga, o largos períodos de sueño durante la noche y puede estar asociada con dificultades en el despertar.

Las personas que padecen de hipersomnia pueden dormirse en cualquier momento y situación, como en el trabajo o mientras conducen.

Los grupos más afectados por la hipersomnia son adolescentes, adultos jóvenes y personas con sobrepeso.

Tipos de hipersomnia

Hipersomnia Idiopática

Es la excesiva somnolencia durante el día durante un largo período de tiempo.

Hipersomnia Recurrente

Implica períodos de somnolencia diurna excesiva, que puede durar de uno a varios días, y se repiten a lo largo de un año o más.

Causas

- Disfunción del sistema nervioso autónomo, o abuso de drogas o de alcohol.

- Un problema físico, tal como un tumor, trauma en la cabeza, o lesión del sistema nervioso central.
- No dormir lo suficiente por la noche (falta de sueño).
- El exceso de peso.
- Alcoholismo o drogadicción.
- Una lesión en la cabeza o una enfermedad neurológica, como la esclerosis múltiple.
- Los medicamentos recetados, como los tranquilizantes.
- Genética (tener un familiar con hipersomnia).

2.2.1.4.3. Narcolepsia

La narcolepsia es un trastorno neurológico crónico causado por la incapacidad cerebral de regular normalmente los ciclos de sueño y despertar. En varios momentos del día, las personas con narcolepsia experimentan impulsos fugaces de dormir. Si el impulso se vuelve abrumador, los pacientes se quedan dormidos durante períodos que duran desde unos segundos a varios minutos. En casos raros, algunas personas pueden permanecer dormidas durante una hora o más.

Además de la somnolencia diurna, tres otros síntomas principales caracterizan frecuentemente a la narcolepsia: cataplejía o la pérdida súbita del tono muscular voluntario, alucinaciones vívidas durante el inicio del sueño o al despertar y breves episodios de parálisis total al comienzo o al final del sueño.

Causas

Los científicos no han logrado aún determinar cuáles son las causas de este trastorno, sin embargo se han señalado algunos de los posibles desencadenantes de esta enfermedad:

- Estrés.

- Cambios súbitos del ritmo vigilia-sueño.
- Mononucleosis infecciosa.
- Traumatismos craneales.
- Asimismo, en algunos casos tiene un origen genético.

2.2.1.4.4. Parasomnias

Son trastornos de la conducta que se dan durante el sueño, que pueden ser de tipo breve y episódico.

Pueden conllevar interrupciones parciales del sueño, pero no suponen una alteración de los mecanismos que regulan el sueño y la vigilia propiamente. Suelen darse en las transiciones entre el sueño y la vigilia, o viceversa, o en ciertas fases del sueño.

Las parasomnias más frecuentes son el sonambulismo y los terrores nocturnos, ambas son expresiones del mismo mecanismo fisiopatológico de base, se incluyen en los trastornos del despertar.

Las pesadillas son otra forma clínica de parasomnia que provocan en cierta medida una alteración del sueño.

2.2.1.4.5. Apnea del sueño

La apnea obstructiva del sueño es una patología respiratoria que se caracteriza porque las personas que lo padecen roncan habitualmente, tienen paradas de la respiración (apneas) de forma repetida y somnolencia durante el día. Como consecuencia, los pacientes que la padecen no pueden descansar bien durante las horas de sueño.

Causas

- Tener el maxilar inferior más corto que el maxilar superior.
- Determinadas formas del paladar o de la vía respiratoria que originan el colapso de la vía.

- Tener un cuello grande.
- Poseer una lengua que puede retrotraerse hasta bloquear la vía respiratoria.
- La obesidad.
- Tener unas amígdalas o vegetaciones grandes que pueden taponar la vía respiratoria.

2.2.2. El vehículo

Uno de los elementos considerado “clave” para los fabricantes de automóviles, que han trabajado durante años para conseguir la mejora tecnológica de sus vehículos en materia de seguridad vial.

Por ese mismo motivo los accidentes causados por fallas mecánicas están en un porcentaje bajo.



Figura 2. Ingreso de Vehículos a Bolivia.
Recuperado de www.emol.com.

En nuestro país y por la idiosincrasia de la sociedad boliviana, es difícil poder acceder a vehículos de casa o “0 km”, a pesar de que están imponiéndose en el mercado de los

vehículos de procedencia China, por este motivo la Policía Bolivia, se encarga de controlar el estado mecánico de los vehículos anualmente. Para obligar a sus conductores a realizar los mantenimientos necesarios y poder reducir de alguna manera las consecuencias de los siniestros automovilísticos.

2.2.2.1. Sistemas de seguridad existentes en el vehículo

El automóvil debe ser siempre lo más seguro para el conductor, así se fue priorizando desde la fabricación masiva de vehículos.

La seguridad vehicular se dividió en dos segmentos que son la seguridad activa y pasiva.

2.2.2.1.1. Seguridad activa

Es el conjunto de los elementos que contribuyen a proporcionar una mayor eficacia y estabilidad al vehículo en marcha.

Sistema de antibloqueo de frenos (ABS)

Previene que las ruedas se bloqueen y asegura una frenada segura, el conductor sigue manteniendo el control sobre el vehículo, el coche permanece estable y la distancia de frenada se reduce considerablemente.

El ABS permite realizar maniobras de esquiava al mismo tiempo que se está actuando con firmeza sobre el pedal de freno.

Control electrónico de estabilidad (ESP)

Actúa de forma automática para frenar de forma selectiva una de las cuatro ruedas del vehículo en situaciones de riesgo en las que se puede superar el límite de adherencia del neumático.

Sensor de presión de los neumáticos

Permite recordar una labor básica de mantenimiento que debe realizarse periódicamente; revisar las presiones de los neumáticos.

Servofreno de emergencia (BAS)

Es un sistema de asistencia de frenada que aumenta la presión realizada sobre el pedal para detener el coche.

Sistemas de frenado automático

Gracias a este dispositivo, los coches son capaces de activar los frenos sin la intervención por parte del conductor.

2.2.2.1.2. Seguridad pasiva

Es el conjunto de elementos que reducen al mínimo los daños que se pueden producir cuando el accidente es inevitable.

Cinturón de seguridad

Dispositivo que evita a los pasajeros salir disparados contra el parabrisas en caso de choque o contra las ventanas laterales en caso de volcar el coche.

Reposacabezas

Permite dirigir la energía del choque a la estructura del asiento. Evita riesgo de lesiones de la columna vertebral.

Airbags

Bolsa de aire que en caso de colisión, amortigua el impacto de los ocupantes contra el volante, panel de instrumentos y contra ventanas laterales en los delanteros y traseros. Este sistema permite reducir el riesgo de muerte en un 30%.

Pretensores

En caso de impacto, tensan al máximo los cinturones de seguridad para evitar cualquier holgura, con ello evitan que los pasajeros se desplacen hacia delante y choquen con el volante, salpicadero, etc.

Cristales

Los cristales del parabrisas están diseñados para que no salten astillas que dañen a los pasajeros, en caso de accidente. Las ventanillas laterales son más débiles, pueden romperse y son la salida más cómoda en caso de vuelco, si las puertas se bloquean.

2.2.2.1.3. Nuevos sistemas de seguridad en el vehículo

La seguridad del vehículo evolucionó de tal manera que en las estadísticas sobre accidentes ocasionados por fallas del automóvil se redujo de gran manera.

El conductor sería entonces el factor vulnerable a ocasionar siniestros automovilísticos, debido a que es propenso a sufrir eventos que salen de su control, como la fatiga y la somnolencia.

Volante que detecta el cansancio

El sistema de fatiga al volante creado por Hoffman y Krippner mide la presión ejercida por las manos del conductor para estudiar su comportamiento.

Compuesto por una serie de sensores alojados en la cara interna del volante, que son capaces de detectar si el conductor empieza a mostrar signos de somnolencia, pérdidas de consciencia o cualquier otra situación que ponga en riesgo su vida.

Una corriente eléctrica de baja intensidad circula por sus sensores, de manera que al mantener la presión se crea un cortocircuito. Cuando es insuficiente o nula, saltan las alertas del sistema para despertar o alertar del problema del conductor.

Cinturón de monitoreo

En España, investigadores del Instituto de Biomecánica de Valencia (IBV), desarrollado unos sensores de ritmo cardíaco y otros similares.

El proyecto es llamado *Harken*, consisten en el diseño de un cinturón de seguridad que mediante la variación del ritmo cardíaco y la respiración, factores que están relacionados con la fatiga, analiza el estado del conductor, activando una alerta en caso de quedarse dormido.

Cámara de monitoreo

Algunos sistema de monitoreo y alerta, trabajan con un sistema de reconocimiento facial, que mediante cámara digitales, permiten detectar síntomas de somnolencia, mediante el análisis de parpadeo, los bostezos y otras expresiones faciales que muestren cansancio y fatiga.

También pueden reconocer si el conductor está estresado, nervioso, colérico, estados emocionales que no permiten una conducción segura.

Las señales de alarma pueden ser luces rojas parpadeantes, mensajes de texto en el cuadro de instrumentos, e instándole a no distraerse, detenerse a descansar, dormir o calmarse.

Algunos vehículos ya vienen con este sistema incorporado de forma experimental, que dentro de unos años podría instalarse en todos los coches.

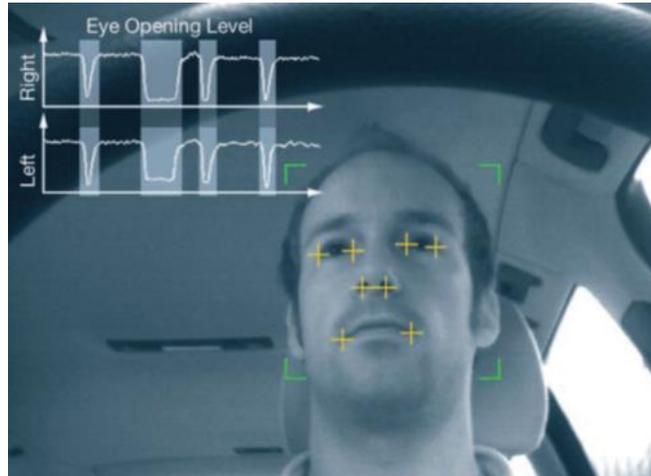


Figura 3. Sistema de reconocimiento facial.
Recuperado de www.centro-zaragoza.com

2.2.2. Las vías

Se puede evidenciar que existe una política de mejoramiento de carreteras en el país, pero que lamentablemente, por la inaccesibilidad de algunas poblaciones, muchas de nuestras vías siguen siendo precarias, factibles a accidentes y carentes de buena señalización y mantenimiento.



Figura 4. Accidente en la carretera de los Yungas.
Recuperado de www.tierraplus.com.bo

2.2.3. El entorno

Es un factor predeterminante y de mucho cuidado, porque sale de los controles policiales, afectando directamente al conductor y al vehículo. Estos son la visibilidad, la temperatura ambiente, la humedad, la lluvia, etc.

2.3. HERRAMIENTAS APLICADAS AL DESARROLLO DEL PROYECTO

2.3.1. Open source

También llamado “Código Abierto” es un término que se utiliza para denominar a cierto tipo de software y hardware que se distribuye mediante una licencia que le permite al usuario final, si tiene los conocimientos necesarios, utilizar el código fuente del programa para estudiarlo, modificarlo y realizar mejoras en el mismo.

2.3.2. Software open source

El *Software Open Source* (SOS) se distribuye bajo una licencia que permite a los usuarios copiar, modificar y redistribuir el software a otros.

El *Software Open Source* se caracteriza por:

- Un bajo costo de implementación y mantenimiento.
- Es un sistema estable con baja incidencia de errores críticos.
- Permitir que las aplicaciones sean copiadas y distribuidas para su mejora continua.
- No es susceptible a ningún virus informático común.

2.3.3. Raspbian

Raspbian es un sistema operativo diseñado para las tarjetas de desarrollo *Raspberry PI*, que está basado en Debian 8 (Jessie) de Linux, que carga por defecto la interfaz gráfica y el escritorio.

CARACTERÍSTICAS DEL SISTEMA OPERATIVO RASPBIAN	
Versión	septiembre de 2016
Fecha de lanzamiento	2016-09-23
Versión del kernel	4.4

Tabla 7. Características Raspbian.
Elaboración propia

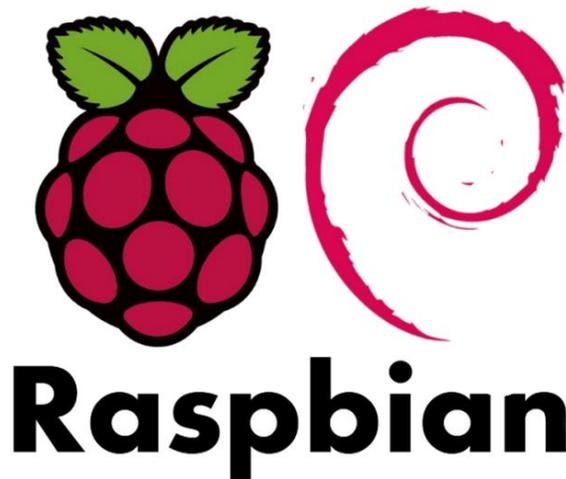


Figura 5. Logo de Raspbian.

Recuperado de blog.jorgeivanmeza.com

2.3.4. Python

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito. Es uno de los lenguajes de programación más empleados para el desarrollo de software.

Python puede ser utilizado en diversas plataformas y sistemas operativos, entre los que podemos destacar *Windows*, *Mac OS X* y *Linux*.

Con este lenguaje podemos desarrollar software para aplicaciones científicas, para comunicaciones de red, para aplicaciones de escritorio con interfaz gráfica de usuario (GUI), para la creación de juegos de *smartphones* y para aplicaciones web.

Python es un lenguaje potente, flexible, con una sintaxis clara y concisa, que no requiere dedicar tiempo a su compilación debido a que es interpretado.

Python es *open source*, cualquiera puede contribuir a su desarrollo y divulgación. No es necesario pagar ninguna licencia para distribuir *software* desarrollado con este lenguaje. Su intérprete también se distribuye de forma gratuita en diferentes plataformas.

CARACTERÍSTICAS DEL LENGUAJE DE PROGRAMACIÓN PYTHON	
Python Versión	2.7.9
TK Versión	8.6.2
IDLE Versión	2.7.9
Lenguaje interpretativo	Un intérprete, actúa de manera casi idéntica a un compilador, con la salvedad de que ejecuta el programa directamente, sin necesidad de generar previamente un ejecutable.
Lenguaje de alto nivel	Es de estructura sintáctica y semántica legible, acorde a las capacidades cognitivas humanas.
Indentación	La indentación es lo que la sangría al lenguaje humano escrito (a nivel formal). En <i>Python</i> la Identación es obligatoria.

Tabla 8. Características de Python.

Elaboración propia

Módulos a utilizar

CARACTERÍSTICAS DE LOS MÓDULOS A UTILIZAR DE PYTHON	
<i>Numpy</i>	Constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.
<i>Cv2</i>	Están son las librerías de <i>Open CV</i> para la visión computacional.
<i>Time</i>	Este módulo ofrece varias funciones relacionadas con el tiempo.
<i>Piserial</i>	Sirve para la comunicación serial.

Tabla 9. Módulos a utilizar de Python.

Elaboración propia



Figura 6. Python lenguaje Multiusuario y Multiplataforma.

Recuperado de geekyhour.blogspot.com

2.3.5. Open CV

OpenCV es una librería de visión por computador de código abierto, la librería está escrita en los lenguajes C y C++ y es compatible con *Linux*, *Windows* y *Mac OS X*.

Cuenta con un desarrollo activo en interfaces para *Python*, *Ruby*, *Matlab* y otros lenguajes.

Ha sido diseñado con un enfoque direccionado a aplicaciones en tiempo real, donde se requiere eficiencia en cuanto a gasto de recursos computacionales, pues está escrito y optimizado en C y puede tomar ventaja de los procesadores con múltiples núcleos.

Uno de los objetivos de *OpenCV* es proveer una infraestructura de visión por computador fácil de utilizar que ayuda a los programadores a desarrollar aplicaciones sofisticadas de CV (*Computer Vision*).

La librería *OpenCV* contiene aproximadamente 500 funciones que abarcan muchas áreas de visión computacional, incluyendo la inspección de productos de fábricas, escaneo médico, seguridad, interfaces de usuario, calibración de cámaras, robótica, entre muchas otras.

Tiene una librería completa de uso general de aprendizaje automático (MLL o *Machine Learning Library*), que es muy útil para cualquier problema de aprendizaje automático. Dicha sub-librería está especializada en el reconocimiento estadístico de patrones y algoritmos de agrupamiento.

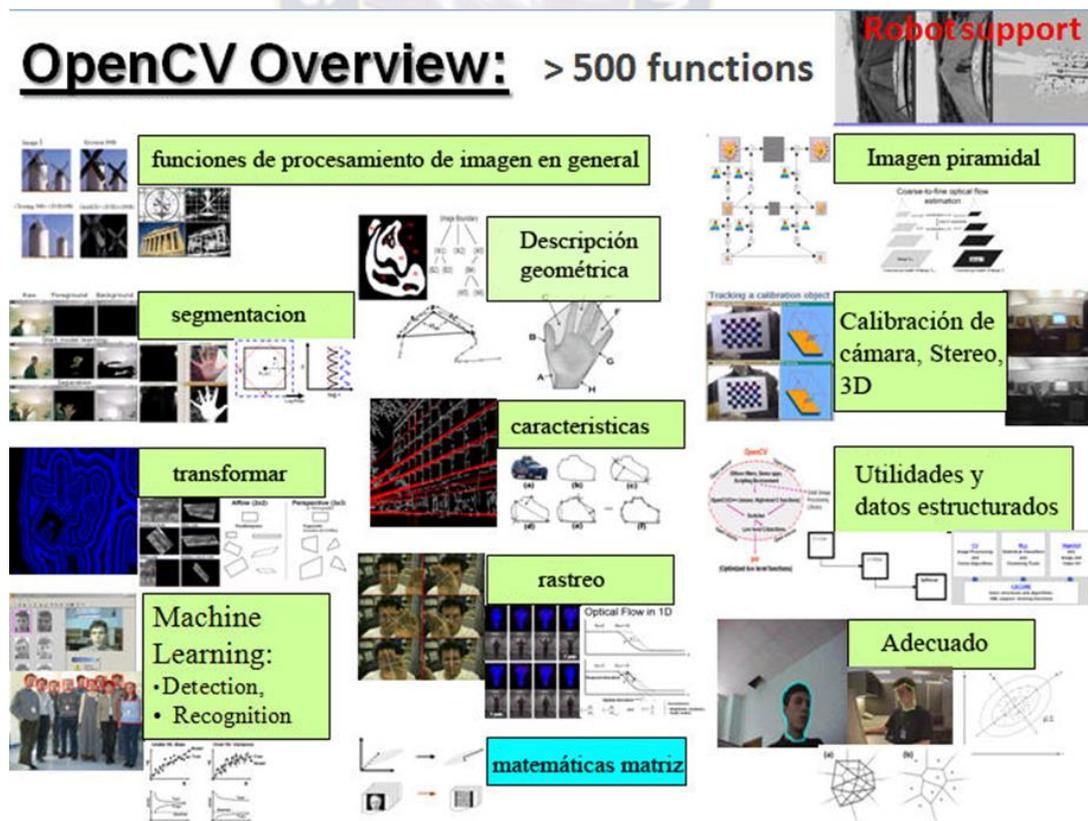


Figura 7. Funcionamiento de Open CV.
Recuperado de osl.uil.es

OpenCV tiene una estructura modular, siendo los principales:

Core

Módulo básico de *OpenCV*, incluye estructuras de datos básicas y las funciones básicas de procesamiento de imágenes. Este módulo también es usado por otros módulos como *highgui*.

Highgui

Módulo que provee interfaz de usuario, códec de imagen, vídeo y la capacidad para capturar imágenes y vídeo eventos del ratón.

Imgproc

Módulo que incluye algoritmos básicos de procesamiento de imágenes, incluyendo filtrado y transformado de imágenes.

Video

Módulo de análisis de vídeo que incluye algoritmos de seguimiento de objetos, entre otros.

Objdetect

Incluye algoritmos de detección y reconocimiento de objetos para objetos estándar.

CARACTERÍSTICAS DE LA LIBRERÍA OPEN CV	
Versión	3.1.0
Lenguajes que soporta	Python, Ruby, Matlab y otros lenguajes
plataformas	GNU/Linux, Mac OS X y Window

Tabla 10. Características de Open CV.
Elaboración propia

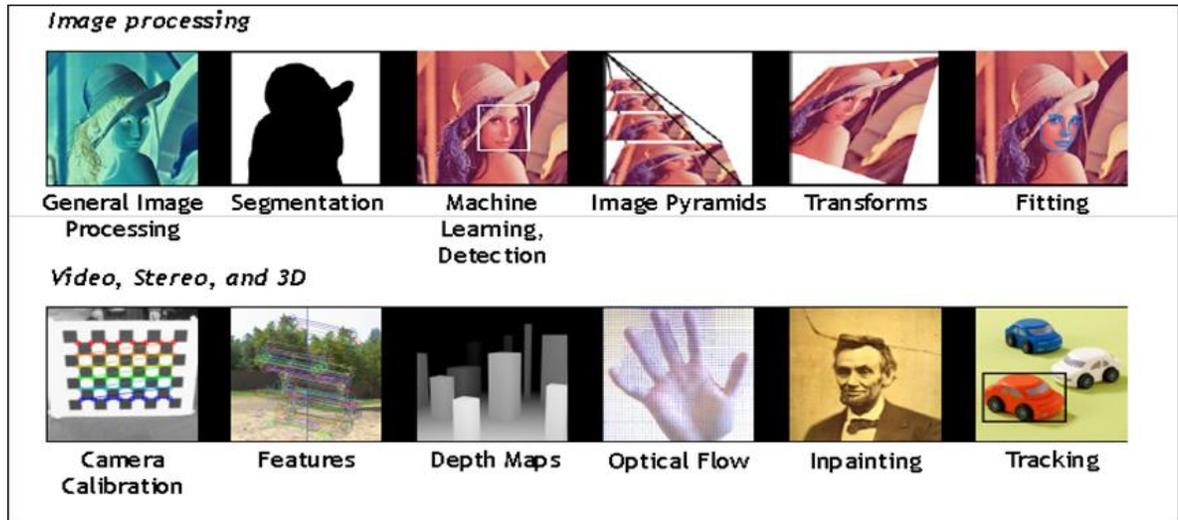


Figura 8. Categorías de Open CV.
Recuperado de osl.uull.es

Funciones de Open CV

Principales funciones que se utilizará:

FUNCIONES DE OPEN CV		
Función Open CV	Tipo	Descripción
cv2.imread()	Auxiliar	Leer una imagen desde una cámara o un archivo.
cv2.imread()	Auxiliar	Transformar una imagen de color a escala de grises u otros.
cv2.circle()	Auxiliar	Dibujar un círculo.
cv2.putText()	Auxiliar	Poner texto dentro de una imagen.
cv2.imshow()	Auxiliar	Mostrar una imagen desde <i>OpenCV</i> .
cv2.VideoCapture()	Auxiliar	Capturar vídeo desde una cámara o un archivo.
cv2.rectangle()	Auxiliar	Dibujar un rectángulo en una imagen.
cv2.waitKey()	Auxiliar	Esperar la pulsación de una determinada tecla para ejecutar una acción.
cv2.destroyAllWindows()	Auxiliar	Cerrar todas las ventanas iniciadas desde <i>OpenCV</i>
cv2.adaptiveThreshold()	Auxiliar	Realizar el balance de una imagen, compensando entre zonas oscuras y claras.
cv2.GaussianBlur()	Auxiliar	Filtrar la imagen usando un cálculo gaussiano del histograma de color o grises de la misma.
cv2.HoughCircles()	Auxiliar	Activar filtro para la detección de círculos

Tabla 11. Funciones de Open CV.

Elaboración propia

Cascadas

Librerías internas de *OpenCV* que facilitan la detección de objetos y/o personas:

LIBRERÍAS A UTILIZAR DE OPEN CV	
haarcascade_frontalcatface.xml	Detector de cara
haarcascade_eye.xml	Detector de ojos
haarcascade_eye_tree_eyeglasses.xml	Detector de ojos con lentes

Tabla 12. Librerías a utilizar de *Open CV*.

Elaboración propia

2.3.6. Hardware open source

Hardware de Fuentes Abiertas (OSHW en inglés) es un término para denominar artefactos tangibles (máquinas, dispositivos, u otros objetos del mundo físico) cuyo diseño ha sido publicado de forma tal que cualquier persona pueda fabricar, modificar, distribuir y usar esos objetos.

El *hardware* es diferente del código (*software*) en la forma que los recursos físicos se dedican a la creación de objetos físicos. De esta manera, personas o entidades que estuvieran produciendo objetos (productos) bajo una licencia OSHW tienen la obligación de no declarar que dichos objetos hayan sido manufacturados, vendidos, garantizados o autorizados en cualquier forma por el diseñador original. Además dichas personas o entidades no deberán usar ninguna de las marcas registradas en posesión del diseñador original.

2.3.7. Raspberry pi 3 modelo B

El pasado 29 de febrero de 2016, fue lanzado el modelo *Raspberry PI 3 B*, día en que el proyecto de *The Raspberry PI Foundation* cumplía 4 años desde que se lanzó el modelo original.

El *Raspberry PI 3B*, ensambla en su circuito un chipset Broadcom BCM2387 con cuatro núcleos ARM Cortex-A53 a 1.2 GHz. Dicho procesador es capaz de mover con soltura

videojuegos y aplicaciones, además dispone de una gran potencia de procesamiento para otro tipo de tareas.

La GPU encargada de los gráficos es la *Broadcom VideoCore IV*, una solución *Dual Core* compatible con *Open GL ES 2.0* y *OpenVG* que permite llegar a resoluciones *Full HD* con soltura.

La memoria RAM DDR2 de un GB que tiene, le permite mover con soltura bastantes sistemas operativos, como *Windows 10 IoT Core* y la distribución *Ubuntu Snappy*, versiones *Slim* de los sistemas operativos a los que emulan.

Dispone de cuatro puertos USB para ampliar el dispositivo con todo tipo de periféricos.

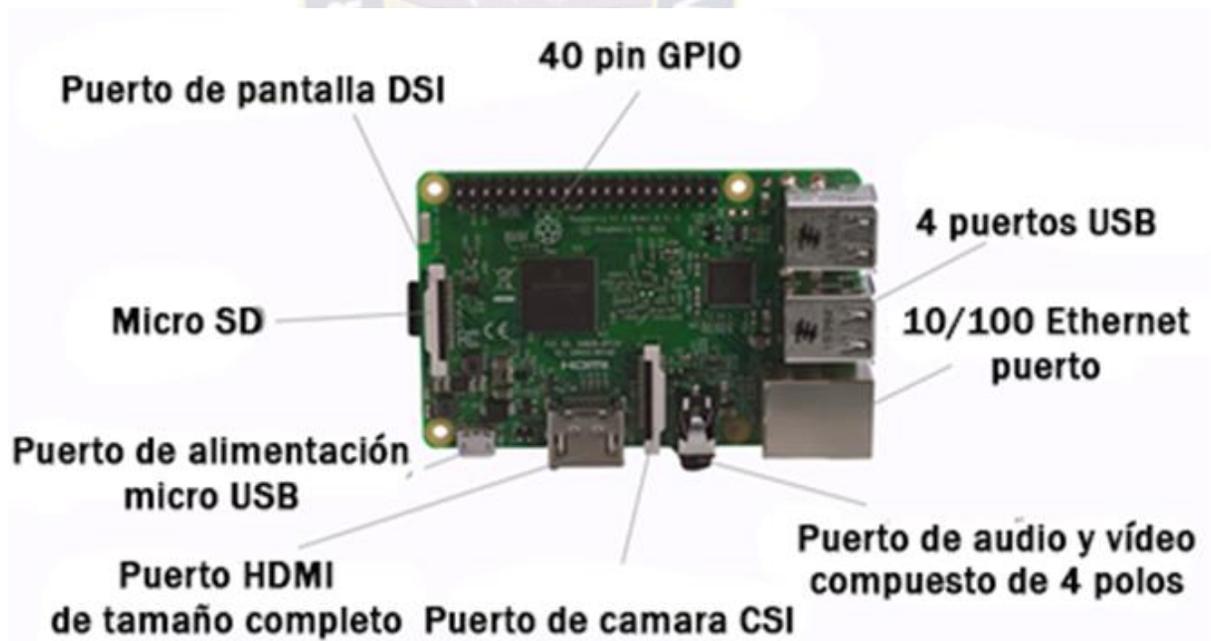


Figura 9. Raspberry PI 3 Modelo B.
Recuperado de thehackernews.com

CARACTERÍSTICAS TÉCNICAS RASPBERRY PI 3 B					
Nombre	Voltaje	Cantidad	Capacidad	Velocidad	Descripción
BCM2387	-	1	-	1,2 GHz	Microprocesador de 64 bits con cuatro núcleos ARM Cortex-A53
Dual Core VideoCore IV	-	1	-	-	Proporciona Open GL ES 2.0, OpenVG acelerado por hardware
RAM	-	1	1GB	-	Memoria RAM tipo DDR2.
Puerto Ethernet	-	1	-	-	Ethernet 10/100 BaseT
Wireless LAN	-	1	-	-	802.11n Wireless LAN (WI FI)
Bluetooth	-	1	-	-	Bluetooth 4.1 última versión
HDMI	-	1	-	-	Puerto full HDMI
Jack de 3,5 mm	-	1	-	-	Conector compuesto de audio y vídeo de 3,5 mm. analógico
Interfaz de la cámara (CSI).	-	1	-	-	Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2)
Interfaz de pantalla (DSI).	-	1	-	-	Pantalla de visualización Conector de la interfaz de serie (DSI) Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj
USB	5v	4	-	-	USB 2.0 donde se puede conectar el teclado mouse y otros elementos
Ranura para tarjetas microSD	-	1	-	-	Del tipo push-pull donde se considera un micro SD 10 capacidad mínima de 8 GB
GPIO	3,3v	40	-	-	Clavijas de 2,54 mm. Proporciona GND y VCC
Alimentación	5V	1	-	-	Conector tipo micro USB, corriente mínima de funcionamiento de 2 Amp.

Tabla 13. Características técnicas Raspberry PI 3 B.

Elaboración propia

2.3.8. Cámara de visión nocturna compatible con Raspberry Pi

Esta cámara es compatible con todas las versiones de *Raspberry Pi* y tiene un modo de visión nocturna infrarroja.

El anillo del foco se puede ajustar según la posición de objetos, cuenta con buena resolución incluso en ambientes oscuros.

Los fotoresistores y la resistencia ajustable pueden detectar la intensidad de luz ambiental y ajustar el umbral del interruptor de luz infrarroja automáticamente.

Sus características son:

CARACTERÍSTICAS TÉCNICAS CÁMARA DE VISIÓN NOCTURNA	
Características (Cámara)	
Cámara	5 mega píxeles
Dispositivo de carga acoplada (CCD)	1/2.7 pulgadas
Longitud Focal	3.6 mm
Resolución máxima	1080 p.
Orificios para tornillos	4
Bus de comunicación	CSI
Características (Leds IR)	
Tamaño	850 nm
Potencia	1w
Orificios para tornillos	2

Tabla 14. Características técnicas Cámara de visión nocturna.

Elaboración propia

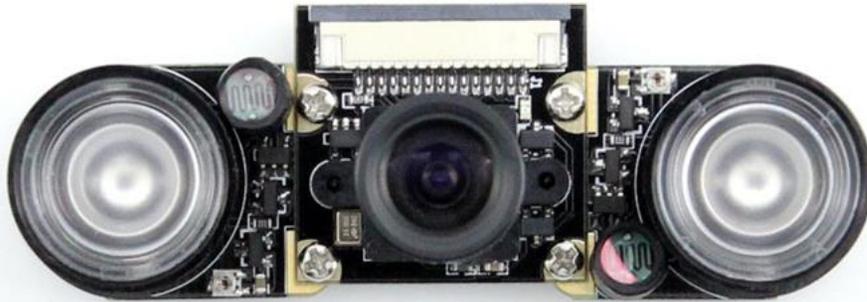


Figura 10. Cámara visión nocturna.
Recuperado de es.aliexpress.com

2.3.9. Modem E303

El dispositivo USB HUAWEI E303 HSPA es un modem de acceso inalámbrico a internet de alta velocidad (HSP), bus universal en serie (USB). Es un terminal inalámbrico multimodo.

CARACTERÍSTICAS DEL MODEM HUAWEI E303	
Protocolo de acceso datos	HSUPA/HSDPA/UMTS/ EDGE/GPRS/GSM
Frecuencias de trabajo	1900/1800/900/850 MHz
Ecuilizador	HSPA
Servicios de descarga de datos (Downlink)	<ul style="list-style-type: none"> - HSPDA de 7.2 Mbit/s - UMTS PS de 384 kbits/s - EDGE de 236.8 kbits/s - GPRS de 85.6 kbits/s
Servicios de subida de datos (Uplink)	<ul style="list-style-type: none"> - HSUPA de 5.76 Mbits/s - EDGE de 236.8 kbit/s(opcional)
SMS interfaz	basado en CS/Packet Switched
ranura	USB estándar (Tipo A)
	Micro SD
Sistemas operativos	Windows 7/ Windows XP/ Vista, Mac OS y Linux

Tabla 15. Características del HUAWEI E303.
Recuperado de consumer.huawei.com



Figura 11. Modem HUAWEI E303.
Recuperado de consumer.huawei.com

2.3.10. Amplificador operacional TDA2822M

El circuito integrado TDA2822M es una versión compacta del CI TDA2822, que cuenta con un total de 16 pines. El TDA2822M es el mismo integrado pero en un encapsulado de 8 pines solamente, siendo el variante en el encapsulado de 16 pines los 4 pines que se emplean para la conexión a tierra (GND) y 5 pines que no tienen conexión alguna, solo están para completar el encapsulado DIP16. En el encapsulado TDA2822M se emplean todos los 8 pines.

Características

Son amplificadores de bajo voltaje, baja distorsión, bajo consumo en reposo y configuración de puente o estéreo. Esto lo hace una excelente opción para amplificadores pequeños y de bajo consumo, sobre todo para dispositivos portátiles o de baja potencia. El TDA2822M es de más baja potencia pero al mismo tiempo permite ensambles y circuitos más compactos.

Especificaciones técnicas

- El voltaje mínimo para el TDA2822M es de 1.8 V y el voltaje máximo es de 15 V.
- La corriente de pico máxima del TDA2822M es de 1 A.

- La disipación máxima de potencia a 50° C para el TDA2822M es de 1 W.
- La salida máxima de potencia por canal a 9V con bocinas de 8 Ω para el TDA2822M es de 1 W.

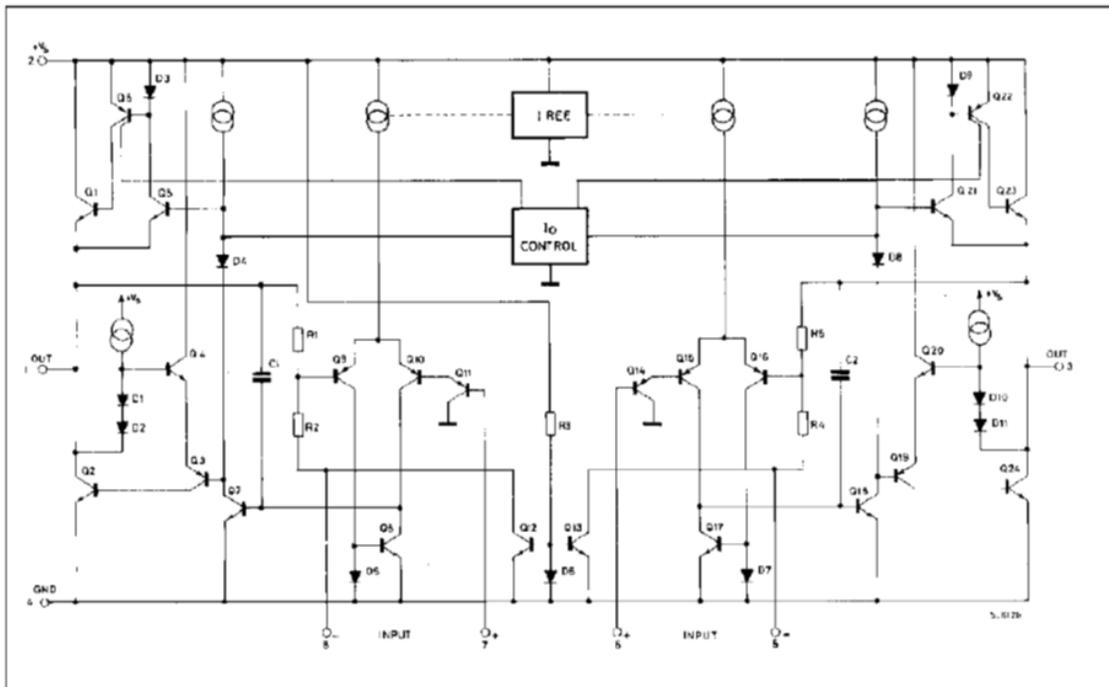


Figura 12. Diagrama esquemático del Amplificador Operacional TDA2822M
Recuperado de SGS-Thomson Microelectronics

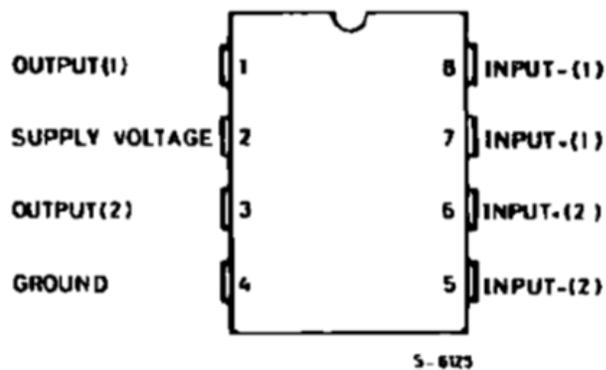


Figura 13. Configuración de los pin del TDA2822M
Recuperado de SGS-Thomson Microelectronics

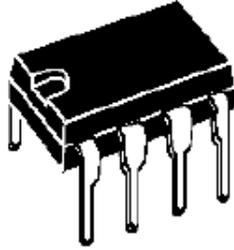


Figura 14. Encapsulado del TDA2822M.
Recuperado de SGS-Thomson Microelectronics

2.3.11. Visión computacional

La visión por computador o interpretación de imágenes, representa una sub-área de la Inteligencia Artificial que tiene como objetivo el análisis y la interpretación de la información visual. La comprensión de la imagen se considera como un proceso que parte en una imagen o secuencia de imágenes y termina en una descripción interna de la escena.

2.3.12. Algoritmo viola y jones

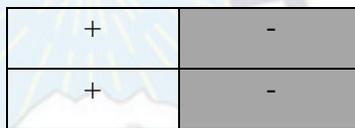
Es un método de detección de objetos que se caracteriza por utilizar bajos recursos computacionales, permitiendo así su empleo en tiempo real. Su desarrollo fue motivado por el problema de la detección de caras, que puede aplicarse a otras clases de objetos que estén caracterizados por patrones típicos de iluminación.

El algoritmo se basa en una serie de clasificadores débiles denominados *Haar-like features*, que se pueden calcular a partir de una imagen general de forma eficiente. La probabilidad de acierto de estos clasificadores en solitario es ligeramente superior a la del azar, por eso se agrupan en una cascada empleando un algoritmo de aprendizaje basado en *AdaBoost* para conseguir un alto rendimiento en la detección así como una alta capacidad discriminativa en las primeras etapas.

2.3.13. Haar-like

Son las características simples que se buscan en las imágenes, que consisten en la diferencia de intensidades luminosas entre regiones rectangulares adyacentes, su posición relativa a la ventana de búsqueda y adquieren un valor numérico resultado de la comparación que evalúan.

Se divide en regiones rectangulares, cada una de estas regiones son de dos tipos, denominadas positivas o negativas



+	-
+	-

Figura 15. Característica Haar.
Recuperado de oa.upm.es

Esta ventana compuesta por píxeles recorre la totalidad de la imagen, sumando cada pixel positivo y negativo, en su grupo respectivo, luego realiza una diferencia al que se le denomina valor de la característica. Se representa con la siguiente ecuación:

$$H(x, y) = \sum_p I(x, y) - \sum_n I(x, y)$$

Dónde:

$I(x, y)$ = Imagen a evaluar.

p = Región positiva.

n = Región Negativa.

$H(x, y)$ = Valor de la características Haar en el punto X y Y.

El método *Haar* se define por parámetros como el tamaño, la orientación y la distribución de las regiones positivas y negativas, estos parámetros van a depender de los rasgos del objeto a detectar, la distribución de intensidades de los píxeles que componen la imagen, tal como bordes, líneas o contornos.

Las cascadas utilizadas por Viola y Jones se componen por el número de regiones para un determinado uso:

DESCRIPCIÓN DE LOS HAAR DE VIOLA Y JONES		
Número de regiones	Orientación de detección	Descripción
Dos	Borde Horizontal: 0 grados Borde Vertical: 90 grados	Características de dos rectángulos cuyo valor es la diferencia entre las sumas de los píxeles contenidos en ambos rectángulos. Las regiones tienen la misma área, forma y son adyacentes.
Tres	Borde Horizontal: 0 grados Borde Vertical: 90 grados	Características de tres rectángulos que calculan la diferencia entre los rectángulos exteriores y el interior multiplicado por un peso para compensar la diferencia de áreas.
Cuatro	Detección líneas diagonales	Características de cuatro rectángulos que computan la diferencia entre pares diagonales de rectángulos.

Tabla 16. Descripción de los Haar de Viola y Jones.
Recuperado de idus.us.es

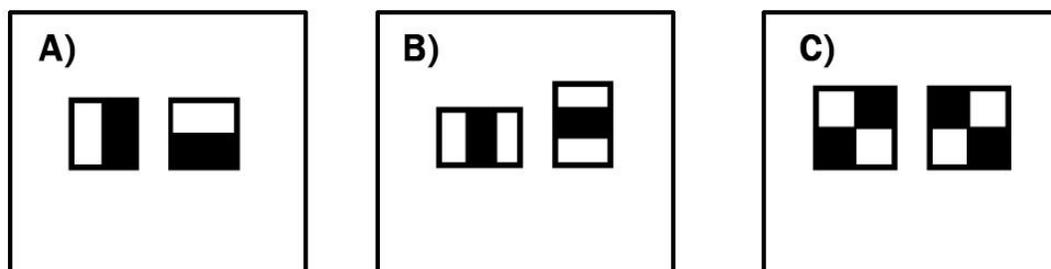


Figura 16. A) Características de dos B) Características de tres C) Características de Cuatro.
Recuperado de oa.upm.es

Después esta técnica de detección se perfecciono con *Lianhar* y *Maydt* en su trabajo², donde añadieron *Haar* inclinados para mejorar la detección de objetos en las imágenes, así con este nuevo conjunto consiguió detectar bordes y líneas de 45 grados.

DESCRIPCIÓN DE LOS HAAR DE LIANHAR Y MAYDT	
Número de regiones	Orientación de detección
Dos	Detectar bordes según la orientación: Borde diagonal: 45 grados
Tres	Detectar líneas según su orientación: Línea diagonal: 45 y 135 grados
Dos	Detectar centros-contornos

Tabla 17. Descripción de los Haar de Lianhar y Maydt.
Recuperado de oa.upm.es

² Lienhart, R. and Maydt, J. (2002) An Extended Set of Haar-Like Features for Rapid Object Detection. 2002 International Conference on Image Processing, Vol. 1, Rochester, 22-25 September 2002, I-900-I-903.
<https://doi.org/10.1109/icip.2002.1038171>

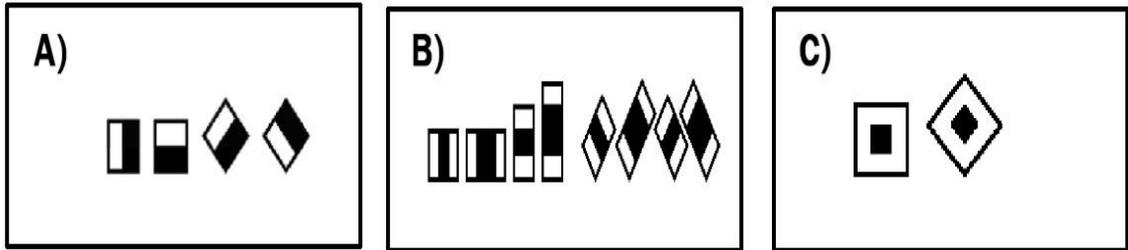


Figura 17. A) *Features* de dos inclinado B) *Features* de tres inclinado C) *Features* de dos centro-contornos.

Recuperado de oa.upm.es

Ejemplo de reconocimiento utilizando *Haar Cascade*:

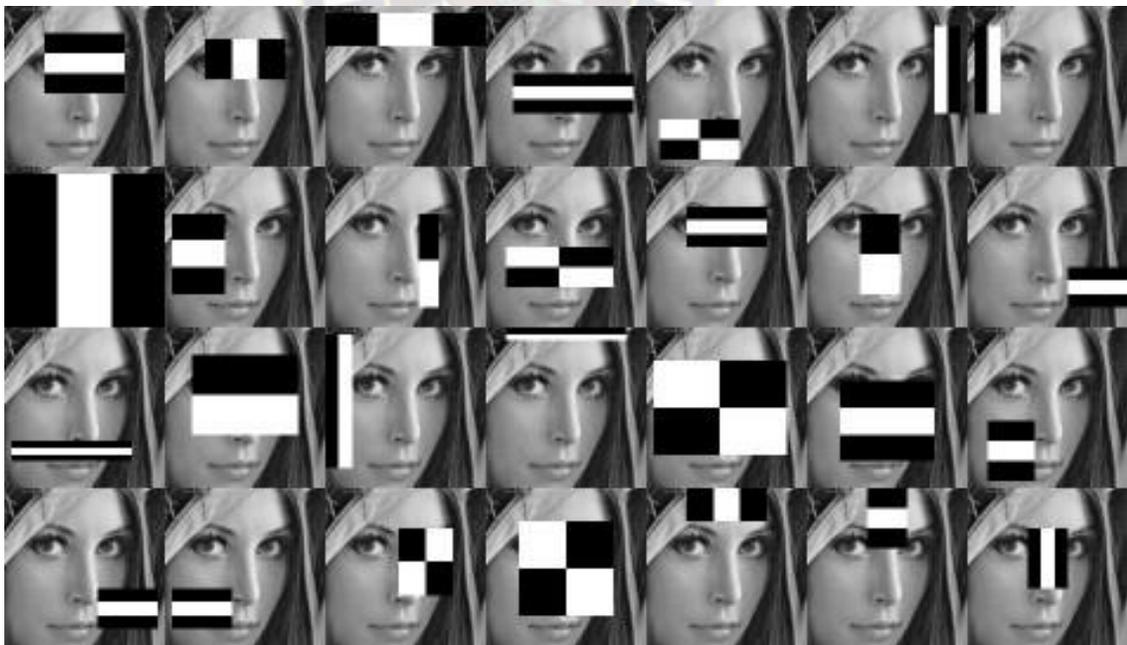


Figura 18. Funcionamiento y detección de Haar cascade.

Recuperado de devless.tumblr.com

2.3.14. Imagen integral

La suma de los píxeles de un rectángulo puede ser calculada de manera muy eficiente empleando una representación intermedia denominada imagen integral. La imagen integral en el punto (x, y) contiene la suma de todos los píxeles que están arriba y hacia la izquierda de ese punto en la imagen original.

$$II(x, y) = \sum_{i \leq x} \sum_{j \leq y} I(i, j)$$

Dónde:

$II(x, y)$ = Imagen integral.

X y Y = Elementos de la imagen.

$I(x, y)$ = Imagen original.

La imagen integral se puede calcular en un solo barrido de la imagen empleando el siguiente par de sentencias recurrentes:

$$S(x, y) = S(x, y - 1) + I(x, y)$$

$$II(x, y) = II(x - 1, y) + S(x, y)$$

Dónde:

$S(x, y)$ = es la suma de los píxeles en X y Y.

$II(x, y)$ = Imagen integral,

Usando la imagen integral, cualquier suma rectangular se puede calcular con cuatro referencias. En la figura 17 se tiene una imagen de la se quiere calcular la suma de los píxeles de la región D.

El valor de la imagen integral en la referencia 1 es la suma de los píxeles de la región A. El valor en la referencia 2 será A+B, en la referencia 3 será A+C en la referencia 4 será A+B+C+D.

De esta manera la suma de los píxeles de la región D se puede calcular como $4+1-(2+3)$ sobre la imagen integral:

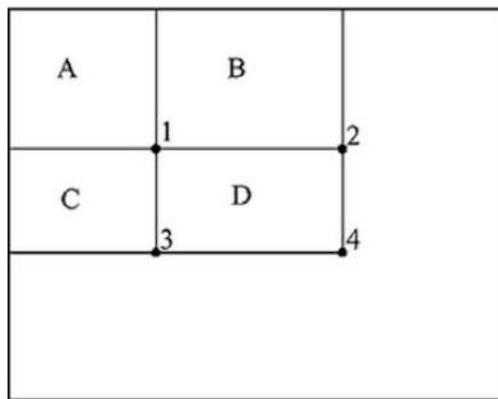


Figura 19. Calculo de una región a partir de una imagen integral.
Recuperado de oa.upm.es

2.3.15. Adaboost

El término *boosting* hace referencia a un tipo de algoritmo cuya finalidad es encontrar una hipótesis fuerte a partir de utilizar hipótesis simples y débiles.

El algoritmo AdaBoost fue creado por *Freund* y *Schapire* y es un diseño mejorado del *boosting* original y de sus diferentes variantes.

AdaBoost es una contracción de *Adaptive Boosting*, el algoritmo busca crear un clasificador fuerte cuya base sea la combinación lineal de clasificadores *weak learner*. Pero también propone entrenar una serie de clasificadores débiles de manera iterativa, de modo que cada nuevo clasificador o *weak learner* se enfoque en los datos que fueron erróneamente clasificados por su predecesor, de esta manera el algoritmo se adapta y logra obtener mejores resultados.

Una de las formas más sencillas de entender la finalidad de este tipo de algoritmos es mediante la analogía del siguiente ejemplo.

Debe considerar un filtro capaz de diferenciar entre un correo *spam* y uno que no lo es, para esto se crean hipótesis acerca de qué tipo de palabras clave contienen este tipo de archivos y así generar un clasificador que logre diferenciar de manera más precisa entre estos dos tipos de correos. Para lograr esto, primero se necesita de un *weak learner*, este es un algoritmo que nos ayuda a encontrar las hipótesis débiles.

A todos los datos se les asigna inicialmente el mismo peso de $(1/m)$, este peso se va actualizando con cada iteración según los ejemplos mal clasificados, de esta manera se busca minimizar el error esperado y enfocarse en clasificar correctamente los datos que ahora tienen mayor peso. De este modo, al final se realiza la suma de todos los clasificadores previamente generados y se obtiene una hipótesis cuya predicción es más acertada.

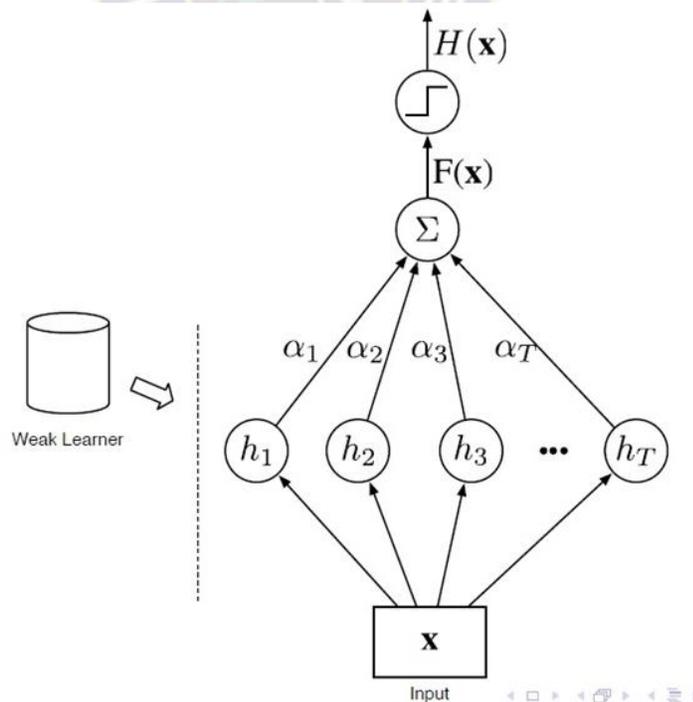


Figura 20. Diagrama de flujo que explica el funcionamiento de AdaBoost.
Recuperado de catarina.udlap.mx

Para que un clasificador sea efectivo y preciso en sus predicciones, se requiere que cumpla las siguientes tres condiciones:

- Debe de haber sido entrenado con suficientes ejemplos-datos.
- Debe de tener un error de entrenamiento bajo.
- Debe de ser “simple”.

2.3.16. Tecnología GSM

El Sistema Global para las comunicaciones Móviles, es un sistema estándar, libre de regalías de telefonía móvil digital, desarrollado por el Instituto Europeo de Estándares de Telecomunicaciones (ETSI).

Fue desarrollado originalmente para transmisión de voz, pero también es capaz de transmitir datos con una carga máxima de 9,6 kbit/s en el enlace descendente y en el enlace ascendente.

Servicios

El sistema de comunicación GSM permite entre otros servicios, efectuar *Roaming* Internacional, bien hacer o recibir en el extranjero llamadas como si se hicieran en el territorio.

Recepción de mensajes cortos alfanuméricos (SMS), posibilitando su lectura en la propia terminal GSM.

Los datos del abonado y la suscripción no se guardan en móvil, si no en una tarjeta inteligente denominada SIM (*Subscriber Identity Module*), que puede memorizar números telefónicos asociados a nombres, dispone, además, de dos códigos de seguridad del PIN y el PUK, sin los cuales es imposible efectuar la llamada.

Otro elemento fundamental en el GSM es el uso de tecnologías criptográficas a escala militar que ofrecen una absoluta seguridad, desde la autenticación de la tarjeta, hasta la conversación.

Calidad de los servicios y seguridad

La calidad de voz telefónica en el sistema GSM debe ser al menos tan buena como la que tenía la primera generación de sistemas analógicos a 900 MHz.

Utiliza tarjeta de usuario para la autenticación encriptada de la validez de la llamada, que facilita una confidencialidad total (voz, datos e identidad del abonado) e imposibilidad de utilización de equipos robados mediante la asignación previa de un número de serie a cada estación móvil.

Frecuencias GSM

El interfaz de radio de GSM se ha implementado en diferentes bandas de frecuencia en distintos países, el motivo es la legislación existente y la disponibilidad de frecuencias.

En la mayor parte de los países se usan las bandas de 900 MHz y 1800 MHz.

En algunos países de América (Canadá, USA) en la banda de 850 MHz y 1900 MHz.

REDES CELULARES GSM					
Banda	Nombre	Canales	Uplink (Mhz)	Downlink (Mhz)	Notas
GSM 850	GSM 850	128-251	824-849	869-894	Usada en los EEUU, Sudamérica y Asia.
GSM 900	P-GSM 900	1-124	890-915	935-960	La banda con que nació GSM en Europa y la más extendida
	E-GSM 900	975-10232	880-890	925-935	E-GSM, extensión de GSM 900
GSM 1800	GSM 1800	512-885	1710-1785	1805-1880	Europa y muchos otros países
GSM 1900	GSM 1900	512-810	1850-1910	1930-1990	Usada en Norteamérica, incompatible con GSM- 1800 por solapamiento de bandas

Tabla 18. Redes celulares GSM.
Recuperado de ocw.uc3m.es

Estructura del sistema GSM

La estructura básica del sistema GSM se organiza como una red de células radioeléctricas continuas que proporcionan cobertura completa al área de servicio.

Cada célula pertenece a una estación base (BTS) que opera en un conjunto de canales de radio diferentes a los usados en las células adyacentes y que se encuentran distribuidas según un plan celular.

Un grupo de BTS se encuentra conectado a un controlador de estaciones base (BSC), encargado de aspectos como el handover (traspaso del móvil de una célula a otra) o el control de potencia de las BTS y de los móviles, prácticamente se encarga del manejo de toda la red de radio.

Una o varias BSCs se conectan a una central de conmutación de móviles (MSC), que viene siendo el corazón del GSM, al ser responsable de la inicialización, enrutamiento, control y finalización de las llamadas, información sobre la tarificación y el interfaz entre diversas redes GSM o entre una de ellas y las redes públicas de telefonía o datos.

Arquitectura de la red GSM

MT / TE (*Mobile Terminal / Terminal Equipment*)

Se comunica con la red móvil a través del interfaz de aire (Um).

Tarjeta SIM (*Subscriber Identity Module*)

Es una tarjeta inteligente desmontable que almacena de forma segura la clave de servicio del suscriptor usada para identificarse en la red.

BTS (*Base Transceiver Station*)

La estación base es el conjunto de la antena y su enlace con el resto de la red, que cubre cierta área geográfica o conocida como celda.

- Radio de cobertura máximo de unos 35 Km.
- Potencia 320 W.
- Cada celda tiene un identificador de celda (CellId).

BSC (*Base Station Controler*)

Agrupar un conjunto de BTS, que elimina complejidad de la BTS y extrae de la MSC las funciones del control de radio.

MSC (*Mobile Switching Centre*)

Agrupar a un conjunto de BSC, cubriendo una amplia zona geográfica, se encarga de iniciar, terminar y canalizar las llamadas a través del BSC y BTS correspondientes al abonado llamado.

VLR (*Visitor Location Register*)

Proporciona un repositorio de datos de suscripción de los móviles que estarían operando en un área de localización, está asociada a un MSC.

HLR (*Home Location Register*)

Proporciona un repositorio central de datos de suscripción donde cada red debe tener al menos un HLR.

AuC (*Authentication Centre*)

Asociado al HLR, contiene las claves individuales de identificación del abonado.

- Clave secreta Ki (128 bits)
- Nunca abandona el AuC ni el SIM.
- El AuC genera un conjunto de claves SRES, Kc y RAND.

EIR (*Equipment Identity Register*)

Repositorio que almacena los IMEI (International Mobile Station Equipment Identity), utilizados por el sistema GSM. El objetivo final es reducir las posibilidades de robo de terminales y fraude.

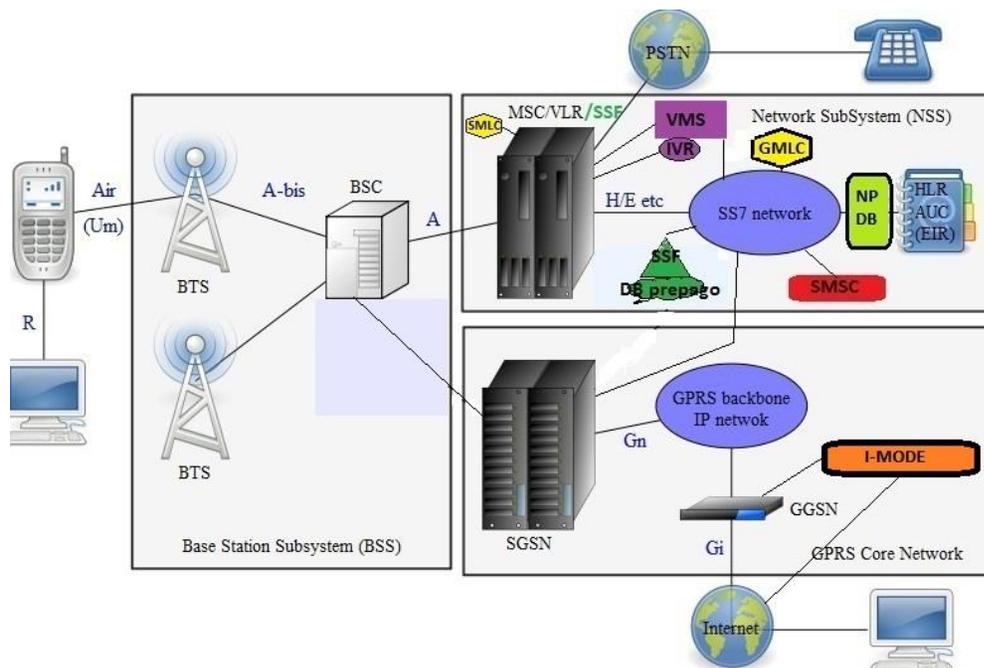


Figura 21. Arquitectura de red GSM.
Recuperado de movilfacil.files.wordpress.com

Comandos AT

Conocidos también como comandos Hayes, en honor a su desarrollador Dennis Hayes, son una serie de instrucciones que conforman un interfaz de comunicación entre usuario y modem.

Aunque la finalidad principal de los comandos AT fue la comunicación con módems, la telefonía móvil GSM/GPRS también adoptó este lenguaje como estándar de comunicación.

En la actualidad, todos los terminales móviles GSM poseen una serie específica de comandos AT que nos permiten configurarlos por medio de estas instrucciones e indicarles una serie de acciones que queremos que ejecuten, tales como marcar un número de teléfono, enviar o leer un SMS, consultar el estado de conexión a la red, leer o escribir en la agenda de contactos, entre otras.

Los comandos AT los más importantes:

AT: Sirve para verificar si el módulo SIM900 está funcionando adecuadamente para entrar en modo comando. Al enviar AT el SIM deberá contestarnos con un OK.

AT+CGMI: Veremos en nombre del fabricante.

ATI: Ver la información del producto.

AT+IPR=?: Preguntar el Baud Rate en el que puede operar el SIM.

AT+IPR?: Sirve para preguntar el Baud Rate actual.

AT+IPR=XXXX: Configurar a la frecuencia deseada.

AT+COPS?: Nombre de la compañía telefónica.

AT+CGSN: Visualizar el IMEI del chip utilizado.

AT+CSCS?: Tipo de texto.

AT+CSCS="XXX": Configurar a tipo de texto.

AT+CMGF?: Ver el formato de un mensaje, ya sea PDU(0) o SMS(1).

AT+CMGS=04455XXXXXXXXXX: Enviar un SMS Se despliega el símbolo mayor que ">", al escribir mensaje y al finalizar presiona Ctrl+Z retornará OK, si el SMS se envió correctamente.

AT+CMGL=ALL: Sirve para ver todos los mensajes que nos han llegado al SIM.

ATD04455XXXXXXXXXX: Sirve para hacer una llamada a cualquier teléfono móvil.

ATA: Sirve para contestar una llamada.

ATH: Sirve para colgar una llamada.

CAPÍTULO III

DESARROLLO DEL PROYECTO

3.1. ANÁLISIS PRELIMINAR

En la actualidad, la mayoría de los vehículos cuentan con seguridad pasiva y existen iniciativas para mejorar la seguridad activa de los mismos, tomando en cuenta el desempeño del conductor como causa de siniestros.

En el presente proyecto, se propone un dispositivo electrónico de monitoreo de la somnolencia y la fatiga del conductor que trabaja con una cámara de video (transductor fotoeléctrico) con visión nocturna en combinación con la librería de visión por computador de código abierto *OpenCV*, como sensor no invasivo, que captura los datos y los envía a una computadora que realiza el procesamiento digital de señales, que en este caso es una tarjeta de desarrollo *Raspberry PI 3B*.

La información es analizada para detectar el rostro del conductor, de forma particular sus ojos y el parpadeo. Si el conductor cierra sus ojos por un tiempo prolongado, una alarma sonora se activa, advirtiéndole al conductor que se mantenga atento durante su conducción.

Si la alarma se activa en dos ocasiones más, significa que el conductor está ante una situación de peligro, por lo que el sistema envía un mensaje de texto a un familiar o contacto externo que pueda brindar el apoyo de fuera del vehículo.

El dispositivo tiene un contador que monitorea el tiempo del conductor frente al volante, tomando como límite tres horas, de acuerdo a la normativa boliviana.

Una vez pasado este tiempo, el sistema activa una alarma sonora de alerta y envía un mensaje de texto al contacto registrado.

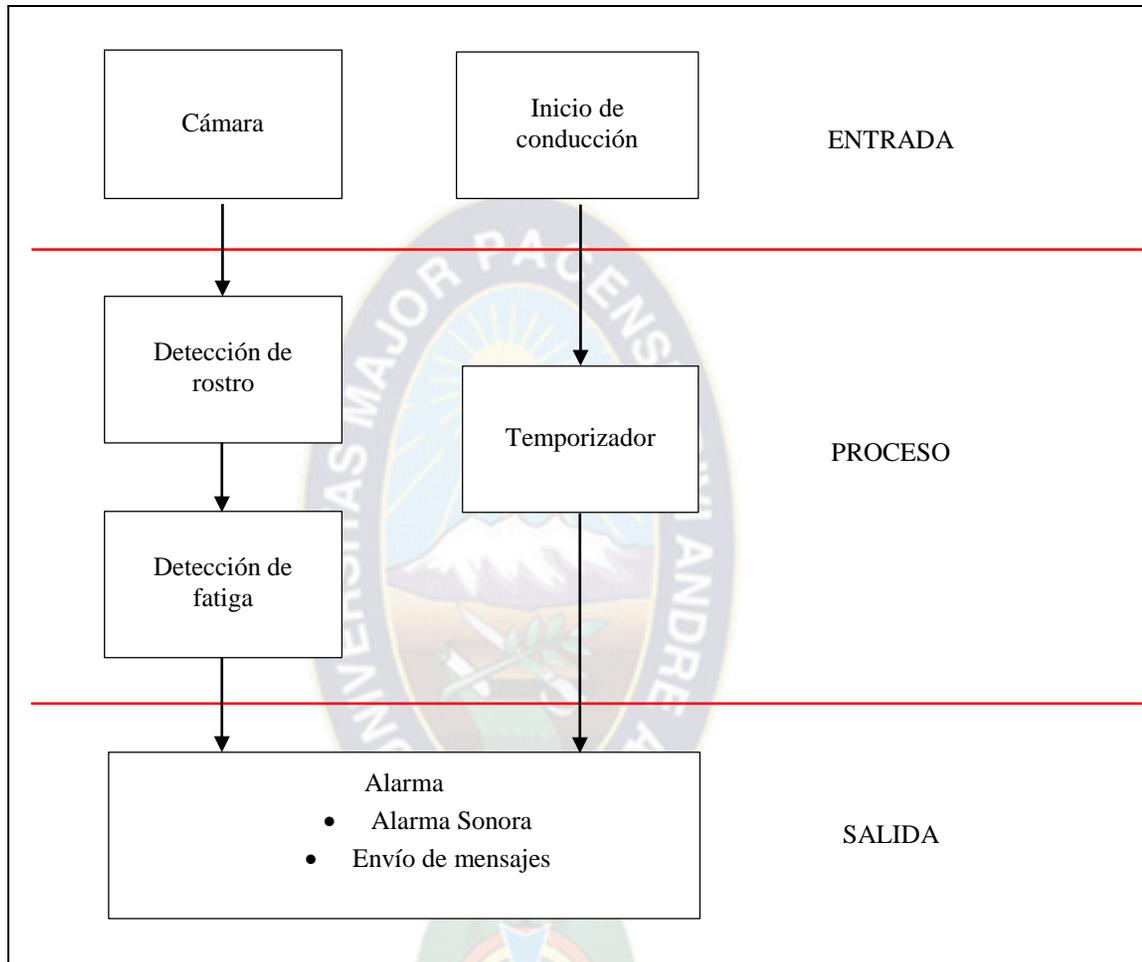


Figura 22. Diagrama funcional del dispositivo.

Elaboración propia

En el diagrama funcional se muestra el funcionamiento general del dispositivo, donde la parte más importante es el proceso de detección del rostro del conductor y por ende consume muchos recursos de un computador.

Para el diseño del dispositivo, se tomaron en cuenta factores como el cambio de entorno como ser el día y la noche.

La posición y movimiento de la cabeza respecto a la posición del dispositivo de monitoreo, que monitorea que el conductor no quite la vista horizonte.

3.2. INGENIERÍA DEL PROYECTO

Uno de los requerimientos más importantes durante el diseño del dispositivo era la capacidad de procesamiento en tiempo real de las imágenes capturadas por la cámara.

Analizando las funcionalidades de diferentes procesadores existentes en el mercado nacional, la tarjeta de desarrollo *Raspberry PI 3B* fue la única que se acomodaba al diseño del dispositivo, debido a su precio accesible, sus características similares a las de un computador y su tamaño que garantiza portabilidad.

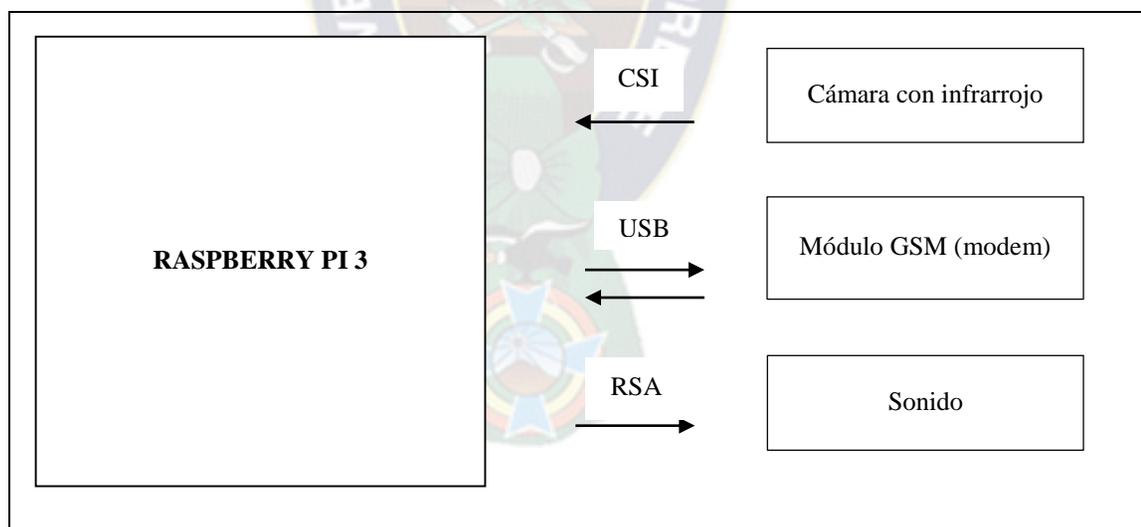


Figura 23. Diagrama conexión del hardware.
Elaboración propia

El Raspberry PI 3B es el elemento fundamental del dispositivo, porque realiza todo el procesamiento de los módulos y el sensor que captura la información por medio de software y analiza los resultados. Activando la alerta en caso de somnolencia, como se muestra en la figura 24.

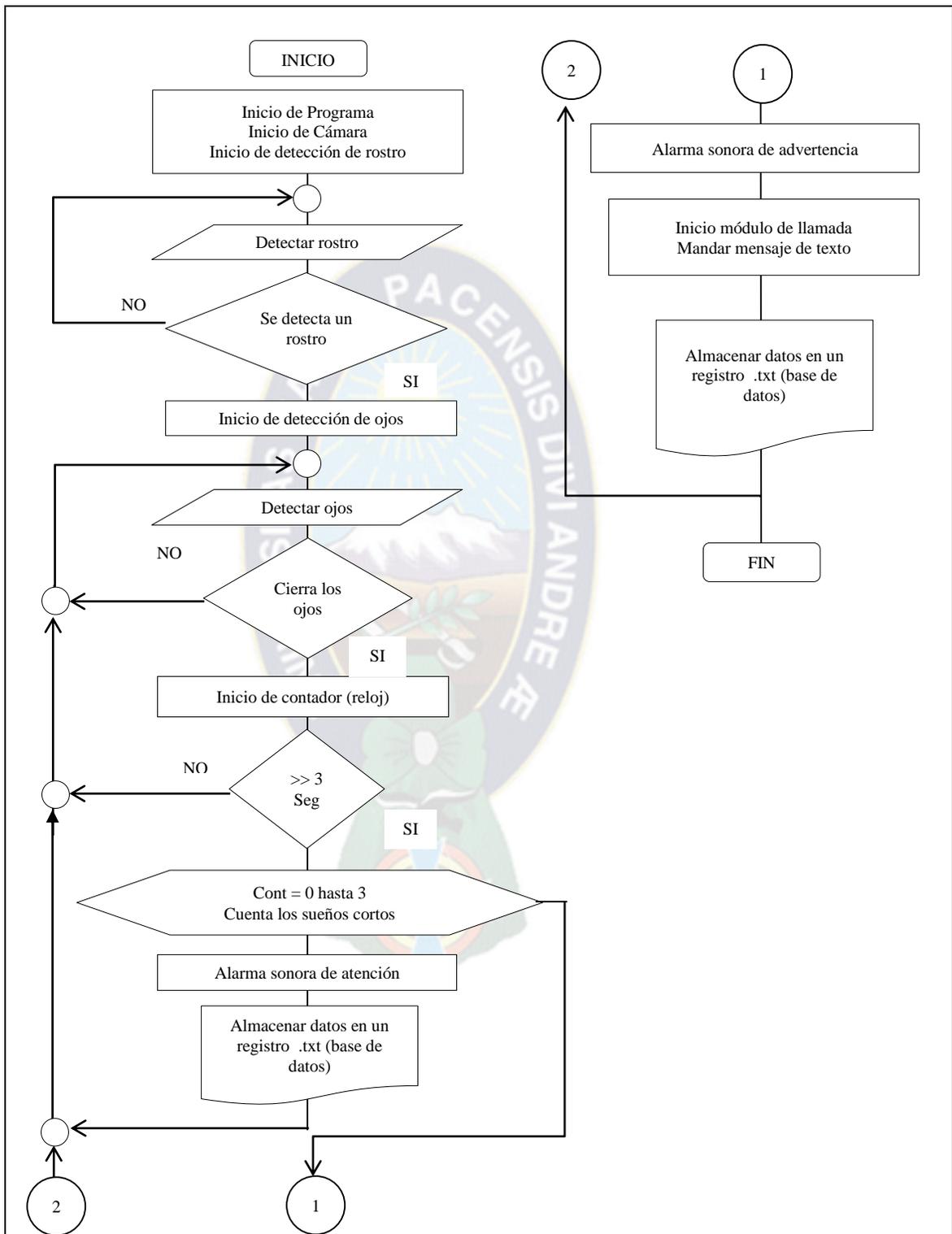


Figura 24. Diagrama del funcionamiento detector de somnolencia.

Elaboración propia

La cámara envía la información (frames) a la tarjeta de desarrollo Raspberry PI 3B, para detectar el rostro del conductor, identificando el iris de los ojos, analizando su forma.

Un ojo abierto tendrá un iris circular, si la persona parpadea y mantiene cerrados los ojos por un período mayor a tres segundos, el dispositivo lo detectará y activará una alarma sonora de aviso fonético.

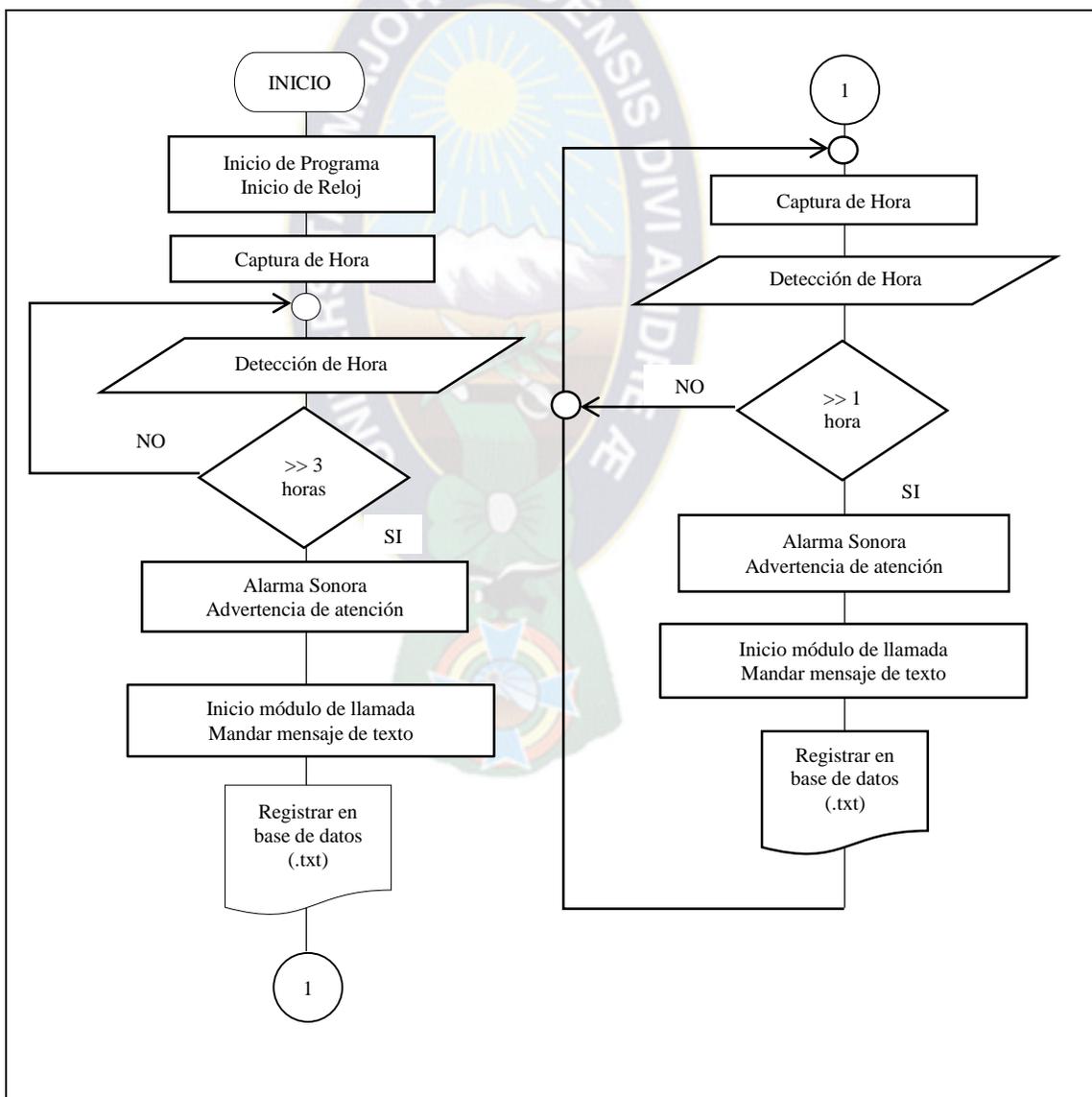


Figura 25 Diagrama flujo del funcionamiento del monitoreo de horas delante el volante.

Elaboración propia

El mismo proceso se realizará dos oportunidades más, pero en la última activación de alarma, el dispositivo enviará un mensaje de texto a un familiar o contacto registrado en el dispositivo, recomendando alertar al conductor sobre su estado de fatiga.

Todos estos sucesos son registrados en un archivo de texto en formato “.txt”, que puede ser útil para el análisis posterior o el desarrollo de futuras mejoras o aplicaciones.

En la figura 25, el diagrama muestra de forma detallada el funcionamiento del dispositivo desde el inicio de la conducción.

Al iniciar la conducción se inicia el programa, un reloj captura la hora, un bloque comparador va comparando esa hora con las horas posteriores durante la conducción hasta que transcurran las tres horas frente al volante.

Luego de transcurridas las tres horas frente al volante, una alarma sonora advierte sobre esta situación y el dispositivo envía un mensaje a un familiar o contacto registrado. Luego de esto, la alarma se activará cada hora hasta que el conductor deje el volante para descansar.

Todos estos sucesos son registrados en un archivo de texto en formato “.txt”, que puede ser útil para el análisis posterior o el desarrollo de futuras mejoras o aplicaciones.

3.2.1. Diseño del hardware

3.2.1.1. Puertos y ranuras de conexión

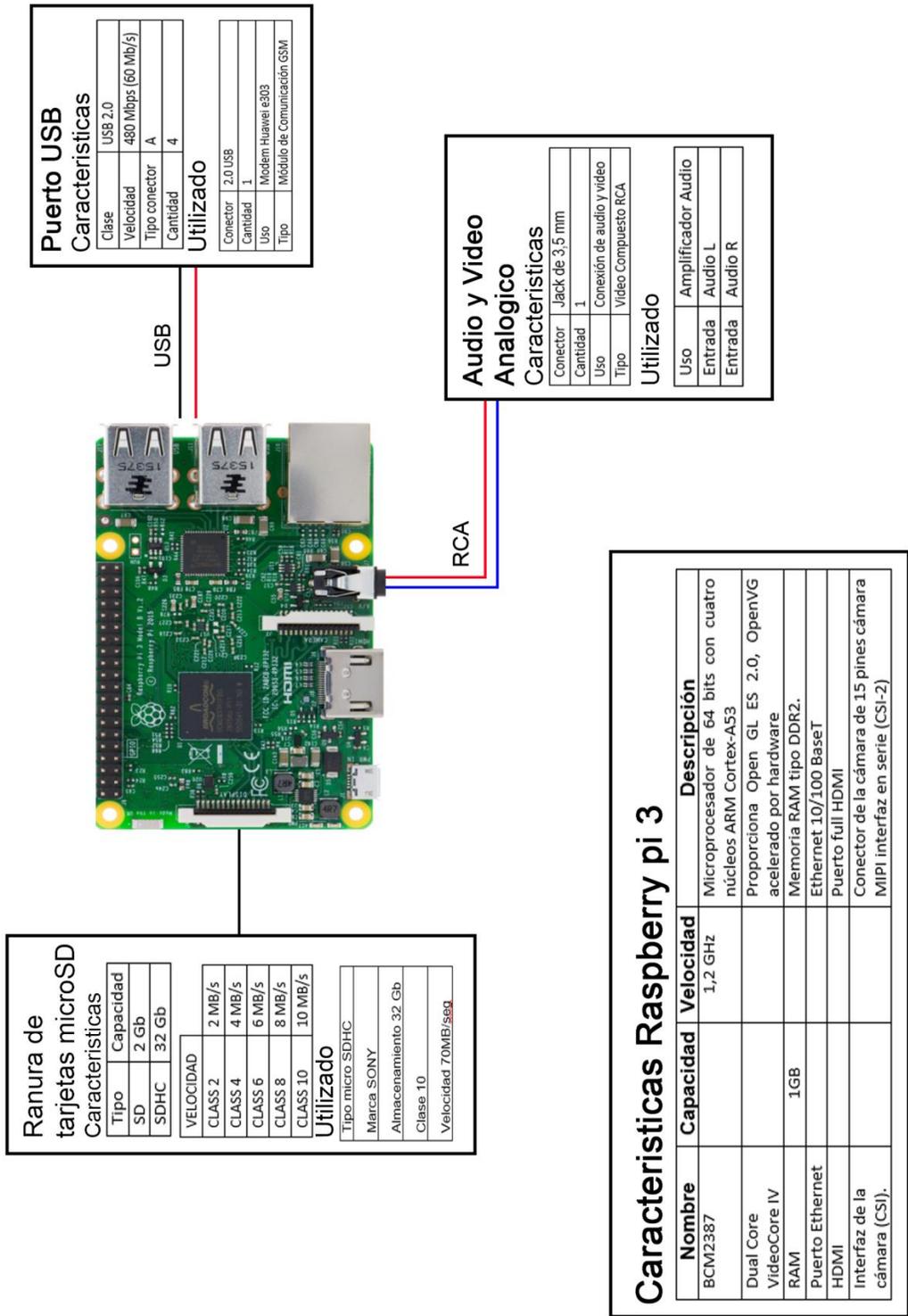


Figura 26. Conexiones y características utilizadas en el Rasperry pi 3 B.

Elaboración Propia

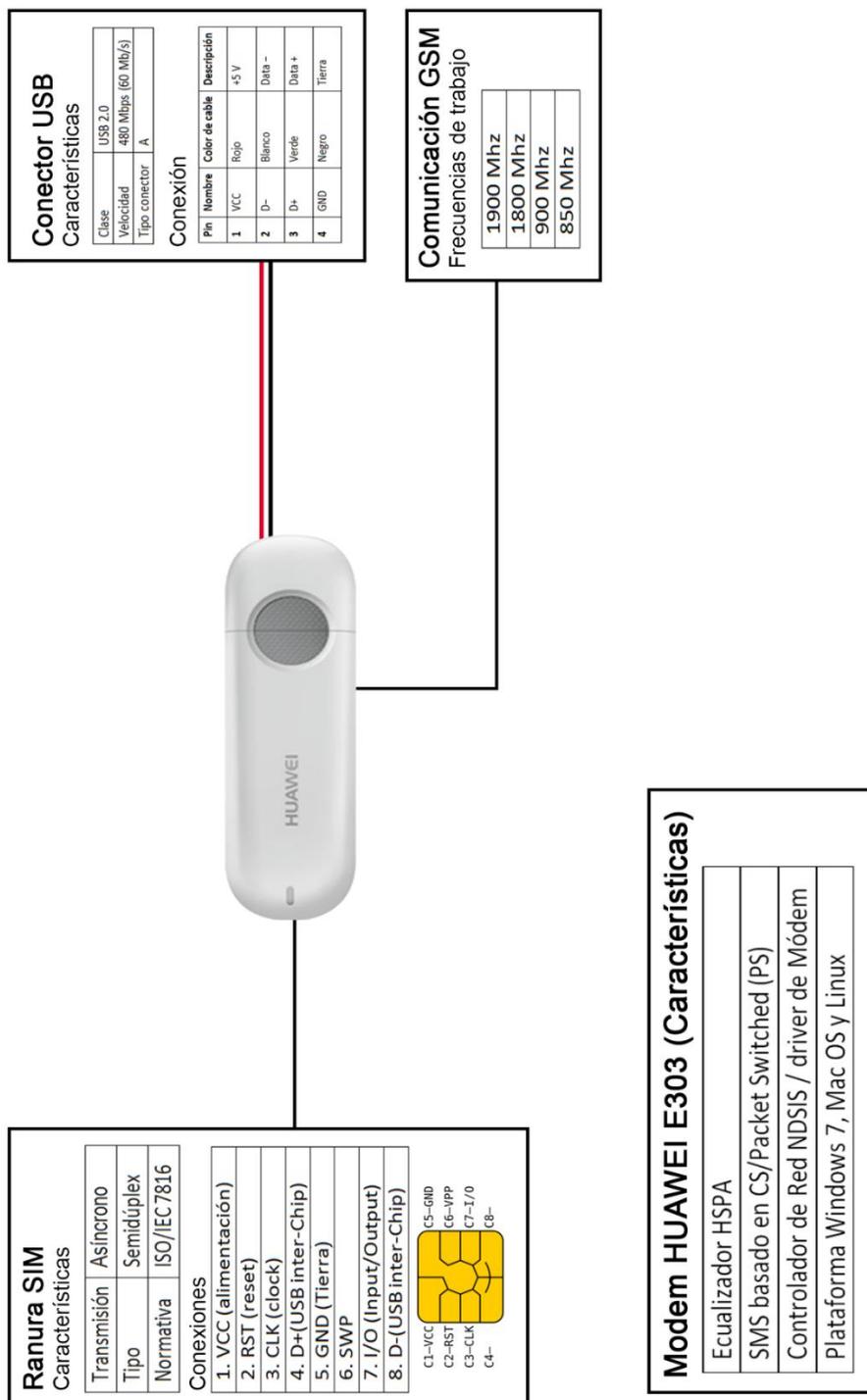


Figura 27. Conexiones y características utilizadas en el modem Huawei e303
Elaboración propia

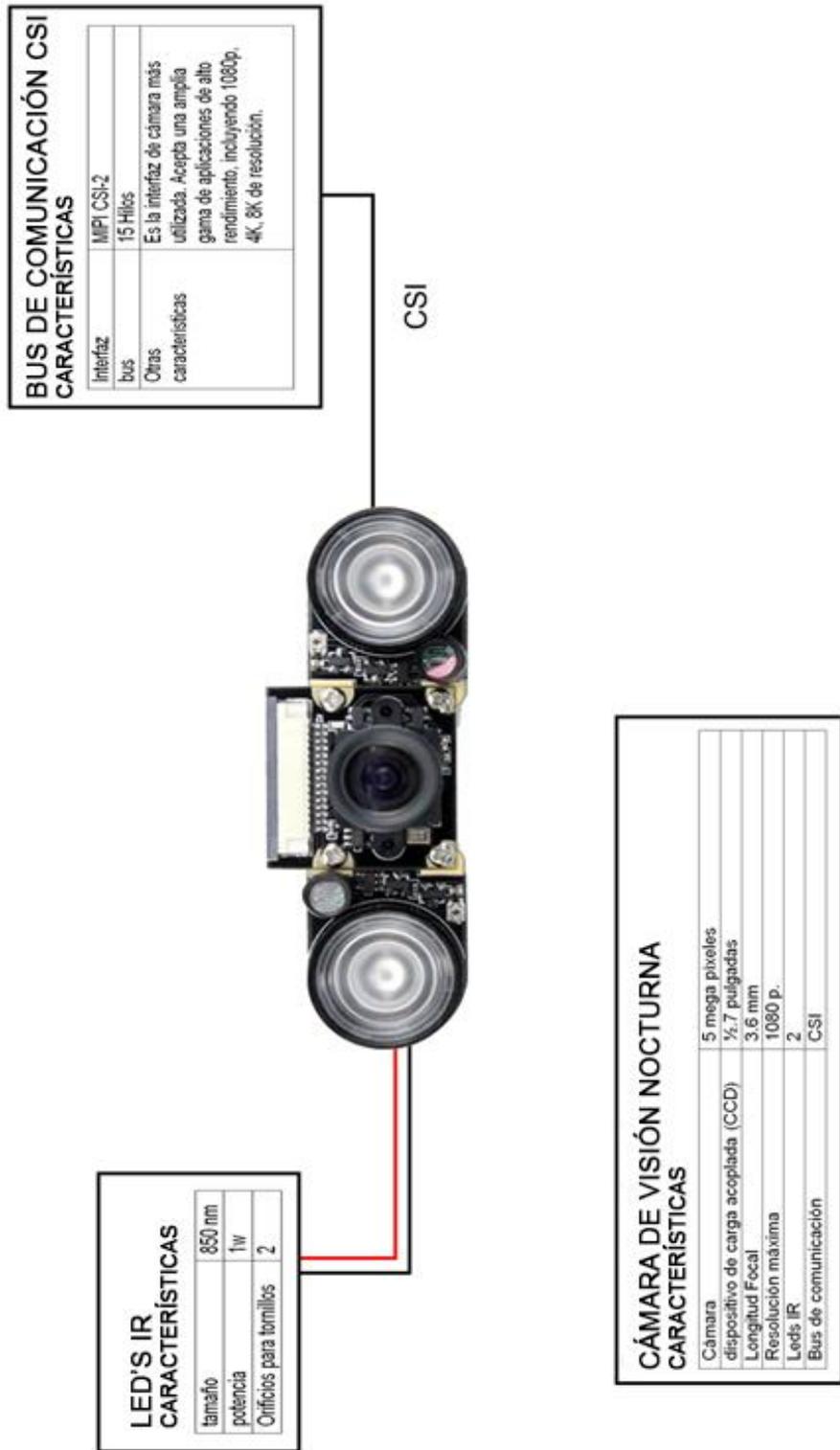


Figura 28. Conexiones y características utilizadas Cámara de visión nocturna.
Elaboración propia

3.2.1.2. Amplificador de audio para el módulo de alerta

Para el diseño del amplificador de audio del módulo de alerta, se está empleando el circuito integrado TDA2822M, tomando como referencia la configuración que proporciona el ST de *SGS-Thomson Microelectronics* en la que se observa un esquema como un amplificador estéreo.

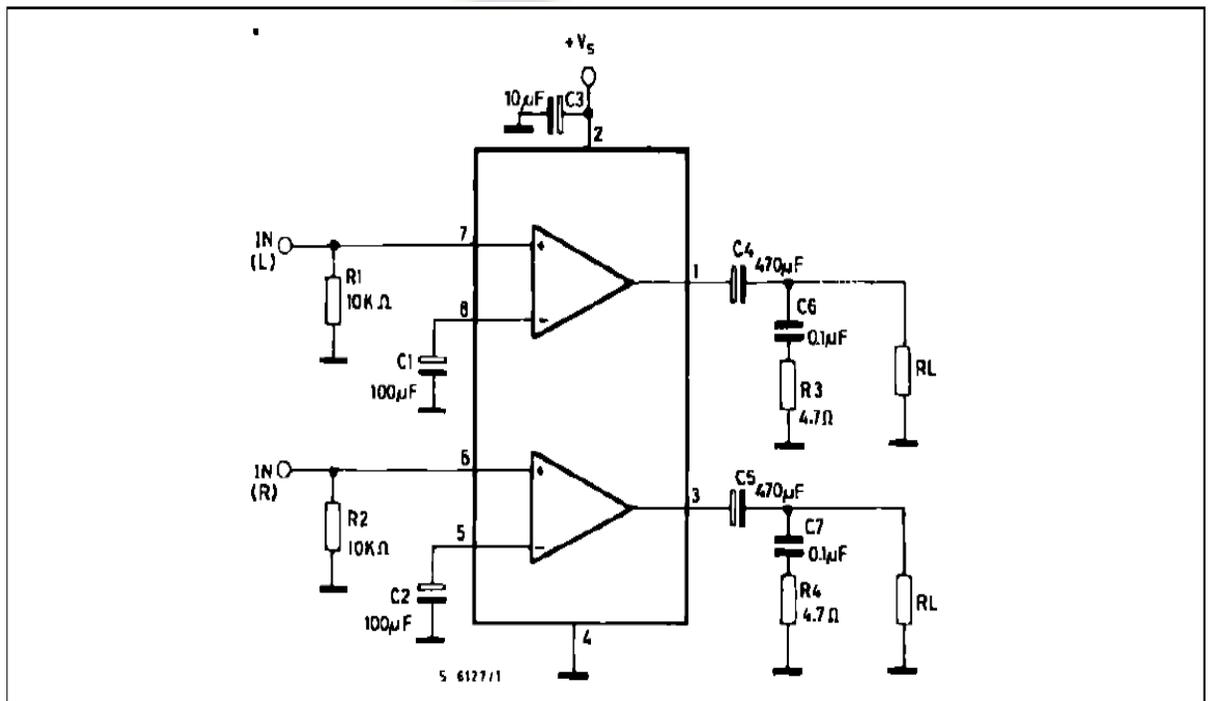


Figura 29. Amplificador de Audio proporcionado por el fabricante.
Recuperado de SGS-Thomson Microelectronics

Para el Diseño del amplificador de baja potencia se reemplazó las resistencias de 10 K ohm de la entrada con dos potenciómetros del mismo valor nominal, esto para poder regular la amplitud de la onda en la entrada (volumen), También se aumentó el valor del capacitor de filtro de alimentación del CI de 10 uf a 470 uf para reducir el ruido de la fuente de poder.

Quedando el circuito de la siguiente manera:

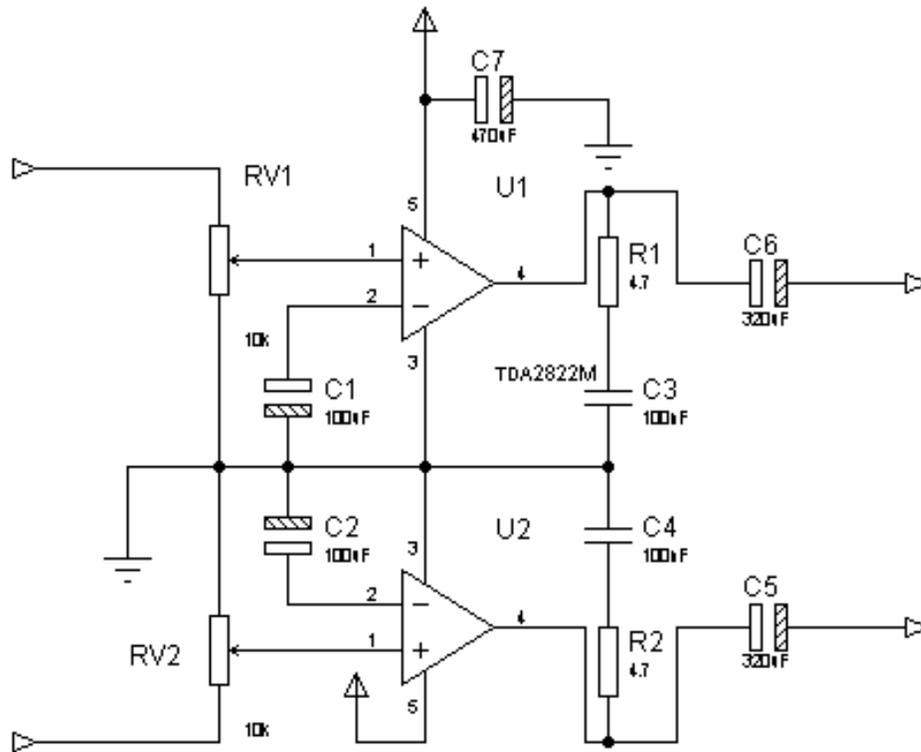


Figura 30. Circuito modificado para cubrir las necesidades del dispositivo.
Elaboración propia

Para poder realizar el ensamble del circuito de prueba listamos la siguiente materiales basado en el diagrama eléctrico mostrado con anterioridad.

Lista de material empleado

COMPONENTES UTILIZADO PARA EL AMPLIFICADOR AUDIO		
Tipo	código	cantidad
CI Amplificador Operacional	TDA2822M	1
Capacitor Electrolítico	100 uf	2
Capacitor Electrolítico	330 uf	2
Capacitor Electrolítico	470 uf	1
Capacitor Cerámico	100 nf	2
Resistencia	4.7 Ohm	1
Potenciómetro	10 K Ohm	2

Tabla 19. Componentes para el amplificador de audio
Elaboración propia

Para el diseño de la placa se utilizó el programa *PCB Wizard* un programa de distribución *Open Source*, de fácil manipulación y que brinda distintos esquemas del circuito obteniendo estos resultados:

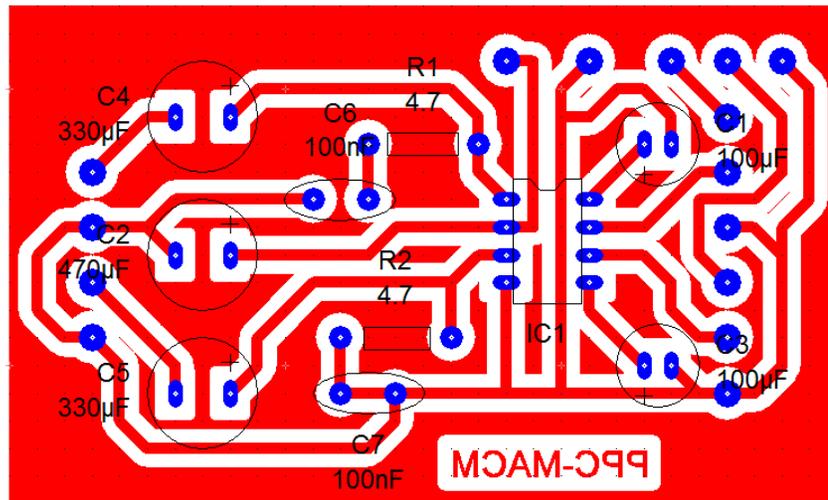


Figura 31. Diseño de la placa en PCB Wizard.
Elaboracion propia

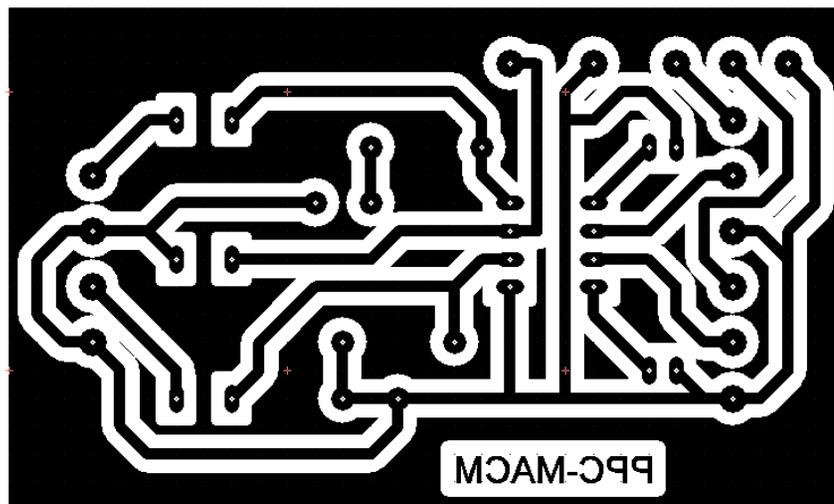


Figura 32. Negativo para realizar el quemado de la placa PBC.
Elaboración propia

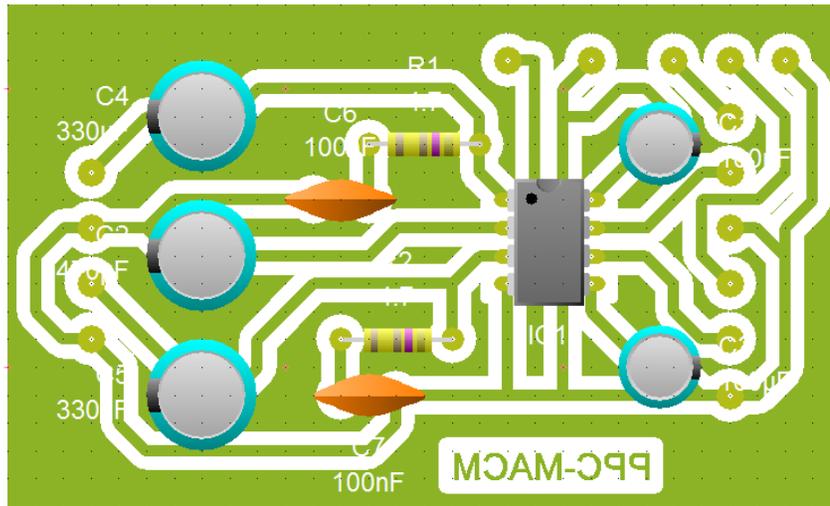


Figura 33. Vista en 3D del Amplificador de Audio.
Elaboración propia

3.2.1.3. Ensamblaje del hardware

Las piezas para el ensamblaje son la placa de desarrollo *Raspberry PI 3* y el amplificador de sonido ya descrito anteriormente sus funcionamientos.

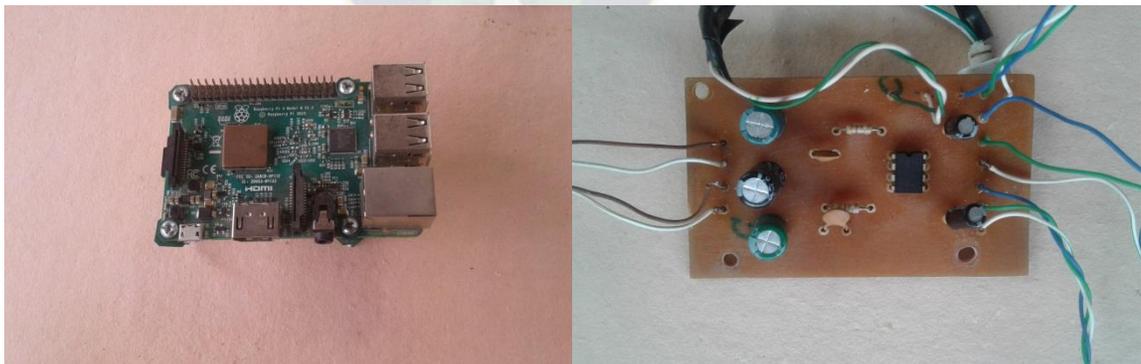


Figura 34. Componentes utilizados en el dispositivo (*Raspberry PI 3B* y Amplificador de audio)
Realización Propia.

Para la protección del dispositivo se diseñó un esqueleto donde se montó todos los elementos que componen lo electrónico, para que así el dispositivo sea manejable y también sea invulnerable a los golpes que se podría producir por la manipulación.

1. Se diseñó el esqueleto con el fin de mantener una estabilidad y estructura para el dispositivo.
2. Posteriormente se sujetó la tarjeta de desarrollo raspberry PI 3B y El amplificador de audio.
3. Se realizó todas las conexiones (alimentación, conexión de audio, cámara ir entre los distintos módulos)
4. Montando después la carcasa con sus respectivos terminales de conexión de salida (audio y los potenciómetros).
5. Para finalizar con el montaje de la cámara en un brazo flexible.

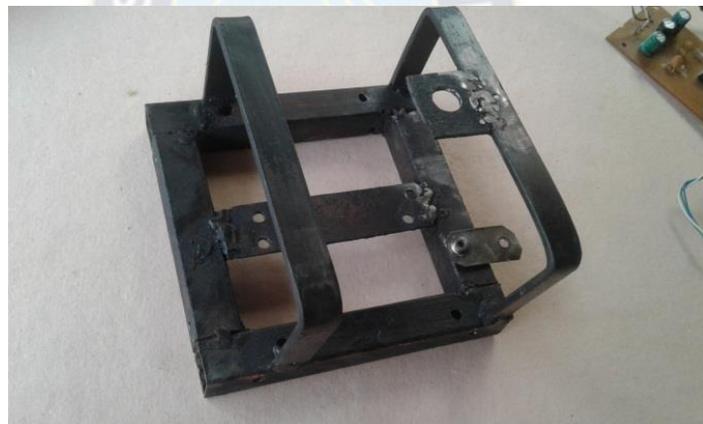


Figura 35. Esqueleto y tapas metálicas del dispositivo.
Elaboración propia.

Posteriormente se procedió al ensamblaje de las piezas:

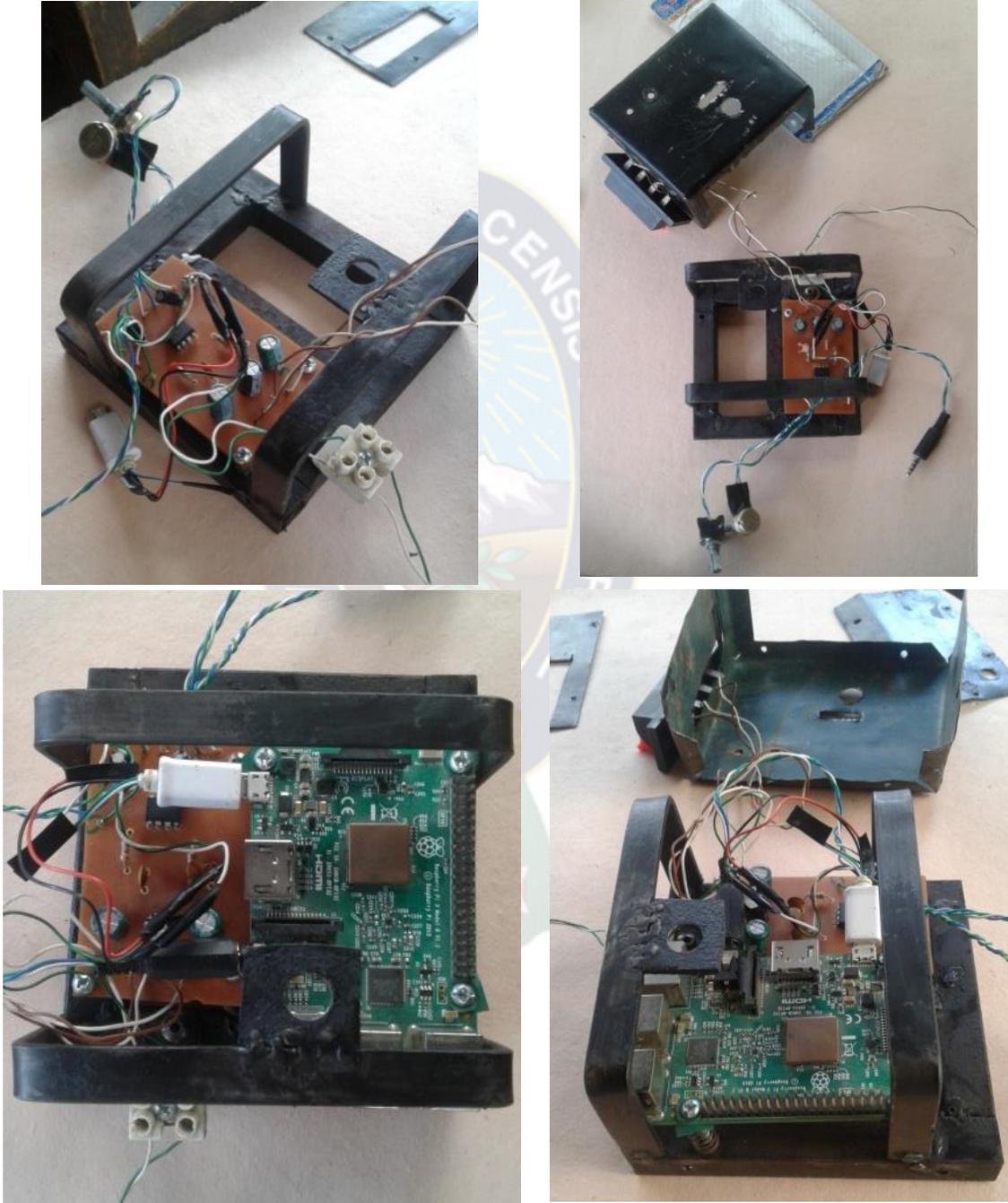


Figura 36. Secuencia de fotos de ensamblaje del dispositivo.
Elaboración propia.



Figura 37. Ensamblado final del dispositivo.
Elaboración propia.

3.2.1.4. Conexión del hardware

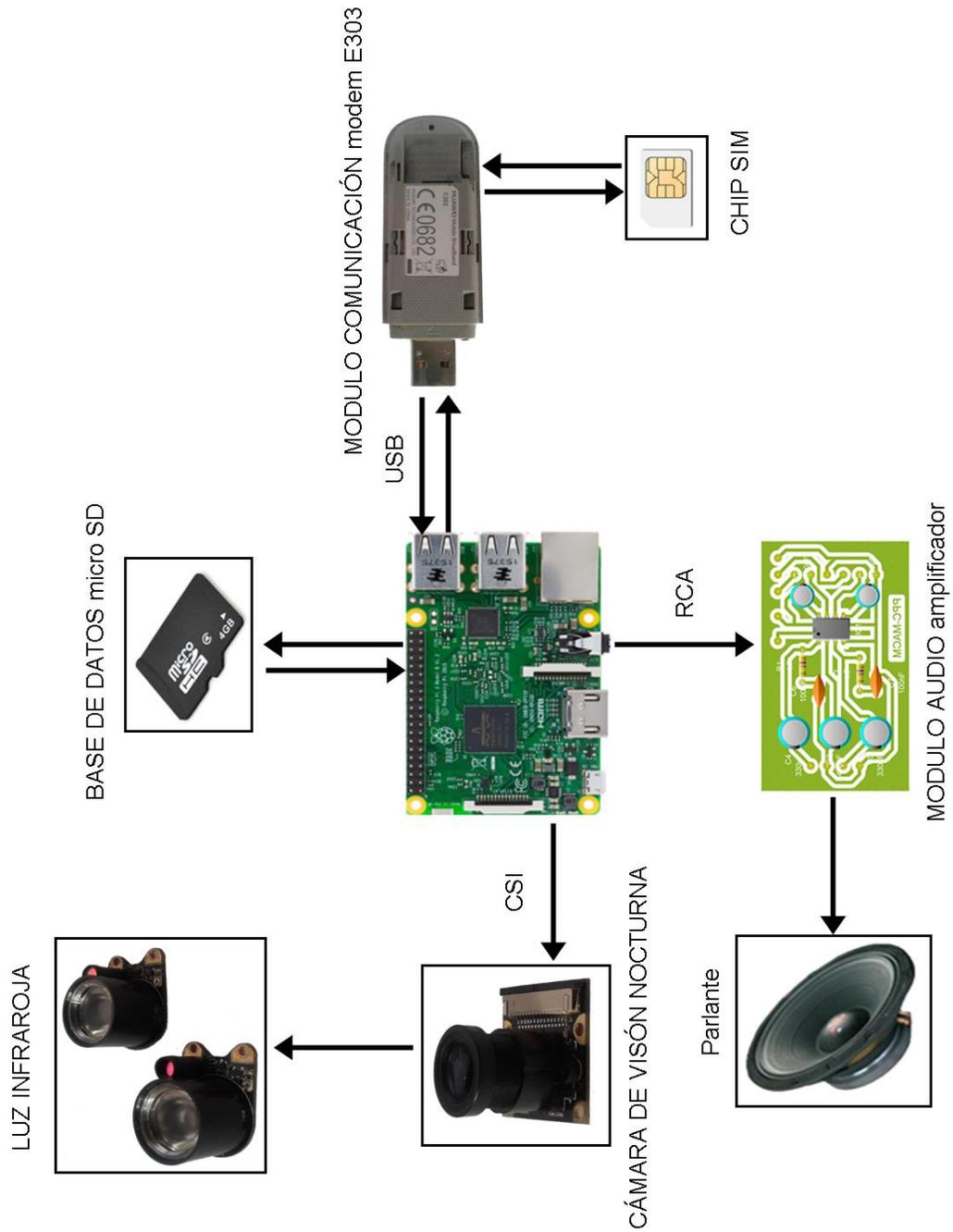


Figura 38. Conexiones generales del hardware.
Elaboración propia.

3.2.2. Diseño software

3.2.2.1. Instalación del sistema operativo raspbian

La tarjeta de desarrollo *Raspberry PI 3B* no viene con un sistema operativo instalado desde fábrica, por lo que es necesario visitar el portal web de este dispositivo, descargar el sistema operativo *Raspbian* desde la sección de soporte el sistema operativo y realizar la clonación de la imagen en una tarjeta de memoria micro SD de almacenamiento mínimo de 8 GB o superior y clase 6.

Por la velocidad de escritura, lectura y necesidad de capacidad para la instalación de paquetes que necesita el dispositivo, se utilizó una tarjeta de memoria micro SD de características especiales descritas a continuación:

- Tipo micro SDHC
- Marca SONY
- Almacenamiento 32 Gb
- Clase 10
- Velocidad 70MB/seg

El sistema operativo *Raspbian* está diseñado a partir del sistema operativo *Debian 8 Jessie*, que facilitaría la instalación de nuevos paquetes por analogía y la cantidad de documentación existente en la web.

Para instalarlo se descargó el archivo comprimido de la página oficial de *Raspberry PI* <https://www.raspberrypi.org/downloads/>.

Para su montaje en la tarjeta de memoria micro SDHC se utilizó la aplicación de código abierto Win32DiskImager que graba imágenes de CD o DVD en la memoria USB o micro SD.

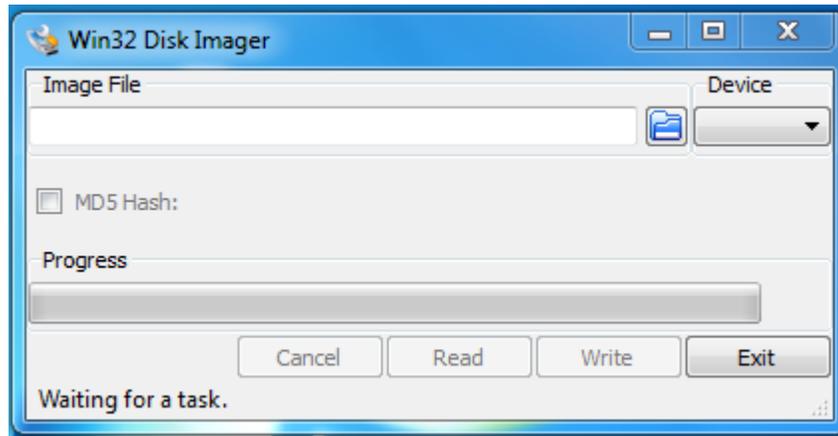


Figura 39. Win32DiskImager.
Recuperado de 1.bp.blogspot.com

Solo se necesita insertar la tarjeta de memoria en la ranura *microSD* de la tarjeta de desarrollo *Raspberry PI 3B* y conectarla a su fuente de alimentación.

El entorno gráfico Linux es agradable hacia el usuario y viene por defecto con varias aplicaciones como un navegador web, Libre Office.

Para el desarrollo del dispositivo se utilizaron las plataformas *Python 2* y *Python 3*.

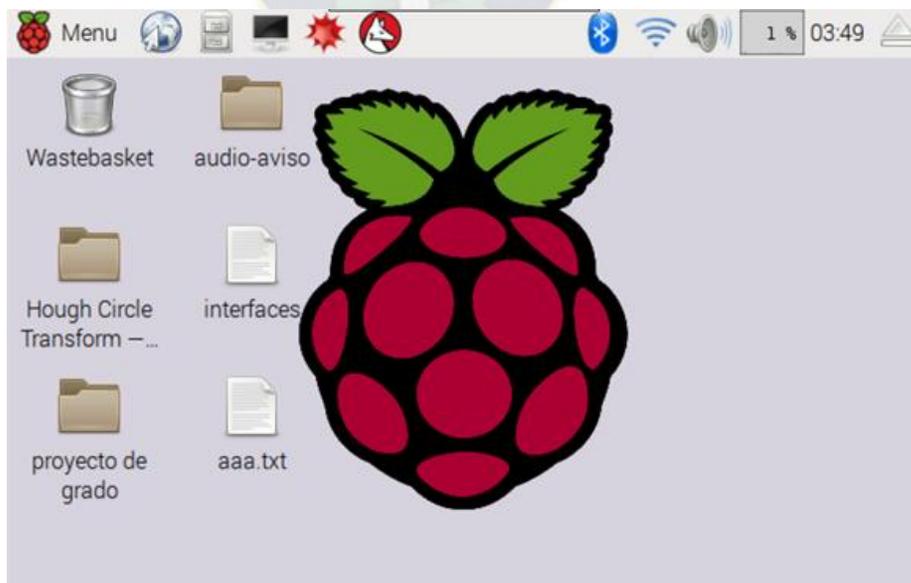


Figura 40. Raspbian.
Elaboración Propia

3.2.2.2. Instalación de la librería open CV

Se necesita instalar la librería de visión computacional que realizara el análisis de las imágenes *Open CV*.

Esta librería es *Open Source* por lo que no requiere pago de licencia y es soportada en los sistemas operativos *Windows* y *Linux* y lenguajes de alto nivel como *C++* y *Python*.

Los pasos de instalación de la librería son los siguientes:

Comando de actualización e instalación del *firmware* de *Raspberry PI 3B*

- \$ sudo apt-get update
- \$ sudo apt-get upgrade
- \$ sudo rpi-update

Comando de instalación de herramientas y paquetes necesarios

- \$ sudo apt-get install build-essential cmake pkg-config

Instalar la biblioteca GTK, que se utiliza para construir interfaces gráficas de usuario (GUI), es indispensable para ver las imágenes en la pantalla.

- \$ sudo apt-get install libgtk2.0-dev

Instalar paquetes de video, estos paquetes se utilizan para cargar archivos de vídeo usando *OpenCV*.

- \$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

Instalar PIP

- \$ wget https://bootstrap.pypa.io/get-pip.py
- \$ sudo python get-pip.py

Instalar los virtualizadores

- `$ sudo pip install virtualenv virtualenvwrapper`
- `$ sudo rm -rf ~/.cache/pip`

Actualizar el perfil y crear el entorno virtual

- `export WORKON_HOME=$HOME/.virtualenvs`
- `source /usr/local/bin/virtualenvwrapper.sh`
- `$ source ~/.profile`
- `$ mkvirtualenv cv`

Instalación de *python 2,7* y *numpy* en el entorno virtual

- `$ sudo apt-get install python2.7-dev`
- `$ pip install numpy`

Descargar Open CV y descomprimirlo

- `$ wget -O opencv-2.4.10.zip`
`http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.10/opencv-2.4.10.zip/download`
- `$ unzip opencv-2.4.10.zip`
- `$ cd opencv-2.4.10`

Configurar la construcción y copilar

- `$ mkdir build`
- `$ cd build`
- `$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D`

```
BUILD_NEW_PYTHON_SUPPORT=ON -D INSTALL_C_EXAMPLES=ON -  
D INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON
```

- \$ make

Instalar *Open Cv*

- \$ sudo make install
- \$ sudo ldconfig

Prueba de funcionamiento

- \$ workon cv
- \$ python
- >>> import cv2

3.2.2.3. Instalación de python

Se debe instalar la versión 2.7 de *Python*, por ser la más estable y documentada.

Este lenguaje de alto nivel cuenta con una serie de librerías que facilitan de gran manera la programación, para este dispositivo se necesita las siguientes librerías:

Numpy

Librería que permite trabajar con matrices y vectores OPEN CV.

Picamera

Librería de control de la cámara Pi con bus de comunicación SCI.

Pydub

Librería que permite desarrollar un entorno de escritorio.

Pyserial

Librería para la comunicación serial.

La instalación es a través de PIP que busca estas librerías desde el repositorio de *Python*:

- \$ pip install numpy
- \$ pip install picamera
- \$ pip install pydub
- \$ pip install pycserial

3.2.2.4. Instalación modem E303

Los Drivers para Linux del modem vienen en el mismo dispositivo, solo deben darse los permisos de instalación, siguiendo los siguientes pasos:

Se debe copiar la carpeta Linux en la ubicación de archivos temporales "/tmp" como usuario *root*.

Después se debe dar los permisos de instalación con los siguientes comandos:

- # cd /tmp/Linux/
- # chmod +x install
- # ./install

Para verificar la instalación, ver los drivers y el bus utilizado se utilizan los siguientes comandos:

- # lsmod | grep cdc
- # lsusb

3.2.2.5. Desarrollo del programa

El desarrollo del programa se realiza en el lenguaje de alto nivel *Python*, porque soporta programación orientación a objetos, programación imperativa y en menor medida,

programación funcional, además de contar con una sintaxis muy visual, gracias a que maneja una sintaxis indentada (con márgenes), que es de carácter obligatorio.

Python convierte el código fuente en una forma intermedia llamada *bytecodes*, después los traduce en el lenguaje nativo de la computadora y los ejecuta.

La realización del programa será modular, que consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más legible y manejable.

El programa tiene la siguiente arquitectura:

- Raíz
- Módulo iris
- Módulo tiempo
- Módulo registro
- Módulo sms
- Módulo sonido

3.2.2.5.1. Módulo de detección del iris del conductor

Es el módulo más importante del programa y encargado del monitoreo del conductor.

El sensor que captura los datos es la cámara PI, que se comunica con la tarjeta de desarrollo mediante un bus CSI, para poder visualizar lo que emite la cámara por *Python* y *Open CV* gracias a la librería *Picamera*, de la siguiente forma:

Librerías utilizadas

```
1   import cv2
2   import picamera
3   import picamera.array
```

```
##### Inicia la cámara #####
```

```
4     with picamera.PiCamera() as camera:
```

```
5     with picamera.array.PiRGBArray(camera) as stream:
```

```
6     camera.resolution = (512, 384)
```

```
##### Visualización en la pantalla del raspberry pi #####
```

```
7     while True:
```

```
8         camera.capture(stream, 'bgr', use_video_port=True)
```

```
9         cv2.imshow('frame', stream.array)
```

Lo que hace este segmento de código es iniciar la cámara Pi en las “líneas 4 y 5”, con una resolución de 512x384 “línea 6”, se utilizó esta resolución donde la imagen no pierde detalles y a la vez no es muy grande para el análisis de imágenes.

Al estar los *frames* en un bucle, estos se repiten constantemente creando imágenes en movimiento “línea 7”. Estas imágenes son capturadas a color “línea 8” y visualizadas en la pantalla del *Raspberry PI 3B* “línea 9”.

Con este fragmento de código se pueden ver las imágenes emitidas por la cámara Pi.

Para la detección del rostro existe un clasificador nativo en Open cv que se encuentra en la carpeta `haarcascades` de open cv y el archivo es:

haarcascade_frontalface_default.xml

La manera de utilizar este haarcascades es de la siguiente:

```
##### Importación de librerías #####
```

```
1     import numpy as np
```

```
2     import cv2
```

```
##### Cuerpo de programa #####
```

```
3     face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

```
4     img = cv2.imread('prueba.jpg')
```

```
5     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
6     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
```

```
7     for (x,y,w,h) in faces:
```

```
8         cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
```

```
9     cv2.imshow('img',img)
```

El haarcascade pertenece a *Open CV* y *numpy* que es una librería de manejo vectorial “línea 1 y 2”, donde la cascada es almacenada en una variable más corta para su facilidad “línea 3”, se carga la imagen de prueba con el nombre prueba.jpg “línea 4” se realiza una copia de la misma en escala de grises “línea 5”, es importante este paso ya que *Open CV* funciona mucho mejor con una imagen sin color.

Una vez obtenida esta imagen, *Python* corre la librería haarcascade de *Open CV* que utiliza el algoritmo viola and jones “línea 6”, posteriormente se detectan todos los rostros en la imagen se nos brinda una coordenada (x, y) ubicado en parte superior de la izquierda del rostro y dos variables (w, h) que son la distancia vertical y horizontal que ocupa el rostro.

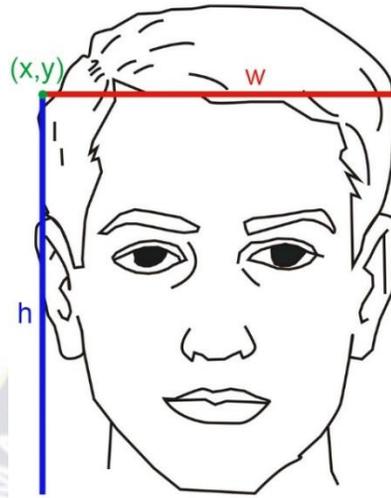


Figura 41. Variables de la cara (x, y, w, h).
Elaboración propia

Dónde:

(x, y): Coordenada de la detección del rostro

w: Longitud horizontal del rostro

h: Longitud vertical del rostro

Obtenidas las coordenadas y distancias, se dibuja un cuadrado donde está todo el rostro, siendo esta una función opcional que permite ver el área de trabajo y el cuadrado es dibujado con una operación matemática donde tenemos los dos vértices equidistantes de un cuadrado:

$$(X1, Y1) = (x, y)$$

$$(X2, Y2) = (x+w, y+h)$$

Con las coordenadas se dibuja un cuadrado en la imagen de color “línea 8” y se muestra en la pantalla “línea 9”.

Con esta operación se define el rostro como área de trabajo, luego se define el área de los ojos.

Haarcascade detecta los ojos pero con un error considerable y al utilizar un nuevo clasificador el dispositivo sufría demasiado retardo, por eso se decidió marcar un área de trabajo con las coordenadas de rostros, considerando como mejores puntos los siguientes:

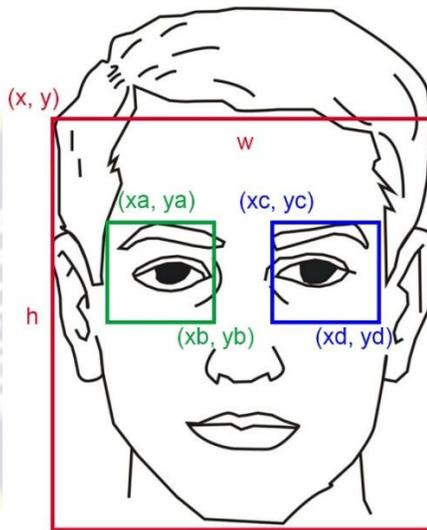


Figura 42. Áreas de detección de la cara y los ojos.
Elaboración propia

Área ojo izquierdo

$$(x_a, y_a) = [(x + (w / 7)), (y + (h / 4))]$$

$$(x_b, y_b) = [(x + (3 * w / 7)), (y + (h / 2))]$$

Área ojo derecho

$$(x_c, y_c) = [(x + (4 * w / 7)), (y + (h / 4))]$$

$$(x_d, y_d) = [(x + (6 * w / 7)), (y + (h / 2))]$$

Contando con el área donde se sitúa los ojos, sin la necesidad de utilizar otro clasificador ahorrando la necesidad de procesos de detección. Nos enfocaremos en la detección del iris en esta área.

Esta detección permitirá saber si la persona cierra los ojos frente al volante durante la conducción, por eso debe enfocarse en la forma que tiene el iris.

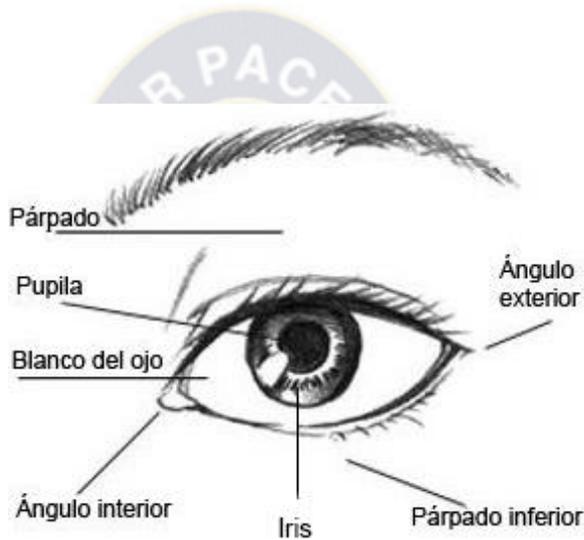


Figura 43. Partes del ojo.
Elaboración propia

En la figura 43 se puede ver que el iris es perfectamente circular, además que posee un color llamativo, diferente al color de piel, rodeado por la esclerótica que es el blanco del ojo, por lo tanto es muy fácil de identificar y monitorear.

La forma de detectar, si el conductor está somnoliento es el monitoreo constante del iris y el descenso del párpado superior, ocultando el iris por un lapso de tiempo mayor a los 3 segundos, donde el dispositivo nos alerta cuando el iris desaparece.

Open CV cuenta con funciones para detectar los círculos, denominado *HoughCircles*, esta función nos devuelve dos resultados que son el punto central del círculo (x,y) y el radio del mismo r.

La transformada de *Hough* es una técnica para la detección de figuras en imágenes digitales. Esta función se basa en la transformada de *Hough*, con esta técnica es posible encontrar todo tipo de figuras que puedan ser expresadas matemáticamente, tales como rectas, circunferencias o elipses.

Esta función tiene la siguiente estructura:

```
cv2.HoughCircles(imagen,metodo,dp,MinDist,param1=None,param2=None,  
minRadius=None,maxRadius=None)
```

Es una de las más difíciles de calcular debido a la cantidad de parámetros que necesita para el buen funcionamiento:

Imagen

La Imagen con que se trabaja y se detecta los círculos, esta necesariamente debe estar en una escala de grises.

Método

Define el método para detectar los círculos en las imágenes. Actualmente, el único método implementado es `cv2.HOUGH_GRADIENT`, que corresponde al método de detección Yuen et al.

DP

Este parámetro es la relación inversa de la resolución del acumulador a la resolución de la imagen, por defecto este es 1.

MinDist

Distancia mínima entre el centro (x, y) las coordenadas de los círculos detectados. Si el `MinDist` es demasiado pequeño, varios círculos en el mismo barrido que el original puede

ser (falsamente) detectado. Si el MinDist es demasiado grande, entonces algunos círculos pueden no detectarse en absoluto.

param1

Valor de gradiente utilizado para manejar la detección de bordes en el método de detección Yuen et al.

param2

Valor umbral del acumulador para el método cv2.HOUGH_GRADIENT. Cuanto menor sea el umbral, se detectarán más círculos (incluyendo círculos falsos).

minRadius

Tamaño mínimo del radio (en píxeles).

maxRadius

Tamaño máximo del radio (en píxeles).

Los valores con mayor rendimiento para esta función son:

DP = 1 valor por defecto

MinDist = 20 por defecto que recomienda Open CV

Param1 = 60 valor con mejores resultados para la detección del borde de la iris

Param2 = 11 valor umbral del método

minRadius = rmi radio mínimo de los círculos que se quieren detectar, definido por la siguiente formula :

$$\mathbf{rmi = \text{redondeo de } (w * 0.02)}$$

maxRadius = rma radio máximo de los círculos que se quieren detectar, definido por la siguiente formula:

$$\text{rma} = \text{redondeo de } (w * 0.06)$$

El segmento de código de cómo se maneja con esta función:

```
##### Importación de librerías #####  
  
1   import cv2  
  
2   import numpy as np  
  
#####Cuerpo de programa #####  
  
3   img = cv2.imread('opencv_logo.png',0)  
  
4   cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)  
  
5   circles=cv2.HoughCircles( img, cv2.HOUGH_GRADIENT, 1, 20, param1=50,  
param2=30, minRadius=0, maxRadius=0)  
  
6   for i in circles[0,:]:  
  
7       cv2.circle(cimg,( i[0],i[1]),i[2],(0,255,0),1)  
  
8       cv2.imshow('detected circles',cimg)
```

Deben importarse las librerías “línea 1 y 2” cargando una imagen y la convirtiéndola en escala de grises “línea 3 y 4”, aplicando la función de detección de círculos “línea 5”, para posteriormente dibujar los círculos detectados en la imagen “línea 6, 7 y 8”.

Llegando a este punto, aplicando todos los algoritmos y métodos mencionados, se puede detectar el rostro del conductor para el monitoreo de somnolencia.

3.2.2.5.2. Reconocimiento de los ojos en función a la variación de la luz

Uno de los factores más influyentes en la somnolencia del conductor es el cambio de luz en el transcurso del día, al ser un factor fuera del entorno de control humano, es necesario realizar las pruebas en un ambiente con variaciones de intensidad de la luz.

Un luxómetro es la herramienta dedicada a la medición de la intensidad de la luz (Lux), por lo inaccesible que es contar con un dispositivo de este tipo, se optó por la utilización de la App Luxómetro que cumple con las siguientes características:



Luxómetro
Crunchy ByteBox
E Todos

Figura 44. Logo App Luxómetro.
Recuperado Play Store.

CARACTERÍSTICAS DE LA APLICACIÓN LUXÓMETRO	
Sistema Operativo	Android
Versión	2.0
Última Actualización	8 marzo del 2016
Programador	crunchybytebox@gmx.de
Permisos	<ul style="list-style-type: none"> • Acceso a red para mostrar los anuncios. • Aplicación Gratuita. • Consulta de modificación a la tarjeta de memoria microSD para guardar grabación.

Tabla 20. Características del App.
Recuperado de Play Store.

La aplicación utiliza las cámaras delantera y posterior del dispositivo móvil, para medir la intensidad de la luz.

Lux es la unidad de medición del nivel de la iluminación, su equivalencia es la siguiente:

$$1 \text{ lux} = 1 \text{ Lumen} / \text{metro}^2$$

lux = Cantidad de luz que tenemos en un metro cuadrado.

Lumen = Unidad del flujo de luminoso.



Figura 45. Entorno grafico utilizando por App.
Recuperado App Luxómetro.

Tabla de comparación

COMPARATIVA DE CANTIDAD DE LUX		
Iluminación	Abreviado	Ejemplo
0.25 lux	250 mlx	Luna llena en una noche despejada
1 lux	1 lx	Luna llena a gran altitud en las latitudes tropicales.
3 lux	3 lx	Limite oscuro del crepúsculo bajo un cielo despejado.
50 lux	50 lx	Sala de una vivienda familiar
80 lux	80 lx	Pasillo/cuarto del baño
400 lux	4 hlx	Oficina bien iluminada
400 lux	4 hlx	Salida o puesta de sol en un día despejado
1000 lux	1 klx	Iluminación habitual en un estudio de televisión

Tabla 21. Tabla comparativa de cantidad de Lux.
Recuperado www.ecured.cu

3.2.2.5.3. Módulo de tiempo

Este módulo está encargado del temporizador que se activa una vez que el conductor inicia la conducción y es el encargado de realizar la comparación constante con el tiempo límite establecido de tres horas, activando una bandera posteriormente se hace esta comparación cada hora, está basado especialmente en las funciones:

`time.time()`

`time.strftime()`

La primera función brinda el valor en segundos del contador interno que tiene la tarjeta de desarrollo, la otra función brinda la fecha y hora.

De todas las directivas que existen para el formato de hora en *Python*, se utilizaran:

- %H - Hora (formato 24 horas)
- %M- Minutos

- %S - Segundos
- %x - Fecha actual

El algoritmo de funcionamiento es el siguiente:

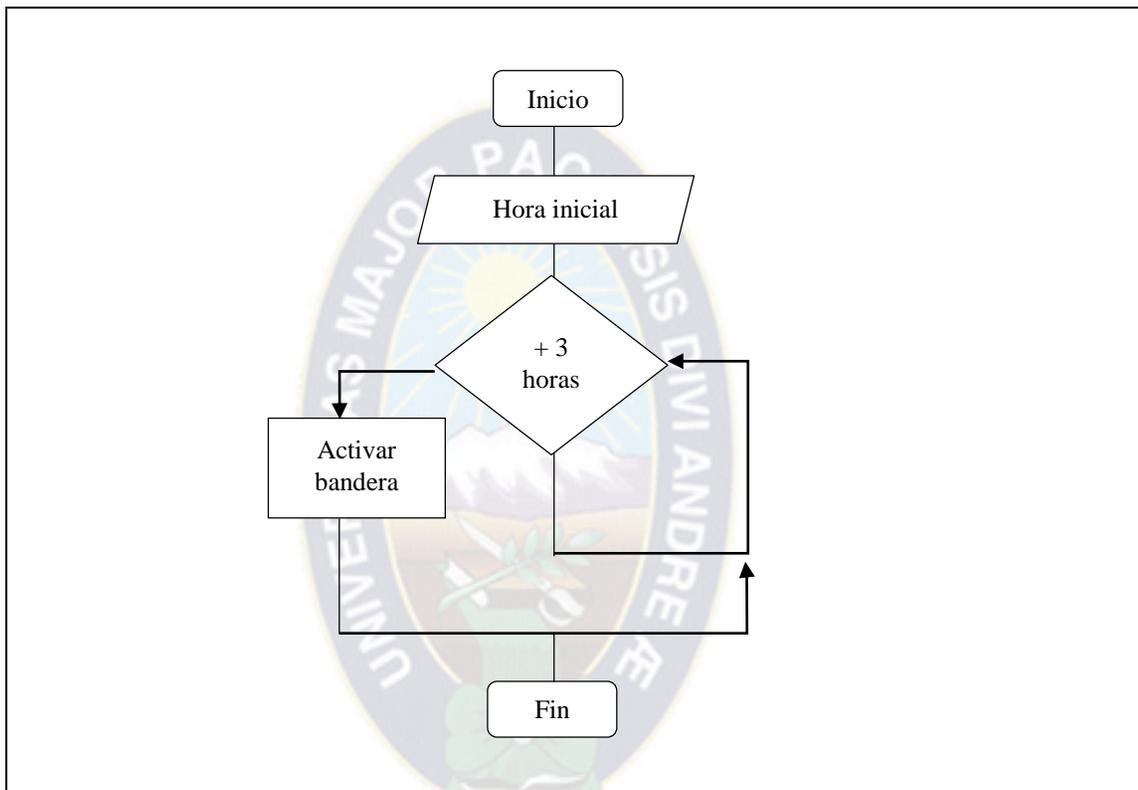


Figura 46. Algoritmo de funcionamiento del timer.

Elaboración propia

3.2.2.5.4. Módulo de registro

Este módulo se encarga de registrar todos los eventos reportados por el dispositivo durante el monitoreo de la somnolencia del conductor, en un archivo de texto en formato “.txt”, con el objetivo de registrar datos que serían útiles para analizar la conducta del conductor y que podrían ser útiles para futuras mejoras del proyecto como el diagnóstico de padecimientos del sueño.

- 1 variable =open("ejemplo.txt","w")
- 2 variable.write(" mensaje para registrar")

3 `variable.close()`

Se crea un archivo .txt en python “línea 1” donde puede escribirse el mensaje que se desee “línea 2”, una vez que acabamos de realizar esta operación procedemos a cerrar el archivo .txt “línea 3”, en el caso del dispositivo, los mensajes de registro son los siguientes:

1. Inicio de conducción.
2. Conductor cierra los ojos aviso 1.
3. Conductor cierra los ojos aviso 2.
4. Conductor sufre cansancio.
5. Se manda un mensaje de texto a un contacto con aviso que el conductor está cansado.
6. Conducción durante tres horas.
7. Conducción más de la hora recomendada.
8. Se manda un mensaje de texto a un contacto con aviso que el conductor excedió las tres horas de conducir.

Estos eventos son registrados con fecha y hora.

3.2.2.5.5. Módulo SMS

Este módulo está encargado de enviar mensajes de textos al familiar o contacto registrado en el dispositivo, el componente encargado de esta transmisión es el modem *Huawei e303*, la comunicación es realizada por los comandos AT que son instrucciones codificadas que conforman un lenguaje de comunicación entre el *Raspberry PI 3B* y un terminal modem, que para establecer esta comunicación se utiliza *Python* y la librería *Piserial*.

Para identificar el puerto conectado al modem se utiliza el siguiente comando:

\$ lsusb

Una vez identificado el puerto, en el dispositivo es “ttyUSB”, se debe indicar a *python* el puerto y los baudios (9600):

```
sSerie = serial.Serial("/dev/ttyUSB0", 9600)
```

Se deben utilizar los comandos AT para mandar órdenes que debe realizar el modem:

ATZ : Resetear el modem

AT+CMGF=1 : Activación del modo para envío de SMS

AT+CMGS=XXXXXXXX : Número al que debe enviarse el SMS

> **Aquí introducimos el mensaje a enviar** : Mensaje

Ctrl+z: Enviar mensaje

Utilizando esta serie de comandos AT se puede enviar un SMS al número deseado.

3.2.2.5.6. Módulo de sonido

Existe una gran variedad de librerías, que permiten reproducir sonidos con *Python*, el problema de estas librerías es que consumen demasiados recursos que afectan la reproducción, haciéndola lenta y entrecortada, por este motivo se vio conveniente utilizar ALSA, un paquete externo a *Python*.

ALSA, Arquitectura de Sonido Avanzada para *Linux* (*Advanced Linux Sound Architecture*), es un *software* de sonido para *GNU/Linux*.

```
1 import os
2 os.system("aplay aviso.wav")
```

Se importa el módulo os, que permite acceder a funcionalidades dependientes del Sistema Operativo, en especial aquellas que refieren información sobre el entorno del mismo y permiten manipular la estructura de directorios “línea 1”, el *software* de sonido ALSA es invocado para reproducir el archivo de sonido “línea 2”.

3.2.3. Presupuesto

El presupuesto en moneda boliviana empleado para el ensamblaje del dispositivo de monitoreo de la somnolencia y fatiga del conductor es el siguiente:

Dispositivo

N°	Descripción	Cantidad	Precio unitario	Valor (Bs.-)
1	Tarjeta de desarrollo Raspberry PI 3B	1	450.00	450.00
2	Cámara PI con visión Nocturna	1	400.00	400.00
3	Memoria micro SD de 32 GB	1	70.00	70.00
4	Modem Huawei	1	90.00	90.00
5	Tarjeta SIM GSM Entel con crédito	1	50.00	50.00
6	Fuente de Alimentación de 5.1 V y 2 A.	1	90.00	90.00
7	Caja metálica para la protección del circuito	1	150.00	150.00
8	Amplificador Operacional TDA2822M	1	5.00	5.00
9	Capacitores Electrolíticos	5	2.00	10.00
10	Capacitores Cerámicos	2	0.50	1.00
11	Resistencia	1	0.30	0.30
12	Potenciómetro	2	2.00	4.00
13	Placa PBC	1	12.00	12.00
Sub-Total				1332.30

Equipamientos

N°	Descripción	Cantidad	Precio unitario	Valor (Bs.-)
1	Teclado USB	1	42.00	42.00
2	Mouse USB	1	35.00	35.00
3	Cable HDMI	1	40.00	40.00
4	Conversor HDMI a VGA	1	70.00	70.00
Sub-Total				187.00

Suministros

N°	Descripción	Cantidad	Precio unitario	Valor (Bs.-)
1	PC Intel(R) Core(TM) I5	1	5000.00	5000.00
2	Impresora Epson L210	1	1300.00	1300.00
3	Suscripción a Internet	1	300.00	300.00
4	Hojas de papel bond, tamaño carta	2	30.00	60.00
Sub-Total				6660.00

Otros

N°	Descripción	Cantidad	Precio unitario	Valor (Bs.-)
1	Transporte	365 (días)	4.00	1460.00
Sub- Total				1460.00

Total

	ITEM	Valor (Bolivianos)
1	Dispositivo	1332.30
2	Equipamientos	187.00
3	Suministros	6660.00
4	Otros	1460.00
Valor total		9639.30

Tabla 22. Presupuestó del proyecto de grado.

Elaboración propia

El monto total gastado para la realización de este proyecto es de **nueve mil seiscientos treintainueve Bolivianos con treinta centavos (9639.30 Bs)**

3.2.4. Cronograma

ACTIVIDADES	Octubre				Noviembre				Diciembre				Enero				Febrero			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Investigación de la normativa y estudios relacionados con el problema del sueño y conductas frente al volante																				
Investigación de las tecnologías para utilizar en el proyecto																				
Implementación de las tecnologías de alerta de la somnolencia y fatiga del conductor																				
<ul style="list-style-type: none"> • Reconocimiento facial 																				
<ul style="list-style-type: none"> • Detección de la iris 																				
<ul style="list-style-type: none"> • Transmisión de mensajes de texto SMS 																				
<ul style="list-style-type: none"> • Integración de los módulos 																				
Prueba y soluciones de errores del dispositivo																				
Obtención de resultados																				
Presentación proyecto																				

Tabla 23. Cronograma
Elaboración propia

CAPITULO IV

PRUEBAS Y RESULTADOS

El objetivo principal del dispositivo es la prevención de siniestros automovilísticos, por lo que debe permanecer en constante monitoreo del rostro del conductor, para activarse la alerta en caso de somnolencia.

Las pruebas de funcionamiento del dispositivo estaban enfocadas a la detección del rostro, el área de los ojos, el parpadeo del conductor y el monitoreo en tiempo real.

Planteando los siguientes casos de prueba:

1. Prueba de reconocimiento facial para diferentes personas.
2. Prueba de detección del iris para diferentes personas.
3. Prueba de detección del iris para diferentes distancia.
4. Prueba de detección del iris para variaciones de luz.

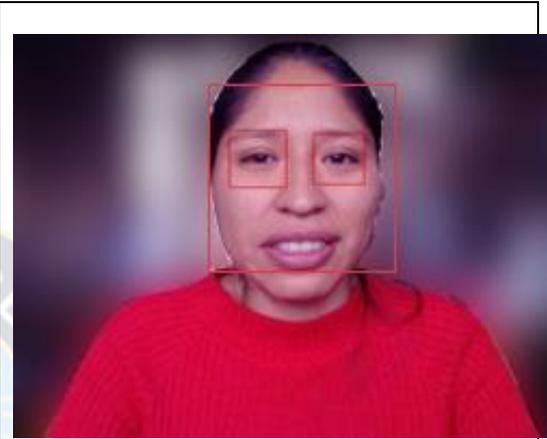
4.1.PRUEBA DE RECONOCIMIENTO FACIAL PARA DIFERENTES PERSONAS

El dispositivo debe detectar el rostro de diferentes personas, graficando un cuadrado para el rostro y un rectángulo en los ojos, sin presentar problemas ante las diferencias de facciones y rasgos de diferentes personas.

Para probar la efectividad del algoritmo y la robustez del dispositivo se realizaron las pruebas con diferentes personas.



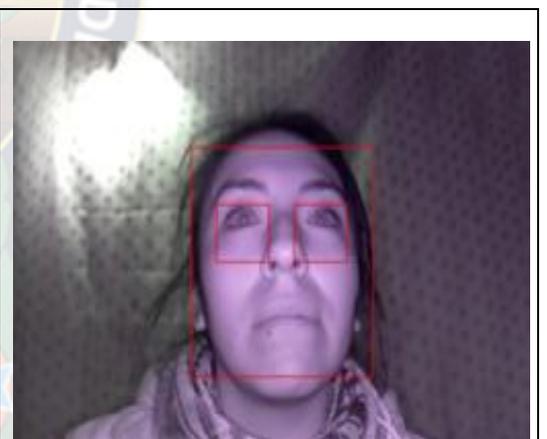
Edad: 29
Sexo: Masculino



Edad: 29
Sexo: Femenino



Edad: 24
Sexo: Masculino



Edad: 26
Sexo: Femenino



Figura 47. Detección de distintas facciones.
Elaboración propia

Las pruebas realizadas para el reconocimiento facial fueron satisfactorias, el dispositivo pudo realizar el reconocimiento del rostro y ojos de diferentes personas en un rango de edad de 24 a 63 años, en ambos sexos.

4.2.PRUEBA DE DETECCIÓN DEL IRIS PARA DIFERENTES PERSONAS

El dispositivo debe reconocer el iris de las personas, graficando un circulo en el lugar del iris, para probar la efectividad del dispositivo se realizaron las pruebas con diferentes personas.

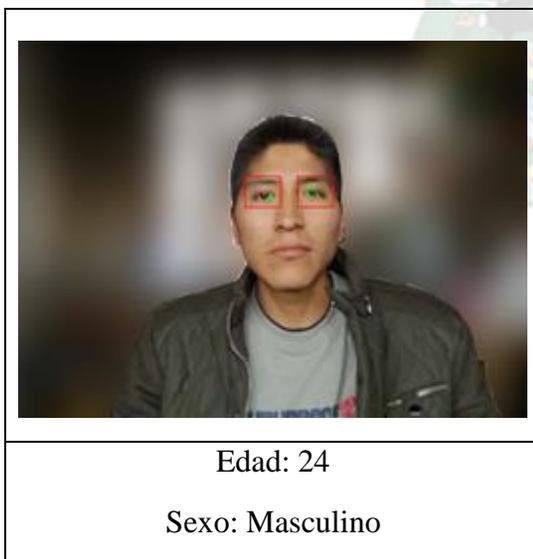
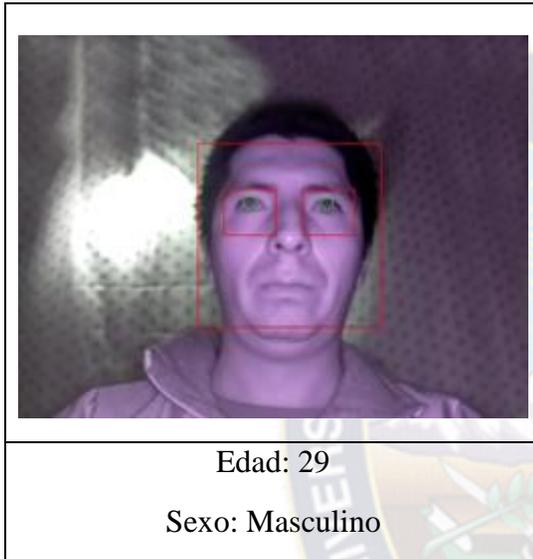




Figura 48. Detección del iris en distintos rostros.
Elaboración propia

Las pruebas realizadas para comprobar la funcionalidad de detección del iris en diferentes personas fueron satisfactorias.

4.3.PRUEBA DE DETECCIÓN DEL IRIS PARA DIFERENTES DISTANCIA

El dispositivo deberá reconocer los ojos del conductor de forma óptima, en un rango de distancia entre la cámara y el conductor de 41 a 50 cm.

Para probar la detección del iris en función de la distancia entre el dispositivo y el conductor, se realizaron 100 pruebas, con distancias entre 30, 35, 40, 45, 50, 55, y 60 cm.

Obteniendo los siguientes resultados:



Figura 49. Pruebas de Distancias para la detectar el iris.
Elaboración propia

N	DISTANCIAS	ACIERTOS
1	30 cm	15/100
2	35 cm	35/100
3	40 cm	78/100
4	45 cm	94/100
5	50 cm	100/100
6	55 cm	100/100
7	60 cm	100/100

Tabla 24. Resultados de aciertos al detectar el pestaño.
Elaboración propia

Las pruebas de funcionamiento del dispositivo en un rango óptimo de 41 a 50 cm entre la cámara y el conductor, fueron satisfactorias.

Rango de distancia de 30 a 40 cm (demasiado cerca)

Se presenta dificultad en la detección del iris del conductor, los gráficos muestran círculos falsos.

Rango de distancia de 50 a 60 cm (demasiado lejos)

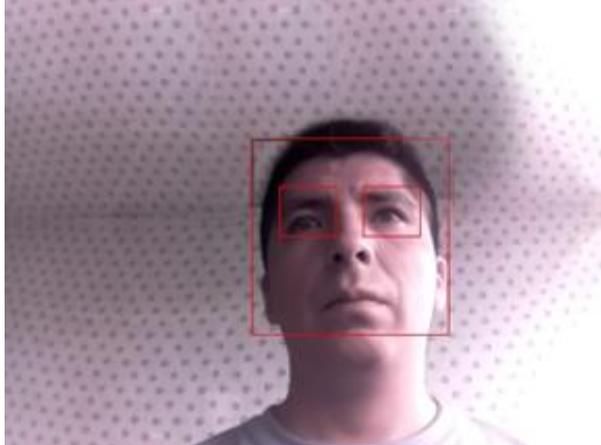
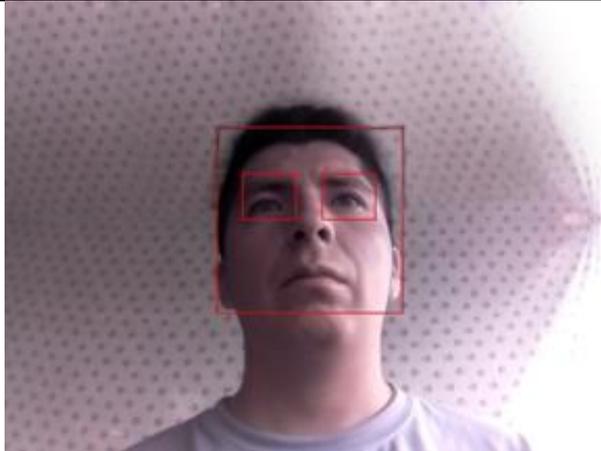
El dispositivo presenta dificultad en la detección de iris del conductor.

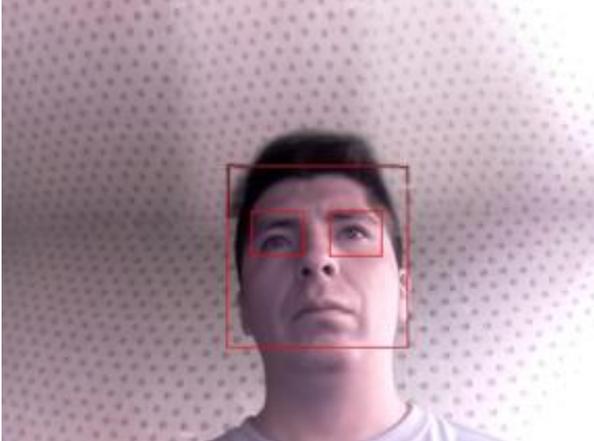
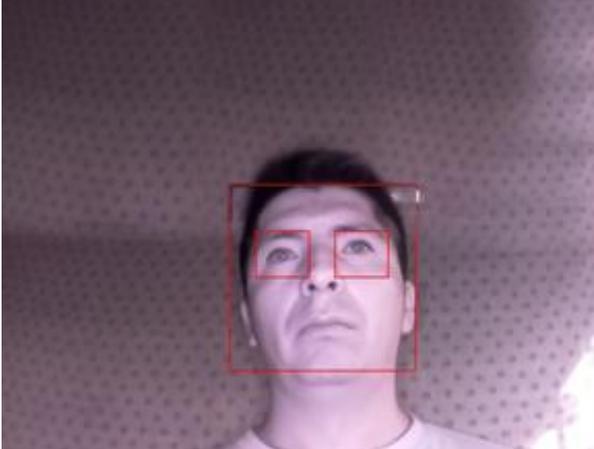
4.4.PRUEBA DE DETECCIÓN DEL IRIS PARA VARIACIONES DE LUZ

El dispositivo debe detectar de forma satisfactoria los ojos y el iris del conductor en cualquier momento del día y la noche.

Para probar esta funcionalidad del dispositivo se realizaron las pruebas en diferentes horarios del día y la noche.

Captura	Horas	Iluminación	Aciertos (de 10 pruebas)										
	11:00 am	4428 lux	<table border="1" data-bbox="1268 531 1487 747"> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> </table> <p data-bbox="1352 751 1425 779">10/10</p>	X	X	X	X	X	X	X	X	X	X
X	X												
X	X												
X	X												
X	X												
X	X												
	12:00 pm	5290 lux	<table border="1" data-bbox="1268 993 1487 1209"> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td></td></tr> <tr><td>X</td><td>X</td></tr> <tr><td></td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> </table> <p data-bbox="1352 1213 1425 1241">8/10</p>	X	X	X		X	X		X	X	X
X	X												
X													
X	X												
	X												
X	X												
	01:00 pm	2134 lux	<table border="1" data-bbox="1268 1455 1487 1671"> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td></td></tr> <tr><td>X</td><td>X</td></tr> </table> <p data-bbox="1352 1675 1425 1703">9/10</p>	X	X	X	X	X	X	X		X	X
X	X												
X	X												
X	X												
X													
X	X												

	02:00 pm	1435 lux	<table border="1" data-bbox="1268 405 1487 621"> <tbody> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td></td><td>X</td></tr> <tr><td>X</td><td></td></tr> <tr><td>X</td><td>X</td></tr> </tbody> </table> <p data-bbox="1360 625 1419 653">8/10</p>	X	X	X	X		X	X		X	X
X	X												
X	X												
	X												
X													
X	X												
	03:00 pm	1342 lux	<table border="1" data-bbox="1268 863 1487 1079"> <tbody> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> </tbody> </table> <p data-bbox="1352 1083 1427 1110">10/10</p>	X	X	X	X	X	X	X	X	X	X
X	X												
X	X												
X	X												
X	X												
X	X												
	04:00 pm	697 lux	<table border="1" data-bbox="1268 1325 1487 1541"> <tbody> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> </tbody> </table> <p data-bbox="1352 1545 1427 1572">10/10</p>	X	X	X	X	X	X	X	X	X	X
X	X												
X	X												
X	X												
X	X												
X	X												

	05:00 pm	666 lux	<table border="1" data-bbox="1268 405 1485 621"> <tbody> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> </tbody> </table> <p data-bbox="1352 625 1425 653">10/10</p>	X	X	X	X	X	X	X	X	X	X
X	X												
X	X												
X	X												
X	X												
X	X												
	06:00 pm	65 lux	<table border="1" data-bbox="1268 863 1485 1079"> <tbody> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> </tbody> </table> <p data-bbox="1352 1083 1425 1110">10/10</p>	X	X	X	X	X	X	X	X	X	X
X	X												
X	X												
X	X												
X	X												
X	X												
	07:00 pm	1 lux	<table border="1" data-bbox="1268 1325 1485 1541"> <tbody> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> </tbody> </table> <p data-bbox="1352 1545 1425 1572">10/10</p>	X	X	X	X	X	X	X	X	X	X
X	X												
X	X												
X	X												
X	X												
X	X												

	08:00 pm	0 lux	<table border="1"> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td>X</td><td>X</td></tr> <tr><td colspan="2">10/10</td></tr> </table>	X	X	X	X	X	X	X	X	X	X	10/10	
X	X														
X	X														
X	X														
X	X														
X	X														
10/10															

Tabla 25. Detección del rostro en distintos cambios de luz.

Elaboración propia

El dispositivo hace la detección de los ojos y el iris del conductor en cualquier momento del día y la noche de forma satisfactoria.

Incluso se pudo comprobar que la detección es más efectiva durante las horas de la noche.

CAPITULO V

MARCO CONCLUSIVO

5.1.CONCLUSIONES

En el diseño final del hardware del dispositivo de monitoreo de la somnolencia del conductor se utilizó una tarjeta de desarrollo *Raspberry PI 3B*, como unidad de procesamiento de datos, una cámara de visión nocturna como periférico de entrada de datos y un módulo de alerta conformado por un modem (*Huawei E303*) y un amplificador de audio.

Se utilizó el lenguaje de programación de alto nivel *Python*, para la elaboración del código que controla el dispositivo. El dispositivo reconoce el iris del conductor de forma satisfactoria y activa el módulo de alerta, en caso de cierre de parpados durante un tiempo mayor a 3 segundos (somnolencia).

Las pruebas de funcionalidad en función de distancia entre la cámara y el conductor, variaciones de luz, monitoreo de cantidad de horas frente al volante y reconocimiento del iris en diferentes personas, fueron satisfactorias, por lo que el dispositivo cumple de forma satisfactoria con los objetivos planteados en el proyecto.

5.2.RECOMENDACIONES

Para una detección óptima de los ojos del conductor, la cámara del dispositivo deberá estar ubicada a una distancia de 40 cm del rostro del conductor.

El suministro de energía ideal para el funcionamiento del dispositivo debe ser mínimo de una tensión constante de 5v y 2 amperios.

El proyecto podría abarcar diferentes aplicaciones y funcionalidades más allá de las presentadas en el proyecto, pero se necesitaría de tiempo e inversión económica mayor.



BIBLIOGRAFÍA

Comisaria Europea del Automóvil. (2017). Seguridad activa y pasiva del vehículo. Recuperado de <https://www.cea-online.es/blog/128-seguridad-activa-y-pasiva-del-vehiculo>

Noticias cohes.com. (2014). Los sistemas de seguridad que no pueden faltar en tu coche. Recuperado de <http://noticias.coches.com/consejos/los-sistemas-de-seguridad-que-no-pueden-faltar-en-tu-coche/108496>

Autopista.es. (2015). El volante que detecta si te cansas conduciendo <http://www.autopista.es/tecnologia/articulo/volante-detecta-cansancio-somnolencia-conductor-sensor-presion-104319>

Barba Guaman, Luis Rodrigo. (2015). Utilización de métodos de visión artificial para PC como apoyo en la automoción. Recuperado de <http://oa.upm.es/44219/>

Parra Barrero Eloy (2015). Aceleración del algoritmo de Viola-Jones mediante rejillas de procesamiento masivamente paralelo en el plano focal. Recuperado de <https://idus.us.es/xmlui/handle/11441/30319>

Lowe, D. (2001). The Viola/Jones Face Detector. Recuperado de <http://www.cs.ubc.ca/~lowe/425/slides/13-ViolaJones.pdf>

Catarina U. (2015). Clasificadores Débiles – AdaBoost. Recuperado de http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/morales_s_aa/capitulo3.pdf

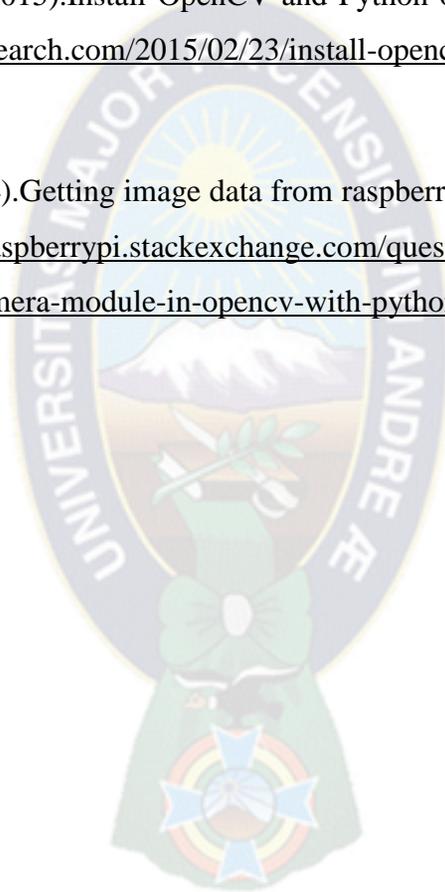
Celeste Campo - Carlos García Rubio. (2011). Arquitecturas y tecnologías inalámbricas. Recuperado de <http://ocw.uc3m.es/ingenieria-telematica/aplicaciones-moviles/material-de-clase-2/inalambricos>

Huawei. (2014). E303 Características. <http://consumer.huawei.com/cl/mobile-broadband/dongles/features/e303-cl.htm>

Electrónica estudiantil. (2015). Instructivo De Comandos AT. <http://www.electronicaestudio.com/docs/ISTD-034.pdf>

Adrian Rosebrock. (2015). Install OpenCV and Python on your Raspberry Pi 2 and B+ <http://www.pyimagesearch.com/2015/02/23/install-opencv-and-python-on-your-raspberry-pi-2-and-b/>

Stackexchange. (2014). Getting image data from raspberry pi camera module in OpenCV with Python. <http://raspberrypi.stackexchange.com/questions/24262/getting-image-data-from-raspberry-pi-camera-module-in-opencv-with-python>



ANEXOS

CÓDIGO MÓDULO RAIZ

```
import time

from modulos.modulo_iris import iris
from modulos.modulo_tiempo import data,time_conduccion
from modulos.modulo_texto import base_dato,nuevo
from modulos.sonido import voz
from modulos.sms import mensaje

#### Inicio ####

dato_1,dato_2 = data(1)#forzamos la salida de fecha
base_dato(dato_1,dato_2,0)#comandos de incoación fecha
tempo = int(time.time())#temporizador de inicio de conducción
tempo = tempo + 60 #primera vez que toca se manda el dato esto es en horas (3)
reg = 4 # sensibilidad de detección
cont = 0 # variable definida
cont_2 = 0 # variable definida
cond = 0 # variable definida
salir = 0 #variable para romper el bucle y salir del programa

##### Cuerpo #####

while(True):

    band_1,tempo,band_2,salir = iris(reg,tempo)
```

```

if band_1 == 1:
    dato_1,dato_2 = data(band_1)
    cont = cont + 1
    if cont == 3:
        base_dato(dato_1,dato_2,3)
        base_dato(dato_1,dato_2,4)
        voz(cont)
        mensaje(1)# envía mensaje de texto que se duerme el conductor
    elif cont >= 4:
        cont = 0
    else:
        base_dato(dato_1,dato_2,cont)
        voz(cont)
#bandera 2 del tiempo de conducción
if band_2 == 1:
    dato_1,dato_2 = data(band_2)
    cont_2 = cont_2 + 1
    if cont_2 >= 2:
        base_dato(dato_1,dato_2,6)
        base_dato(dato_1,dato_2,7)
        cond = 5
        voz(cond)# reproduce los audio según "cond"
        cond = 0

```

```

    mensaje(3)# mensaje de texto que el conductor conduce más de tres horas
else:
    base_dato(dato_1,dato_2,5)
    base_dato(dato_1,dato_2,7)
    cond = 4
    voz(cond)# aquí reproduce los audio según "cond"
    cond = 0
    mensaje(2)#mensaje de texto que se excedió las tres horas de conducción
if salir == 1:
    break

```

CÓDIGO MÓDULO IRIS

```

import cv2
import picamera
import picamera.array
import numpy as np
import time
from modulos.modulo_tiempo import time_conduccion
def iris(n,tempo):
    n1 = 0
    n2 = 0
    nn = 0

```

```

bandera = 0

salir = 0

face_cascade = cv2.CascadeClassifier('/home/pi/opencv-
3.1.0/data/haarcascades/haarcascade_frontalface_default.xml')

eye_cascade = cv2.CascadeClassifier('/home/pi/opencv-
3.1.0/data/haarcascades/haarcascade_eye.xml')

with picamera.PiCamera() as camera:

    with picamera.array.PiRGBArray(camera) as stream:

        #camera.resolution = (640, 480)
        camera.resolution = (512, 384)

        while True:

            # aqui se agrega el temporizador
            tempo,ban = time_conduccion(tempo)

            #####

            camera.capture(stream, 'bgr', use_video_port=True)

            img = stream.array

            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

            faces = face_cascade.detectMultiScale(gray, 1.3, 5)

            for (x,y,w,h) in faces:

                #cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),1)

                #ojo 1 cordenadas

                xa = x+(w/7)

                ya = y+(h/4)

                xb = x+(3*w/7)

```

```

yb = y+(h/2)

#ojo 2 cordenadas

xc = x+(4*w/7)

yc = y+(h/4)

xd = x+(6*w/7)

yd = y+(h/2)

rma = int(round(w*0.06))
rmi = int(round(w*0.02))

#cv2.rectangle(img,(xa,ya),(xb,yb),(0,0,255),1)

roi1_gray = gray[ya:yb, xa:xb]
roi1_color = img[ya:yb, xa:xb]

#cv2.rectangle(img,(xc,yc),(xd,yd),(0,0,255),1)

roi2_gray = gray[yc:yd, xc:xd]
roi2_color = img[yc:yd, xc:xd]

circulo1 =
cv2.HoughCircles(roi1_gray,cv2.HOUGH_GRADIENT,1,20,param1=60,param2=11,mi
nRadius=rmi,maxRadius=rma)

circulo2 =
cv2.HoughCircles(roi2_gray,cv2.HOUGH_GRADIENT,1,20,param1=60,param2=11,mi
nRadius=rmi,maxRadius=rma)

#dibujo circulo ojo 1

if circulo1 is None:

    nn = nn + 1

    n1 = n1 + 1

```

```

print (n1)

print ("Cerrado")

else:

    nn = 0

    n2 = n2 + 1

    print (n2)

    print ("abierto")

    for i in circulo1[0,:]:

        cv2.circle(roi1_color,(i[0],i[1]),i[2],(0,255,0),1)

#dibujo circulo ojo 2

if circulo2 is None:

    nn = nn + 1

    n1 = n1 + 1

    print (n1)

    print ("Cerrado")

else:

    nn = 0

    n2 = n2 + 1

    print (n2)

    print ("Abierto")

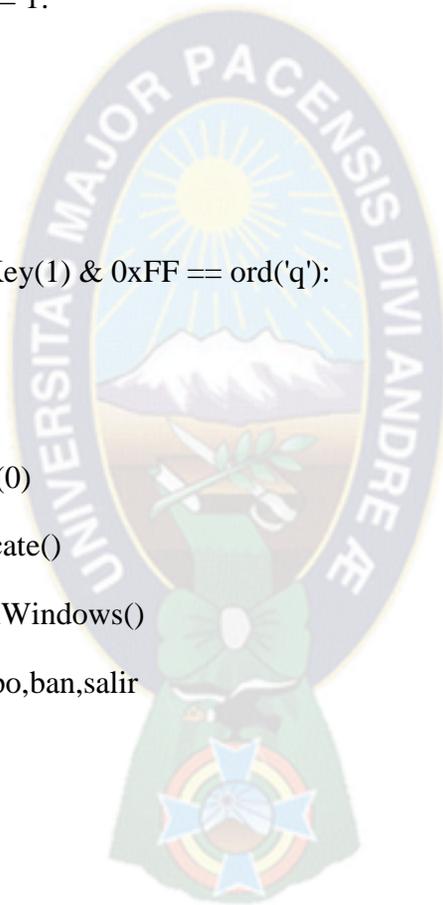
    for i in circulo2[0,:]:

        cv2.circle(roi2_color,(i[0],i[1]),i[2],(0,255,0),1)

# Sentencia encargada de la sensibilidad de detección ALARMA

```

```
    if nn >= n:
        bandera = 1
        nn = 0
cv2.imshow('camara', img)
if bandera == 1:
    break
if ban == 1:
    break
if cv2.waitKey(1) & 0xFF == ord('q'):
    salir = 1
    break
stream.seek(0)
stream.truncate()
cv2.destroyAllWindows()
return bandera,tempo,ban,salir
```



CÓDIGO MÓDULO TIEMPO

```
import time

def data(n):

    if n == 1:

        ## Representación del tiempo

        fecha = ("Fecha " + time.strftime("%x"))

        ## Representación de la hora

        hora = ("Hora " + time.strftime("%H:%M:%S"))

        return fecha,hora

def time_conduccion(tempo):

    bandera = 0

    tiempo = int(time.time())

    if tiempo >= tempo:

        print "---Alerta de tiempo de Conduccion---"

        tiempo = tiempo + 30#una vez llegado al número inicial se suma 10 seg

        bandera = 1

    return tiempo,bandera
```

CÓDIGO MÓDULO REGISTRO

```
def nuevo(n):  
    f=open("datos_conduccion.txt","w")  
    f.write(" FECHA \t HORA \tSITUACION\n")  
    f.close()  
  
def base_dato(n1,n2,sit):  
    print n1,n2  
    nn1 = repr(n1)  
    nn2 = repr(n2)  
    f=open("datos_conduccion.txt","a")  
    f.write(nn1 + "\t" + nn2 + "\t" )  
    if sit == 0:  
        f.write("\t Inicio de conduccion\n")  
    elif sit == 1:  
        f.write("\t Conductor cierra los ojos aviso 1\n")  
    elif sit == 2:  
        f.write("\t Conductor cierra los ojos aviso 2\n")  
    elif sit == 3:  
        f.write("\t Conductor sufre cansancio\n")  
    elif sit == 4:  
        f.write("\t Se manda un mensaje de texto a un contacto con aviso que el conductor  
esta cansado\n")
```

```
elif sit == 5:  
    f.write("\t Conduccion durante tres horas\n")  
elif sit == 6:  
    f.write("\t Conduccion mas de la hora recomendada\n")  
elif sit == 7:  
    f.write("\t Se manda un mensaje de texto a un contacto con aviso que el conductor  
excedio las tres horas de conducir\n")  
f.close()
```



CÓDIGO MÓDULO SMS

```
import serial

from curses import ascii

def mensaje(aviso):

    sSerie = serial.Serial("/dev/ttyUSB0", 9600)

    try:

        sSerie.write("ATZ\r\n")# receteamos el moden

        sSerie.write("AT+CMGF=1\r\n")# colocamos en modo para enviar SMS

        #sSerie.write("AT+CSCS=\"IRA\"\r\n")# cambia al formato internacional

        sSerie.write("AT+CMGS=\"75844304\"\r\n")#número para envió de SMS

        if aviso == 1:

            sSerie.write("Se advierte que el conductor esta sufriendo cansancio" +
            ascii.ctrl("z"))

        elif aviso == 2:

            sSerie.write("Se advierte que el conductor ha conducido por tres horas seguidas"
            + ascii.ctrl("z"))

        elif aviso == 3:

            sSerie.write("Se advierte que el conductor a conducido mas de tres horas
            recomendadas" + ascii.ctrl("z"))

        print sSerie.readline()

        print sSerie.readline()

        print sSerie.readline()

    except ValueError:
```

```
print "error"
```

CÓDIGO MÓDULO SONIDO

```
import os

def voz(avisos):
    if avisos == 1:
        os.system("aplay /home/pi/avisos_1.wav")
    elif avisos == 2:
        os.system("aplay /home/pi/avisos_2.WAV")
    elif avisos == 3:
        os.system("aplay /home/pi/avisos_3.WAV")
    elif avisos == 4:
        os.system("aplay /home/pi/avisos_4.WAV")
    elif avisos == 5:
        os.system("aplay /home/pi/avisos_5.WAV")
    elif avisos == 6:
        os.system("aplay /home/pi/avisos_6.WAV")
```