



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UNA ALARMA BASADA EN LA  
DETECCIÓN DE SOMNOLENCIA PARA CONDUCTORES  
DE VEHICULOS MEDIANTE VISIÓN ARTIFICIAL

AUTOR

Jhon Henry Revelo Rosero

AÑO

2019



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UNA ALARMA BASADA EN LA DETECCIÓN DE  
SOMNOLENCIA PARA CONDUCTORES DE VEHICULOS MEDIANTE VISION  
ARTIFICIAL

Trabajo de Titulación presentado en conformidad a los requisitos establecidos  
para optar por el título de Ingeniero en Redes y Telecomunicaciones.

Profesor Guía

Mg. Iván Ricardo Sánchez Salazar

Autor

Jhon Henry Revelo Rosero

Año

2019

### **DECLARACIÓN PROFESOR GUÍA.**

“Declaro haber dirigido este trabajo, Implementación de una alarma basada en la detección de somnolencia para conductores de vehículos mediante visión artificial, a través de reuniones periódicas con el estudiante Jhon Henry Revelo Rosero, en el semestre 201920, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

---

Iván Ricardo Sánchez Salazar  
Magíster en Calidad, Seguridad y Ambiente.  
C.C. 18030456142

## **DECLARACIÓN PROFESOR CORRECTOR.**

“Declaro haber revisado este trabajo, Implementación de una alarma basada en la detección de somnolencia para conductores de vehículos mediante visión artificial, del estudiante Jhon Henry Revelo Rosero, en el semestre 201920, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

---

Nathaly Verónica Orozco Garzón

Doctora en Ingeniería Eléctrica en el Área de Telecomunicaciones y  
Telemática.

C.C. 1720938586

### **DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE.**

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”.

---

Jhon Henry Revelo Rosero

C.C. 0401680533

## AGRADECIMIENTOS

Agradezco a Dios por guiarme siempre en cada etapa de mi vida.

A mi familia quienes de alguna u otra forma me han brindado su apoyo incondicionalmente en todo momento.

A mí madre quien es el pilar fundamental de mi vida y es la persona que me motiva a superarme cada día.

A mis profesores, quienes contribuyeron en mi formación profesional.

## DEDICATORIA

Dedico este trabajo a mi familia, en especial a mi madre que con su cariño, esfuerzo diario y apoyo incondicional ha permitido que yo pudiera alcanzar este logro tan importante.

## RESUMEN

El presente proyecto elabora un sistema de alarma auditiva la cual permite alertar a los conductores de vehículos si presentan estado de somnolencia, por medio de la utilización de procesamiento de imágenes, permitiendo el seguimiento de las coordenadas del rostro al igual que los ejes de los ojos y posterior a ello el pestañeo, el método usado es CONT este contador se guía por el tiempo que la etapa de detección de la relación de los ojos no detecta a estos abiertos. Mediante el desarrollo del proyecto se pretende que el sistema sea flexible con ello no obstaculice con la visibilidad del conductor a la hora de conducir ya sea realizado en cualquier periodo del día.

Para el desarrollo del proyecto fue indispensable establecer los requisitos funcionales con ello se logró una correcta elección de elementos usados en el sistema. Asimismo, se tomó en cuenta un estudio sobre la situación actual permitiendo definir los parámetros para la elaboración del diseño del proyecto. El sistema funciona mediante el monitoreo sobre la fisonomía de los ojos y exceso de pestañeo hacia el conductor, al momento de mostrar signos de somnolencia o mostrar cambios fisonómicos el sistema procede a la activación de una alerta auditiva.

Los elementos utilizados para la elaboración del proyecto tanto en software como hardware fueron; Raspberry Pi 3 modelo b +, cámara Pi NoIR V2, *software* de procesamiento *OpenCV* y el software de programación Python estos elementos ayudaron a cumplir con las necesidades que presenta el sistema de alarma.

El prototipo es acoplable al vehículo, de esta manera se logrará demostrar la funcionalidad del sistema. Los resultados alcanzados por medio de cada fase de funcionalidad se deducen que los conductores de vehículos presentan sueño en un horario no tan habitual como son altas horas de la noche. El prototipo reacciona de una manera eficiente bajo la existencia de estado de somnolencia, dando como resultados de fallas casi nulas, por esto el sistema es totalmente funcional frente a los conductores de vehículos.

## **ABSTRACT**

The present project develops an audible alarm system which allows vehicle drivers to be alerted if they have a state of drowsiness, through the use of image processing, allowing the tracking of the coordinates of the face as well as the axes of the eyes and after that the blinking, the method used is CONT this counter is guided by the time that the stage of detection of the relationship of the eyes does not detect these open ones. Through the development of the project, it is intended that the system be flexible with it not hindering the visibility of the driver when driving either made at any time of the day.

For the development of the project, it was essential to establish the functional requirements, thereby achieving a correct choice of elements used in the system. Likewise, a study on the current situation was taken into account, allowing to define the parameters for the elaboration of the project design. The system works by monitoring the physiognomy of the eyes and excessive blinking towards the driver, at the time of showing signs of drowsiness or showing physiognomic changes the system proceeds to the activation of an auditory alert.

The elements used for the development of the project in both software and hardware were; Raspberry Pi 3 model b +, camera Pi NoIR V2, OpenCV processing software and Python programming software these elements help meet the needs of the alarm system.

The prototype is attachable to the vehicle; in this way it will be possible to demonstrate the functionality of the system. The results achieved through each phase of functionality are deduced that drivers of vehicles have sleep in a not so usual schedule as they are late at night. The prototype reacts in an efficient way under the existence of state of drowsiness, giving like results of almost null faults, for this reason the system is totally functional in front of the drivers of vehicles.

# ÍNDICE

1	Introducción.....	1
1.1	Objetivo General.....	2
1.2	Objetivos Específicos .....	2
1.3	Alcance.....	2
1.4	Justificación .....	3
2	Marco Teórico .....	3
2.1	Distracción y somnolencia al conducir .....	4
2.1.1	Distracción al conducir .....	4
2.1.2	Tipos de distracción .....	5
2.1.2.1	Distracción interna.....	5
2.1.2.2	Distracción externa.....	6
2.1.3	Somnolencia al conducir .....	6
2.1.3.1	Estados fisiológicos a causa de somnolencia .....	7
2.2	Proceso de conducción en horas del día .....	7
2.3	Sistemas para la detección de somnolencia .....	8
2.3.1	Sistema basado en medición de parámetros .....	9
2.3.2	Sistema basado en el reconocimiento facial .....	10
2.4	Visión artificial.....	11
2.5	Procesamiento de imágenes .....	11
2.6	Algoritmos para la detección facial .....	12
2.6.1	Algoritmo de Viola & Jones .....	12
2.6.2	Algoritmo patrón binario local.....	15
2.6.3	Algoritmo de enfoque basado en características elásticas.....	16
2.6.4	Selección del algoritmo para el análisis facial del proyecto.....	17

2.7 Tecnologías basadas en la detección de somnolencia y distracción .....	19
2.8 Software libre.....	22
2.9 Hardware .....	23
2.9.1 Comparación entre herramientas de hardware .....	24
3 Metodología.....	25
3.1 Modelo en V .....	25
3.2 Descripción de proceso de funcionamiento del dispositivo ....	27
3.2.1 Diagrama de flujo del sistema de detección de estado de somnolencia .....	32
3.3 Requerimientos .....	33
3.3.1 Requerimientos de usuario .....	34
3.3.2 Requerimientos del sistema .....	34
3.3.3 Requerimientos de arquitectura .....	35
3.4 Selección de hardware y software .....	36
3.4.1 Selección de <i>software</i> .....	37
3.4.2 Selección de hardware.....	38
4 Diseño .....	41
4.1 Ensamblaje del sistema.....	41
4.2 Instalación del sistema Raspbian .....	43
4.3 Configuración de programas para el sistema.....	45
4.3.1 Software Raspberry Pi camera .....	45
4.3.2 Python3.....	46
4.3.3 OpenCV .....	46
4.3.4 Archivos <i>Haar_Cascade</i> .....	49

4.3.5 Inicio automático del sistema .....	49
4.3.6 Selección de tonos por parte del conductor. ....	50
4.4 Diagrama de bloques.....	51
4.4.1 Bloque 1: Inicio de cámara y obtención de imágenes .....	51
4.4.2 Bloque 2: Tratamiento de imágenes.....	53
4.4.3 Bloque 3. Detección de rostro, ojos y parpadeo.....	54
4.4.4 Bloque 4. Análisis fisiológico y alerta. ....	56
<b>5 Implementación .....</b>	<b>57</b>
5.1 Diagrama del circuito .....	57
5.2 Ubicación de la cámara .....	58
5.3 Código fuente .....	59
5.4 Verificación del código.....	60
5.4.1 Código reconocimiento facial .....	60
5.4.2 Código reconocimiento ojos .....	60
5.4.3 Código de reconocimiento de estado de somnolencia.....	61
5.4.4 Código de reconocimiento de los ojos mediante el uso de lentes ....	62
5.4.5 Código de reconocimiento de somnolencia usando lentes. ....	62
<b>6 Resultados y Discusión .....</b>	<b>63</b>
6.1 Pruebas del Sistema.....	63
6.1.1 Pruebas realizadas en la fase 1 .....	65
6.1.2 Pruebas realizadas en la fase 2 .....	66
6.1.3 Pruebas realizadas en la fase 3 .....	67
6.1.4 Pruebas realizadas en la fase 4 .....	67
6.1.5 Pruebas realizadas en la fase 5 .....	72
6.2. Análisis de los Resultados.....	72
<b>7 Conclusiones y Recomendaciones .....</b>	<b>74</b>

7.1 Conclusiones .....	74
7.2 Recomendaciones.....	76
REFERENCIAS.....	78
ANEXOS .....	81

## ÍNDICE DE FIGURAS

Figura 1. Porcentaje de accidentes de tránsito en horas nocturnas.....	8
Figura 2. Puntos reconocibles del rostro. ....	11
Figura 3. Análisis sobre las características tipo Haar.....	13
<i>Figura 4. Formula 3 imagen integral.....</i>	<i>14</i>
Figura 5. Etapa cascada de clasificadores.....	15
Figura 6. Comportamiento del algoritmo patrón binario local. ....	16
Figura 7. Algoritmo basado en la extracción de características.. ....	17
Figura 8. Sistema detector de fatiga Bosch integrado por Volkswagen.. ....	20
Figura 9. Sistema detector de somnolencia. ....	21
Figura 10. Ícono OpenCV.....	22
Figura 11. Ícono Python. ....	23
Figura 12. Vista superior placa Raspberry Pi 3 Modelo B +.....	23
Figura 13. Ciclo de vida de desarrollo basado al modelo V. ....	25
Figura 14. Fases que contiene una metodología basada en el modelo en V... ..	26
Figura 15. Instalación del dispositivo en la parte interna vehículo.....	27
Figura 16. Diagrama general de funcionamiento del dispositivo. ....	28
Figura 17. Conexión de la Pi cámara a la unidad de procesamiento. ....	29

Figura 18. Zumbador utilizado para el sistema de detección de somnolencia .	30
Figura 19. Conexión electrónica del dispositivo. ....	30
Figura 20. Diagrama de flujo del proceso de control de somnolencia. ....	32
Figura 21. Conexión de la cámara a través del puerto por defecto raspberry. .	41
Figura 22. Distribución del puerto GPIO Raspberry Pi 3 Modelo b +.. ....	42
Figura 23. Instalación de sistema operativo Raspbian. ....	43
Figura 24. Conexión microSD en Raspberry Pi 3 Modelo b +.. ....	44
Figura 25. Escritorio Raspbian. ....	44
Figura 26. Ventana de comandos Raspbian. ....	45
Figura 27. Habilitación Pi camera mediante consola.....	46
Figura 28. Instalación de OpenCV en el sistema operativo Raspbian.....	47
Figura 29. Inicialización del sistema en modo automático. ....	50
Figura 30. Selección de tonos de alarma mediante pulsador.....	50
Figura 31. Diagrama de bloques. ....	51
Figura 32. Inicialización y captura de imágenes.....	52
Figura 33. Diagrama de bloques 1. ....	52
Figura 34. Imagen en escala de grises. ....	53
Figura 35. Diagrama de bloques 2. ....	54
Figura 36. Detección de ojos por parte del sistema. ....	55

Figura 37. Diagrama de bloques 3. ....	55
Figura 38. Analisis fisiológico .....	56
Figura 39. Diagrama de bloques 4. ....	57
Figura 40. Diagrama circuital del sistema de detección de somnolencia. ....	58
Figura 41. Ubicación específica de la cámara anti somnolencia. ....	58
Figura 42. Ensayo de código fase uno. ....	60
Figura 43. Ensayo de código fase dos. ....	61
Figura 44. Ensayo de código fase tres. ....	61
Figura 45. Ensayo de código fase cuatro. ....	62
Figura 46. Ensayo de código fase cinco.....	63
Figura 47. Análisis del dispositivo en la semana 1.....	69
Figura 48. Análisi del dispositivo en la semana 2.....	70
Figura 49. Análisis del dispositivo en la semana 3.....	71
Figura 50. Porcentaje de efectividad del dispositivo.....	71

## ÍNDICE DE TABLAS

Tabla 1. Características principales de los algoritmos. ....	18
Tabla 2. Tabla comparativa de hardware libre en el mercado.....	24
Tabla 3. Características de los elementos del sistema. ....	29
Tabla 4. Análisis de requerimientos de usuario.....	34
Tabla 5. Análisis de requerimientos del sistema. ....	34
Tabla 6. Análisis de requerimientos de arquitectura.....	35
Tabla 7. Selección de software de procesamiento de imágenes.....	37
Tabla 8. Selección de software de programación.....	38
Tabla 9. Selección de hardware. ....	39
Tabla 10. Selección de la cámara que usará el sistema. ....	40
Tabla 11. Software y hardware utilizados en el sistema de somnolencia.....	40
Tabla 12. Fase de pruebas de funcionamiento del dispositivo. ....	64
Tabla 13. Análisis de datos fase 1.....	66
Tabla 14. Análisis de datos fase 2.....	66
Tabla 15. Análisis de datos fase 3.....	67
Tabla 16. Análisis de datos fase 4.....	68
Tabla 17. Análisis de datos fase 4 durante la semana 1. ....	68

Tabla 18. Análisis de datos fase 4 durante la semana 2. ....	69
Tabla 19. Análisis de datos fase 4 durante la semana 3. ....	70
Tabla 20. Análisis de datos fase 5.....	72

# 1 Introducción

Los accidentes de tránsito son cada día más frecuentes con la aparición de nuevos vehículos en los parques automotores de las ciudades de Ecuador; existen diferentes razones para que se generen accidentes de tránsito como son: exceso de velocidad, conducir con llantas en mas estado, somnolencia, etc., la somnolencia de los choferes es una causa importante por la que provoca accidentes de tránsito y es causante de al rededor del 10% de este tipo de tragedias en el país según un estudio estadístico realizado en el 2010 por la Agencia Nacional de Transito (ANT), por lo cual se ha realizado la presente investigación con el fin de contribuir a erradicar este tipo de siniestros.

En la presente investigación se pretende evitar accidentes de tránsito por estado de somnolencia a través de la implementación de un sistema electrónico que permita verificar el estado del conductor mediante visión artificial, este tipo de sistema se lo realizó mediante la utilización de una placa Raspberry Pi 3 modelo b + y una cámara de alto rendimiento compatible con la unidad de procesamiento para captar los rasgos fisiológicos del conductor y en caso de presentar un estado en el que se pueda verificar que es somnolencia el dispositivo active una alarma con el fin de alertar al conductor para evitar siniestros por motivo de este factor.

Actualmente se está estudiando las mejores formas en las que un dispositivo electrónico sea capaz de reconocer el estado anímico de las personas mediante visión artificial, es por eso que el presente tema indagó en diversos algoritmos que permiten reconocer el estado de las personas sobre todo la somnolencia que presentan los conductores a la hora de trasladar sus vehículos a largas distancias con el fin de avisar mediante alertas sonoras un posible estado de somnolencia que podría causar un accidente de tránsito el cual podría cobrar varias vidas inocentes.

## 1.1 Objetivo General

Implementación de un sistema de alarma basado en la detección de somnolencia para conductores de vehículos mediante visión artificial.

## 1.2 Objetivos Específicos

- Analizar los algoritmos de detección del estado de somnolencia, así como también el procesamiento de imágenes y visión artificial.
- Analizar los componentes necesarios en Hardware que sean compatibles con el algoritmo de detección de estado de somnolencia.
- Diseñar un prototipo con dispositivos electrónicos que alerten al conductor al momento de presentar un estado de somnolencia frente al volante.
- Implementación de sistema de alarma y monitoreo facial que permita detectar los rasgos fisiológicos del conductor al instante de conducir un vehículo.
- Realizar pruebas de funcionamiento que permita verificar el desempeño del sistema.

## 1.3 Alcance

Se analizarán las diferentes formas de detección de signos de somnolencia en conductores. Para el efecto se investigará la bibliografía acerca de las causas de somnolencia, así como también revisará la literatura acerca de procesamiento de imágenes y visión artificial que permitan identificar las mejores soluciones al problema planteado.

Estudiar los algoritmos necesarios para su respectivo funcionamiento, hardware en este caso cámara, unidad de procesamiento, alarma de detección, que permitan adaptarse al prototipo y de esta manera poder detectar el estado de somnolencia en conductores.

Diseño e implementación de un prototipo para detectar el estado de somnolencia en conductores, basado en el uso de visión artificial. Posterior a ello se realizarán

pruebas de funcionamiento en el vehículo a implementar para comprobar el funcionamiento y buen desempeño del prototipo.

## **1.4 Justificación**

Con el presente proyecto se busca beneficiar a los conductores, así como también a personas que hagan uso de este sistema, mediante la interacción entre hardware y software permiten alertar los conductores cuando estos presenten cansancio y desconcentración a la hora de conducir. También se busca reducir los índices de muertes a causas de este efecto en el Ecuador.

Conforme a estadísticas de estudios notificados por la Organización Mundial de la Salud (OMS), se tasa que 1,25 millones de personas mueren por causa de accidentes de tránsito. De estos accidentes en su mayoría son a causa de la inatención de conductores, cuyos factores principales son la somnolencia como la distracción. Se deduce que la inatención provoca entre un 25% y 75% de accidentes y casi accidentes en el Mundo (OMS, 2014).

Este proyecto al tratarse de la prevención de muertes de personas a causa de somnolencia en conductores quedaría como un proyecto piloto que pueda ser implementado en cualquier tipo de vehículo, y como siguiente fase a futuro sería la comercialización de este sistema permitiendo a la Universidad de las Américas sobresalir como un referente de tecnología y soluciones para la sociedad.

## **2 Marco Teórico**

En el presente capítulo referente al marco teórico se detallará de forma clara temas relacionados a los algoritmos a utilizar en el sistema, signos por los cuales se presenta este síntoma de forma real, igualmente estudios sobre el procesamiento de imágenes su uso y su forma de aplicación con ello se logrará determinar si un conductor posee fatiga o sueño y parámetros utilizados para el desarrollo del sistema.

## **2.1 Distracción y somnolencia al conducir**

### **2.1.1 Distracción al conducir**

Las principales causas por que se generan los accidentes de tránsito es la distracción en la conducción, de tal manera que en el 2017 el 33% de accidentes de tránsito fueron producidos por la distracción superando accidentes de tránsito producidos por exceso velocidad que tiene un porcentaje de 29% y accidentes generados por conductores en estado de embriaguez que tiene un porcentaje de 26%. Por este motivo, la Dirección General de Tráfico realizó campañas sobre la concienciación de estos accidentes que son peligrosos denominados distracción frente al volante (DGT, 2016).

La distracción es producida en el momento en que la atención del conductor es dirigida a otro lado diferente a la conducción ya sea ocasionada por una acción secundaria o evento reflejado al momento de la conducción, pese a que la información que existe no es apropiada, y no es significativa para los conductores, se estima que llegan a distraerse por acciones secundarias, prácticamente el 30% de conductores se distraen al momento de conducir un vehículo.

La distracción en la conducción ha llevado a realizar varios estudios y de esta manera obtener su concepción, existen diferentes descripciones las cuales unas se describen de forma parcial, y otras se orientan en los aspectos ocasionados por algún fenómeno ya sea por efectos o distractores para el conductor. Existen también descripciones que limitan el significado de distracción, en la que se enfocan en acciones sustitutas ejemplo utilización de aparatos electrónicos.

Debido a la perplejidad para definir un concepto específico de distracción se ha llegado a la aceptación de que el descuido al momento de conducir un vehículo se relaciona algún acontecimiento, acción, objeto o individuo, adentro o afuera del vehículo, interrumpen con la atención del conductor y no se realiza su acción primordial que es conducir. Debido a este factor como es la distracción en la

mayoría de los casos puede generar accidentes de tránsito (Gras, Pedra, & Font-Mayolas, 2008).

Durante la conducción el estado físico de la persona juega un factor significativo la obtención de información y la capacidad de aceptación. Es importante tomar en cuenta también el estado del conductor el cual puede llegar a obtener fatiga con ello es complicado mantener una atención completa a la hora de conducir. Por lo que la capacidad de respuesta llega afectar de forma negativa, cometiendo errores y ocasionando accidentes de tránsito en las vías.

### **2.1.2 Tipos de distracción**

Las distracciones del conductor pueden ser tanto internas como externas al vehículo, esto quiere decir que un conductor no acepta el peligro que puede originar una distracción debido a la acción de actividades secundarias. La labor de conducir conlleva a que el conductor establezca y llegue a ordenar toda la información, estímulos que adquiere del entorno en el cual se encuentra. Por lo tanto, se exige que el conductor logre mantener un nivel correcto de atención selectiva al momento de conducir.

Para comprender mejor la naturaleza de la distracción del conductor, el término hace referencia comúnmente al tipo visual, cognitivo, biomecánico y auditivo; si bien los cuatro tipos son útiles, es importante reconocer que estos tipos de distracción no se excluyen mutuamente.

#### **2.1.2.1 Distracción interna**

Incluye todo lo que se encuentra en el interior del vehículo, incluso las propias condiciones y acciones de los ocupantes del vehículo y del conductor, un dispositivo puede distraer visualmente de manera particular, también es probable que incorpore elementos como la distracción biomecánica, por ejemplo, manipulación de los controles. sin embargo, en situaciones donde un conductor se distrae con los pasajeros, no está claro si esto es visual, es decir, el conductor movió su cabeza para mirar a su pasajero y, por lo tanto, no estaba observando

la carretera o el auditorio, es decir, el conductor fue sorprendido por un comentario de su pasajero, o cognitivo, es decir, la conversación fue mentalmente agotadora e impactada en la concentración del conductor (Rastreator, 2018).

### **2.1.2.2 Distracción externa**

Incluye todo lo que aparece fuera del vehículo, por ejemplo. Cuando el conductor mira edificios, personas o situaciones fuera del vehículo, así como vallas publicitarias, anuncios que son "exitosos" desde una perspectiva de *marketing*, buscando señales de tránsito cuando se pierde, es razonable suponer que la distracción es de naturaleza visual y cognitiva, y es poco probable que sea biomecánica o auditiva, el comportamiento impredecible de un peatón en estado de embriaguez (Montes, Ledesma, & Poó, 2014).

### **2.1.3 Somnolencia al conducir**

La somnolencia y el sueño al conducir se asemejan como las causas que se encuentran detrás de los accidentes irremediables e incidentes en las vías ocasionados por conductores aparentemente profesionales. Se ha relacionado a la fatiga con los sucesos de tránsito, estos se generan por motivos de exceso de velocidad a la hora de conducir, conducción extensa y conducción nocturna, según estudios el uso de medicamentos, la falta de sueño, la somnolencia y conducir en la noche generan un 20% de accidentes de tránsito relacionado en las vías (ANT, 2017).

En conductores profesionales los trastornos de sueño deben ser estudiados de forma significativa con el fin de disminuir los accidentes de tránsito. Se definen como factores de riesgo el hecho de conducir con sueño, generando entre el 15% y 30% del total de accidentes de tránsito, y los accidentes de tránsito asociados mediante los efectos del alcohol son superiores a los que generan conducir en estado de somnolencia, estos datos se obtuvieron de un estudio realizado a nivel mundial por la *National Transportation Safety Board* (Race, 2017).

Un conductor en estado de somnolencia puede llegar a poseer inexactitud de atención, falta de respuesta, derivación de agotamiento frente al volante, de esta forma pueden llegar ocasionar accidentes de tránsito. Los sucesos ocasionados al estado de somnolencia pueden generar como consecuencias altas tasas de víctimas mortales en las vías.

### **2.1.3.1 Estados fisiológicos a causa de somnolencia**

A través de las características visuales comunes con las cuales una persona presenta somnolencia son: el cabeceo, bostezo, pestañeo, expresión facial y movimientos inquietos al ser constantes (MAPFRE, 2017).

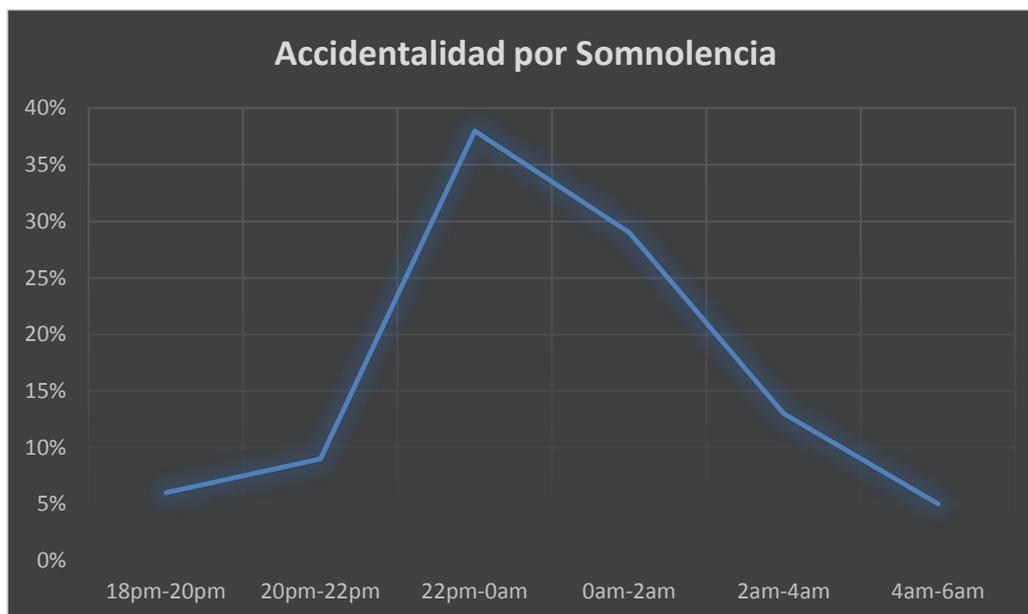
La somnolencia es una queja frecuente en la neurológica y puede ser debida a múltiples trastornos. Puede definirse como la sensación subjetiva de sueño a una hora inapropiada. Dentro de los síndromes con somnolencia diurna excesiva podemos encontrar trastornos intrínsecos del sueño como la narcolepsia o hipersomnia secundaria a otras enfermedades o como efecto secundario de diversos fármacos

La somnolencia en su mayoría afecta a las operaciones simples como son las neurocognitivas y psicomotoras, esto quiere decir que afecta tanto a lo que es el tiempo de respuesta, la visibilidad, falta de atención, que como resultado puede llegar a causar accidentes de tránsito. Muchos de los conductores conllevan al excesivo pestañeo, al igual que el cabeceo y posterior a ello sueño el cual es afectado en sus horas de trabajo. La somnolencia en su mayoría genera la inexactitud de capacidad de reacción al momento de tomar volante, lo que hace que produzca altas tasas de mortalidad.

## **2.2 Proceso de conducción en horas del día**

Se estima que las principales horas en la mañana como son de 2:00 – 5:00am y las principales horas en la tarde como son de 14:30 – 16:00pm son ciclos los cuales los conductores presentan somnolencia. Como se puede observar en la figura 1, se presenta el porcentaje de accidentes de tránsito en horas nocturnas,

en donde se evidencia que desde las 22h00 hasta 02h00 existe mayor índice de accidentes.



*Figura 1.* Porcentaje de accidentes de tránsito en horas nocturnas.

En su mayoría los conductores laboran en horas del día, de igual manera existe conductores que trabajan vespertinamente donde presentan todo su rendimiento en horas de la tarde, existen de igual forma horarios que no benefician los conductores como son la madrugada y en general las altas horas de la noche.

Entre más tiempo el conductor lleve despierto, es más probable que se resista al sueño debido a que no ha tenido ningún espacio de descanso, de esta manera las horas de descanso no han sido recuperadas, dando a conocer que la conducción es una acción fuertemente peligrosa (DGT, 2016).

### **2.3 Sistemas para la detección de somnolencia**

La producción automotora ha tenido un gran efecto sobre el comportamiento del conductor en este caso sobre su conducta, dando a conocer la adaptación tanto el conductor como el vehículo (*Human- Machine Interface*), esto se logra gracias a la programación informática o a su vez sobre mensajes de asistencia

permitiendo al conductor un manejo más seguro frente al volante (MAPFRE, 2017).

Actualmente se puede encontrar todo tipo de sistemas para detectar la fatiga y el cansancio del conductor, por ejemplo, sistemas de visión por computador, sistemas de análisis de señales bioeléctricas, lo que los diferencia a cada uno de estos sistemas son los tipos de datos establecidos al conductor al momento de conducir esto ayuda al conductor a que no logre perder su concentración o a su vez que el conductor entre en estado de somnolencia.

Según seguimientos al comportamiento del conductor se pudo evidenciar que existen diferentes sistemas, uno de ellos denominado antifatiga el cual genera información mediante un aviso de alerta ubicada en la parte interna del vehículo y sobre las medidas de conducción. Por otro lado, existen sistemas que se basan específicamente en el monitoreo de los rasgos fisiológicos del conductor a la hora de conducir como son sistema de monitoreo de fatiga, GPS monitoreo entre otros.

### **2.3.1 Sistema basado en medición de parámetros**

Los sistemas de medición de parámetros se basan en la creación de un perfil alusivo al comportamiento del conductor, esta investigación se genera sobre la labor realizada por parte del conductor como es el dominio al momento de conducir; esta investigación obtenida con respecto al perfil de conducción hace referencia a la velocidad empleada por el conductor, manejo frente al volante, utilización de freno y la aceleración, por otra parte, de igual manera se desarrollan evaluaciones sobre el ambiente vehicular como es el viento, estado de semaforización, límites de carril al momento de sobrepasar etc (Rosales, Rosales, & Rojas, 2013).

El monitoreo mediante dispositivos electrónicos es importante ya que es posible determinar el estado fisiológico del conductor de esta manera saber si presenta fatiga o sueño, si es así de serlo el dispositivo emitirá una alarma la que permite alertar al conductor.

### **2.3.2 Sistema basado en el reconocimiento facial**

Así como los sistemas por medición de parámetros, existen sistemas que se caracterizan en el monitoreo de los rasgos faciales, estos sistemas se orientan en los rasgos fisiológicos del conductor como son los ojos y el parpadeo de la persona, partiendo de estos parámetros se puede llegar a determinar si el conductor presenta estado de somnolencia; para determinar estos datos se utiliza cámaras ubicadas dentro del vehículo, utilizando los infrarrojos los que se encuentran orientados a los ojos del conductor se monitorearan los rasgos fisiológicos, posterior a ello estos datos son procesados en la unidad de control (UC), estos datos obtenidos verificaran si el conductor presenta estado de somnolencia o fatiga.

Para el ensamblaje de un sistema de reconocimiento facial se exige el uso de una cámara de alta definición que se adapte a cualquier estado del día ya sea día o noche, mediante el dispositivo se puede determinar si el conductor se encuentra distraído o si presenta sueño. Para verificar si el conductor está con sueño se analiza cada cuadro captado cada segundo y verificando si la apertura de cada ojo es suficiente para aumentar el valor de la variable o no; si el número de cuadros de conteo por ojos cerrados excede una constante, se activará una alarma la cual permita alertar al conductor (Barba, 2019).

Por otro lado, para poder determinar si el conductor se encuentra distraído, el sistema hace uso de la dirección logrando captar sus globos oculares. Por lo que el conductor tiene que estar concentrado en dirección a la parte frontal del automóvil, caso de no estarlo el sistema mediante un contador proporciona un tiempo el cual se define siempre que su mirada no se encuentre dirigida hacia la carretera. Por lo que si el tiempo es sumamente prolongado este indica al conductor que se encuentra en estado de distracción, este tipo de sistema actúa a través de una de una señal visual el cual se encuentra ubicada en el tablero del automóvil. Como se puede observar en la figura 2, se detalla la forma en como el sistema realiza el reconocimiento partiendo desde el rostro y posterior a

ello el reconocimiento de los ojos y con ello saber si el conductor presenta sueño o fatiga.



*Figura 2.* Puntos reconocibles del rostro.

Tomado de (Ávila, 2016)

## **2.4 Visión artificial**

La visión artificial es una disciplina en desarrollo, que apunta a construir modelos computacionales de las funcionalidades visuales en humanos, que trabaja mediante el procesamiento de imágenes donde la señal de entrada capta la imagen en escala de rojo, verde y azul (RGB) estos son captados mediante dispositivos que disponen de cámaras (Contaval, 2016). La visión artificial se enfoca con imágenes digitalizadas; es decir usa escala de grises con ello obtiene un mayor reconocimiento al momento de exponerlo frente a una persona u objeto.

## **2.5 Procesamiento de imágenes**

El procesamiento de imágenes ya sea dirigido al algoritmo de patrones binarios locales o Viola & Jones estos algoritmos siguen un patrón que se encuentra ya establecido permitiendo de esta manera capturar la imagen de entrada y

posterior a ello esta pasa a ser digitalizada. El proceso de digitalización de imágenes lleva consigo los siguientes parámetros como son: captura de imagen, procesamiento de imagen y por último la visualización de imagen capturada.

## **2.6 Algoritmos para la detección facial**

El *software* de detección facial sirve para verificar si una fotografía cualquiera contiene o no uno o varios rostros de personas, este tipo de *software* es importante para la realización de proyectos que requieran este tipo de características como son la captura de imágenes y procesamiento de imágenes, generalmente se lo realiza por categorización binaria mediante clasificadores que permite minimizar errores; para este tipo de reconocimiento es necesario que el algoritmo minimice errores frecuentes como lo son los falsos positivos y negativos para que el código de programación tenga un rendimiento aceptable; este proceso requiere una codificación numérica que sea exacta ya que debe ser capaz de diferenciar rostros humanos de objetos.

### **2.6.1 Algoritmo de Viola & Jones**

El algoritmo Viola & Jones se utiliza para detectar los rostros humanos a partir de una imagen, en donde el sistema toma algunas imágenes faciales y no faciales como entrada, posteriormente iniciará la fase de entrenamiento en la que el sistema detecta el rostro. En fase de formación se incluyen dos tipos de conjuntos como son el conjunto de imágenes positivas y el conjunto de imágenes negativas. Las imágenes positivas se refieren a imágenes con rostros y en lo que es imágenes negativas se refiere a las imágenes sin rostro. En la fase de formación, se recogen todas las características que son relacionadas con las imágenes de rostro y todas estas características se almacenan en un archivo el que es comparado mediante los clasificadores en cascada. Después de la formación la siguiente fase es la fase de prueba la que conlleva a detectar los rostros finales u objetos (ProgramaciónExtrema, 2018). El algoritmo de Viola & Jones tiene cuatro etapas las cuales son:

a) Características de tipo *Haar*

Las características de tipo Haar son combinaciones de rectángulos del mismo tamaño, adyacentes horizontal o verticalmente. Los rectángulos en negro representan zonas con una contribución positiva; mientras que, los rectángulos blancos representan zonas con una contribución negativa al filtro o resultado final (RicardoGeek, 2019). Las características tipo Haar están compuestas por tres características las cuales son: dos rectángulos, tres rectángulos, cuatro rectángulos (horizontal – vertical). En la figura 3 se puede observar los tipos de características Haar las cuales están aplicadas al rostros y ojos relacionadas a dos y tres rectángulos con ello se logra comparar la intensidad de la zona asignada, el resultado del filtro es la diferencia de los pixeles de una imagen tanto de la zona negra como blanca (ROBOLOGS, 2014).

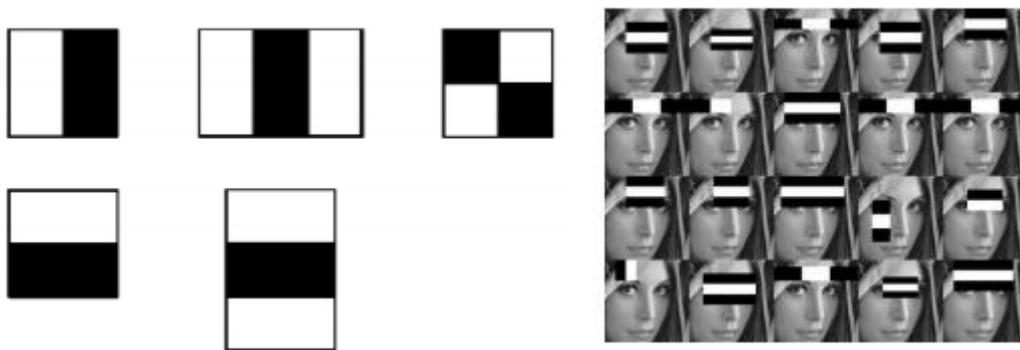


Figura 3. Análisis sobre las características tipo *Haar*.

Tomado de (ROBOLOGS, 2014)

b) Imagen integral

Viola & Jones usaron una representación de imagen llamada imagen integral, también conocida como tabla de área sumada, para determinar la presencia o ausencia de cientos de características similares a Haar en cada ubicación de imagen. Para realizar el cálculo se determina un punto  $x, y$ ; que se establece su valor en la suma de los pixeles ubicados por encima y a su izquierda de dicho punto incluyéndose, convirtiendo la imagen original en una imagen integral

(STACKPOINTERS, 2018). En la figura 4 se puede observar, que la primera tabla es una imagen original y la segunda tabla es una imagen integral resultado de la aplicación de la primera tabla con ello se puede aplicar las características de Haar y lograr una mejor detección de rostro sobre la imagen.

$$I(x, y) = \sum_{x' \leq x, y' \leq y} I_m(x', y')$$

1	1	1
1	1	1
1	1	1

1	2	3
2	4	6
3	6	9

Figura 4. Formula 3 imagen integral.

Tomado de (STACKPOINTERS, 2018)

#### c) Impulso adaptativo

El impulso adaptativo es capaz de elegir entre un gran conjunto de filtros, las características Haar, para seleccionar en cada momento cuál de ellos se ajusta mejor, para que se clasifiquen satisfactoriamente los diferentes elementos que vamos a clasificar. En este procedimiento se obtiene después de algunas iteraciones, donde se prueba los clasificadores de Haar para cada una de las muestras, un clasificador que separa entre muestras positivas y muestras negativas. Primeramente, se genera una detección con algún clasificador, donde los elementos mal clasificados aumentan su tamaño, para que el siguiente clasificador que intervenga, de más importancia a que, la clasificación de los elementos con mayor tamaño sea la correcta.

#### d) Cascada de clasificadores

Se describe al algoritmo para construir una cascada de clasificadores que logra un mayor rendimiento de detección mientras reduce radicalmente el cálculo negativo de dicha imagen, la idea clave es que, si son más pequeños los clasificadores, más se pueden construir clasificadores eficientes e impulsados que rechacen muchas de las imágenes negativas mientras detecta casi todas las instancias positivas. Los clasificadores más simples rechazan la mayoría de las imágenes en cambio los clasificadores complejos son llamados para lograr bajo falsos positivos. (Ibermática, 2011) Las etapas en la cascada se construyen mediante entrenamiento de clasificadores utilizando impulsos adaptivos. Comenzando con una de dos características del clasificador fuerte, se puede obtener un filtro facial eficaz ajustando al umbral del clasificador fuerte para minimizar falsos negativos. En la figura 5 se puede observar, que el primer clasificador es capaz de eliminar las muestras negativas fáciles, el otro clasificador elimina las muestras negativas más difíciles logrando de esta manera establecer muestras de forma eficiente sin limitación de errores.

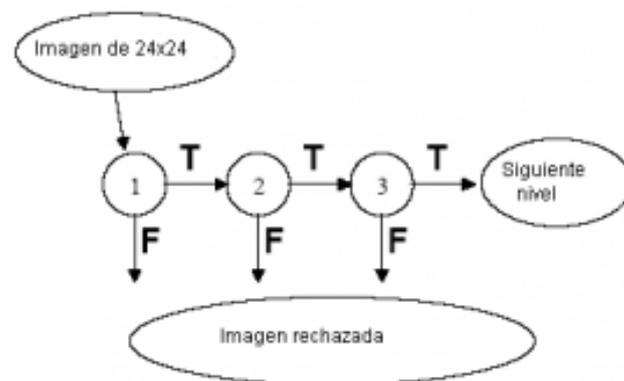


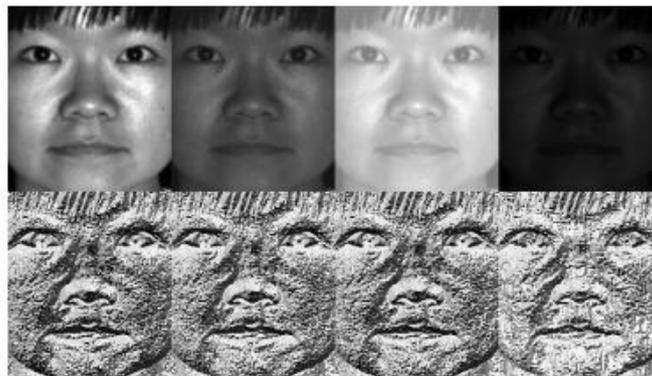
Figura 5. Etapa cascada de clasificadores.

Tomado de (Ibermática, 2011)

#### 2.6.2 Algoritmo patrón binario local

Durante los últimos años, patrones binarios locales (LBP) han despertado un interés referente al procesamiento de imágenes y visión artificial. LBP es un

método no paramétrico que resume las estructuras locales de las imágenes de manera eficiente comparando cada píxel con píxeles adyacentes. Las propiedades más importantes de LBP son: su tolerancia a la iluminación monotónica y su simplicidad computacional. LBP fue originalmente propuesto para análisis de texturas, debido a que es potente para describir las estructuras locales. Ha sido ampliamente explotado en muchas aplicaciones, por ejemplo, análisis de imagen, recuperación de imagen y vídeo, modelado de entorno, inspección visual, y teledetección (Troya, 2016). En la figura 6 se puede observar el comportamiento del algoritmo el cual transforma una imagen basada en escala de grises en texturas locales facilitando de esta manera el comportamiento de detección. El análisis de imagen facial basado en LBP ha sido uno de los más populares y exitosos en los últimos años.



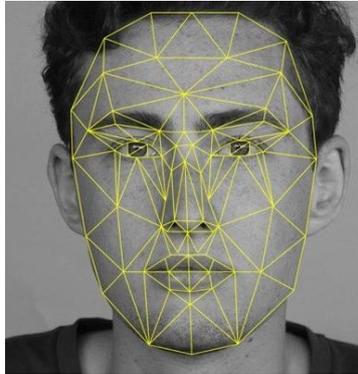
*Figura 6.* Comportamiento del algoritmo patrón binario local.

Tomado de (Troya, 2016)

### **2.6.3 Algoritmo de enfoque basado en características elásticas**

El algoritmo EBGGM es único en relación con los demás ya que percibe los rostros y contrasta sus partes; en segundo lugar, los resultados son buenos en relación con diferentes algoritmos evaluados en las pruebas (AMICUS, 2015). Presenta código abierto ampliado que incorpora cuatro algoritmos de referencia y una disposición de los dispositivos y secuencias de comandos que se pueden utilizar para evaluar la ejecución del algoritmo de reconocimiento facial. En la figura 7

podemos observar que el algoritmo EBGM se encarga en tomar los puntos geométricos faciales para la extracción de características de las distintas partes del rostro.



*Figura 7.* Algoritmo basado en la extracción de características.

Tomado de (AMICUS, 2015)

#### **2.6.4 Selección del algoritmo para el análisis facial del proyecto**

En el ítem de resolución es evidente que, por su bajo coste de procesamiento y utilización de memoria, el algoritmo de Viola & Jones es el más concreto para la realización del presente sistema ya que analiza la imagen en fotogramas de 20 x 20 píxeles que es considerable y válido para la realización del proyecto.

En el ítem de dato temporal, se puede verificar que los 3 algoritmos son similares por lo que en este punto cualquier algoritmo es una buena opción para sistema.

En el ítem de rasgos el algoritmo que mejor se adapta al ambiente del reconocimiento facial de rostros en tiempo real es el algoritmo de Viola & Jones, puesto que va a interactuar directamente con el rostro del conductor a una corta distancia para verificar su estado de somnolencia permitiendo una adaptación oportuna al cambio de iluminación y entorno de trabajo.

En el ítem de adaptación al entorno el algoritmo que sobresale es el de Viola Jones, debido a que no genera tanta sensibilidad al cambio de iluminación y

generalmente este algoritmo tiene eficiente respuesta en la noche ya que es el momento en que los conductores más padecen de sueño.

La tabla 1, muestra las comparaciones de los tres tipos de algoritmos que son más utilizados hoy en día para detectar los rasgos faciales de individuos u objetos, entre ellos sobresale el algoritmo de Viola & Jones por su adaptación al entorno y empleabilidad en tiempo real.

Tabla 1.

*Características principales de los algoritmos*

<b>Algoritmo</b>	<b>Viola &amp; Jones</b>	<b>Patrón Binario Local</b>	<b>Enfoque Características Elásticas</b>
<b>Resolución</b>	Detección de objetos y rostros destacado por su bajo coste. Tamaño de imagen se divide en fotogramas de 20 x 20 pixeles.	Detección de textura a nivel local en escala de grises, basándose en diferentes aplicaciones referente al tratamiento de imágenes en fotogramas de 16 x 16 pixeles.	Detección de reconocimiento facial empleado en puntos importantes y no en la totalidad del rostro por fotogramas de 40 x 40 pixeles.
<b>Tiempo</b>	Empleabilidad en tiempo real a la velocidad del procesador.	Empleabilidad en tiempo real a la velocidad del procesador.	Empleabilidad en tiempo real a la velocidad del procesador.
<b>Rasgos</b>	Emplea filtros de Haar, sin la necesidad de puntos concordancia.	Le cuesta combatir con el paso del tiempo referente a las imágenes en la distribución de rasgos.	En su mayoría muchos de los grafos es necesario colocar de forma manual al rostro.

---

<b>Iluminación y adaptación del entorno</b>	Sensible a cambios de iluminación se adapta fácilmente.	Presenta problemas al momento de que existan alteraciones de iluminación muy contundente.	Es supremamente sensible referente a las condiciones de iluminación.
---	---	---	--

---

## 2.7 Tecnologías basadas en la detección de somnolencia y distracción

Es común que en las tecnologías usadas para la detección de somnolencia se tiene sistemas de alarmas como por ejemplo sistema anti somnolencia que responden automáticamente a cambios en la presión del volante por parte del conductor, existen otros sistemas que permiten alertar al conductor a tal punto que el vehículo no llegue a desviarse de su carril, es decir corrige la conducta de conducción del conductor este sistema se denominan *Driver Alert System* (DAS).

A continuación, se describirán otros sistemas basados en la detección de somnolencia como:

### a) Tecnología Raspberry Pi 3

La tecnología Raspberry está administrada por Raspbian, el cual es un sistema operativo basado en Linux Debian que se encarga de controlar el dispositivo; con la intervención de una cámara de video, librería *OpenCV* que permite realizar el procesamiento de imágenes y Python que es un lenguaje de programación eficiente, permiten determinar el estado del conductor a la hora de conducir un vehículo ya que mediante la visión artificial que presenta este dispositivo, se puede detectar el estado de somnolencia que presenta un conductor.

Este sistema contiene una cámara de alto rendimiento cuya posición está ubicada frente al rostro del conductor, el sistema se alimenta mediante la batería del vehículo.

## b) Detector de fatiga

Este sistema de detección de fatiga actúa cuando surgen variaciones referentes a la conducta normal de conducción, la forma de advertir al conductor se la realiza por medio de una alerta visual y también acústica. La disminución de la concentración y la fatiga comprometen el comportamiento de la dirección y el tiempo de respuesta del conductor, las habilidades motoras se deterioran y el comportamiento de la dirección se vuelve menos preciso. El conductor corrige pequeños errores de dirección con más frecuencia. Los signos típicos de disminución de la concentración son fases durante las cuales el conductor apenas está conduciendo, combinado con movimientos de dirección leves, pero rápidos y abruptos, para mantener al automóvil en la pista. Este tipo de sistema puede analizar un valor estimado de fatiga en el conductor. Si el valor estimado sobrepasa un valor definido en el sistema, una señal visual y acústica advierte al conductor que debe tomar un pequeño descanso (Autocasión, 2015). La función de detección de somnolencia del conductor aborda un aspecto importante de la condición del conductor por lo que puede contribuir a mejorar la seguridad vial (LogiNews, 2015). En la figura 8 se puede observar, un sistema de detección de fatiga de la marca *Bosch*, este sistema se incorporó en muchos de sus modelos automovilísticos por su alto rendimiento y eficiencia. Ejemplo *Volkswagen Golf Variant*.



Figura 8. Sistema detector de fatiga Bosch integrado por Volkswagen.

Tomado de (LogiNews, 2015)

c) Dispositivo práctico para la somnolencia

Mediante estudios realizados por alumnos pertenecientes al departamento de software de la Universidad de Minnesota, descubrieron una estructura en la cual monitorear la somnolencia en conductores de vehículos. Dicho sistema está formado por diferentes dispositivos electrónicos los cuales son cámaras de vigilancia las cuales permiten monitorear el rostro del individuo y de esta manera establecer si el conductor presenta síntomas de sueño, este sistema actúa de manera instantánea en cuestiones de segundos, algo importante a recalcar sobre este sistema es que examina de forma clara como es el color de pixeles del rostro del individuo (GPSTEC, 2015). Como se puede observar en la figura 9, el sistema examina los ojos del conductor de esta forma analiza si se encuentran abiertos o cerrados, con ello el sistema actúa y permite saber si el conductor presenta somnolencia o fatiga. El sistema resultó ser eficaz ya que en pruebas de funcionamiento resulto dar resultados positivos. El sistema utiliza un software *OpenCV*, igualmente un algoritmo denominado *facial landmark predictor*, permitiendo de esta manera saber si los conductores presentan fatiga.



Figura 9. Sistema detector de somnolencia.

Tomado de (GPSTEC, 2015)

## 2.8 Software libre

### a) *OpenCV*

Es un software que está orientado al desarrollo de aplicaciones de reconocimiento de imágenes, estos software actualmente son conocidos herramientas para el desarrollo de visión artificial, lo cual es de gran importancia ya que permite reconocer los rasgos fisiológicos que presenta una persona y así generar aplicaciones que administren y diferencien los diferentes estados de ánimo de una persona mediante un escaneo o relieve en 3D; esta librería ofrece varios algoritmos que son capaces de reconocer rasgos de personas, animales, cosas y hasta paisajes con el fin de orientarlos a la realidad aumentada (OpenCV, 2019). En la figura 10, es posible observar un logo el cual caracteriza a *OpenCV* por su eficiencia en el proceso de visión artificial.



*Figura 10. Ícono OpenCV.*

Tomado de (OpenCV, 2019)

### b) Python

Es un lenguaje de programación sencillo y fácil de utilizar ya que ofrece opciones dinámicas de estructuración de código, su sintaxis es fácil de comprender y la forma de estructurar su código es mediante la sangría que se le aplica al código; de esta forma los programadores pueden entender el código de forma rápida y es mucho más sencillo que otros lenguajes de programación. En la figura 11, es posible observar el logo el cual caracteriza a Python como un lenguaje de programación de alto rendimiento (Python, s.f.).



Figura 11. Ícono Python.

Tomado de (Python, s.f.)

## 2.9 Hardware

### a) Raspberry Pi 3 modelo b +

La Raspberry Pi 3 modelo b + es una tarjeta pequeña la cual cabe en la palma de la mano y su funcionamiento permite la instalación de software Linux el cual es de fácil administración. El sistema, presenta varios puertos de entrada y salida de información, entre ellos cuatro puertos *universal serial bus* (USB), al igual que puertos *high-definition multimedia interface* (HDMI), un puerto de red, Wifi y una administración de 40 pines en su puerto GPIO los cuales se pueden usar como entradas o salidas para múltiples usos relacionados con proyectos de electrónica (PiFoundation, s.f.). En la figura 12 se observa, la representación de una placa Raspberry Pi 3 modelo b +.



Figura 12. Vista superior placa Raspberry Pi 3 Modelo B +.

Tomado de (PiFoundation, s.f.)

### 2.9.1 Comparación entre herramientas de hardware

La placa Raspberry Pi 3 modelo b + fue la placa seleccionada para el desarrollo del sistema ya que su precio es intermedio entre las otras placas que fueron objeto de estudio, sus dimensiones también son similares pero las ventajas están en la memoria RAM, acceso a Ethernet y Wifi, conexión Bluetooth, velocidad de sincronización. Lo más importante es que maneja un sistema operativo independiente por lo que toda la configuración depende de sí misma, además como es un dispositivo autónomo se puede observar que presenta 4 conectores USB con los cuales se puede acceder a varios dispositivos como impresoras, teclados, cámaras, etc., y mediante su puerto HDMI se puede conectar cualquier tipo de monitor para observar el entorno del sistema operativo.

En la tabla 2, es posible observar una comparativa de distintas placas, las cuales son adaptables a cualquier tipo de proyectos variando entre sus características principales: precio, entorno de desarrollo y memoria.

Tabla 2.

*Tabla comparativa de hardware libre en el mercado*

<b>Características</b>	<b>Arduino</b>	<b>Raspberry Pi Modelo 3 B</b>	<b>Intel Galileo</b>
<b>Precio</b>	\$30	\$50	\$60
<b>Dimensiones</b>	7.6 x 1.9 x 6.4 cm	8.5 x 5.6 x 1.7cm	3.55 x 2.5 x 3.9 cm
<b>Memoria</b>	0.002 MB	1Gb LPDDR2	1 GB
<b>GPIO</b>	14	40	40
<b>Sincronización</b>	16 MHZ	1.2 GHZ	500 MHZ, 100 MHZ
<b>Conexión internet</b>	No tiene conexión	802.11 b/g/n	Dual - Band (2.4 y 5 Ghz) Wifi. Bluetooth 4.0
<b>Entrada de voltaje</b>	7, 12 v	5 v	3.3 y 4.5 v
<b>Micro SD</b>	32KB	MicroSD c	4GB Emmc
<b>Entrada USB</b>	No tiene entrada USB	Cuatro entradas USB	Una entrada USB
<b>Sistema Operativo</b>	Ninguno	Linux	Linux v 1.6

<b>Entorno de desarrollo integrado</b>	Arduino ID	Scratch, IDEL y Linux	Arduino ID, Eclipse, Intel XDK
--	------------	-----------------------	--------------------------------

### 3 Metodología

La metodología para desarrollo de *software* consiste una manera ordenada referente a la realización y gestión de proyectos con el fin de que los proyectos sean creados de forma eficiente y posterior a ello obtengan gran probabilidad de éxito, de esta manera se divide la realización del proyecto en fases designadas como etapas, las acciones pertenecientes a cada una de ellas ayudaran a establecer tanto entradas como salidas para cada etapa, que ayudara a normalizar la forma para la gestión de desarrollo del proyecto.

El proceso de ingeniería aplicada a la construcción de dispositivos es una metodología utilizada como complemento en el diseño y manufactura de productos y artefactos, especialmente cuando estos elementos están compuestos por dispositivos electrónicos complejos.

#### 3.1 Modelo en V

El modelo en V permite realizar y validar las etapas de desarrollo de una investigación de tal forma que cada etapa inicial es útil para posteriormente validar y verificar las etapas finales de tal forma retroactiva. En la figura 13 se puede observar, de forma clara como es el ciclo de vida del modelo en V.

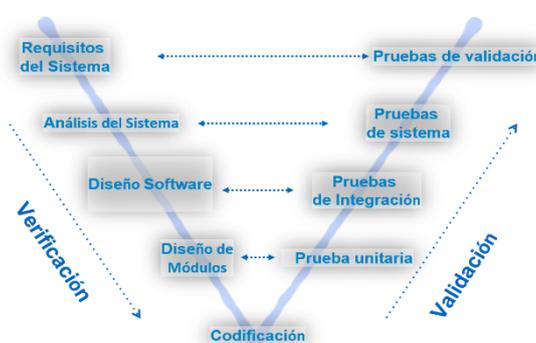


Figura 13. Ciclo de vida de desarrollo basado al modelo V.

Para la realización del presente proyecto se seleccionó la metodología en V ya que presenta un proceso efectivo para la ejecución de proyectos y cumple con los diferentes objetivos del sistema, el modelo en V consta de dos partes que son la verificación y la validación las cuales van entrelazadas de forma que la validación depende de cada etapa de la verificación de tal forma que los requisitos del sistema que se deben detallar al inicio de un trabajo de investigación son el soporte para que al final las pruebas de validación sean positivas y permitan que el sistema sea funcional, mediante el análisis del sistema se hacen posibles estas, posteriormente las pruebas del sistema, mediante el diseño del software se realizarán las pruebas de integración y mediante el diseño de los módulos se lograra realizar las pruebas unitarias. A continuación, en la figura 14 se presenta las etapas que contiene el modelo V.

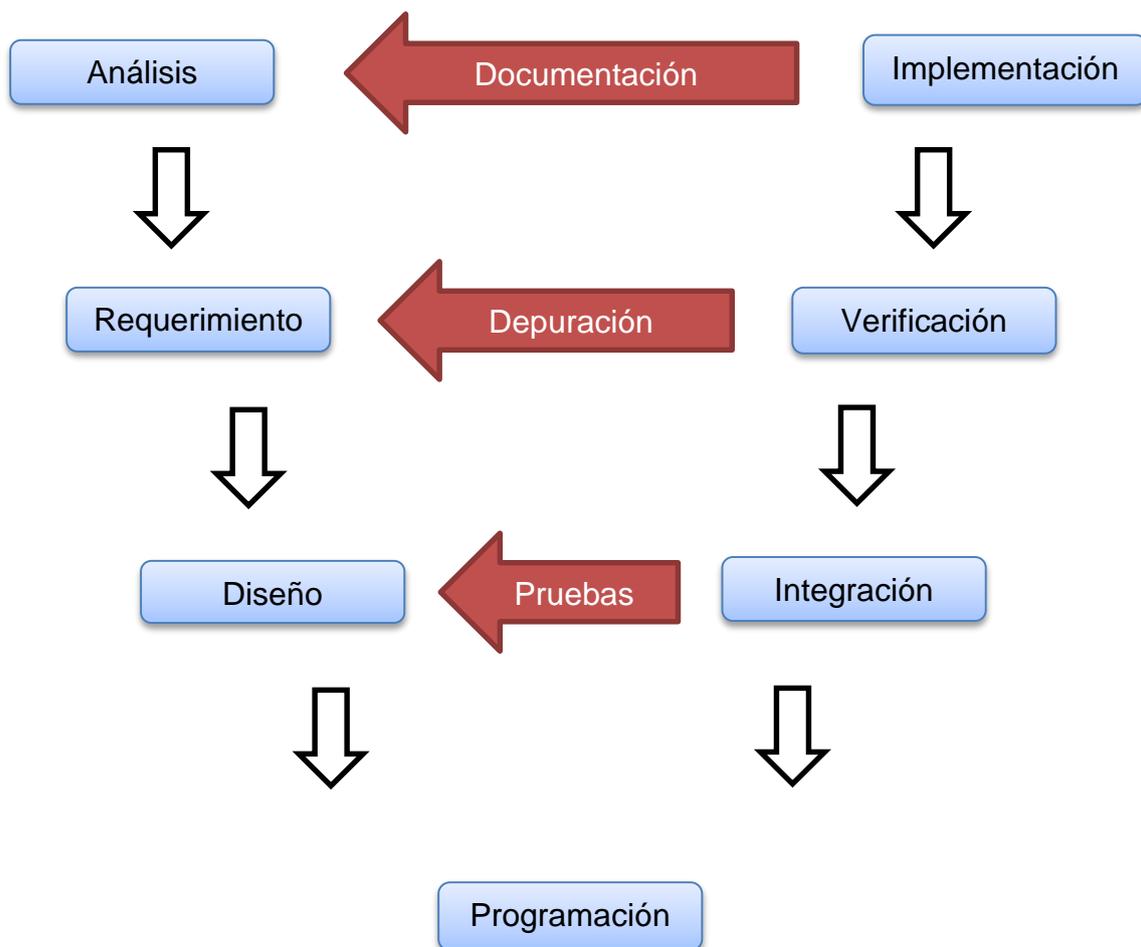


Figura 14. Fases que contiene una metodología basada en el modelo en V.

### 3.2 Descripción de proceso de funcionamiento del dispositivo

La ingeniería permite el análisis, representación y reproducción de productos a partir de la manipulación de la electrónica aplicada, el proceso de ingeniería implica el estudio del funcionamiento de dispositivos Raspberry el cual permite dar pautas a la configuración y la aplicación de *software* Python que permita mediante el uso de una cámara inteligente y de igual forma la aceptación a los algoritmos el reconocimiento facial, aplicado a los conductores de vehículos; por lo que es posible aplicar la ingeniería empleando la tecnología electrónica para realizar la construcción del dispositivo acorde a las necesidades.

Para el funcionamiento del dispositivo primero este debe ser instalado en el vehículo y conectado a la corriente de la batería, el dispositivo Raspberry funciona mediante la alimentación del cable USB el cual tiene una salida de 5 voltios por lo que es necesario conectar el dispositivo a la batería mediante un regulador de corriente; el dispositivo debe encenderse cuando el vehículo se encienda por lo que se debe conectar al *switch* del vehículo; el dispositivo debe estar sujeto dentro del tablero del vehículo y mediante un bus de datos se conecta la cámara la cual se instalará en el tablero frente al rostro del conductor, la bocina la cual actúa como alarma se conecta al dispositivo mediante el puerto GPIO de nuestra unidad de procesamiento. En la figura 15 se puede observar, la ubicación específica del sistema con respecto al vehículo.



Figura 15. Instalación del dispositivo en la parte interna del vehículo.

Una vez instalado el dispositivo, el proceso de funcionamiento es el siguiente: el conductor se sube al vehículo, posteriormente enciende el vehículo y en ese instante se ejecuta el dispositivo ejecutando el programa de forma automática, una vez el programa esté en funcionamiento va a estar detectando el rostro y los ojos del conductor con el fin de verificar su estado de somnolencia, si el dispositivo detecta que el conductor cerró los ojos, automáticamente el sistema ejecuta una alarma auditiva la cual permite alertar al conductor que presento estado de somnolencia; al detener y apagar el vehículo, el dispositivo también se apaga; a continuación, en la figura 16 se muestra la conexión del dispositivo la que está conformado de los siguientes elementos para su correcto funcionamiento, cámara, inversor de voltaje, cable mini USB, zumbador y unidad de procesamiento.

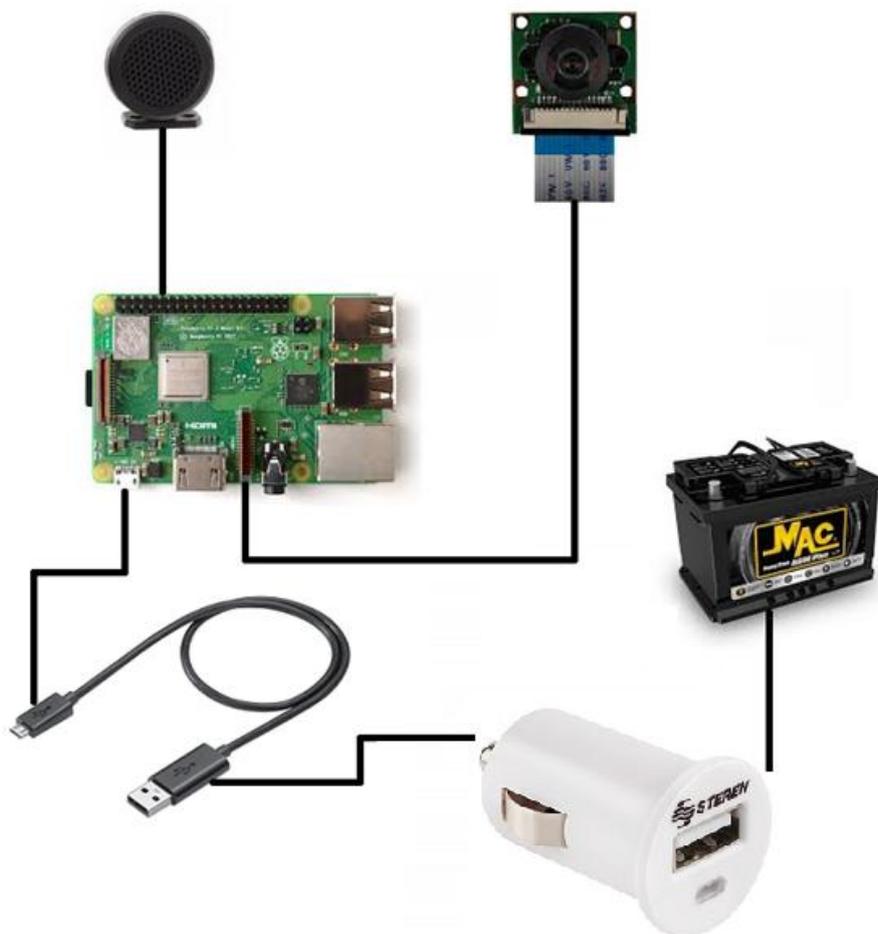


Figura 16. Diagrama general de funcionamiento del dispositivo.

Tabla 3.

*Características de los elementos del sistema*

Raspberry	Broadcom BCM2837B0 1.4 GHz 1GBRAM LPDDR2 SDRAM Wifi, Bluetooth, Ethernet Puerto GPIO de 40 pines Puerto HDMI Puertos USB 2.0 Entrada microSD	<b>3.2.1. Algoritmo de detección de somnolencia.</b> - Depende directamente de la conexión y configuración de la placa Raspberry Pi 3 b+ y la Pi Cámara para su funcionamiento y así el dispositivo pueda visualizar el estado anímico del conductor y predecir cuando entre en estado de somnolencia para dar una alerta y prevenir al conductor.
Cámara	Imagen HD 8 megapíxeles Sensor CMOS Sony Cable de 15 pines	



Figura 17. Conexión de la Pi cámara a la unidad de procesamiento.

Bocina	<ul style="list-style-type: none"> <li>{ Mini bocina de 5 voltios</li> <li>{ Pitido de alerta</li> </ul>	<ul style="list-style-type: none"> <li>} <b>3.2.2. Alarma de alerta.</b> - Permite dar una alerta auditiva al conductor.</li> </ul>
--------	--	---



Figura 18. Zumbador utilizado para el sistema de detección de somnolencia.

Fuente	<ul style="list-style-type: none"> <li>{ Conversor de corriente</li> <li>{ Entrada 12 voltios y 24 voltios</li> <li>{ Salida de 5 voltios</li> </ul>	<ul style="list-style-type: none"> <li>} <b>3.2.3. Conexión con el Vehículo.</b> - Permite conectar el dispositivo a la batería del vehículo mediante un adaptador de corriente que permite reducir de 12 a 5 voltios.</li> </ul>
Batería	<ul style="list-style-type: none"> <li>{ 12 voltios DC</li> <li>{ De vehículo</li> </ul>	



Figura 19. Conexión electrónica del dispositivo.

El algoritmo para la detección de somnolencia depende principalmente del hardware, en este caso la placa Raspberry Pi 3 modelo b + y la cámara Pi NoIR.

El funcionamiento pertinente del *software* varía de la velocidad de procesamiento de imágenes, procesador y la resolución de imagen que presente la cámara, para que así el sistema sea seguro y oportuno a la hora de detectar el estado de somnolencia de los conductores.

Para ejecutar la alarma de detección de somnolencia fue necesario utilizar un zumbador de 5 voltios que se encuentra conectada al puerto GPIO de la placa Raspberry con el fin de que mediante este puerto se pueda enviar información a la bocina para que emita el pitido de alerta y así el conductor se despierte en caso de presentar somnolencia frente al volante.

Para realizar la conexión del dispositivo al vehículo, se debe tener en cuenta que la batería de todo vehículo pequeño es de 12 voltios y los vehículos de carga que son más grandes utilizan una batería de 24 voltios por lo que es necesario utilizar un convertor de corriente de 12 o 24 voltios a 5 voltios que es la corriente con la cual funciona la placa Raspberry Pi 3 modelo b + que se utilizó en el desarrollo del presente proyecto.

El convertor de corriente debe ir conectado a la batería a través del *switch* de encendido del vehículo, con el fin de que la corriente pase y active la placa siempre y cuando el vehículo se encienda, de lo contrario la placa estará siempre apagada con el vehículo; es decir, que, si el conductor enciende el vehículo, automáticamente la placa se enciende y procede a controlar el estado de somnolencia del conductor.

Para iniciar el programa en modo automático es necesario desarrollar un código script con el cual el sistema operativo Raspbian, se ejecutará automáticamente permitiendo el control de somnolencia hacia los conductores.

## 3.2.1 Diagrama de flujo del sistema de detección de estado de somnolencia

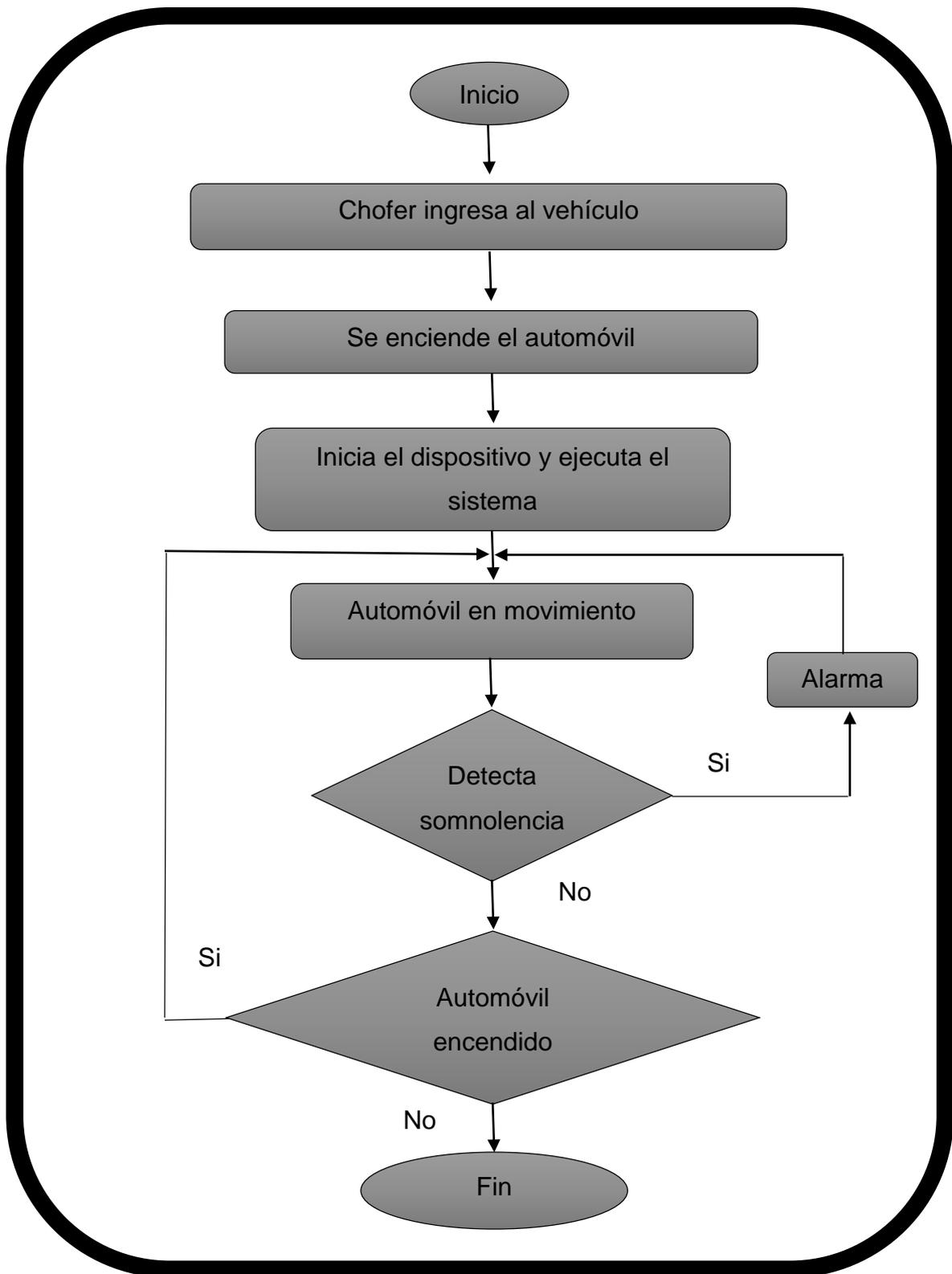


Figura 20. Diagrama de flujo del proceso de control de somnolencia.

En la figura 20 se observa, el diagrama de flujo de datos respecto al proceso de control de somnolencia, al inicio el dispositivo se encuentra apagado, el conductor ingresa al vehículo posteriormente enciende el vehículo; en este momento automáticamente se enciende el dispositivo e inicia con el proceso de detección de estado de somnolencia, posteriormente al momento que el vehículo se encuentre en movimiento generado por parte del conductor el sistema encuentra en alerta constante; el dispositivo y la cámara mediante visión artificial siempre están calculando el promedio del diámetro de apertura de los ojos, si los ojos permanecen cerrados durante un lapso promedio de tiempo; es decir, cuando el conductor no está parpadeando y sus ojos se encuentren cerrados es un indicio de que el conductor ingresó en un estado de somnolencia; si el sistema verifica que el conductor entro en ese estado, automáticamente el sistema activa una alarma auditiva que alertará al conductor hasta que ya no presente estado de somnolencia y continúe con su labor de manejo; para el caso de que no se detecte ningún estado de somnolencia en el conductor y si el automóvil siga encendido el dispositivo continuará detectando el estado anímico del conductor; una vez que el conductor llega a su destino; el detiene el vehículo, apaga el vehículo, el dispositivo automáticamente se apagará y el sistema terminará con el proceso de detección de estado de somnolencia, este sistema es sumamente preciso y eficiente a la hora de detección.

### **3.3 Requerimientos**

Para el análisis de los requerimientos del presente proyecto se estableció el estándar IEEE 29148:2011, este estándar comprende instrucciones acatados a los procedimientos relativos de la ingeniería como son proyectos, sistemas, servicios de software. Mediante la presente normativa se establecen requisitos sumamente eficientes generando atributos y condiciones a lo largo del periodo de la vida. Es importante tener en cuenta que la IEEE 29148:2011 posibilita la dirección para establecer los requisitos desarrollados en la ingeniería al igual que los procesamientos en gestión de las labores a desarrollar (Ortiz & Fernández, 2017)

### 3.3.1 Requerimientos de usuario

En la tabla 4 se determinan las limitaciones y operaciones permitidas por parte de los usuarios.

Tabla 4.

*Análisis de requerimientos de usuario*

<b>Requerimientos de Usuario</b>				
<b>#</b>	<b>Requerimiento</b>	<b>Precedencia</b>		
		<b>Excelente</b>	<b>Bueno</b>	<b>Regular</b>
1	El usuario no puede manipular la cámara.	X		
2	Es importante que el usuario permanezca en la zona de reconocimiento.	X		
3	La alarma permite alertar si el usuario que utiliza el sistema tenga completamente los ojos cerrados.	X		
4	En la noche la iluminación tiene que ser moderada.		X	

### 3.3.2 Requerimientos del sistema

Para definir los requerimientos de sistema primeramente se deben establecer los requerimientos funcionales del sistema de esta manera se llega a la obtención de procesos y propiedades que se deben otorgar. La tabla 5 se muestra un análisis respecto a la funcionalidad con ello determinan las limitaciones y operaciones permitidas de los resultados técnicos.

Tabla 5.

*Análisis de requerimientos del sistema*

<b>Requerimientos del Sistema</b>				
<b>#</b>	<b>Requerimiento</b>	<b>Precedencia</b>		
		<b>Excelente</b>	<b>Bueno</b>	<b>Regular</b>
1	Vínculo a batería de 12 voltios.	X		

2	Vínculo del dispositivo mediante un convertor de voltaje de 12 voltios a 5 voltios.	X
3	Detección facial mediante el uso de cámara Pi NoIR.	X
4	Tono de aviso especificado por el conductor.	X
5	Alerta auditiva desactivada al momento de detectar los ojos del conductor abiertos.	X
6	El sistema estará alerta cuando se desactive la alarma auditiva.	X
7	Integración de cámara Pi NoIR a la unidad de procesamiento.	X

### 3.3.3 Requerimientos de arquitectura

Para definir los requerimientos de arquitectura se debe tomar en cuenta el hardware a utilizar, tiene que adaptarse a las características planteadas por el sistema. La tabla 6 se presenta los requisitos de arquitectura del sistema tanto de software como hardware al igual el que el sistema electrónico.

Tabla 6.

#### *Análisis de requerimientos de arquitectura*

Requerimientos de Arquitectura				
#	Requerimiento	Precedencia		
		Excelente	Bueno	Regular
Requerimientos de Software.				
1	Sistema operativo Raspbian basado en Linux Debian.	X		

2	Software el cual trabaje mediante la utilización de procesamiento de imágenes en especial la fisonomía de los ojos.	X
3	Software para tratamiento de imágenes.	X
4	Software para ejecutar programas desde el dispositivo.	X

#### **Requerimientos de Hardware**

5	Dispositivo con puerto para administrar cámara a utilizar en el sistema.	X
6	Procesador capaz de soportar análisis de imágenes en tiempo real.	X
7	La cámara debe ser pequeña y de alto rendimiento.	X
8	Compatibilidad entre las entradas de la cámara con la unidad de procesamiento.	X

#### **Requerimientos Electrónicos**

9	Adaptador de corriente de 12 voltios a 5 voltios.	X
---	---	---

### **3.4 Selección de hardware y software**

Luego de verificar los requerimientos de usuario, requerimientos del sistema y arquitectura planteada tanto a *software* y *hardware*, se procede a seleccionar los diferentes programas necesarios para la programación y el procesamiento de imágenes, posterior a ello la selección de los elementos necesarios que cumplan con las características planteadas y la compatibilidad entre *software* y *hardware*.

### 3.4.1 Selección de *software*

La parte más importante para la realización del sistema es la elección de *software*, ya que existen diferentes alternativas de *software* que reconocen la detección de objetos y procesamiento de imágenes son limitadas, hay que escoger la que más satisfaga con los requisitos del sistema. En la tabla 7 se tiene dos alternativas de *software* de procesamiento para imágenes, estos tipos de *software* se asemejan a los requerimientos antes señalados.

Para la elección del *software* se tomaron en cuenta dos aspectos, la primera opción es el funcionamiento de *software* en *Open Source*, lo cual las dos opciones propuestas cumplen con esta opción, y como segunda opción se tomó en cuenta el análisis de imágenes en tiempo real y se obtuvo como resultado que la mejor opción es el *software OpenCV* ya que permite un mejor análisis de imágenes en tiempo real y se adapta a los requerimientos.

Tabla 7.

*Selección de software de procesamiento de imágenes*

Software analizador de imágenes.	Requerimientos		Evaluación
	OP 1	OP 2	
<i>OpenCV</i>	1	1	2
<i>SimpleCV</i>	1	0	1

1 -Satisface

0 -No satisface

Para la selección del *software* de programación se tomaron en cuenta dos aspectos, primeramente, la compatibilidad con *OpenCV* y como segundo aspecto que sea de código abierto; a continuación, en la tabla 8 se presentan las dos alternativas de programación para trabajar en el sistema.

Orientados en los requerimientos de *software* establecidos en la tabla 6 para la selección del *software* de programación se valoraron dos aspectos fundamentales en primer lugar la compatibilidad con *OpenCV* lo cual dio como resultado que los dos programas analizados son compatibles, y como segunda opción la ejecución de scripts en menor tiempo a diferencia de otros lenguajes de programación lo cual dio como resultado que el *software* de programación seleccionado es Python esto se debe al uso de scripts para la ejecución.

Tabla 8.

*Selección de software de programación*

Software de programación	Requerimientos		Evaluación
	OP 1	OP 2	
Python	1	1	2
Java	1	0	1

1 -Satisface

0 -No satisface

### 3.4.2 Selección de hardware

Para la selección del hardware se tomó en cuenta la velocidad del procesador, memoria RAM disponible en la unidad de procesamiento; a continuación, la tabla 9 detalla cómo se valoró a los distintos sistemas embebidos, además se tomó mucho en cuenta que se cumplan los requisitos de hardware presentados con anterioridad.

Enfocados en los requerimientos de hardware para la selección del hardware se tomaron en cuenta dos aspectos importantes las cuales fueron la como primero la compatibilidad con la cámara de alto rendimiento en la cual la única opción de hardware que no cumplió con este ítem fue Intel Galileo por lo que este dispositivo no puede ser utilizado en el presente proyecto; como segundo aspecto es la velocidad de procesamiento de imágenes en tiempo real, para lo

cual se analizó que Arduino Mega no cumplen con este ítem, por lo que se seleccionó la placa Raspberry Pi 3 modelo b + ya que cumple con los ítems analizados y mencionados con anterioridad.

Tabla 9.

*Selección de hardware*

Hardware	Requerimientos		Evaluación
	OP 1	OP 2	
Raspberry Pi 3 Modelo B +	1	1	2
Arduino Mega	1	0	1
Intel Galileo	0	1	1

1 -Satisface

0 -No Satisface

Una vez seleccionado la unidad de procesamiento con el que se va a trabajar en el sistema se procede a seleccionar la mejor opción de cámara de alto rendimiento, para lo cual se realiza el análisis como se puede observar en la tabla 10.

Orientados en los requerimientos de hardware para la selección de la cámara con la que trabajara el sistema se tomaron en cuenta dos opciones de las cuales como primera opción fue el tamaño y tipo de cámara en la cual no cumple con este ítem es la cámara *IP D-Link* ya que son cámaras grandes que ocupan mucho espacio y su calidad de imagen es normal; como segunda opción se tomó en cuenta la compatibilidad con puerto USB o puerto de conexión del dispositivo de hardware seleccionado lo cual es favorable para todas las cámaras restantes; por estas razones las dos cámaras Raspberry pueden ser utilizadas y se recomienda cualquiera de ellas a diferencia que la Pi cámara V2 trabaja en modo diurno y la Pi NoIR cámara V2 trabaja en modo diurno como nocturno.

Tabla 10.

*Selección de la cámara que usará el sistema*

Hardware	Requerimientos		Evaluación
	OP 1	OP 2	
Pi cámara V2	1	1	2
Pi NoIR cámara V2	1	1	1
Cámara IP D-Link	0	1	1
1 -Satisface			
0 -No Satisface			

En tabla 11 se puede observar, de forma clara sobre la selección de *software* respecto a la programación como el tratamiento de imágenes a ser utilizados en el sistema, de igual forma un detalle claro sobre el hardware a utilizar y la cámara de alto rendimiento, con ello el sistema logre detectar el estado de somnolencia en conductor de vehículos.

La selección del hardware fue realizada a partir de la verificación de memoria RAM y velocidad de procesamiento por lo que la opción indicada fue trabajar con Raspberry tanto en la unidad de procesamiento como en la cámara de alto rendimiento y en cuestión de *software* se seleccionó un software libre ya que los dispositivos Raspberry funcionan mediante plataforma Linux en este caso el sistema operativo es Raspbian por lo que el *software* seleccionado fue Python y *OpenCV*.

Tabla 11.

*Software y hardware utilizados en el sistema de somnolencia*

Detalle	Selección software y hardware
<i>Software</i> de reconocimiento facial	OpenCV

---

Software de programación	Python
Hardware	Raspberry Pi 3 modelo B+
Cámara	Raspberry Pi cámara V2

---

## 4 Diseño

Para el diseño del dispositivo de alerta basada en la detección de somnolencia para conductores de vehículos mediante el uso de visión artificial, fue necesario dividir en tres etapas, entre las cuales como primera etapa está el ensamblaje de todos los dispositivos a utilizar como: Raspberry Pi cámara y la alerta auditiva en la unidad de procesamiento, la siguiente etapa fue la instalación del sistema Raspbian en la memoria microSD y como etapa final la configuración de los diferentes programas que permiten el funcionamiento del sistema para la detección de estado de somnolencia.

### 4.1 Ensamblaje del sistema

Para el ensamblaje del sistema fueron necesarias los siguientes elementos: unidad de procesamiento, cámara y un zumbador; la unidad de procesamiento trae por defecto un pin para conexión del bus de datos, la conexión de la cámara es sencilla y se la realiza tal como se muestra en la figura 21.



*Figura 21.* Conexión de la cámara a través del puerto por defecto Raspberry.

Tomado de (Raspiman, 2017)

Posteriormente se procede a conectar la bocina en el puerto GPIO de la unidad de procesamiento, este puerto trae 40 pines los cuales están divididos de la siguiente forma: dos puertos de salida de voltaje de 3.3 voltios los cuales son el puerto 1 y 17, dos puertos de salida de voltaje de 5 voltios los cuales son el puerto 2 y 4, ocho puertos *ground* para descarga los cuales permiten la base de cualquier voltaje los cuales son los puertos 6, 9, 14, 20, 25, 30, 34 y 39, dos puertos para comunicación en transmisión y recepción los cuales son los puertos 8 y 10, dos puertos de reserva que son el puerto 27 y 28, veinte y cuatro puertos de entrada y salida tanto analógicos como digitales que son los puertos sobrantes; a continuación, se muestra en la figura 22 un diagrama referente al puerto GPIO y un detalle sobre las partes internas de la Raspberry.

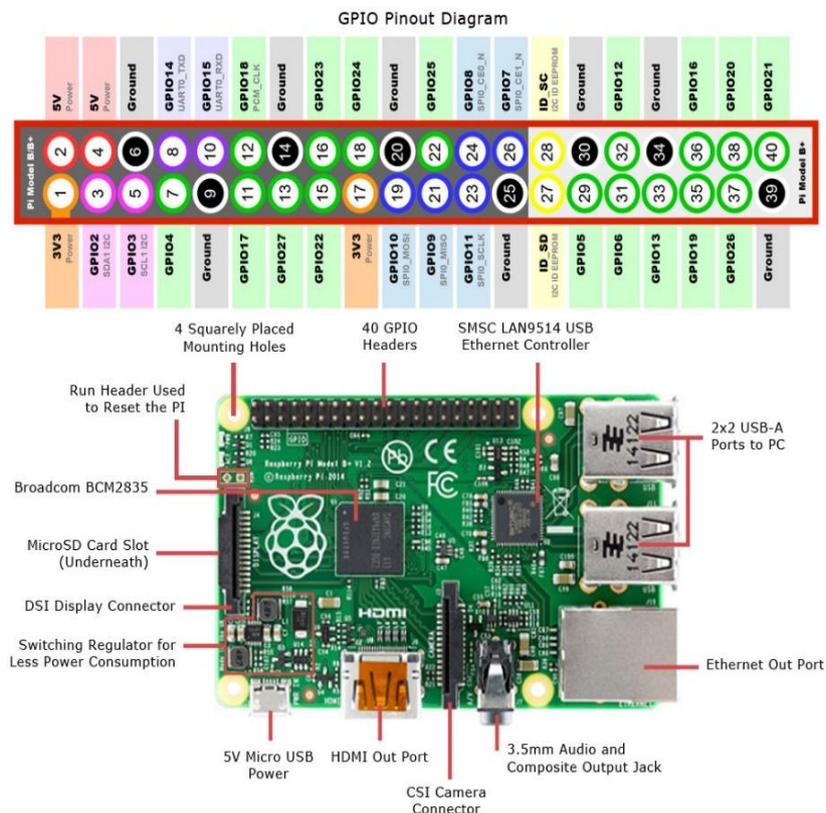


Figura 22. Distribución del puerto GPIO Raspberry Pi 3 Modelo b +.

Tomado de (NEOTEO, 2016)

Para el presente proyecto se conectó la bocina en el puerto número 7 definido como salida digital y en el puerto número 9 definido como descarga de corriente;

de tal forma que cuando por el puerto 7 se envía corriente, la bocina se activa y si no se envía corriente la bocina permanece apagada.

## 4.2 Instalación del sistema Raspbian

Para la instalación del sistema operativo Raspbian fue necesario en primer lugar descargar el *software* Raspbian *Stretch With Desktop*, posteriormente se procede a grabarlo en la microSD mediante un *software* que permite grabar la imagen del sistema en la microSD la cual debe tener un mínimo de 32 Gb de espacio o superior; para el presente proyecto la grabación de la imagen se lo realizó en el sistema operativo Windows 10 mediante el programa *Win32DiskManager* cuya configuración se muestra en la figura 23.

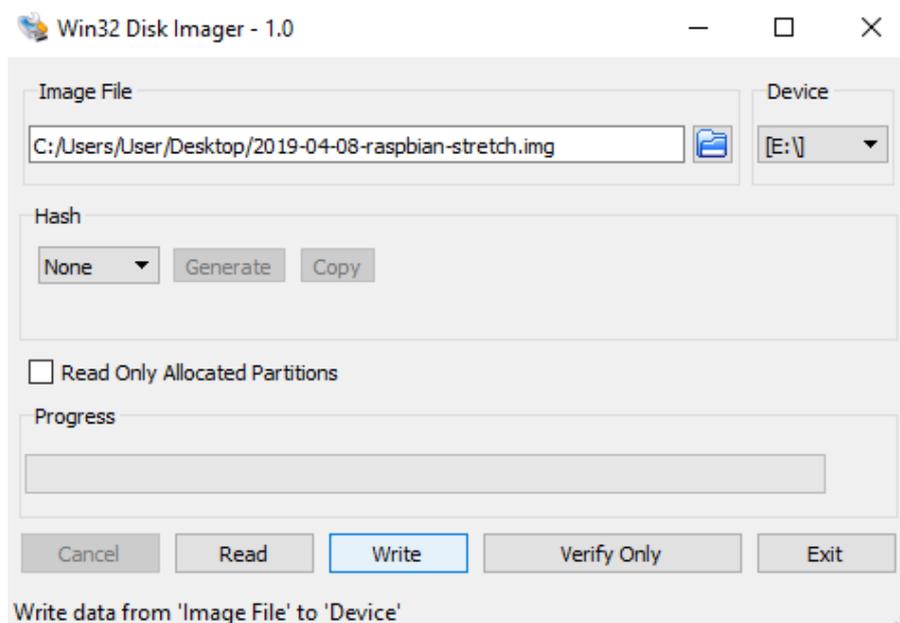


Figura 23. Instalación de sistema operativo Raspbian.

Una vez seleccionado el archivo de imagen del sistema operativo Raspbian y la unidad en la que se instalará, se procede a quemar el sistema Raspbian en la microSD; luego de que llegue al 100% el sistema quedará grabado de forma correcta y estará listo para su funcionamiento en la unidad de procesamiento; posteriormente se inserta la microSD grabada en la Raspberry Pi 3 modelo b + tal como se muestra en la figura 24.

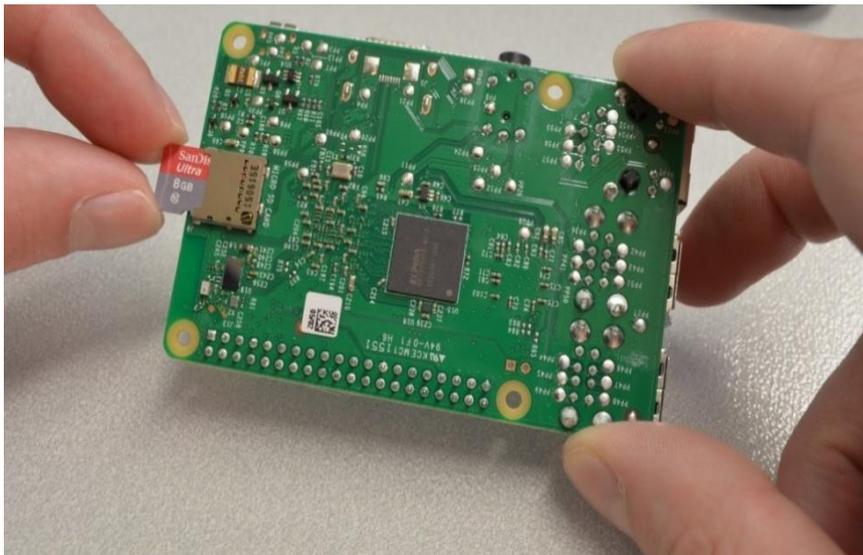


Figura 24. Conexión microSD en Raspberry Pi 3 modelo b +.

A continuación, se conecta el dispositivo a un puerto HDMI, un teclado, un *mouse* y la fuente de corriente, luego se enciende el dispositivo y se procede a configurar el sistema operativo, finalmente ya está listo el sistema para su funcionamiento. En la figura 25 se puede observar, la interfaz con respecto al escritorio del sistema operativo Raspbian.

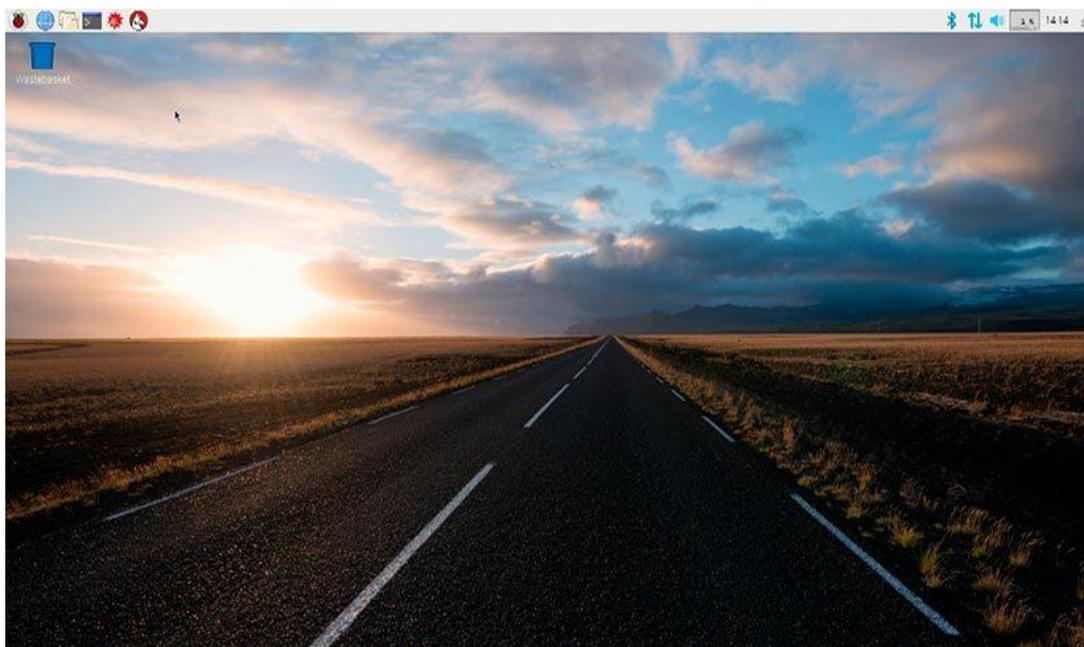


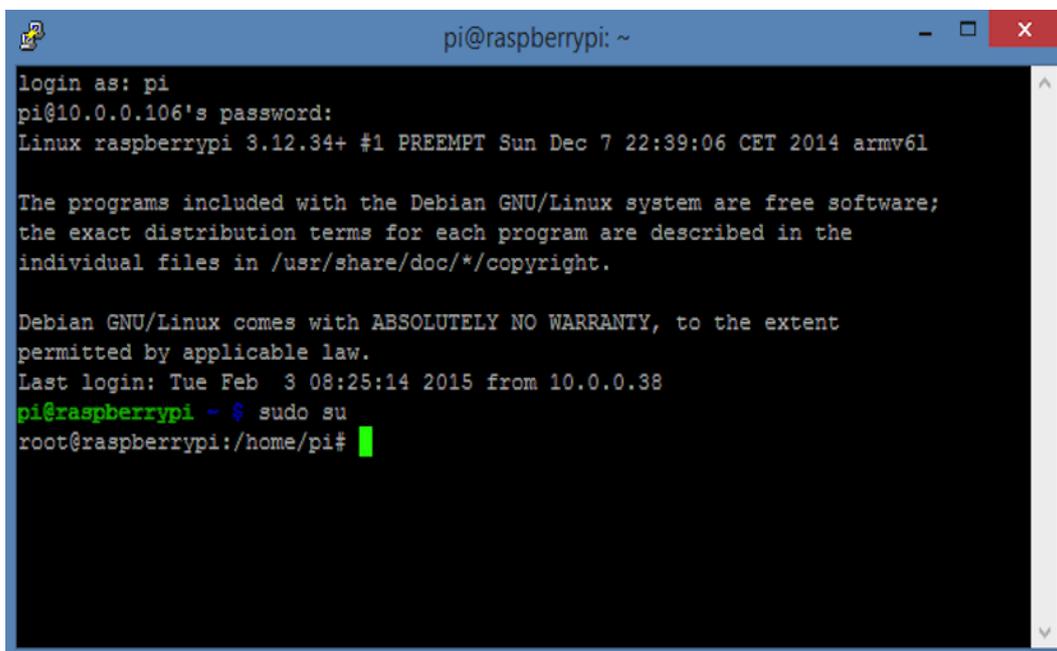
Figura 25. Escritorio Raspbian.

## 4.3 Configuración de programas para el sistema

Los programas necesarios para el funcionamiento del sistema embebido son:

- Software Pi camera
- Python3
- OpenCV
- Archivos *Haar\_Cascade*

Todos los programas anteriormente indicados deben ser instalados mediante la ventana de comandos la cual se muestra en la figura 26.



```
pi@raspberrypi: ~
login as: pi
pi@10.0.0.106's password:
Linux raspberrypi 3.12.34+ #1 PREEMPT Sun Dec 7 22:39:06 CET 2014 armv61

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Feb  3 08:25:14 2015 from 10.0.0.38
pi@raspberrypi ~ $ sudo su
root@raspberrypi:/home/pi#
```

Figura 26. Ventana de comandos Raspbian.

### 4.3.1 Software Raspberry Pi camera

Para la instalación del *software* Pi cámara fue necesario la instalación varias librerías, y configuraciones necesarias con ello la cámara pueda trabajar de la mejor manera frente al sistema de somnolencia, cuando se requiere dar uso de la cámara Pi V2 se necesita habilitar la cámara, para lo cual se debe ingresar al terminal y escribir el siguiente comando `sudo apt-get install python-picamera` al

ejecutar este comando se desplegará varias opciones de configuración del sistema Raspbian como se presenta en la figura 27.

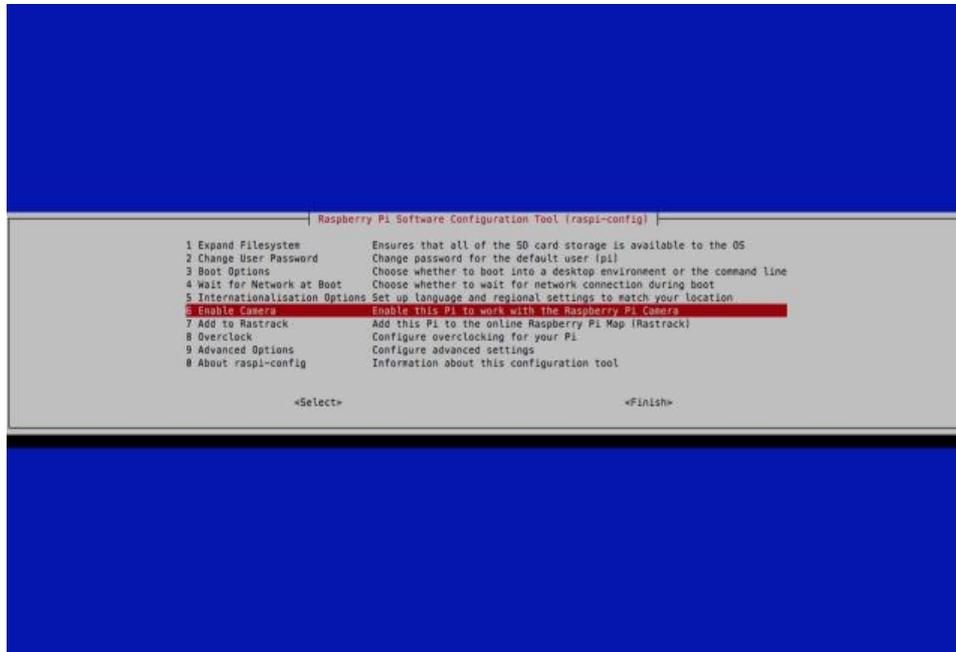


Figura 27. Habilitación Pi camera mediante consola.

### 4.3.2 Python3

Para la instalación de Python3 fueron necesarios instalar varios comandos los cuales permitieron actualizar las librerías del sistema operativo y de igual manera obtener todos los archivos necesarios para la actualización de Python3; el sistema trae instalado por defecto Python2, fue necesario actualizar la versión de Python3 para la realización del proyecto debido al uso de procesamiento de imágenes.

### 4.3.3 OpenCV

Para la instalación de *OpenCV* fue necesario seguir varias etapas de configuración las cuales permitieron configurar la librería en el sistema operativo Raspbian; las etapas para la configuración de la librería *OpenCV* se detallan a continuación en la figura 28.

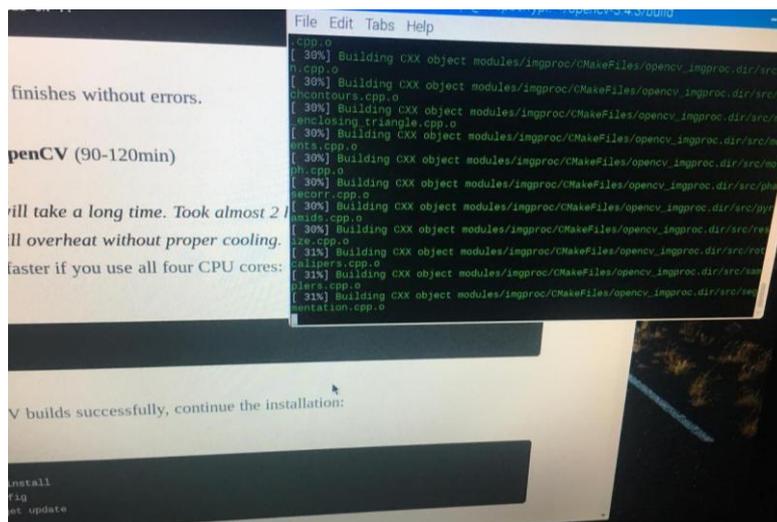


Figura 28. Instalación de *OpenCV* en el sistema operativo Raspbian.

### Paso 1: Asegurarse de que el sistema operativo esté actualizado

Primeramente, se actualiza el sistema operativo para verificar si existe alguna actualización importante y así permanezca el sistema siempre con sus últimas versiones.

### Paso 2: Configurar SSH y utilidades

En este paso se configuran los servicios y utilidades que deben estar presentes para la instalación de la librería *OpenCV*.

### Paso 3: Configurar swap y asignación de memoria GPU

En este paso se instalan ciertas características que requiere la librería y se aumenta la memoria a 128Gb para que la librería obtenga mayor espacio de memoria y el dispositivo no se quede bloqueado por falta de memoria.

### Paso 4: Instalar dependencias

Se instalan algunas dependencias necesarias para que la instalación de la librería sea correcta respecto a *OpenCV*, caso contrario algunas características de la librería *OpenCV* no podrían funcionar ya que no tendrían soporte para

realizar la configuración de su tipo de código de funcionamiento y la instalación de la librería sería fallida.

### **Paso 5: Instalar Python 3**

Se debe instalar Python3 en caso de no estar instalado, puesto que la librería *OpenCV* 4.0 necesita de este sistema para su funcionamiento.

### **Paso 6: Instalar pip3.**

Es necesario actualizar el sistema pip a pip3 el cual permite un mejor funcionamiento y rendimiento de Python3.

### **Paso 7: Obtener el último código fuente de OpenCV 4.0.**

En este paso se descarga desde la página oficial la librería *OpenCV* 4.0 y *OpenCV Contrib* para posteriormente instalarla en el sistema operativo.

### **Paso 8: Instalar Numpy y Scipy.**

En el paso 8 se instala el programa Numpy que es indispensable para el funcionamiento de la librería *OpenCV* y el programa Scipy que de igual manera contribuye al funcionamiento de la librería *OpenCV* 4.0.

### **Paso 9: Compilar OpenCV.**

A continuación, se ingresa a la carpeta *OpenCV* 4.0 y se inicia el proceso de compilación de los archivos que componen la librería *OpenCV* 4.0.

### **Paso 10: Construir OpenCV.**

Finalmente, se construye por completo la librería *OpenCV* para que forme parte del sistema operativo y Python; es decir, reconstruye el código de todos los archivos necesarios del sistema operativo para que puedan estar acordes con los procesos que realiza la librería en su funcionamiento y administración para el procesamiento de imágenes.

#### 4.3.4 Archivos *Haar\_Cascade*

Son archivos que permiten realizar la detección de patrones que presentan diferentes objetos, para el caso de la realización del presente proyecto fue necesario utilizar los archivos *Haar\_Cascade* que permiten detectar patrones generales de rostro de una o varias personas al igual fueron necesarios utilizar los archivos que permitan detectar los patrones de los ojos, mediante coordenadas del ojo izquierdo como del ojo derecho.

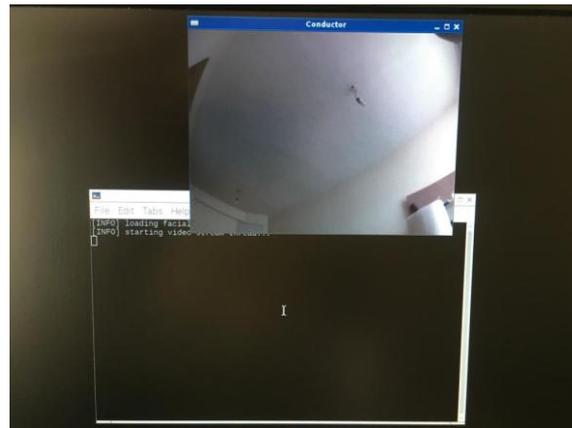
Los archivos *Haar\_Cascade* con los cuales se trabajó se encuentran en internet, el usuario también puede trabajar con la dirección URL o a su vez pueden descargar los archivos y tenerlos en la unidad de procesamiento.

Los archivos pueden ser descargados desde la dirección URL y ubicarlos en la carpeta del programa escrito en Python3; los archivos *Haar\_Cascade* que se deben tener descargados para el funcionamiento del sistema son los que permiten el reconocimiento del rostro de forma frontal.

En caso de no descargar los archivos, se puede hacer una referencia directa a la dirección URL de cada archivo, de tal forma que con tan solo dar la ruta URL del archivo, el sistema funcionará de la misma forma que si se descargaran estos archivos en el equipo, por lo que la forma que el usuario seleccione debe tenerse en cuenta a la hora de realizar la llamada en el programa escrito en Python puesto que si no se da la ruta correcta de los archivos el sistema no funcionará.

#### 4.3.5 Inicio automático del sistema

El sistema debe iniciarse de forma automática al momento de encender el vehículo por parte del conductor, esto se logró mediante la ejecución de un *script* que permite iniciar automáticamente el programa de Python; este proceso fue realizado mediante el comando *crontab -e* y el comando *@reboot sudo python3*, seguido de la dirección en donde se encuentre el archivo principal de Python. La figura 29 muestra, el ambiente sobre el inicio automático del sistema.



*Figura 29.* Inicialización del sistema en modo automático.

#### **4.3.6 Selección de tonos por parte del conductor.**

El conductor tiene la facilidad de seleccionar el tono de alarma mediante un pulsador el cual estará ubicado en el tablero al alcance del conductor, en el sistema están detallados tres tonos diferentes de alarma y cada usuario puede seleccionar el que desee; entre los tonos que están dispuestos en el sistema se presenta un tono continuo, un tono pausado y otro pausado por un tiempo más prolongado; el pulsador está conectado en el puerto GPIO en los pines 3 y 6, el pin 3 definido como salida de voltaje y el pin 6 definido como descarga de corriente eléctrica. En la figura 30 se puede observar, un pulsador conectado a la unidad de procesamiento el cual permite seleccionar tres tipos de tonos establecidos por el sistema embebido.



*Figura 30.* Selección de tonos de alarma mediante pulsador.

## 4.4 Diagrama de bloques

En el presente diagrama de bloques se puede observar cuatro etapas el cual conforman al sistema, cada una de estas etapas están conformada por subprocesos, de esta manera el sistema actuara frente al individuo de una manera eficiente sabiendo detectar si un usuario presenta estado de somnolencia. En la figura 31 se puede observar cada etapa del diagrama de bloques.

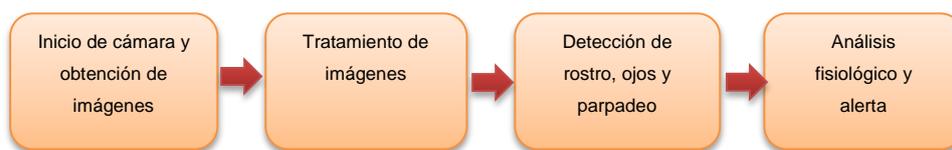


Figura 31. Diagrama de bloques.

En el diagrama de bloques se puede observar que el primer bloque se da inicio a la cámara y la obtención de imágenes, para lo cual es necesario iniciar el sistema mediante un script que permita el encendido de la cámara de forma instantánea para que trabaje constantemente mientras el automóvil este encendido. En el segundo bloque ya se administra el tratamiento de imágenes mediante el *software OpenCV* el cual es el encargado de analizar y medir el nivel de somnolencia. En el tercer bloque se analiza el rostro y los ojos del conductor en tiempo real con el fin de verificar el estado de ánimo del conductor. En el cuarto bloque se analiza la fisiología del conductor; es decir, su estado de ánimo, en caso de que éste presente estado de somnolencia automáticamente se activara una la alarma auditiva que alerte al conductor.

### 4.4.1 Bloque 1: Inicio de cámara y obtención de imágenes

Primeramente, se debe hacer funcionar la cámara en este caso la utilizada en el sistema para el proceso de captura de imágenes, posterior a ello, se procede a instalar las diferentes librerías que permitan el funcionamiento del dispositivo y de esta manera mediante los clasificadores utilizados poder detectar el estado de somnolencia en los conductores. En la figura 32 se puede observar, una

cámara que cuenta con un sensor digital que está conformado de celdas fotosensibles, las cuales analizan la luz y es ahí donde se capta una pequeña escena de la imagen que se requiere obtener en forma de señales luminosas para más luego poder transformarlas en señales analógicas.



Figura 32. Inicialización y captura de imágenes.

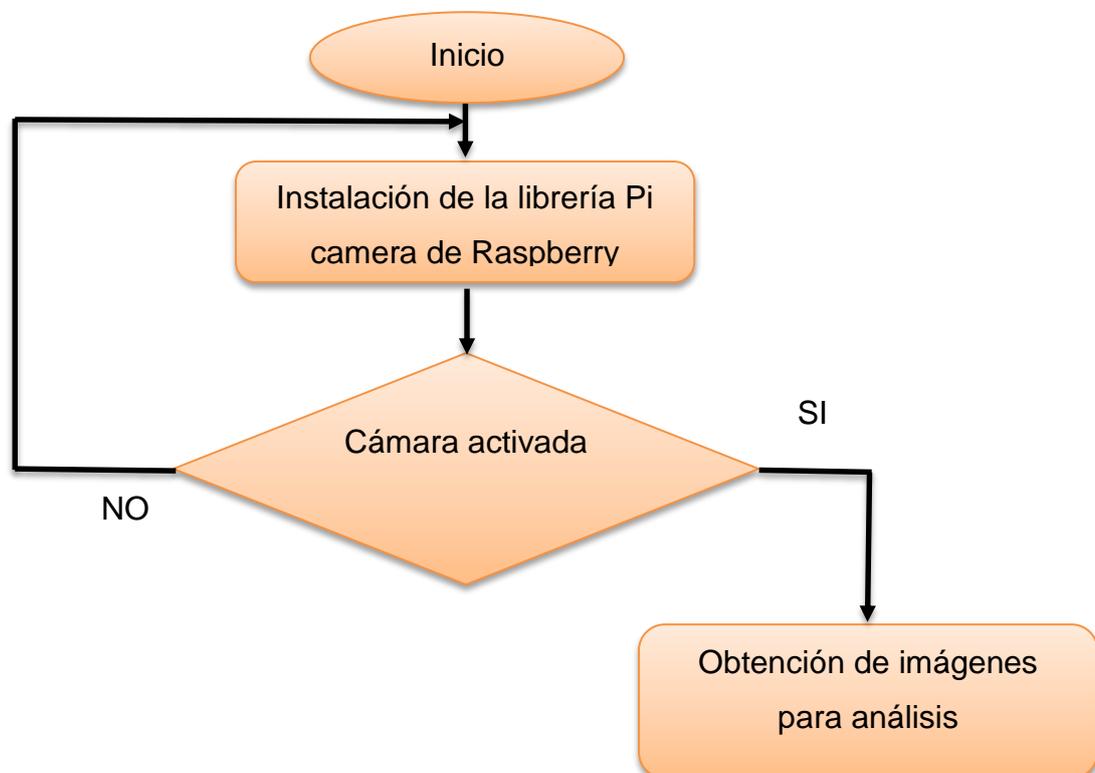


Figura 33. Diagrama de bloques 1.

#### 4.4.2 Bloque 2: Tratamiento de imágenes.

En esta etapa se puede verificar la calibración de las imágenes tanto en el brillo al igual que el contraste con el fin de ser analizadas para obtener un resultado sin dificultades ni deformaciones de tal forma que el resultado del análisis de las imágenes sea óptimo y no presente fallas puesto que si se diera algún error el resultado podría ser fatal. En la figura 34 se puede observar, la conversión de colores a escala de grises que ayuda a saber cuáles son las fronteras de la imagen a analizar, esta técnica se halla dentro de las operaciones de *OpenCV* dando uso de la función `cvtColor` se puede dar o realizar varias transformaciones a una imagen, entre estas se encuentran: `CV_BGR2GRAY` `CV_RGB2GRAY`, `CV_GRAY2BGR`, `CV_GRAY2RGB`.



Figura 34. Imagen en escala de grises.

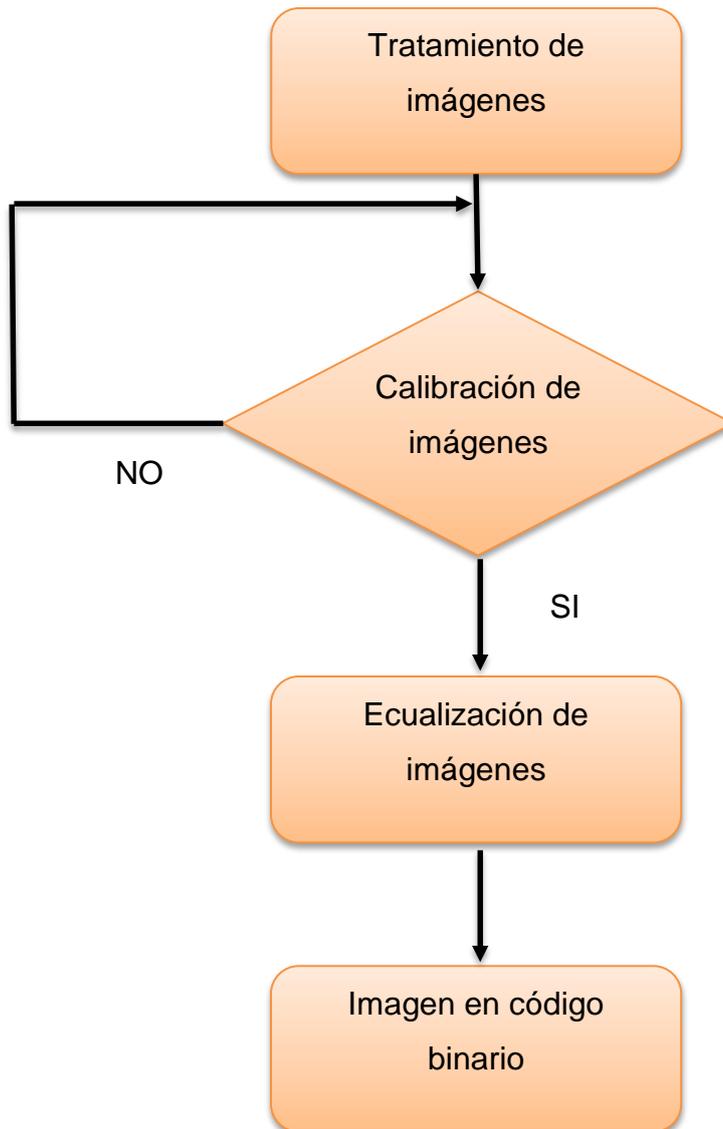


Figura 35. Diagrama de bloques 2.

#### 4.4.3 Bloque 3. Detección de rostro, ojos y parpadeo.

En la Figura 36 se puede observar, que el sistema realiza la detección de los ojos y su gráfico de contorno para verificar posteriormente el promedio del diámetro de apertura de los ojos, si el conductor presenta los ojos cerrados se activara una alarma la cual permita alertar al conductor que presenta estado de somnolencia. En la figura 36 se puede observar, que cuando el conductor sobrepase el umbral de parpadeo definido en la programación del sistema de detección de somnolencia este activara una alarma de advertencia.



Figura 36. Detección de ojos por parte del sistema.

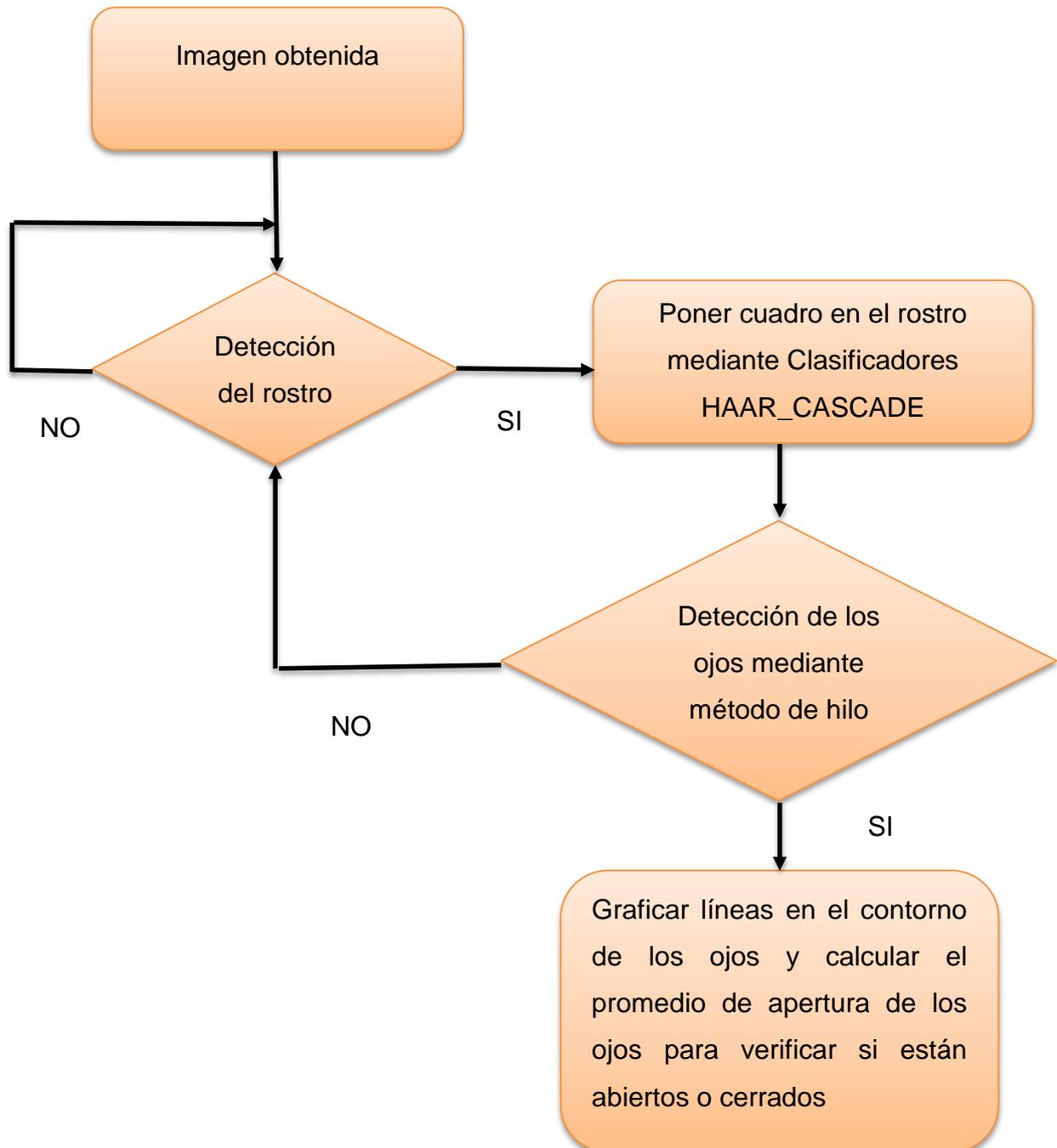


Figura 37. Diagrama de bloques 3.

#### 4.4.4 Bloque 4. Análisis fisiológico y alerta.

Para la detección y análisis fisiológicos del estado de somnolencia de un conductor, es necesario verificar el valor de apertura del ojo mediante un cálculo matemático el cual permite conocer la distancia entre la parte superior e inferior de los ojos, si el ojo está abierto la distancia excede un valor constante y si los ojos están cerrados el valor calculado será menor que el de la constante; entonces para el caso que el valor calculado sea menor que la constante de somnolencia, la alarma se activa y esta no deja de alertar hasta que el conductor abra los ojos. La figura 38 muestra, de forma clara como el sistema al momento que el conductor presenta estado de somnolencia activará una alerta auditiva la cual permita alertar al conductor sobre su estado de somnolencia, la alarma del dispositivo se desactivará al momento que el conductor vuelva a su estado normal.



Figura 38. Análisis fisiológico.

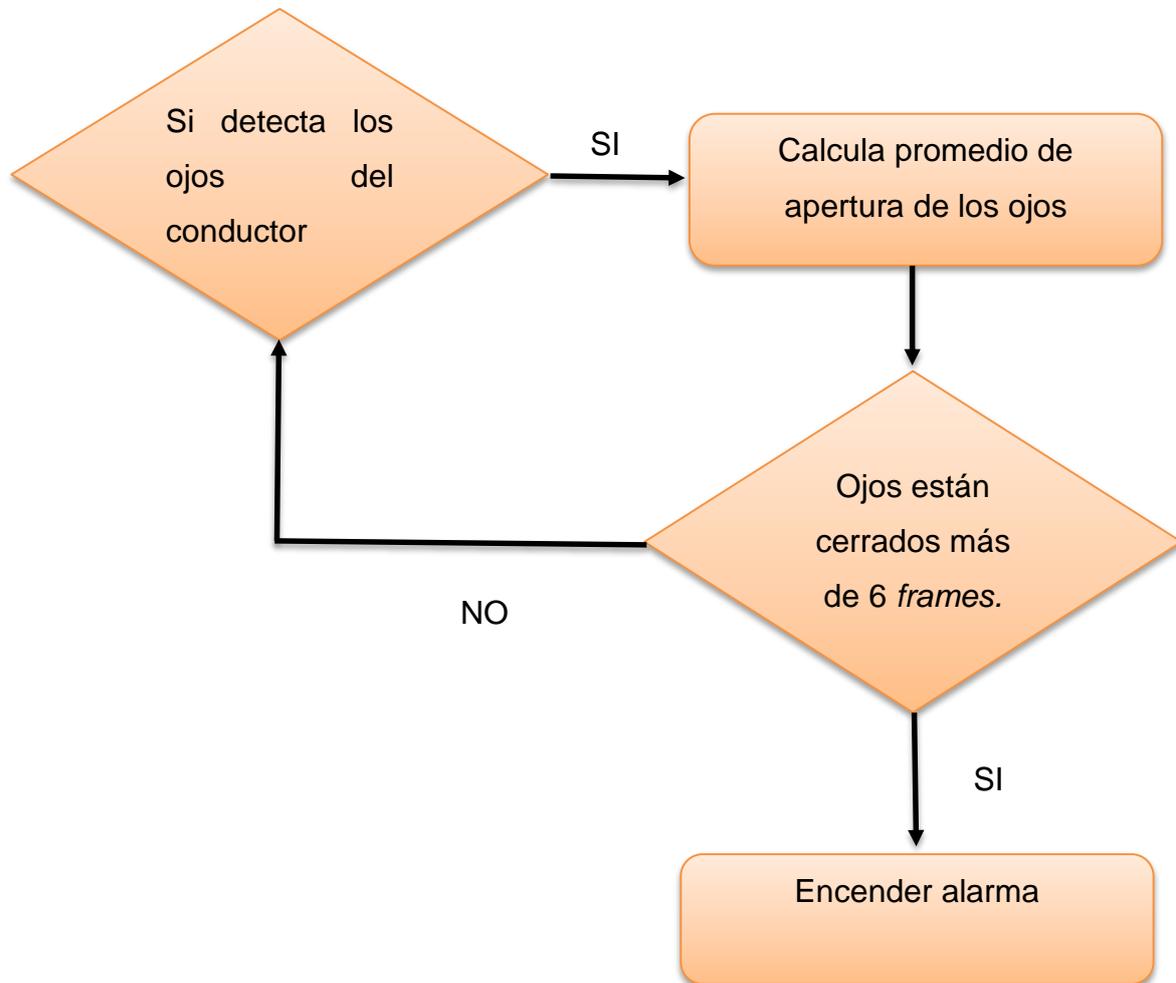


Figura 39. Diagrama de bloques 4.

## 5 Implementación

Para la implementación del sistema se parte desde el análisis sobre los diagramas de bloques para posteriormente analizar el código fuente que permite el funcionamiento de dicho sistema.

### 5.1 Diagrama del circuito

En la figura 40 se puede observar el diagrama del circuito electrónico para lo cual se utilizó una unidad de procesamiento y una cámara de alto rendimiento adicional a ello una fuente de alimentación la cual posee entrada USB de 5 voltios que desde la batería del vehículo alimenta a la unidad de procesamiento.

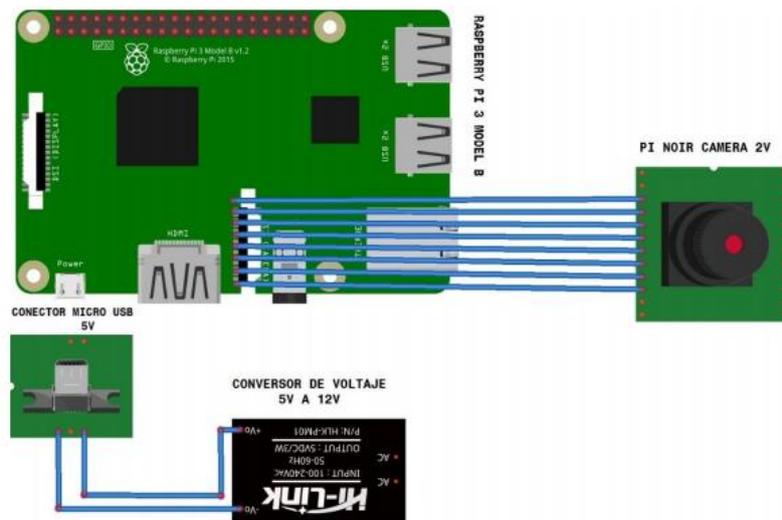


Figura 40. Diagrama circuital del sistema de detección de somnolencia.

## 5.2 Ubicación de la cámara

La cámara tiene que ser ubicada en un punto específico donde puede estar captando el rostro del conductor con el fin de verificar su estado; es necesario que la cámara sea ubicada en la parte frontal del tablero o en la parte central con un ángulo de rotación que apunte directamente al rostro del conductor. En la figura 41 se puede observar, la ubicación de la cámara dirigida al conductor logrando de esta manera un mejor panorama de monitoreo.



Figura 41. Ubicación específica de la cámara anti somnolencia.

### 5.3 Código fuente

El código fuente es el que permitió el control del dispositivo y la realización del proceso mediante visión artificial; a continuación, se detalla el código fuente de forma detallada.

Para la descripción del código primeramente se deben Importar las librerías necesarias para el funcionamiento del sistema, se pueden observar algunas librerías como la *imutils* que sirve para el control de video, GPIO se utiliza para controlar el puerto tanto de entradas como salidas de información, también se deben referenciar las librerías *Cv2* y *dblib* que permiten el reconocimiento facial; luego, se inicializa el puerto GPIO y se le configura el pin 7 como salida de información y se lo inicializa en *False*; es decir, que inicialmente no enviará corriente a la alarma por lo que se mantendrá en silencio.

Posteriormente se hace referencia a los archivos de clasificación *Haar* que permiten el reconocimiento del rostro y ojos, luego de esto se inicia un bucle *while* que permite que el sistema trabaje de forma constante sin detenerse, dentro del bloque *while* deben ir las instrucciones de reconocimiento del rostro y de los ojos.

Dentro del bloque *while* se debe poner el código que permite el reconocimiento del rostro y se declaran variables con el fin de crear las líneas del contorno de cada ojo; posteriormente, se declara un bucle *for* para calcular y delimitar el contorno del ojo; primeramente, se grafica el contorno de cada ojo mediante el método de hilos y luego se procede a realizar el cálculo de apertura del ojo para lo cual se hace obtiene el diámetro de apertura de cada ojo y se hace un promedio del valor de apertura, si el promedio es inferior a 0,23 significa que los ojos están cerrados caso contrario los ojos están abiertos.

Al momento que el sistema capta que los ojos están cerrados una variable inicializada en cero se incrementa mientras los ojos sigan cerrados, al momento en que la variable llegue al número 6, la alarma se activará automáticamente.

Cuando el conductor se despierte por causa de la alarma, la variable vuelve automáticamente a cero y se inician nuevamente los cálculos; el sistema se va a encontrar siempre bajo este modelo calculando siempre el promedio de apertura de los ojos y contando las *frames* en donde los ojos se encuentran cerrados.

## 5.4 Verificación del código

### 5.4.1 Código reconocimiento facial

Una vez que el código fuente del sistema fue realizado y el resultado de todas las configuraciones del sistema operativo fueron correctas, se dio inicio a la primera ejecución del sistema el cual inició en su primera fase que fue el reconocimiento del rostro; a continuación, en la figura 42 se muestra la captura de rostro encerrado en un cuadro verde, lo cual indica que este proceso fue satisfactorio.

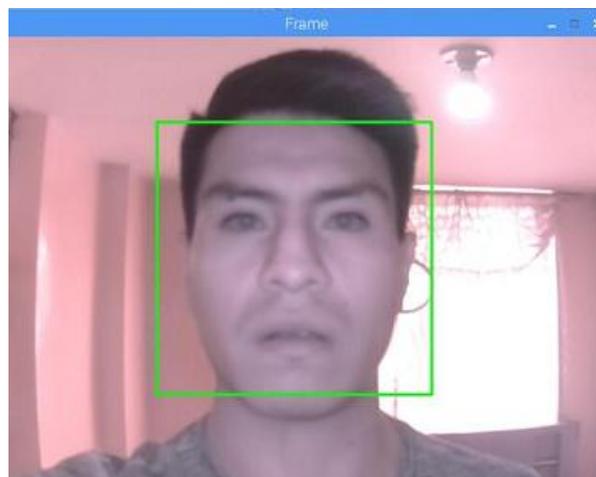


Figura 42. Ensayo de código fase uno.

### 5.4.2 Código reconocimiento ojos

Luego del reconocimiento del rostro, el siguiente paso fue captar los ojos mediante la gráfica de contorno de cada ojo, para lo cual fue necesario dibujar líneas en el contorno de los ojos mediante el método de hilos; a continuación, en

la figura 43 se muestra el resultado del reconocimiento de ojos de manera satisfactoria.

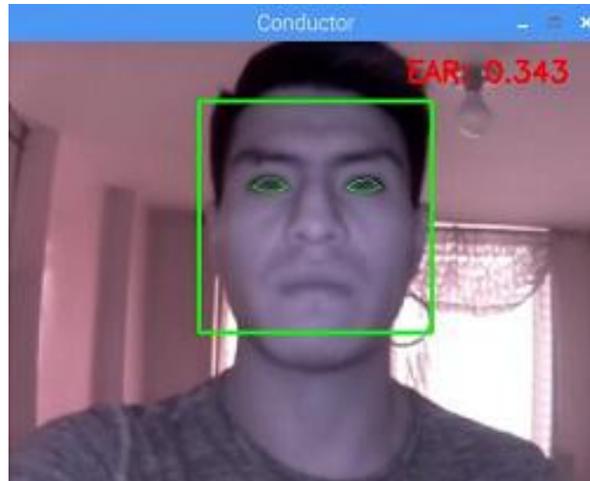


Figura 43. Ensayo de código fase dos.

#### 5.4.3 Código de reconocimiento de estado de somnolencia

Luego del reconocimiento del rostro y ojos, el siguiente paso fue calcular el promedio de apertura de los ojos mediante el diámetro del ojo desde la parte superior hasta la parte inferior, este proceso devuelve un número el cual, si es menor a un rango en el que el ojo esté cerrado, automáticamente se ejecuta una alerta; a continuación, en la figura 44 se muestra el resultado de dicha prueba la cual fue satisfactoria.

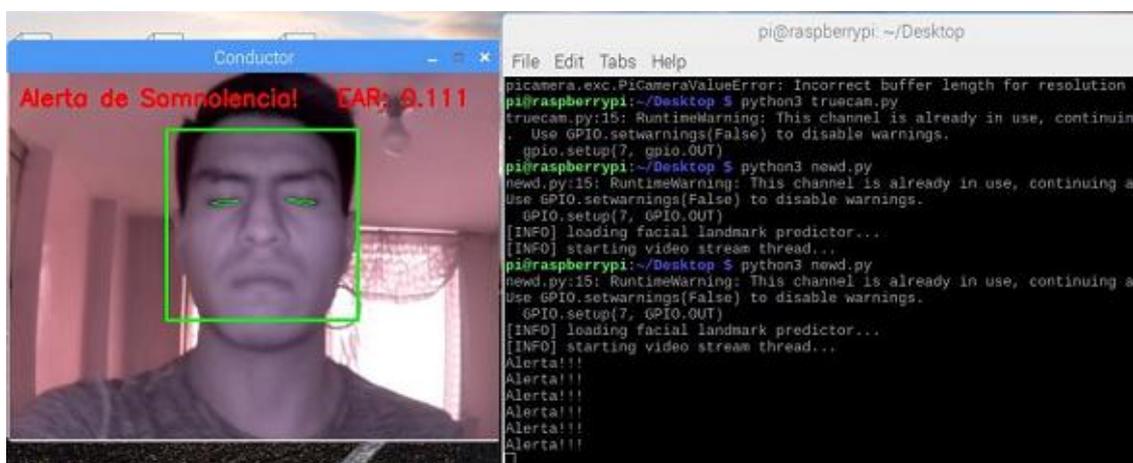


Figura 44. Ensayo de código fase tres.

#### 5.4.4 Código de reconocimiento de los ojos mediante el uso de lentes

Luego del reconocimiento del rostro, ojos y somnolencia, el siguiente paso es verificar el funcionamiento del código mediante la utilización lentes ya que algunos conductores los utilizan para manejar sus vehículos, por lo que el sistema debe ser capaz de reconocer y poder detectar el estado de somnolencia; a continuación, en la figura 45 se muestra el resultado de la fase 4 el cual fue satisfactorio.



Figura 45. Ensayo de código fase cuatro.

#### 5.4.5 Código de reconocimiento de somnolencia usando lentes.

Luego del reconocimiento de los ojos con la utilizando lentes, el siguiente paso es verificar el funcionamiento del código al captar si el conductor presenta somnolencia; a continuación, en la figura 46 se muestra que el sistema trabaja de manera eficiente, logrando de esta manera saber si un conductor presenta o no estado de somnolencia.

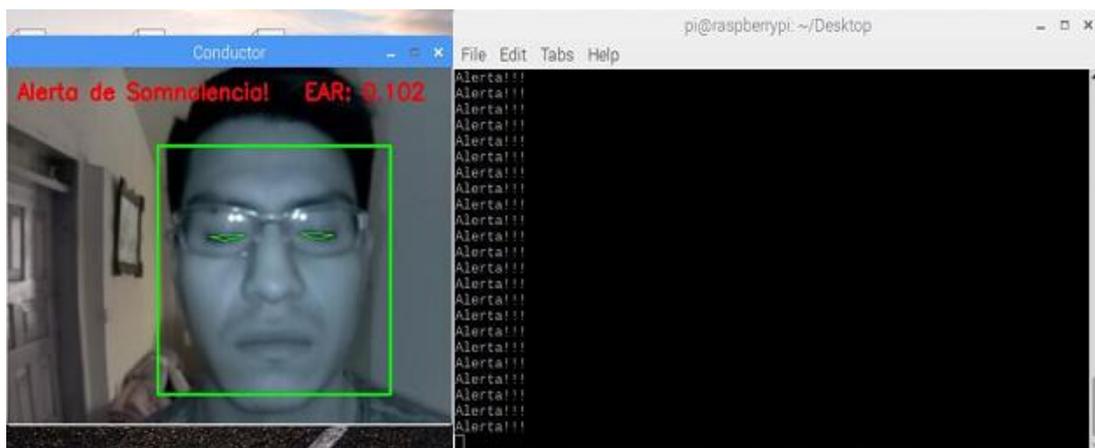


Figura 46. Ensayo de código fase cinco.

## 6 Resultados y Discusión

Las pruebas de funcionamiento fueron realizadas con el sistema de monitoreo facial con diferentes conductores los cuales son choferes profesionales experimentados, se realizó el control del dispositivo y su funcionamiento tanto en la noche como en el día; también se realizaron pruebas de funcionamiento en varios horarios para obtener resultados reales en el monitoreo sobre el estado de los conductores de vehículos.

Las pruebas realizadas para la validación del funcionamiento del dispositivo se las hizo en diferentes fases según el diagrama de bloques, cada etapa verifica los resultados que se presentan en el algoritmo que verifica el funcionamiento óptimo del sistema; para la realización de las pruebas se tomó en cuenta varios conductores los cuales son choferes profesionales y ellos fueron los que validaron el funcionamiento del dispositivo anti somnolencia.

### 6.1 Pruebas del Sistema

Luego de finalizar la etapa de diseño e implementación se realiza la etapa de las pruebas, donde se valida la funcionalidad del dispositivo, las pruebas sirven para verificar si el sistema realiza el monitoreo fisiológico hacia los conductores los cuales presentan diferentes rasgos físicos.

Las pruebas realizadas sirvieron para comprobar el funcionamiento del dispositivo y el sistema en tiempo real; es decir, se hicieron varias pruebas en diferentes horarios en las carreteras mientras el automóvil estaba en movimiento lo cual fue satisfactorio para verificar el funcionamiento del dispositivo, ya que al presentar fallas se las corrigió y se volvió a probar el sistema para modificar las fallas de código y obtener como resultado un excelente rendimiento.

Las pruebas fueron realizadas tomando como práctica el uso de varios vehículos con el fin de ubicar el dispositivo de manera perfecta en los diferentes ambientes dentro de cada automóvil, y también se lo hizo en los diferentes horarios como es la mañana, tarde y noche para conocer los efectos de la luz y de la oscuridad; finalmente se comprobó el funcionamiento del sistema para el caso de que el conductor use lentes y se lo hizo en diferentes horas del día y los resultados fueron favorables para el uso de lentes.

Para la realización de las pruebas se siguió un plan de actividades, en donde se analizan los conductores con los cuales se realizaron las pruebas, el área en donde se hizo las pruebas de funcionalidad fue en los vehículos de los usuarios y el tiempo de aplicación de cada prueba dependió del funcionamiento del sistema y su adaptabilidad. En la tabla 12 se muestra el resumen de las pruebas realizadas en cada fase de funcionamiento.

Tabla 11.

*Fase de pruebas de funcionamiento del dispositivo*

<b>Plan Actividades Pruebas de Funcionamiento</b>					
	<b>Fase 1</b>	<b>Fase 2</b>	<b>Fase 3</b>	<b>Fase 4</b>	<b>Fase 5</b>
<b>Comprobación</b>	Eficacia de imagen	Detección de rostro	Detección de los ojos	Comprobar apertura del ojo	Alarma auditiva
<b>Individuos</b>	Conductores	Conductores	Conductores	Conductores	Conductores

Área	Vehículo	Vehículo	Vehículo	Vehículo	Vehículo
<b>Efectos</b>	Imagen óptima	Detección facial de rostro efectiva	Detección efectiva de los ojos	Diámetro de apertura del ojo	Alerta activada
<b>Tiempo</b>	10 días	15 días	15 días	15 días	10 días

Según las actividades propuestas se evidenció que en la primera fase existieron problemas por exceso de luz sobre todo en el día, por lo que fue necesario corregir este error mediante el paso de la imagen a escala de grises, posteriormente a ello la imagen se presentó estable y el porcentaje de error fue del 5%; en la fase 2 puesto que la cámara tiene instalado infrarrojos, el rostro es reconocido de todas las formas posibles menos cuando se le da a la cámara un lado del rostro por lo que el promedio de error fue del 8%; en la fase 3 y por la razón anterior los ojos son reconocidos perfectamente con o sin utilizar lentes, y también se realizaron las pruebas durante el día y la noche por lo que el promedio de error es del 10% ya que al voltear el rostro en ocasiones no se reconocen los ojos; en la fase 4 se realizó el promedio de apertura de los ojos por lo que cuando están abiertos existe un valor de diámetro promedio de 0,45 centímetros pero al cerrarlos este valor cae a los 0,28 centímetros, entonces si el diámetro promedio es inferior a 0,3 el sistema activa la alarma, por lo que el error promedio es del 10% según la luz del día o el enfoque de las luces de los vehículos cuando se utiliza lentes; en la fase 6 se verificó la alerta y activación de la alarma y se comprobó que funciona de manera correcta por lo que el error promedio en esta etapa es del 5%.

### 6.1.1 Pruebas realizadas en la fase 1

Las pruebas realizadas en la fase 1 se las hizo en el día, tarde y noche y los resultados fueron 95% positivos, el promedio de fallas fue por causa de exceso de luz en el vehículo especialmente en horas de la mañana, a continuación, la

tabla 13 se indican los resultados obtenidos en la fase 1 respecto a las pruebas de funcionamiento.

Tabla 12.

*Análisis de datos fase 1*

	Fase 1			Total
	Comprobación de funcionamiento 1	Comprobación de funcionamiento 2	Comprobación de funcionamiento 3	
Programa	Matutino	Vespertino	Noche	
Pruebas	20	20	20	60
Exactas	18	20	19	57
Inexactas	2	0	1	3
Exitosas	90%	100%	95%	95%
Fallidas	10%	0%	5%	5%

### 6.1.2 Pruebas realizadas en la fase 2

Las pruebas realizadas en la fase 2 se basaron en la detección del rostro del conductor, al igual que en la anterior fase se las realizó en la mañana, tarde y noche y los resultados fueron 92% positivos, el promedio de fallas fue por causa del movimiento que realizaron los conductores al manejar y voltear el rostro para verificar que no haya vehículos al cambiar de carril o ingresar a una vía en particular, a continuación, la tabla 14 se indican los resultados obtenidos de la fase 2 respecto a las pruebas de funcionamiento.

Tabla 13.

*Análisis de datos fase 2*

	Fase 2			Total
	Comprobación de funcionamiento 1	Comprobación de funcionamiento 2	Comprobación de funcionamiento 3	
Programa	Matutino	Vespertino	Noche	
Pruebas	25	25	25	75
Exactas	23	24	22	69
Inexactas	2	1	3	6

Exitosas	92%	96%	88%	92%
Fallidas	8%	4%	12%	8%

### 6.1.3 Pruebas realizadas en la fase 3

Las pruebas realizadas en la fase 3 se basaron en la detección del área de los ojos del conductor, al igual que en las anteriores fases se las realizó en la mañana, tarde y noche y los resultados fueron 95% positivos, reflejando que el número de falsos positivos que presenta el sistema es mínimo, esto quiere decir que la detección facial del sistema tiene un alto grado de efectividad., a continuación, la tabla 15 se indican los resultados obtenidos de la fase 3 respecto a las pruebas de funcionamiento.

Tabla 15.

#### *Análisis de datos fase 3*

	Fase 3			Total
	Comprobación de funcionamiento 1	Comprobación de funcionamiento 2	Comprobación de funcionamiento 3	
	Matutino	Vespertino	Noche	
Programa Pruebas	20	20	20	60
Exactas	17	19	18	57
Inexactas	3	1	2	3
Exitosas	85%	95%	90%	95%
Fallidas	15%	5%	10%	5%

### 6.1.4 Pruebas realizadas en la fase 4

Las pruebas realizadas en la fase 4 al igual que en las anteriores fases se las realizó en la mañana, tarde y noche y los resultados fueron 90% positivos, el promedio de fallas fue por causa del exceso de luz para el caso del día y de la luz directa de los vehículos para el caso de la noche y cuando el conductor utiliza lentes, a continuación, la tabla 16 se indican los resultados obtenidos de la fase 4.

Tabla 16.

*Análisis de datos fase 4*

	Fase 4			Total
	Comprobación de funcionamiento 1	Comprobación de funcionamiento 2	Comprobación de funcionamiento 3	
	Matutino	Vespertino	Noche	
Programa	Matutino	Vespertino	Noche	
Pruebas	20	20	20	60
Exactas	18	19	17	56
Inexactas	2	1	3	4
Exitosas	90%	95%	85%	95%
Fallidas	10%	5%	15%	5%

Puesto que la fase 4 se la realizó con varios conductores en distintos horarios, fue necesario realizar un análisis semanal para verificar los resultados obtenidos; a continuación, en la tabla 16 se muestran las pruebas de funcionamiento realizadas en el transcurso de la semana 1.

Se debe aclarar que, mediante las fallas presentadas durante las pruebas de funcionamiento del sistema en tiempo real, se realizaron mejoras en el código de programación para corregir cada una de las fallas lo cual permitió que en la realización de posteriores pruebas se mejore el sistema y tenga un menor porcentaje de fallas y de esta manera el sistema funcione de forma correcta y no presente fallas a la hora de que el usuario conduzca; a continuación, la tabla 17 se presentan los datos de la prueba realizada en la semana 1.

Tabla 17.

*Análisis de datos fase 4 durante la semana 1*

Fase 4. Sistema anti somnolencia	
<b>Semana 1</b>	Sitio: Vehículos particulares circulando por las calles de la ciudad
	Individuos: Conductores de vehículos particulares
	Horario de prueba de funcionamiento: Matutino, vespertino y noche

---

Duración de la prueba de funcionamiento: 15 minutos

---

A continuación, en la figura 47 se muestran las fallas que presentó el sistema en la semana 1 de pruebas realizadas en vehículos particulares.



*Figura 47.* Análisis del dispositivo en la semana 1.

La semana 2 de pruebas de funcionamiento se la realizó con conductores de servicio público dentro de la ciudad y al igual que en la semana 1 se realizaron las pruebas en la mañana, tarde y noche; a continuación, la tabla 18 se presentan los datos de la prueba realizada en la semana 2.

*Tabla 14.*

*Análisis de datos fase 4 durante la semana 2*

<b>Fase 4. Sistema anti somnolencia</b>	
<b>Semana 2</b>	Sitio: Vehículos de servicio público circulando por las calles de la ciudad
	Individuos: Conductores de vehículos de servicio público
	Horario de prueba de funcionamiento: Matutino, vespertino y noche
	Duración de prueba de funcionamiento: 20 minutos

A continuación, la figura 48 se muestran las fallas que presentó el sistema en la semana 2 de pruebas realizadas en vehículos de servicio público.



*Figura 48.* Análisis del dispositivo en la semana 2.

La semana 3 de pruebas se la realizó con conductores de vehículos de carga y buses dentro de la ciudad y al igual que en la semana 1 se realizaron las pruebas en la mañana, tarde y noche; a continuación, la tabla 19 se presentan los datos de la prueba realizada.

Tabla 19.

*Análisis de datos de la fase 4 durante la semana 3*

<b>Fase 4. Sistema anti somnolencia</b>	
<b>Semana 3</b>	Sitio: Vehículos de carga y buses circulando por las calles de la ciudad
	Individuos: Conductores de vehículos de carga y buses
	Horario de prueba de funcionamiento: Matutino, vespertino y noche
	Duración de prueba de funcionamiento: de 20 minutos de

A continuación, la figura 49 se muestran las fallas que presentó el sistema en la semana 3 de pruebas realizadas en vehículos de carga y buses.



Figura 49. Análisis del dispositivo en la semana 3.

Finalmente, la figura 50 se muestra el gráfico de efectividad del funcionamiento del dispositivo en las pruebas realizadas en la fase 4 las cuales se hicieron en varios vehículos con diferentes conductores y en distintos horarios del día, por lo que los resultados son diferentes y el porcentaje de error de maximiza en vehículos pesados como buses ya que su luminosidad es mayor.

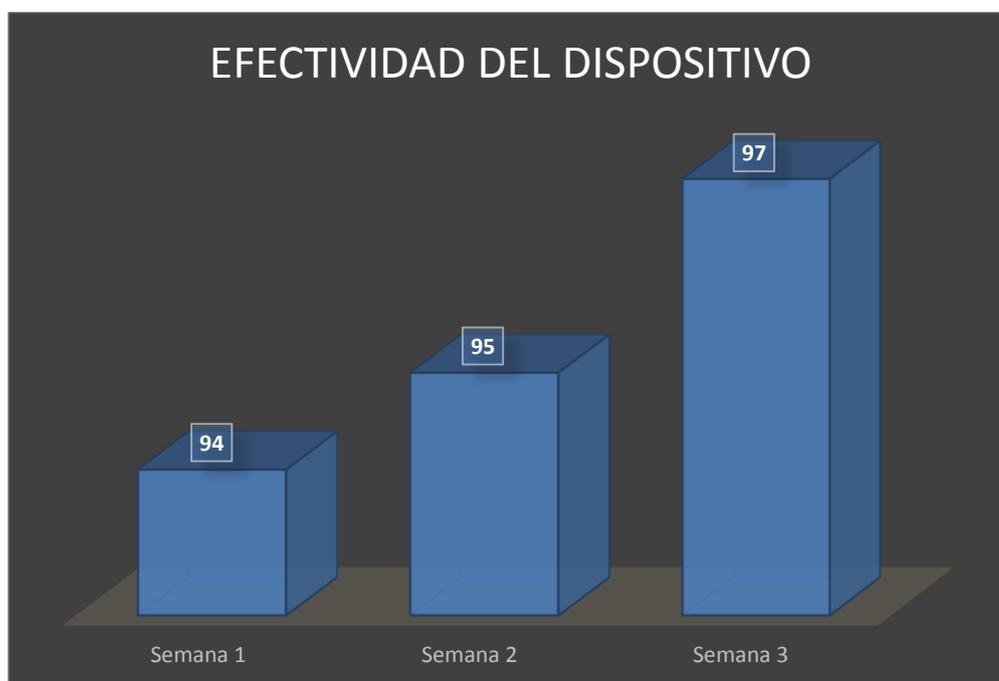


Figura 50. Porcentaje de efectividad del dispositivo.

### 6.1.5 Pruebas realizadas en la fase 5

Las pruebas realizadas en la etapa 5 al igual que en las anteriores fases se las realizó en la mañana, tarde y noche y los resultados fueron 95% positivos, el promedio de fallas fue por causa del exceso de luz para el caso del día y de la luz directa de los vehículos para el caso de la noche y cuando el conductor utiliza lentes por lo que la alarma del dispositivo no se activó debidamente, a continuación, en la tabla 19 se indican los resultados obtenidos en la fase 5 respecto a las pruebas de funcionamiento.

Tabla 20.

#### *Análisis de datos fase 5*

<b>Fase 5</b>				
	<b>Comprobación de funcionamiento 1</b>	<b>Comprobación de funcionamiento 2</b>	<b>Comprobación de funcionamiento 3</b>	<b>Total</b>
<b>Horario</b>	<b>Matutino</b>	<b>Vespertino</b>	<b>Noche</b>	
Pruebas	20	20	20	60
Exactas	19	19	19	57
Inexactas	1	1	1	3
Exitosas	95%	95%	95%	95%
Fallidas	5%	5%	5%	5%

### 6.2. Análisis de los Resultados.

Luego de la realización de las pruebas y teniendo en cuenta los resultados presentes en la fase 4 se puede decir que el sistema funcionó de forma efectiva con ciertos momentos de fallas por cuestión del exceso de luminosidad o de la luz directa de otros vehículos en la noche cuando el conductor usa lentes, pues el dispositivo confunde la luz reflejada en los lentes con un ojo abierto, entonces en estas circunstancias presenta un fallo, pero en general el dispositivo se comportó de forma eficiente en la práctica.

En las pruebas realizadas también se pudo observar que tras conducir durante un lapso largo de tiempo, la mayoría de conductores presentaron mayor estado

de somnolencia durante las horas de la noche, ya que el sistema advirtió de este estado mediante la activación de la alarma, permitiendo así que el conductor se alertara y abriera los ojos de forma inmediata; de esta manera se pudo evidenciar que el dispositivo anti somnolencia funciona correctamente y alerta al conductor en caso de somnolencia; se puede decir que en casos extremos no va a evitar al 100% los accidentes de tránsito pero si ayudará a los conductores a estar en constante alerta y así disminuir el porcentaje de accidentes de tránsito ocurridos por el factor de somnolencia de los conductores.

El dispositivo debe instalarse juntamente con un ventilador ya que cuando los conductores deben viajar a largas distancias el sistema debe estar alerta siempre y sin un ventilador calentará demasiado y es posible que se sobrecaliente y se apague, lo que ocasionaría que el sistema falle, entonces mediante la ayuda de un ventilador este problema se evitará por completo y el sistema estará siempre alerta.

Al verificar los resultados de las pruebas se obtiene un aproximado al 10% de fallos lo cual es conveniente para corregir los errores iniciales del sistema; luego de corregir cada error el programa tubo un porcentaje mayor de aciertos y el porcentaje de errores bajó al 5%; este porcentaje de error se dio finalmente puesto que el sistema para el caso en el que el conductor utilice lentes en la noche, cuando un vehículo viene de frente y con luz alta, los lentes reflejan esta luz y el dispositivo la asimila como un ojo abierto; otra de las causas se presenta en horas de la mañana cuando sale el sol y llega de frente al rostro, al usar lentes de igual forma refleja la luz y el sistema lo reconoce como ojo abierto.

Este tipo de errores es común en el uso de cámaras de video ya que cuando se refleja la luz en el lente genera un brillo que ni siquiera con los propios ojos se puede verificar si el conductor tiene los ojos abiertos o cerrados.

## 7 Conclusiones y Recomendaciones

### 7.1 Conclusiones

El código de programación de Python interactúa directamente con la librería *OpenCV* la cual es compatible con la plataforma y otras librerías lo que permite realizar la clasificación de los puntos en el rostro, el código al igual interactúa con los archivos de clasificación de estados fisiológicos, lo cual permite un buen funcionamiento del sistema al reconocer el rostro y ojos del conductor.

Se evidencio que las principales causas por las que un conductor pierde la concentración y llega a caer en un estado de somnolencia es debido al uso de fármacos, el uso de estupefacientes, un largo horario de trabajo y además la falta de sueño provocan el estado de somnolencia.

Mediante el uso de los clasificadores *Haar* se puede realizar la detección de los rasgos faciales que presentan los conductores ya que estos rasgos varían según la luz a la que el dispositivo esté sometido y el estado fisiológico de cada persona.

El algoritmo de Viola & Jones presenta un gran beneficio para la programación de código de visión artificial, ya que se puede clasificar los rasgos fisiológicos de forma sencilla y el resultado es satisfactorio considerando para que el sistema funciona de forma coherente.

En el proceso de la conexión de la unidad de procesamiento y sus elementos se concluye que Raspberry es un minicomputador de características capaces de abarcar varios procesos a la vez y su velocidad es óptima a la hora de ejecutar programas de gran consumo de memoria, por lo que es un dispositivo fácil de interactuar y agradable al momento de trabajar.

El sistema operativo Raspbian basado en Linux Debian ofrece un entorno gráfico para su mejor administración y configuración, lo cual lo hace fácil de utilizar y eficiente para trabajar.

Para la configuración de todos los programas en el sistema operativo Raspbian, es necesario realizarlo desde la consola de comandos para un mejor procesamiento de datos ya que en el entorno gráfico es posible que se presenten problemas debido a que ocupa gran espacio en memoria RAM.

La cámara como parte del dispositivo se encarga de procesar imágenes y mediante el algoritmo utilizado se verifica si los ojos del conductor permanecen abiertos o cerrados para posterior a ello emitir una alerta de aviso.

El funcionamiento del sistema fue comprobado en tiempo real en conductores de vehículos que utilizan lentes, así como también en conductores que no los utilizan, donde el sistema logro detectar el estado de somnolencia de manera eficiente en ambos casos de conductores mencionados anteriormente.

Mediante la configuración de los tonos de alerta que tiene el sistema, el conductor interactúa con el sistema dándole la oportunidad de cambiar el tono de alarma a su gusto en tres tonos diferentes según la preferencia de cada usuario.

Mediante las pruebas realizadas se llegó a la conclusión de que el hardware y *software* funcionan de forma eficaz en la parte interna del vehículo esto ya sea en el día como en la noche, y por parte del usuario al momento de utilizar lentes, o a su vez que un vehículo se dirija directamente con luz alta hacia el vehículo que cuente con el sistema.

Las pruebas de funcionamiento fueron satisfactorias y constataron el buen rendimiento del sistema, mediante los aciertos positivos y fallos por parte del sistema se logró mejorar el código de programación y así llegar a un sistema sin fallas.

El sistema anti somnolencia, verifica constantemente el rostro del conductor y los rasgos fisiológicos, especialmente se basa en los ojos de cada conductor para de esta manera verificar si estos se encuentran abiertos o cerrados; para

esto se debe diferenciar entre el parpadeo y el cierre completo de los ojos para saber cuándo es necesario activar la alerta auditiva.

La implementación de un sistema de alarma anti somnolencia en vehículos es positivo y necesario puesto que se evitará un gran porcentaje de accidentes de tránsito provocados por este factor.

## **7.2 Recomendaciones**

Se recomienda ubicar el sistema de manera correcta de tal forma que no interfiera con la visibilidad del conductor, colocando la cámara en un punto específico donde pueda captar el rostro del conductor permitiendo de esta manera el monitoreo de los rasgos fisiológicos.

Se debe utilizar un convertidor de voltaje para la instalación eléctrica del dispositivo con el vehículo ya que el automotor trabaja con 12 voltios y el sistema funciona con 5 voltios, caso contrario el dispositivo podrían ser afectados por un cambio brusco de voltaje.

Para el funcionamiento óptimo de la presente aplicación se recomienda usar una placa Raspberry Pi 3 modelo b + con su respectivo sistema operativo Raspbian, debido a que este tipo de dispositivo es compatible con el sistema antes mencionado y cuenta con las características técnicas adecuadas para que el dispositivo funcione como un pequeño computador.

Para una mejor captación de imagen en la noche, se recomienda instalar infrarrojos a la cámara de alto rendimiento para que el reconocimiento sea óptimo y no existan fallas durante la noche mientras el conductor está manejando el vehículo.

Se recomienda al usuario configurar el tono de alarma a su gusto mediante el pulsador que presenta el dispositivo ya que existen tres tonos de alarma por lo cual el usuario podrá utilizar el que guste.

Durante la noche para los conductores que utilicen lentes, se debe tener en cuenta que la luz alta y directa de otros vehículos puede ocasionar que el sistema capte la luz reflejada como un ojo abierto por lo que el usuario debe estar alerta en este proceso.

## REFERENCIAS

- AMICUS. (2015). *Facial Recognition*. Recuperado el 10 de Mayo de 2019, de <https://www.amicusint.org/articles/2015/10/13/face-recognition>
- ANT. (2017). Agencia Nacional De Transito. Recuperado el 2 de Abril de 2019, de <https://www.ant.gob.ec/>
- Autocasión. (2015). Los sistemas de asistencia a la conducción más usados según Bosch. Recuperado el 6 de Mayo de 2019, de <https://www.autocasion.com/actualidad/noticias/los-sistemas-de-asistencia-a-la-conduccion-mas-usados-segun-bosch>
- Barba, M. (2019). El potencial de la tecnología de reconocimiento facial. Recuperado el 18 de Abril de 2019, de <https://blogthinkbig.com/potencial-reconocimiento-facial>
- Contaval. (2016). Qué es la visión artificial y para qué sirve. Recuperado el 24 de Abril de 2019, de <https://www.contaval.es/que-es-la-vision-artificial-y-para-que-sirve/>
- DGT. (2016). Dirección General de Tránsito. Recuperado el 30 de Marzo de 2019, de <http://www.dgt.es/es/>
- GPSTEC. (2015). Sensor de Fatiga del Conductor. Recuperado el 20 de Abril de 2019, de <http://gpstec.cl/sensor-de-fatiga/>
- Gras, M., Pedra, M., & Font-Mayolas, S. (2008). La distracción de los conductores: un riesgo no percibido. Recuperado el 10 de Mayo de 2019, de [http://imagenes.racc.es/pub/ficheros/adjuntos/adjuntos\\_esp\\_distraccions\\_web\\_jzq\\_62fb66d0.pdf](http://imagenes.racc.es/pub/ficheros/adjuntos/adjuntos_esp_distraccions_web_jzq_62fb66d0.pdf)
- Ibermática. (2011). Sistema de seguimiento de la mirada. Recuperado el 8 de Mayo de 2019, de <http://rtdibermatica.com/?p=648>

- LogiNews. (2015). Volkswagen integra detector de fatiga Bosch en sus vehículos. Recuperado el 20 de Mayo de 2019, de <https://noticiaslogisticaytransporte.com/empresas/06/11/2015/volkswagen-integra-detector-de-fatiga-bosch-en-sus-vehiculos/61187.html>
- MAPFRE. (2017). Sueño y fatiga, y su influencia al volante. Recuperado el 8 de Junio de 2019, de [https://www.fundacionmapfre.org/fundacion/es\\_es/programas/seguridad-vial/movilidad-segura-salud/temas-conduccion-segura/enfermedades-neurologicas/sueno-fatiga.jsp](https://www.fundacionmapfre.org/fundacion/es_es/programas/seguridad-vial/movilidad-segura-salud/temas-conduccion-segura/enfermedades-neurologicas/sueno-fatiga.jsp)
- Montes, S., Ledesma, R., & Poó, F. (2014). Estudio y prevención de la distracción e inatención en la conducción. Recuperado el 28 de Marzo de 2019, de <http://www.scielo.org.co/pdf/apl/v32n1/v32n1a09.pdf>
- OMS. (2014). Organización Mundial de la Salud . Recuperado el 12 de Mayo de 2019, de <https://www.who.int/es/>
- OpenCV. (2019). *OpenCV*. Recuperado el 30 de Mayo de 2019, de <https://opencv.org/>
- Ortiz, F., & Fernández, J. (2017). Modelo de análisis de requisitos orientados al desarrollo de recursos tiflo-tecnológicos. *DESARROLLO E INNOVACIÓN EN INGENIERÍA*, 149.
- PiFoundation. (s.f.). Raspberry Pi. Recuperado el 29 de Mayo de 2019, de <https://www.raspberrypi.org/>
- ProgramaciónExtrema. (2018). El algoritmo de Viola & Jones. Recuperado el 27 de Abril de 2019, de <http://programacionextrema.es/2018/02/27/algoritmo-viola-jones-diario-programador/>
- Python. (s.f.). PYTHON. Recuperado el 28 de Mayo de 2019, de <https://www.python.org/>

- Race. (2017). La fatiga en la conducción: consejos y recomendaciones. Recuperado el 5 de Abril de 2019, de <https://www.race.es/fatiga-conduccion-consejos-recomendaciones>
- Rastreator. (2018). Distracciones más comunes al volante. Recuperado el 20 de Mayo de 2019, de <https://www.rastreator.com/seguros-de-coche/articulos-destacados/las-distracciones-mas-comunes-al-volante.aspx>
- RicardoGeek. (2019). Detección De Caras Con OpenCV y Python. Recuperado el 15 de Junio de 2019, de <https://ricardogeek.com/deteccion-de-caras-con-opencv-y-python/>
- ROBOLOGS. (2014). Reconocimiento facial con OpenCV + webcam. Recuperado el 2 de Mayo de 2019, de <https://robologs.net/2014/05/26/reconocimiento-facial-con-opencv-webcam/>
- Rosales, M., Rosales, M., & Rojas, E. (2013). Accidentes de carretera y su relación con cansancio y somnolencia en conductores. *Revista Médica Herediana*, 48-59.
- STACKPOINTERS. (2018). *Object Detection Framework*. Recuperado el 4 de Mayo de 2019, de <http://stackpointers.com/python/log/>
- Troya, C. (2016). *LBP y ULBP – Local Binary Patterns y Uniform Local Binary Patterns*. Recuperado el 20 e Abril de 2019, de <https://cesartroyasherdek.wordpress.com/2016/02/26/deteccion-de-objetos-vi/>

## **ANEXOS**

## Anexo 1 Código Python

El siguiente código muestra las líneas más importantes escritas en el sistema con el fin de describir su funcionamiento y así verificar que el dispositivo funcione de forma correcta.

```
# Se deben importar las librerías principales para el reconocimiento facial.
```

```
import RPi.GPIO as GPIO
```

```
import numpy as np
```

```
import cv2
```

```
# A continuación se inicializa el puerto GPIO el cual es un puerto que tiene entradas y salidas de voltaje para trabajar con dispositivos electrónicos externos a la placa, se utiliza el puerto 7 como salida de voltaje para prender o apagar la alarma.
```

```
GPIO.setmode(GPIO.BOARD)
```

```
GPIO.setup(7, GPIO.OUT)
```

```
GPIO.output(7, False)
```

```
# Posteriormente se crean dos constantes las cuales sirven para ubicar el promedio mínimo de apertura de los ojos para considerar que el conductor esté despierto o dormido y también se considera el número mínimo de cuadros que el sistema debe considerar para verificar el estado del conductor.
```

```
Prom_ojo = 0.22
```

```
Cuadros_ojo = 8
```

# A continuación, se inicia una variable de tipo contador en cero, para que cuando el sistema considere que los ojos están cerrados, la variable aumente hasta llegar al número 16 y así se active la alarma.

```
Contador = 0
```

# Se inicializa el bucle while, el cual debe ejecutar el código interno de forma constante.

```
while True:
```

# Se transforma cada cuadro de imagen a formato de escala de grises para que el identificador Haar pueda reconocer el rostro y los ojos de mejor forma.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

# Inicia bucle For para la detección del rostro y ojos.

```
for (x, y, w, h) in rectas:
```

# Calcula el promedio de apertura de los ojos.

```
diametro = (ojo_izq + ojo_der) / 2.0
```

# A continuación, se grafica el contorno del ojo para mostrarlo en pantalla.

```
contornoojoizq = cv2.convexHull(ojo_izq)
```

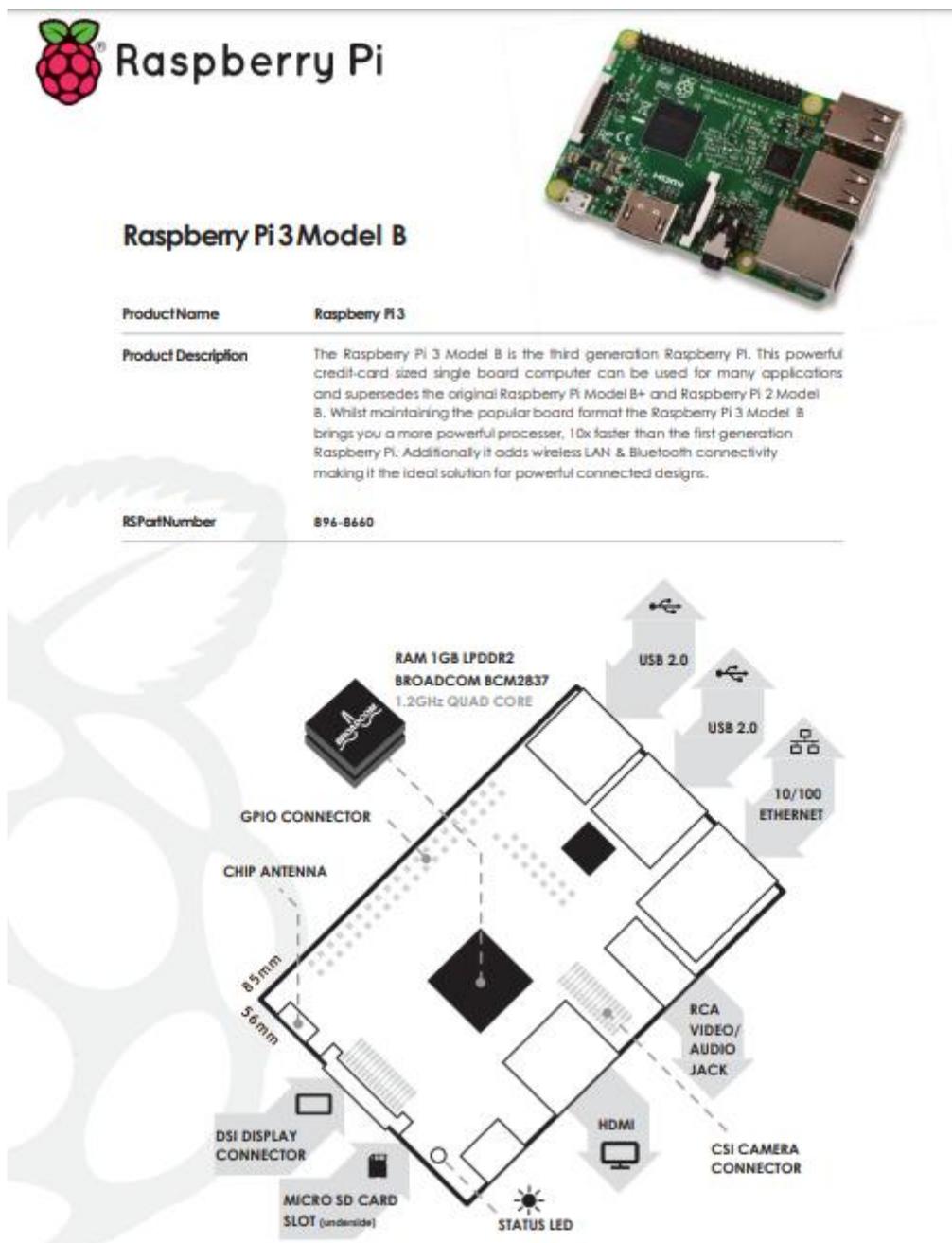
```
cv2.drawContours(cuadro, [contornoojoizq], -1, (0, 255, 0), 1)
```

# Se muestra en pantalla la imagen resultante con la detección del rostro encerrado en un rectángulo y el gráfico del contorno de los ojos

```
cv2.imshow("Cuadro", cuadro)
```

## Anexo 2 *Datasheet* elementos del sistema

- *Datasheet* referente a la placa Raspberry Pi 3 modelo b + utilizada como unidad de procesamiento en el sistema de detección de somnolencia



- *Datasheet* referente a la cámara Pi NoIR V2 Raspberry utilizada en el sistema de detección de somnolencia



# Raspberry Pi

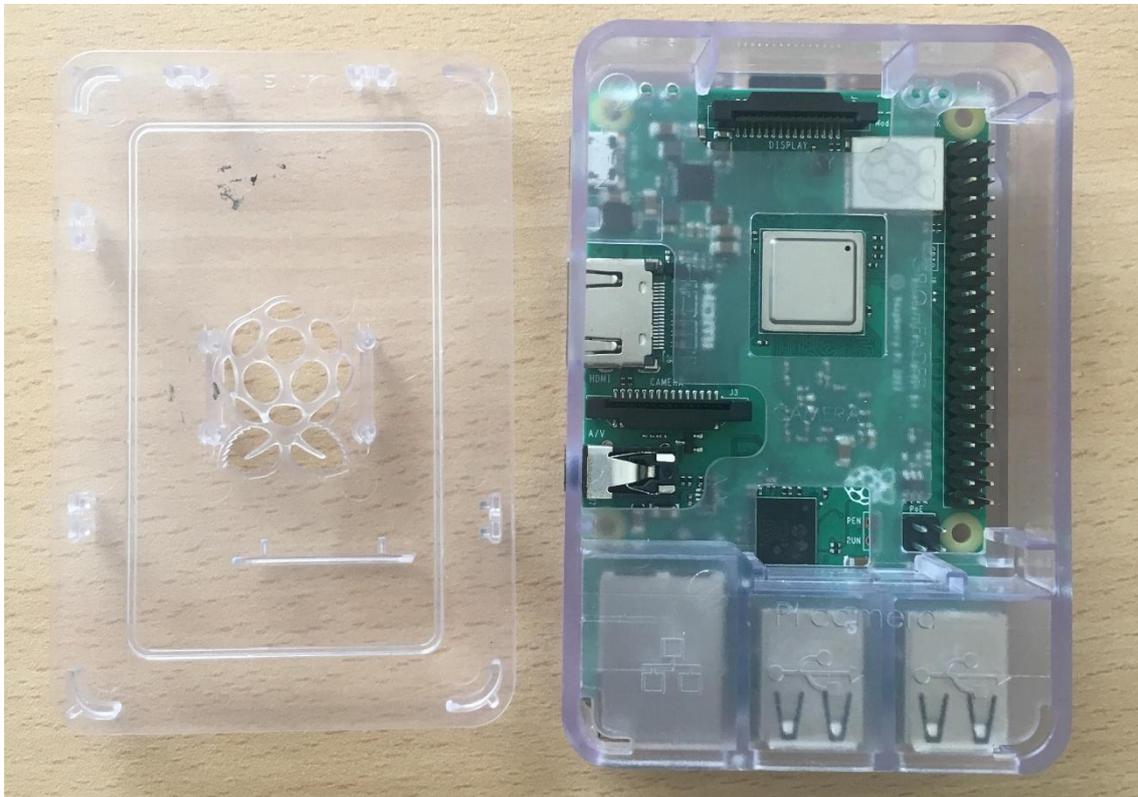


## Pi NoIR

<b>Product Name</b>	<b>Pi NoIR - Raspberry Pi Infrared Camera Module</b>
<b>Product Description</b>	Featuring the same 5 megapixel image sensor as the standard Raspberry Pi camera with the infrared cut-off filter removed to increase IR light sensitivity. The Pi NoIR is compatible with Raspberry Pi model A & model B and provides high definition, high sensitivity, low crosstalk and low noise image capture in an ultra small and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector which is capable of extremely high data rates and it exclusively carries pixel data to the BCM2835 processor.
<b>RS Part Number</b>	<b>790-2811</b>
<b>Specifications</b>	
<b>Image Sensor</b>	Omnivision 5647 CMOS image sensor in a fixed-focus module with IR blocking filter removed
<b>Resolution</b>	5-megapixel
<b>Still picture resolution</b>	2592 x 1944
<b>Max image transfer rate</b>	1080p: 30fps (encode and decode) 720p: 60fps
<b>Connection to Raspberry Pi</b>	15 Pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2)
<b>Image control functions</b>	Automatic exposure control Automatic white balance Automatic band filter Automatic 50/60 Hz luminance detection Automatic black level calibration
<b>Temp range</b>	Operating: -30° to 70° Stable image: 0° to 50°

### Anexo 03 Fotografías del sistema realizado

- Dispositivo 1 Raspberry Pi 3 modelo b +



- Dispositivo 2 Camera Pi NoIR V2 la cual tiene que ir acoplada a la unidad de procesamiento como es la Raspberry pi 3 b +



- Dispositivo 3 Buzzer o Zumbador el cual tiene que estar conectado en los puertos GPIO de la Raspberry Pi modelo b + (puerto 4 (+) – puerto 5(-))



- Dispositivo 4 Conversor de voltaje 12 voltios a 5 voltios conectado directamente a unidad de procesamiento Raspberry Pi 3 modelo b +



- Dispositivo 5 Pulsador el cual tiene que estar conectado en los puertos GPIO de la Raspberry Pi modelo b + (puerto 3 – puerto 6)



#### Anexo 4 Dispositivos conectados al sistema embebido

- Sistema embebido mediante la conexión de todos los elementos necesarios para la detección de estado de somnolencia



- Sistema embebido implementado en el vehículo mediante la conexión de todos los elementos necesarios para la detección de estado de somnolencia





### Anexo 5 Pruebas y validación de funcionamiento

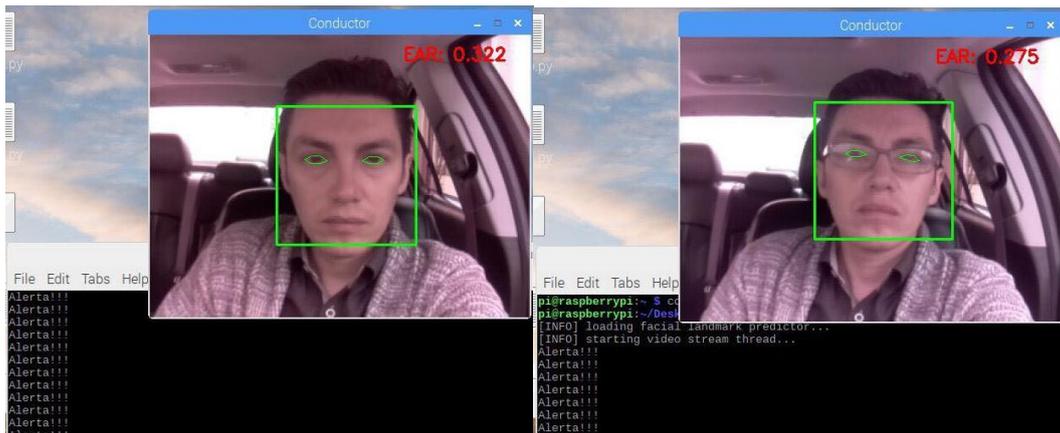
- Vehículos en los cuales se efectuó las pruebas y validación de funcionamiento del sistema embebido





- Resultados obtenidos de las pruebas de funcionamiento

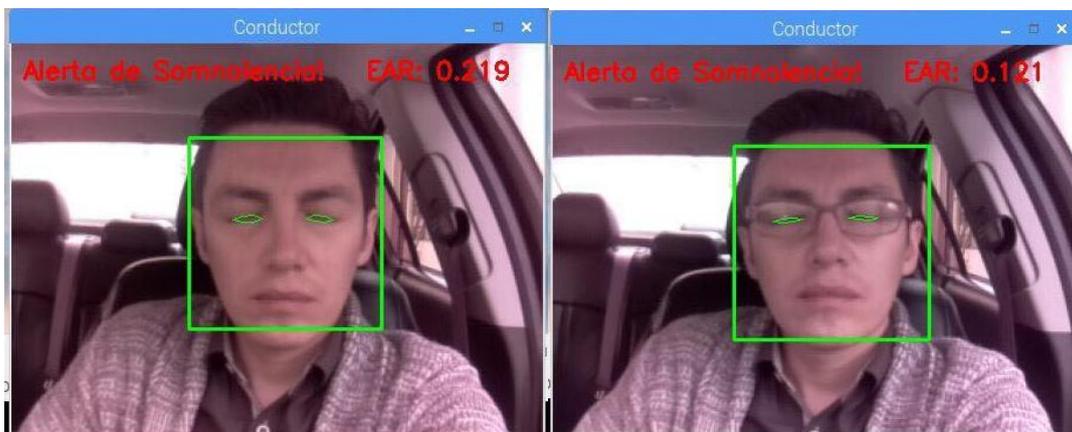
En la fase 1 y 2, las pruebas de funcionamiento se las realizo con conductores de vehículos profesionales el resultado de estas pruebas fue de un (86%), esto se debió a que la luz dentro del habitáculo influye en el procesamiento de las imágenes lo que para la fase 3 y 4 este factor fue modificado.



Fase 1

Fase 2

En la fase 3 y 4 las pruebas de funcionamiento se las realizo con conductores de vehículos profesionales donde el resultado de estas pruebas fue de un (97%), siendo el sistema sumamente preciso a la hora de detectar estado de somnolencia



Fase 3

Fase 4

