



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

**“IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE**  
**ALERTA PARA CONDUCTORES DISTRAÍDOS Y**  
**SOMNOLIENTOS DE VEHÍCULOS BASADO EN VISIÓN**  
**ARTIFICIAL”**

**TRABAJO DE TITULACIÓN**  
**TIPO: DISPOSITIVO TECNOLÓGICO**

Presentado para optar al grado académico de:

**INGENIERA EN ELECTRÓNICA, CONTROL Y REDES**  
**INDUSTRIALES**

**AUTORA: BERTHA MARISOL ARÉVALO TENELEMA**  
**DIRECTOR: ING. PABLO EDUARDO LOZADA YÁNEZ**

**Riobamba – Ecuador**

**2019**

**©2018, Bertha Marisol Arévalo Tenelema**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca los Derechos del Autor.

Bertha Arévalo

Yo, Bertha Marisol Arévalo Tenelema, declaro que el presente trabajo de titulación es de mi autoría y el resultado del mismo son auténtico. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor (a) asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

**Bertha Marisol Arévalo Tenelema**

**020233241-7**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

El Tribunal del Trabajo de Titulación certifica que: El dispositivo tecnológico “IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA PARA CONDUCTORES DISTRAÍDOS Y SOMNOLIENTOS DE VEHÍCULOS BASADO EN VISIÓN ARTIFICIAL”, de responsabilidad de la Srta. Bertha Marisol Arévalo Tenelema, ha sido minuciosamente revisado por los miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

**NOMBRE**

**FIRMA**

**FECHA**

Ing. Washington Luna

**PRESIDENTE DE TRIBUNAL**

\_\_\_\_\_

\_\_\_\_\_

Ing. Pablo Lozada

**DIRECTOR DEL TRABAJO  
DE TITULACIÓN**

\_\_\_\_\_

\_\_\_\_\_

Ing. José Luis Tinajero.

**MIEMBRO DEL TRIBUNAL**

\_\_\_\_\_

\_\_\_\_\_

## **DEDICATORIA**

Este logro se lo dedico primeramente a Dios por bendecirme y guiarme en cada momento de mi vida, además por brindarme su sabiduría para seguir adelante en cada uno de los pasos dados en el transcurso de mis estudios, a mis padres por su amor y su apoyo incondicional sin ellos no habría logrado mi meta, a mis queridas hermanas y mi esposo por sus consejos de nunca rendirme y seguir adelante, a mi querido hijo mi inspiración y mi fuerza de continuar en la lucha para conseguir este sueño y a todos aquellos ellos por su cariño incondicional se los dedico con amor.

**Bertha Arévalo**

## **AGRADECIMIENTO**

A Dios por permitirme dar un paso muy importante en mi vida, a mis padres por todo el esfuerzo realizado para poder permitirme estudiar y alcanzar mis sueños, a mis hermanas, mi esposo y amigos por su constante apoyo en todos estos años de estudio, al Ing. Pablo Lozada por el apoyo brindado para el desarrollo del presente trabajo de titulación y a todos quienes estuvieron ahí incondicionalmente se los agradezco.

**Bertha Arévalo**

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE GRÁFICOS.....	xi
ÍNDICE DE ANEXOS.....	xii
ÍNDICE DE ABREVIATURAS.....	xiii
RESUMEN .....	xiv
SUMMARY .....	xv
INTRODUCCIÓN.....	1
<b>CAPÍTULO I</b>	
<b>1. MARCO TEÓRICO.....</b>	<b>4</b>
<b>1.1 Somnolencia .....</b>	<b>4</b>
<i>1.1.1 Relación de la somnolencia con los accidentes de tránsito.....</i>	<i>5</i>
<i>1.1.2 Sueño.....</i>	<i>5</i>
<i>1.1.3 Fatiga .....</i>	<i>6</i>
<i>1.1.4 Alteraciones de la somnolencia en la conducción.....</i>	<i>8</i>
<i>1.1.5 Causas de la aparición de somnolencia .....</i>	<i>8</i>
<b>1.2 Distracción.....</b>	<b>9</b>
<i>1.2.1 Nivel de atención en la conducción .....</i>	<i>10</i>
<i>1.2.2 Tipos de la distracción en la conducción.....</i>	<i>11</i>
<b>1.3 Visión artificial.....</b>	<b>12</b>
<i>1.3.1 Etapas de un sistema de visión artificial.....</i>	<i>12</i>
<i>1.3.1.1 Adquisición de imágenes .....</i>	<i>13</i>
<i>1.3.1.2 Pre procesamiento.....</i>	<i>15</i>
<i>1.3.1.3 Segmentación.....</i>	<i>17</i>
<i>1.3.1.4 Representación y descripción.....</i>	<i>17</i>
<i>1.3.1.5 Reconocimiento e interpretación.....</i>	<i>18</i>
<i>1.3.2 Iluminación.....</i>	<i>18</i>
<b>1.4 Patrones de reconocimiento para personas somnolientos y distraídas .....</b>	<b>18</b>
<i>1.4.1 Detector de caras: Algoritmo de Viola Jones.....</i>	<i>18</i>
<i>1.4.1.1 Integral de la imagen.....</i>	<i>19</i>
<i>1.4.1.2 Características de rectángulos – Características Haar.....</i>	<i>19</i>
<i>1.4.1.3 AdaBoost.....</i>	<i>20</i>
<i>1.4.1.4 Cascada de decisión .....</i>	<i>21</i>

<b>1.5</b>	<b>Hardware del prototipo.....</b>	<b>21</b>
<b>1.5.1</b>	<b>Raspberry Pi.....</b>	<b>22</b>
1.5.1.1	Terminales de entrada y salida.....	22
1.5.1.2	Definiciones básicas de las partes.....	23
<b>1.5.2</b>	<b>Dispositivo de Captura.....</b>	<b>23</b>
1.5.2.1	Cámara de Raspberry.....	24
1.5.2.2	Conexión de la cámara a la Raspberry.....	25
<b>1.6</b>	<b>Software libre.....</b>	<b>25</b>
<b>1.6.1</b>	<b>Python.....</b>	<b>26</b>
<b>1.6.2</b>	<b>Open CV, librería para el procesamiento digital de imágenes.....</b>	<b>26</b>
1.6.2.1	Estructura modular de Open CV.....	26
1.6.2.2	OpenCV-Python.....	27
1.6.2.3	Paquetes adicionales en OpenCV-Python.....	27
<b>1.6.3</b>	<b>PyCharm.....</b>	<b>28</b>
1.6.3.1	Características de PyCharm.....	28
 <b>CAPÍTULO II</b>		
<b>2.</b>	<b>MARCO METODOLÓGICO.....</b>	<b>29</b>
<b>2.1</b>	<b>Requerimientos del sistema.....</b>	<b>29</b>
<b>2.2</b>	<b>Concepción general del diseño de <i>hardware</i> del prototipo.....</b>	<b>29</b>
<b>2.3</b>	<b>Descripción de los elementos <i>hardware</i> del prototipo.....</b>	<b>30</b>
2.3.1	Selección de la tarjeta Raspberry Pi.....	30
2.3.2	Cámara Pi NoiR.....	31
2.3.3	Buzzer.....	32
2.3.4	Fuente de alimentación.....	33
2.3.5	Inversor de corriente.....	33
<b>2.4</b>	<b>Diagrama de conexiones del prototipo.....</b>	<b>34</b>
<b>2.5</b>	<b>Selección del <i>software</i> del dispositivo.....</b>	<b>35</b>
<b>2.6</b>	<b>Requerimiento del <i>software</i>.....</b>	<b>35</b>
2.6.1	Software para el procesamiento de imágenes.....	36
2.6.2	Análisis de OpenCV.....	36
<b>2.7</b>	<b>Entrenamiento del nuevo clasificador.....</b>	<b>36</b>
<b>2.8</b>	<b>Programación en el IDE de Python.....</b>	<b>37</b>
2.8.1	Adquisición de la imagen.....	38
2.8.2	Pre-procesamiento e identificación.....	39
2.8.3	Activación de alarma.....	40



### **CAPÍTULO III**

<b>3.</b>	<b>ANÁLISIS Y RESULTADOS.....</b>	<b>42</b>
<b>3.1</b>	<b>Ubicación del dispositivo en el vehículo.....</b>	<b>42</b>
<b>3.2</b>	<b>Metodología para las pruebas.....</b>	<b>43</b>
<b>3.2.1</b>	<i>Pruebas detección de somnolencia – Cierre de Ojos.....</i>	<i>43</i>
<b>3.2.2</b>	<i>Pruebas detección de asentamiento de cabeza.....</i>	<i>45</i>
<b>3.2.3</b>	<i>Pruebas detección del uso del teléfono celular.....</i>	<i>47</i>
<b>3.2.4</b>	<i>Pruebas de detección con respecto a la iluminación.....</i>	<i>50</i>
<b>3.2.4.1</b>	<i>Pruebas de iluminación para la detección de cierre de ojos y asentamiento de cabeza .....</i>	<i>50</i>
<b>3.2.4.2</b>	<i>Pruebas de iluminación para la detección del teléfono celular.....</i>	<i>51</i>
<b>3.2.5</b>	<i>Prueba de tiempo de reconocimiento.....</i>	<i>52</i>
<b>3.3</b>	<b>Análisis de costos.....</b>	<b>53</b>
	<b>CONCLUSIONES.....</b>	<b>54</b>
	<b>RECOMENDACIONES.....</b>	<b>55</b>
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

<b>Tabla 1-1:</b>	Comparación de tarjetas de desarrollo.....	21
<b>Tabla 2-1:</b>	Comparación de las cámaras .....	24
<b>Tabla 1-2:</b>	Características de los modelos de Raspberry .....	31
<b>Tabla 2-2:</b>	Características técnicas de la cámara Pi NoiR.....	32
<b>Tabla 3-2:</b>	Características técnicas del Buzzer.....	33
<b>Tabla 4-2:</b>	Características técnicas de cargador adaptador .....	33
<b>Tabla 5-2:</b>	Características técnicas del inversor de corriente.....	34
<b>Tabla 1-3:</b>	Detección de somnolencia - Cierre de ojos .....	44
<b>Tabla 2-3:</b>	Detección de Asentamiento de la cabeza .....	46
<b>Tabla 3-3:</b>	Detección del uso del teléfono celular.....	48
<b>Tabla 4-3:</b>	Pruebas de iluminación para la detección de cierre de ojos y asentamiento de cabeza .....	50
<b>Tabla 5-3:</b>	Pruebas de iluminación en la detección del teléfono celular .....	51
<b>Tabla 6-3:</b>	Datos obtenidos por el tiempo de reconocimiento de objetos. ....	52
<b>Tabla 7-3:</b>	Costos del Hardware.....	53

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b>	Conductor dormido .....	6
<b>Figura 2-1:</b>	Conductor bostezando por el cansancio .....	6
<b>Figura 3-1:</b>	Distracción del conductor por uso del teléfono celular.....	10
<b>Figura 4-1:</b>	Etapas de un sistema de visión artificial .....	13
<b>Figura 5-1:</b>	Imagen digital y su estructura .....	14
<b>Figura 6-1:</b>	Imagen original y con ruido impulsivo .....	16
<b>Figura 7-1:</b>	Imagen original vs imagen con ruido gaussiano .....	16
<b>Figura 8-1:</b>	Clasificador Haar básico .....	19
<b>Figura 9-1:</b>	Tarjeta Raspberry Pi e identificación de puertos .....	22
<b>Figura 10-1:</b>	Cámara Pi NoiR .....	24
<b>Figura 1-2:</b>	Concepción general del prototipo .....	30
<b>Figura 4-2:</b>	Raspberry Pi 3 B .....	31
<b>Figura 5-2:</b>	Cámara Pi NoiR .....	32
<b>Figura 6-2:</b>	Buzzer .....	32
<b>Figura 7-2:</b>	Cardador de la Raspberry.....	33
<b>Figura 8-2:</b>	Inversor de corriente .....	34
<b>Figura 9-2:</b>	Diagrama de conexiones del prototipo.....	35
<b>Figura 2-2:</b>	Recolección de imágenes positivas.....	37
<b>Figura 3-2:</b>	Conjunto de imágenes negativas.....	37
<b>Figura 10-2:</b>	Diagrama de flujo para la adquisición de la imagen .....	38
<b>Figura 11-2:</b>	Diagrama de flujo del pre-procesamiento de imagen .....	39
<b>Figura 12-2:</b>	Diagrama de flujo para la activación de la alarma .....	40
<b>Figura 1-3:</b>	Ubicación del dispositivo en el vehículo .....	43
<b>Figura 2-3:</b>	Luxómetro de la marca AIIGSUN .....	50
<b>Figura 3-3:</b>	Datos del tiempo de reconocimiento mostrados en Python .....	52

## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-3:</b>	Numero de aciertos en la detección de somnolencia - cierre de ojos .....	45
<b>Gráfico 2-3:</b>	Eficiencia en la detección de somnolencia – cierre de ojos .....	45
<b>Gráfico 3-3:</b>	Numero de aciertos en la detección de asentamiento de cabeza .....	47
<b>Gráfico 4-3:</b>	Eficiencia en la detección de asentamiento de cabeza .....	47
<b>Gráfico 5-3:</b>	Numero de aciertos en la detección del uso del teléfono celular.....	49
<b>Gráfico 6-3:</b>	Eficiencia en la detección del teléfono celular .....	49
<b>Gráfico 7-3:</b>	Grafico de resultados de las pruebas de iluminación .....	51
<b>Gráfico 8-3:</b>	Grafico de las pruebas de iluminación respecto al teléfono celular .....	52
<b>Gráfico 9-3:</b>	Tiempo promedio del reconocimiento de rostro, ojos y teléfono celular .....	53

## ÍNDICE DE ANEXOS

- ANEXO A:** Especificaciones físicas de uso de la Raspberry Pi 3 B
- ANEXO B:** Diagrama de esquema interna de la Raspberry Pi 3 B
- ANEXO C:** Datasheet de la GPIO de la Raspberry Pi 3 B
- ANEXO D:** Datasheet de los pines de la Cámara de Raspberry PiNoIR V2
- ANEXO E:** Diagrama de esquema interna de la Cámara de PiNoIR V2
- ANEXO F:** Líneas de Código en Python del sistema de alerta

## ÍNDICE DE ABREVIATURAS

- VA:** Visión Artificial
- VNC:** Virtual Network Computing
- XML:** Lenguaje de Marca Extensible.
- PDI:** Procesamiento Digital de Imágenes
- RAM:** Random Access Memory
- GPIO:** Entrada/Salida de Propósito General
- SO:** Sistema Operativo
- OMS:** Organización Mundial de Salud
- ANT:** Agencia Nacional de Transito

## **RESUMEN**

En el presente trabajo de titulación se implementó un prototipo de sistema de alerta para conductores de vehículos que detecta y advierte sobre la presencia de distracción y somnolencia usando visión artificial. El sistema consta de cuatro etapas: entrenamiento del clasificador para el teléfono celular, adquisición de la imagen, procesamiento, detección y activación de la alarma. En la primera etapa se realizó el entrenamiento del clasificador utilizando la técnica Haartraining para obtener características únicas del objeto en estudio, en la segunda etapa se utilizó una tarjeta de desarrollo Raspberry Pi 3 B, el cual recibe el video en tiempo real grabado por la cámara, la tercera etapa se encarga de procesarlo buscando características del estado de los ojos, asentamiento de cabeza y del teléfono celular; una vez realizado el procesamiento, se ejecuta la detección de somnolencia y distracción para que luego entre en funcionamiento la etapa de alerta. El objetivo se selecciona mediante la fijación en los ojos, en el cual este identifica la somnolencia o distracción una vez detectados estos puntos se enviará una señal a los puertos de la Raspberry en donde se emitirá una señal de alerta. El algoritmo del sistema se desarrolló en Python por su diversidad de librerías, además se utilizó la librería OPEN CV por la variedad de comandos para el procesamiento de imágenes. Como resultado de las pruebas se determinó que la distancia optima es de 60 cm y no debe existir objetos en la línea imaginaria entre la cámara y el rostro del conductor. La velocidad que le tomó al sistema en procesar y enviar la información es de 3.38 milisegundos y el sistema funciona con una iluminación mayor a 2000 luxes. Se recomienda ampliar el tiempo de pruebas del prototipo con la finalidad de identificar mejoras a su diseño.

**PALABRAS CLAVES:** <ROBÓTICA>, <VISIÓN ARTIFICIAL>, <SISTEMA DE ALERTA>, <ACCIDENTES DE TRÁNSITO>, <PROCESAMIENTO DE IMAGENES>, <RASPBERRY (SOFTWARE-HARDWARE)>, <PYTHON (SOFTWARE)>, <OPENCV (SOFTWARE)>.

## SUMMARY

In this titling work, a prototype warning system for vehicle drivers was implemented that detects and alerts about the presence of distraction and drowsiness using artificial vision. The system consists of four stages: qualifier training for the cell phone, image acquisition, processing, detection, and alarm activation. In the first stage, the preparation of the classifier was performed using the Haartraining technique to obtain unique characteristics of the object under study; in the second stage, a Raspberry Pi 3 B processor was used, which receives the real-time video recorded by the camera. The third stage is responsible for processing the video looking for characteristics of the state of the eyes, head settlement and cell phone; Once the processing has been carried out, the detection of drowsiness and distraction is carried out so that the alert stage will then start operating. The objective is selected by fixing the eyes, in which it identifies their state (open or closed), head settlement, and detection of cell phone use. Once these points are detected, a signal will be sent to the ports of the Raspberry, where an alert message will be emitted. The system algorithm was developed in Python for its diversity of libraries. Besides, the OPEN CV library was used for the variety of commands for image processing. As a result of the tests it was determined that the optimal distance is 60 cm and there should be not objects in the imaginary line between the camera and the driver's face. The speed that allows the system to process and send the information is 3.38 miliseconds and the system works with lighting greater than 2000 lux. It is recommended to extend the testing time of the prototype with the proposal to identify improvements to its design.

**Keywords:** <ROBOTICS>, <ARTIFICIAL VISION>, <ALERT SYSTEM>, <TRAFFIC ACCIDENTS>, <IMAGE PROCESSING>, <RASPBERRY (SOFTWARE-HARDWARE)>, <PYTHON (SOFTWARE)>, <OPENCV (SOFTWARE)>.



## INTRODUCCIÓN

Según la agencia nacional de tránsito del Ecuador, en el año 2018 se registró 25.530 siniestros en las carreteras del país dejando 2.151 personas fallecidas, siendo la conducción desatenta 40,03%, y la somnolencia o mal estado físico 3,07%, una de las causas de accidentes de tránsito. (Agencia Nacional de Transito, 2018)

Según la OMS la distracción es *“el desvío de la atención de todas aquellas actividades esenciales para una conducción segura hacia otra actividad diferente”*. Cuando el conductor se distrae, su atención se ve temporalmente dividida entre la "tarea principal", esto es, conducir, y otras "tareas secundarias" no relacionadas con la conducción. Así como durante una conversación por celular, los recursos cognitivos del conductor (es decir, la capacidad de reflexión) se utilizan para analizar la situación de la conducción y la conversación el cual está teniendo lugar. En consecuencia, la percepción del conductor en el entorno, su capacidad para tomar decisiones y su desempeño en la conducción se ven afectados. (OMS, 2015)

La somnolencia es el adormecimiento diurno, es decir la persona siente mucho sueño durante el día y duerma más durante la noche, además de los trastornos del ciclo sueño-vigilia provocados por los cambios de turno en el trabajo o los viajes como consecuencia produce la reducción de los reflejos y capacidad de concentración disminuida por lo tanto la Organización Mundial de la Salud (OMS) insiste en que dormir no es un placer sino una necesidad. (OMS, 2015)

En la investigación CONducir EBrio Y DEtección DE SOMnoLENCIA se presentó un sistema para determinar la somnolencia del conductor la cual se basó en la detección en tiempo real de la cabeza, cara y boca, en donde se utilizó el algoritmo de cascada Haar o clasificador para la detección de cara y ojos. También se realizaron plantillas para la coincidencia en la región de la boca para la detección de bostezos. Este sistema tuvo un sensor de detección de alcohol el cual comprueba si el conductor está ebrio o no, cubriendo así las principales razones de los accidentes de tránsito (Charniya, N. N., & Nair, V. R., 2017).

En la investigación DEtección DE SOMnoLENCIA MULTIMODAL DEL CONDUCTOR, se realizó un estudio de viabilidad que presenta un sistema discreto donde se combinan una cámara en el tablero de montaje junto con un reloj, este sistema se usó para registrar las señales ópticas. Además se realizó un análisis estadístico con un sistema híbrido de detección de somnolencia, donde se aprovecha las medidas fisiológicas y conductuales, lo que determinó de forma fiable el nivel de somnolencia con una precisión que varía de 86% a 96% dependiendo el número de niveles y el método de clasificador utilizado (Lemkaddem, A. et al. , 2018).

La empresa Optalert, desarrolló el SISTEMA DE DETECCIÓN DEL ADORMECIMIENTO Y ADVERTENCIA TEMPRANA; este sistema está diseñado para detectar los indicios tempranos del adormecimiento antes de percatarse si la persona se encuentra cansada o no. Este sistema usa la tecnología de oculografía de reflectancia infrarroja (IR), la cual se encuentra alojado en un par de gafas en donde se monitorea los movimientos de los ojos y los párpados de los conductores de manera continua sin interferir con su desempeño ni con la conducción. Esta tecnología permitió identificar diversas variables oculares. Además este sistema de detección ya se encuentra activa en una empresa minera de Chile (Optalert, 2016).

Como se conoce en el mercado ya existen vehículos en diferentes marcas como: Volkswagen, Mazda, Volvo entre otros que cuentan con un sistema avanzado de asistencia al conductor con funciones para detectar la somnolencia o fatiga, pero el costo del mismo aumenta y resulta inaccesibles para la mayoría de las personas que optan por vehículos que no cuentan con estos beneficios del avance tecnológico.

Por ende, este proyecto de titulación responde a la necesidad de implementar un prototipo para la detección de somnolencia y distracción, que sea de bajo costo y de fácil acceso, con el objetivo de alertar la pérdida de atención en el conductor, el cual ayudará en la disminución de accidentes de tránsito, con la puesta en marcha de este sistema.

## **OBJETIVOS**

### **Objetivo general**

Implementar un prototipo de sistema de alerta para conductores distraídos y somnolientos de vehículos basado en visión artificial

### **Objetivos específicos**

- Investigar los síntomas de somnolencia y distracción de un conductor.
- Analizar los requerimientos de *hardware* y *software* necesarios para el sistema de alerta para conductores somnolientos y distraídos.
- Probar los algoritmos que permitan identificar los síntomas de los conductores somnolientos y distraídos.
- Evaluar si el prototipo implementado cumple con los requerimientos planteados.

El presente documento se detalla en forma escrita tres capítulos, donde en el Primer capítulo llamado Marco Teórico, se describe aspectos con respecto, a la somnolencia, distracción y los efectos que produce, la visión artificial y sus etapas, componentes de hardware y software. El segundo capítulo denominado marco metodológico hace referencia al desarrollo del sistema, sus 2 etapas, concepción, la descripción de los elementos hardware y software utilizado y el desarrollo del algoritmo para la detección de somnolencia y distracción. Finalmente, el Tercer capítulo llamado Pruebas y resultados, se realiza la validación del sistema, donde se realizan pruebas de detección, iluminación, tiempo de procesamiento y costos. Las conclusiones han sido desarrolladas de acuerdo a los resultados obtenidos de las pruebas del sistema, y las recomendaciones van relacionadas a investigaciones futuras que pueden mejorar el sistema.

# CAPÍTULO I

## 1. MARCO TEÓRICO

El presente capítulo se hace referencia a una revisión bibliográfica de los aspectos tomados en cuenta para llevar a cabo el trabajo de titulación. Se analiza los conceptos generales acerca de la somnolencia, efectos y la distracción. Se especifica las características de la visión artificial con sus respectivas etapas, además también se hace énfasis en las técnicas de procesamiento, finalmente se determina el *hardware* y *software* necesarios para implementar el prototipo.

### 1.1 Somnolencia

Es conocida como la propensión a dormirse o la habilidad de transición de la vigilia al sueño. La presencia e intensidad de esta necesidad puede ser inferida por cuán rápidamente se inicia el sueño, cuán fácilmente es interrumpido y cuánto tiempo se duerme (Guevara Ipushima, K. & Torres Jara, E, 2016).

Las principales causas de somnolencia son la disminución de la cantidad de horas de sueño, disrupción del ritmo circadiano, uso de algunos medicamentos y pobre calidad de sueño, éste se expresa como una fragmentación del sueño que se manifiesta en múltiples despertares o una progresión anormal de los estados y etapas del sueño. (Somnolence and sleep quality in medical students during hospital practices and holidays, 2014). Entre los efectos de encontrarse somnoliento son:

- Disminución de la vigilia
- Disminución del tiempo de reacción
- Disminución de la coordinación psicomotora
- Disminución del procesamiento de información

Para el conductor el efecto principal es la retirada progresiva de la atención en las demandas de la carretera, tráfico y señalización, lo que provoca un bajo rendimiento de conducción provocando accidentes. La mayoría de personas que presentan somnolencia presentan los siguientes signos:

- Parpadeo frecuente
- Frotarse los ojos
- Bostezos repetidos

- Inclinación de la cabeza
- Distracciones

### ***1.1.1 Relación de la somnolencia con los accidentes de tránsito***

Según la Organización Mundial de Salud (OMS) una de las causas más frecuentes que producen accidentes de tránsito es la presencia de somnolencia y distracción en conductores (OMS, 2015). La Agencia Nacional Tránsito afirma que la conducción desatenta es 40,03%, y la somnolencia o mal estado físico 3,07% de los accidentes donde tienen relación con esta causa (Agencia Nacional de Tránsito, 2018). Los accidentes de tránsito causados por somnolencia han aumentado el interés en desarrollar el sistema de detección de somnolencia y distracción.

### ***1.1.2 Sueño.***

Es un estado de “desconexión con la realidad”, una necesidad fundamental e imprescindible del ser humano; que permite recuperar las funciones físicas y psicológicas. También es considerado como un proceso fisiológico de vital importancia para la salud integral de los seres humanos (P. Carrillo Mora, J. Ramírez Peris, & K. Magaña Vázquez, 2015).

Además, cabe recalcar cuáles son sus características conductuales:

- Existe una disminución de la conciencia y de la reactividad a los estímulos externos.
- Se trata de un proceso fácilmente reversible, este lo diferencia de estados patológicos cursando con alteraciones del estado de alerta como el estupor y el coma.
- Generalmente se asocia a inmovilidad y relajación muscular.
- Suele presentarse con una periodicidad circadiana, por lo común en relación con los ciclos de luz y oscuridad.
- Durante el sueño los individuos adquieren una postura estereotipada.
- La ausencia de sueño, induce distintas alteraciones conductuales, psicológicas y fisiológicas; además de generar una “deuda” acumulativa de sueño el cual eventualmente deberá recuperarse a esto se conoce como rebote de sueño.

En la figura 1-1 se observa a un conductor dormido donde este lleva a la pérdida de la concentración en el desenvolvimiento en la conducción y por ello se producen los accidentes de tránsito.



**Figura 1-1:** Conductor dormido

**Fuente:** (Optalert, 2018)

### **1.1.3** *Fatiga*

Se refiere a la sensación de agotamiento y esta produce inhabilidad para realizar una actividad física como se esperaría realizarlo, esta es peligrosa por sí misma y como antesala del sueño, además la fatiga es la sensación de cansancio extremo, agotamiento o debilidad que puede hacer que las tareas cotidianas se tornen más difíciles (Union de Mutuas, 2017).

La fatiga reduce drásticamente nuestra capacidad de reacción y sobre todo disminuye nuestra capacidad para procesar la enorme cantidad de información al momento de conducir. Además, disminuye el campo de visión, alarga los tiempos de reacción y dificulta la coordinación psicomotriz y la capacidad muscular. Por lo tanto, se recomienda si una persona se encuentra fatigada descansar las 8 horas recomendables, las cuales son adecuadas para un mejor descanso, en la figura 2-1 se visualiza a un conductor bostezando por el cansancio (Rosales Mayor, E. & Rey de Castro Mujica, J., 2018).



**Figura 2-1:** Conductor bostezando por el cansancio

**Fuente:** (Union de Mutuas, 2017)

Conducir exige esfuerzo físico y mental, y los síntomas de fatiga que activan la alarma de que el sueño se apodera del conductor son los siguientes (Union de Mutuas, 2017):

- Pérdida de concentración
- Picor de ojos y parpadeo constante, pesadez en los ojos o visión borrosa y agotamiento.
- Dolor de cabeza y sensación de brazos o pies dormidos.
- Calambres musculares, dolores de espalda o zumbido de oídos.
- Necesidad de moverse en el asiento.
- Sequedad en la boca o deshidratación.
- Variación involuntaria de la velocidad.
- Dificultad para recordar los últimos kilómetros recorridos.
- Somnolencia o reacciones lentas o inseguras.

Según el artículo de Fatiga y Cansancio de la Corporación de Unión de Mutuas *“El cansancio al conducir puede ser potenciado por ciertos factores relacionados con la conducción como son los siguientes:”*

**Factores externos:** son aquellos relacionados con el estado de la vía y del entorno. Se presentan cuando hay mucho tráfico, en retenciones, en la conducción por carreteras poco conocidas o en mal estado, en situaciones de climatología adversa, etc. (Union de Mutuas, 2017).

**Factores del vehículo:** al conducir un automóvil mal ventilado, el exceso de temperatura dentro del habitáculo, la falta de iluminación de los faros, la conducción nocturna en un coche deficiente, el exceso de ruido, las vibraciones del motor, los defectos en dirección y suspensión, el diseño poco ergonómico. Es decir, todo vehículo en condiciones desfavorables para circular, puede aumentar el cansancio, la fatiga y el sueño (Union de Mutuas, 2017).

**Factores del conductor:** son aquellos conductores agotados que circulan a velocidades excesivas o tienen alteradas sus capacidades mentales convirtiéndose en un individuo fatigado, para evitar accidentes este tiene que reposar antes de ponerse al volante (Union de Mutuas, 2017).

**Otros factores:** son los factores clásicos, como el hambre, la somnolencia, la ingesta de alcohol y drogas, las enfermedades, los cambios en los hábitos, la postura inadecuada mientras se conduce, la falta de automatización de procesos de la conducción (Union de Mutuas, 2017).

#### **1.1.4 Alteraciones de la somnolencia en la conducción**

El acto de dormir es una de las necesidades primarias más importantes que tiene el ser humano. Si no se duerme o no se descansa lo suficiente, el organismo reacciona con una serie de desajustes altamente peligrosos para el conductor. Por lo tanto las alteraciones producidas por la somnolencia y las cuales afectan al conductor de un vehículo son (Unión de Mutuas, 2017):

- Se disminuye considerablemente el tiempo de reacción del conductor en la carretera.
- Se le hace más difícil mantener la concentración en la carretera. Por ello las distracciones pueden aparecerse con mayor facilidad.
- La toma de decisiones es más lenta y tarda más tiempo en procesar la información que recoge del ambiente y por lo tanto su reacción no es instantánea, lo cual se debe hacer en milisegundos.
- La capacidad de reacción es lenta, sus movimientos se vuelven más lentos y menos precisos.
- Ejecuta los movimientos de forma automática. Esto como consecuencia llevará a realizar una maniobra basándose en el hábito antes que la necesidad de situación real.
- Aparición de microsueños.
- Alteración de las funciones sensoriales.
- Alteración de la percepción donde se identifica de forma errónea cualquier objeto en el entorno de la vía.
- Cambios en el comportamiento donde se llega a sentir tensión, nervios e incluso agresividad. Por tanto, es posible que su comportamiento se vuelva más arriesgado cuando ya se esté acercando a su destino.

#### **1.1.5 Causas de la aparición de somnolencia**

En general, todo aquello que reduzca la cantidad o la calidad del sueño nocturno, impidiendo así la adecuada recuperación de nuestro organismo, va a provocarnos una mayor somnolencia al día siguiente. Las causas son:



**Privación del sueño:** Las personas adultas necesitan dormir entre 7 y 9 horas para estar a pleno rendimiento al día siguiente, por lo tanto dormir poco provocará una fuerte somnolencia que como ya hemos señalado, altera gravemente la capacidad para conducir (Miro, E.; Cano Lozano, M.; & Buela Casal, G., 2009).

**El sueño fragmentado:** Es despertarse varias veces durante la noche el cual impide que una persona tenga suficiente cantidad de sueño profundo y que su descanso sea reparador lo que afecta negativamente a su estado de ánimo, rendimiento, etc. Es importante señalar que para sufrir los efectos del sueño fragmentado no es necesario llegar a despertarse durante la noche. En muchas ocasiones lo que sucede es que el sueño se vuelve muy ligero y no se alcanzan las fases más profundas (Medina O., Sánchez N., et al, 2015).

**Cambios en el horario del sueño:** Son aquellas personas que cambian frecuentemente sus horas de sueño y por lo tanto suelen pasar por periodos de somnolencia mientras están despiertos, esto se puede dar en personas que trabajan con turnos de trabajo y choferes de vehículos (Trastornos del sueño: ¿qué son y cuáles son sus consecuencias?, 2018). Alterar el llamado ciclo sueño-vigilia sin precauciones puede tener consecuencias para la seguridad en el tráfico, por lo que las personas que se ven obligadas a hacer esto deben extremar las precauciones en la carretera (Guevara Ipushima, K. & Torres Jara, E, 2016).

**Sustancias con estupefacientes:** El alcohol y ciertos medicamentos pueden favorecer la aparición de la somnolencia. Entre estos destacan los antihistamínicos como los utilizados en los resfriados y las alergias, algunos antidepresivos y, especialmente, muchos fármacos recetados en el tratamiento de la ansiedad. Bajo la influencia de estas sustancias se debe evitar la conducción ya que constituyen un claro riesgo para la seguridad vial (P. Carrillo Mora, J. Ramírez Peris, & K. Magaña Vázquez, 2015).

**Trastornos del sueño:** Alteran directamente el ciclo sueño-vigilia que influyen muy negativamente en la conducción y, por ello, son especialmente peligrosos para la seguridad en el tráfico. Tanto es así que algunos de estos trastornos, como la narcolepsia, impiden legalmente conducir (Cabanye, 2015).

## 1.2 Distracción

Según la Organización Mundial de la Salud (OMS) dice: *“la distracción es el desvío de la atención de todas aquellas actividades esenciales, cuando una persona se distrae su atención se ve temporalmente dividida entre la tarea principal y otras tareas secundarias que se ponen a realizar por motivos desconocidos o por un afán que la mayor parte de la gente tiene al celular y es por esta razón que hay muchos accidentes de tránsito”*. (OMS, 2015)

En el artículo distracciones al volante realizado por el ministerio de interior y la agencia nacional de tránsito de Madrid define la distracción como “cosa atrayente el cual capta su atención apartándola de aquello esencial, es decir divide la atención de la tarea principal como sería la de conducir, como se visualiza en la 3-1” (Sánchez Pérez P., 2017).



**Figura 3-1:** Distracción del conductor por uso del teléfono celular  
Fuente: (Sánchez Pérez, 2018)

### **1.2.1 Nivel de atención en la conducción**

La tarea de conducir implica que el conductor organice y coordine toda la información, los estímulos que le llegan. Exige por tanto de él, mantener un nivel adecuado de atención selectiva, mantenida y dividida de los cuales se dice a continuación (OMS, 2015):

**Atención selectiva:** Es aquella donde el conductor percibe estímulos y selecciona solo aquellos que aportan la información necesaria para ejecutar su tarea, desechando todo aquello que no es necesario. Cuando el conductor tiene mayor experiencia al volante resulta más fácil discriminar aquella información que no es necesaria (OMS, 2015).

**Atención mantenida:** Durante la conducción, es aquella que se prolonga en el tiempo, es decir se mantiene y da respuesta a las diferentes situaciones que se presentan durante el viaje en el automóvil. Como la atención selectiva esta también se ve afectada por los factores externos y propios del conductor la cual dificulta su concentración (Sánchez Pérez P., 2017).

**Atención dividida:** La información que percibe el conductor resultará más sencilla si se percibiera por dos vías diferentes como la auditiva y visual, es decir mientras conduce se debe dividir la atención en varias fuentes de información, o cuando la tarea no requiere mucho esfuerzo

por parte del conductor, lo que le posibilita atender a una nueva tarea (Sánchez Pérez P., 2017). Los factores de la atención dividida son:

- *Factores externos:* Facilitan al conductor perder su atención, como las vías que nos son muy familiares o que nos resultan monótonas pues estas hacen que bajemos la guardia y no prestemos atención, una señalización excesiva, las situaciones que son ajenas al tráfico como la publicidad, el móvil o las propias de la conducción como pueden ser los accidentes, en los que solemos mirar a ver qué es lo que ha sucedido (Sánchez Pérez, 2018).
- *Factores internos:* Entre las circunstancias personales del conductor se tienen la propia capacidad del mismo, hay personas que se distraen con mayor facilidad que otras. El estado en el que se encuentre el conductor es un factor determinante, si tenemos sueño o sentimos fatiga nos es más difícil mantener la atención. El alcohol, las drogas y los medicamentos también son factores facilitadores (Sánchez Pérez, 2018).

### **1.2.2 Tipos de la distracción en la conducción**

En el Ecuador, la distracción al volante es uno de los factores que mayor número de accidentes causa. Por lo tanto las distracciones en la conducción pueden ser de cuatro tipos (OMS, 2015):

- *Visuales:* es aquella en donde el conductor aparta la vista de la carretera para realizar una tarea no relacionada con la conducción como leer mensajes del teléfono celular, conversar con el copiloto, búsqueda en la guantera del automóvil, etc. (OMS, 2015).
- *Cognitivas:* es reflexionar sobre un tema de conversación del cual se esté hablando por teléfono, en lugar de analizar la situación de la carretera (OMS, 2015).
- *Físicas:* cuando un conductor sujeta o manipula un aparato en lugar de conducir con ambas manos al volante como cuando marca un número de teléfono o se inclina para sintonizar una emisora de radio, tiene comida o bebida, etc. son los efectos más frecuentes por los que ocurren accidentes (OMS, 2015).
- *Auditivas:* poseer aparatos con el volumen tan alto que no es posible escucha otros sonidos como la sirena de una ambulancia, policía, etc. (OMS, 2015).

Cabe indicar que existen vehículos con sistema de detección de somnolencia, pero su costo es muy alto, por lo tanto se ve la necesidad de crear un sistema que ayude a alertar a los conductores de vehículos. Para lo cual se procede al estudio de la Visión Artificial.

### **1.3 Visión artificial**

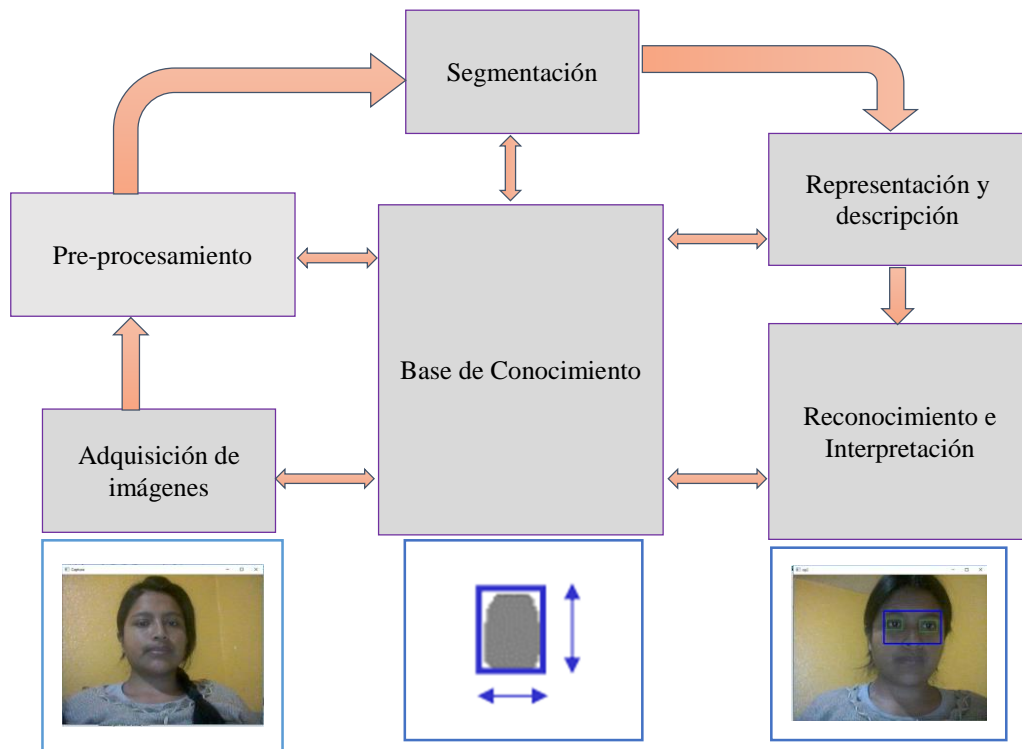
También conocida como visión por computadora, forma parte de la inteligencia artificial y consiste en un proceso computacional con el objetivo de la adquisición, procesamiento, análisis y explicación de cualquier tipo de información espacial del mundo real obtenida a través de imágenes digitalizadas (Sánchez Calle, 2005). Trata de producir en una maquina los sentidos de los humanos como son los ojos para comprender el mundo que nos rodea, esta podrá percibir y entender una imagen o secuencia de imágenes y actuar según convenga en una determinada situación (Gómez Rodríguez, F. & Domínguez, M. J., 2016).

Entre las aplicaciones más utilizadas en la visión artificial son:

- Detección de rasgos faciales.
- Reconocimientos de rostros.
- Análisis médico por imágenes.
- Interpretación de video vigilancia.
- Inspección industrial y control de calidad.
- Visión en máquinas y robots.
- Detección de Movimiento e Intrusos

#### ***1.3.1 Etapas de un sistema de visión artificial***

Las técnicas de visión artificial siguen normalmente un proceso desde la adquisición de la imagen hasta el reconocimiento e interpretación. En la Figura 4-1 se observa las etapas de un sistema de visión artificial y las cuales se detallan a continuación:



**Figura 4-1:** Etapas de un sistema de visión artificial  
**Fuente:** (Gómez Rodríguez, F. & Domínguez, M. J., 2016)  
**Realizado por:** Bertha Arévalo, 2019

### 1.3.1.1 Adquisición de imágenes

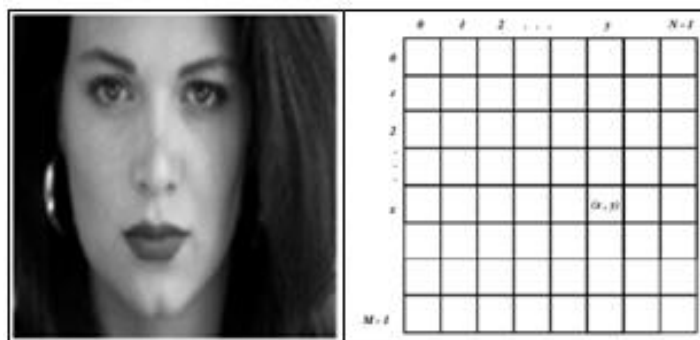
Se encarga de capturar una imagen y luego la digitaliza para lograr un cambio de analógica a digital, la imagen digitalizada es bidimensional, la misma que está constituida por píxeles. La digitalización de imágenes es diferenciada por dos procesos (Castejon, Ordieres, Martinez, & Alba, 2006) (Castejon, M. et al., 2006):

- *Muestreo:* es capaz de reducir la resolución espacial de la misma, además mediante este proceso se obtiene la imagen en una matriz de  $m$  filas y  $n$  columnas en donde se determina el tamaño del pixel y valor de brillo.
- *Cuantización:* es la imposibilidad de tener un rango indefinido de valores de medida para la intensidad de brillo de los pixeles, consiste en asignar valores a cada uno de los elementos de la matriz, los mismos representan el valor de la variable física en un determinado punto (Castejon, Ordieres, Martinez, & Alba, 2006) (Castejon, M. et al., 2006).

Para la adquisición de imágenes se utilizan dispositivos de captura que cuente buena resolución donde este adquirirá la escena y después enviara a una unidad donde pueda ser procesada (Sánchez Calle, 2005). Los más comunes son (Enrique Sucar & Gomez, 2005) (Sucar, E. L. and Gomez, G., 2005):

- *Fax*: convierte textos e imágenes impresos en papel y los transmite como información digitalizada a través de la línea telefónica hasta otro aparato de fax para imprimir la información.
- *Tabla digitalizadora*: se realizan dibujos y son visualizados en la pantalla de un ordenador. Se utiliza, por lo general, asociada a programas de dibujo técnico o artístico.
- Escáner: transforma la información impresa de un papel en un mapa de bits para poder ser luego almacenadas, además permiten una copia digital de alguna imagen o documento.
- *Cámaras digitales*: Cámaras fotográficas o de vídeo, guarda información ya digitalizada. La conversión de la información es analógica (Enrique Sucar & Gomez, 2005) (Sucar, E. L. and Gomez, G., 2005).

Una vez adquirida la imagen digital este es representada como una matriz bidimensional expresada como:  $f(x, y)$ , como se observa en la figura 5 -1, donde  $x, y$  corresponden a un punto en el plano cartesiano con el nivel de brillantez o intensidad de dicho punto sea a color o en escala de grises definido como la amplitud  $f$ , si estos elementos son finitos, es decir, una imagen digital tiene un número discreto de puntos denominados píxeles, cada pixel adquiere valores de colores codificados (R: red, G: green, B: blue). La frecuencia de actualización de las imágenes oscila, en el entorno digital, entre los 15 y los 60 FPS (frames por segundo). El rango más común se encuentra entre 25 y 30 FPS (Maduell García, E., 2016).



**Figura 5-1:** Imagen digital y su estructura  
Fuente: (Maduell García, E., 2016)

Las imágenes se las clasifican en dos grupos:

- *Imágenes Vectoriales*: Son aquellas imágenes capaces de conservar la nitidez de los bordes y no pierden los detalles cuando se modifica o altera el tamaño de la imagen, la resolución es independiente, es decir la información de cada punto es relativa (Tinajero, 2018).

- *Imágenes Raster o Mapa de Bits*: Este tipo de imágenes tiene un número fijo de píxeles donde dependen de la resolución, es decir pierden detalles y se ven dentadas cuando estas son ampliadas (Tinajero, 2018).

### 1.3.1.2 Pre procesamiento

Es utilizado para eliminar ruido en donde se incluyen operaciones de mejora en la fotografía adquirida como es de atenuar las imperfecciones debido a la función de transferencia del sistema de captación de imágenes, de regularizar la imagen, de mejorar el contraste o de optimizar la distribución de la intensidad o de realzar algunas características de la imagen, como bordes o áreas (Maduell García, E., 2016). En el procesamiento digital de imágenes (PDI) se maneja con cuatro tipos la cuales son: (Ferrer, 2018)

- *Imágenes RGB (red-green-blue)*: Este tipo de imágenes utiliza tres canales para reproducir los colores, es decir 8 bits por canal. Además son capaz de reproducir hasta 16.7 millones de colores soportando algunos formatos como: JPG, BMP, PNG, etc.
- *Imágenes indexadas*: reducen los colores de la imagen a un máximo de 256, además admiten los formatos GIF y PNG-8 y muchas aplicaciones multimedia.
- *Imágenes en escala de grises*: utilizan distintos tonos de gris, en imágenes de 8 bits, tienen hasta 256 tonos de gris, además cada píxel tiene un valor de brillo comprendido entre 0 (negro) y 255 (blanco).
- *Imágenes binarias*: tienen una profundidad de color de 1 bit (Ferrer, 2018).

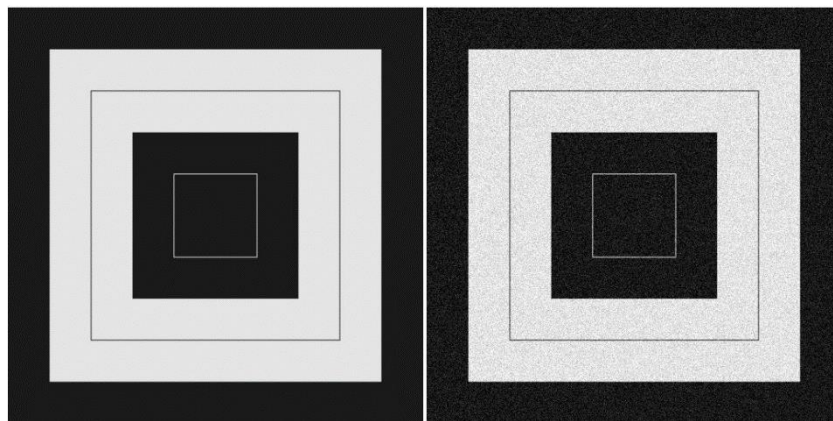
Cuando se captura la imagen este es adquirido normalmente con ruido, de los cuales estos pueden ser:

- *Ruido salt-pepper o ruido impulsivo*: El valor de cada píxel no tiene relación con el valor ideal, sino con el valor del ruido, estos valores pueden ser muy altos o bajos (puntos blancos y/o negros) donde se distribuyen de manera aleatoria por la imagen. Este aparece producto de interferencias atmosféricas, o por acciones hechas por el humano (motores de coches, etc.), como se observa en la figura 6-1 (Castejon, M. et al., 2006).



**Figura 6-1:** Imagen original y con ruido impulsivo  
**Fuente:** (Maduell García, E., 2016)

- *Ruido Gaussiano:* los píxeles de la imagen toman pequeñas variaciones en su valor en base a una distribución normal o gaussiana. El efecto de este ruido genera valores aleatorios los cuales variarán respecto al valor original del píxel, este ruido generalmente se produce por diversas ganancias existentes en la cámara, perturbaciones en el medio de transmisión o ruido en los digitalizadores (Sada, S. and Sanz Delgado, J., 2015), como se observa en la figura 7-1.



**Figura 7-1:** Imagen original vs imagen con ruido gaussiano  
**Fuente:** (Maduell García, E., 2016)

- *Ruido Multiplicativo:* el resultado final de la imagen es el resultado de la multiplicación de dos imágenes (Tinajero, 2018).

Los tipos de filtros útiles para mejorar la imagen con ruido son (Vélez Serrano, J.F., et al., 2003):

- *Filtro pasa bajo:* Es utilizado para alisar o suavizar la imagen. Tiende a retener la información de baja frecuencia en la imagen mientras se reduce la información de alta frecuencia. El alisado se consigue al disminuir la disparidad entre los valores de píxeles promediando los píxeles cercanos (López Mena, 2017). En este tipo de filtros existen dos clases: el filtro de la



mediana el cual se basa en sustituir el valor de un píxel por el de la mediana del conjunto formado por el mismo y sus ocho vecinos y el filtro de media o suavizado el cual compara la intensidad de un píxel con la de sus 8 vecinos. Si la diferencia es superior a cierto umbral se sustituye tal píxel por el valor alto de los píxeles vecinos, en otro caso se mantiene su valor de intensidad.

- *Filtro pasa alto:* Es utilizado para afilar la imagen, esto se consigue aumentando el contraste entre áreas contiguas con poca variación en el brillo. El filtro de énfasis de altas frecuencias acentúa las altas frecuencias y preserva las bajas frecuencias, para esto se debe multiplicar a la imagen original de la expresión del filtro pasa alto normal por una constante mayor a uno. Esto permite mantener las regiones homogéneas y resalta las discontinuidades.
- *Filtros detectores de borde:* Este tipo de filtros obtiene la gradiente en una sola dirección y promedia la información de la perpendicular. Entre estos filtros están el filtro de Sobel, Canny, Kirsh y Prewitt (Castejon, M. et al., 2006).

#### *1.3.1.3 Segmentación*

Fracciona una imagen en partes constituyentes u objetos, lo que consiste en diferenciar los distintos objetos que componen la señal y determina su posición con respecto al fondo de la imagen (Sánchez Calle, 2005).

El objetivo es conocer perfectamente los objetos que hay para extraer las características propias de cada uno de ellos. Además, cada píxel de la imagen debe tener una etiqueta que los defina, de forma que simplemente por agrupación de puntos con la misma etiqueta y conectados espacialmente, se pueda determinar la lista de objetos. Existen varias técnicas: umbralización, determinación de discontinuidades, crecimiento de regiones, uso del color o de movimiento, etc. (Gómez Rodríguez, F. & Domínguez, M. J., 2016).

#### *1.3.1.4 Representación y descripción*

Consiste en extraer características de un objeto para reconocerlo por regiones, una región se puede describir por la forma de su frontera o por sus características internas. La identificación de objetos o regiones en una imagen a menudo se pueden realizar usando descriptores de textura. Básicamente son de tipo morfológico, tales como área, perímetro, excentricidad, momentos de

inercia, esqueletos, pero también se puede emplear características basadas en la textura o en el color (Gómez Rodríguez, F. & Domínguez, M. J., 2016).

#### *1.3.1.5 Reconocimiento e interpretación*

El reconocimiento es el proceso que asigna una etiqueta a un objeto basada en la información que proporcionan los descriptores (clasificación), mientras que la interpretación lleva a asignar significado al conjunto de objetos reconocidos. Para clasificar una imagen se comparan todas y cada una de sus características locales con todas y cada una de las extraídas del conjunto de aprendizaje. Las imágenes se clasifican en la clase que mayor número de coincidencias tenga (Maduell García, E., 2016).

#### **1.3.2 Iluminación**

Es la parte más crítica dentro de un sistema de visión. Las cámaras capturan la luz reflejada de los objetos. El propósito de la iluminación utilizada en las aplicaciones de visión es controlar la forma en que la cámara va a ver el objeto (Fontanellas, 2014).

La iluminación adecuada puede mejorar una imagen de forma que se anulen algunas características y se mejoren otras, silueteando una pieza que oscurece los detalles superficiales para permitir la medición de sus bordes.

### **1.4 Patrones de reconocimiento para personas somnolientos y distraídas**

Las metodologías para reconocimiento de rostros se clasifican en dos grupos: en base a holística, en donde se considera al rostro como un todo y con enfoque analítico, basados en análisis de características de las partes del rostro como ojos, nariz, boca, etc. También se establece una clase distinta, en donde el reconocimiento se basa en la aplicación de un modelo.

#### **1.4.1 Detector de caras: Algoritmo de Viola Jones**

Paul Viola y Michael Jones desarrollaron este algoritmo en 2001. Este sistema de detección de rostros representa un gran avance debido a su rapidez para identificar caras humanas, este algoritmo realiza la clasificación mediante características extraídas en una escala de grises, este no utiliza directamente la imagen sino una representación de la misma, llamada imagen integral. El algoritmo divide esta última, en subregiones de distinto tamaño y emplea una serie de

clasificadores en cascada donde tienen ciertas características gráficas (Facial parts detection using Viola Jones algorithm, 2017).

El algoritmo busca combinaciones de estos patrones y al no procesar la totalidad de la imagen original sino las subregiones, presentan una posibilidad de poseer un rostro, disminuyendo el gasto computacional de manera considerable (Espinoza Olguín, E. & Jorquera Guillen, P. I., 2015). Los algoritmos del método de Viola Jones se detallan a continuación:

#### *1.4.1.1 Integral de la imagen*

Esta es usada para la rápida detección de las características a diferentes escalas, no trabaja directamente con los valores de intensidad si no con una imagen acumulativa que se construye a partir de operaciones básicas (Faces Detection in Digital Images Using Cascade Classifiers, 2008).

Es una representación de la imagen original, donde se extraen las características. Esta imagen es obtenida a partir de un algoritmo con la suma de los valores de un rectángulo dentro de una imagen (Espinoza Olguín, E. & Jorquera Guillen, P. I., 2015).

#### *1.4.1.2 Características de rectángulos – Características Haar*

Son descriptores para obtener información de una zona en particular mediante operaciones aritméticas, además tiene gran eficiencia de cálculo. Estas características son rasgos muy simples para la búsqueda en las imágenes la cual consisten en la diferencia de intensidades luminosas entre regiones rectangulares adyacentes. Las características quedan definidas por unos rectángulos y su posición relativa a la ventana de búsqueda adquiriendo un valor numérico. En la figura 8-1 se observa el clasificador de Haar básico (Barrero Parra, 2015).



**Figura 8-1:** Clasificador Haar básico  
**Fuente:** (Espinoza Olguín, E. & Jorquera Guillen, P. I., 2015)

La extracción de características es realizada aplicando filtros con bases Haar. En el algoritmo de Viola-Jones se usan tres características de Haar:

- *Característica de dos rectángulos:* es la diferencia entre la suma de los píxeles de ambas regiones rectangulares.
- *Característica de tres rectángulos:* Es la suma de los píxeles de los rectángulos exteriores menos la suma del rectángulo central.
- *Característica de cuatro rectángulos:* Es la diferencia entre los pares diagonales de los rectángulos.

#### 1.4.1.3 AdaBoost

Fue introducido por Freund y Schapire en 1995 resolviendo muchas de las dificultades prácticas asociadas al proceso de boosting el cual consiste en tomar una serie de clasificadores débiles y combinarlos para construir un clasificador fuerte con la precisión deseada (Barrero Parra, 2015), donde selecciona un pequeño set de características para entrenar el clasificador. Para seleccionar las características y entrenar al clasificador, se combina una selección de funciones débiles de clasificación para formar un clasificador fuerte. Las funciones débiles están diseñadas para seleccionar una característica de Haar para separar de mejor forma los ejemplos positivos y negativos. (Espinoza Olgún, E. & Jorquera Guillen, P. I., 2015)

Los pasos del algoritmo son:

- Se tienen las imágenes  $(X_1, Y_1), \dots, (X_n, Y_n)$  donde  $Y_i = 0, 1$  para ejemplos negativos y positivos respectivamente.
- Inicializar los pesos, dados por  $\frac{1}{2} m$  y  $\frac{1}{2} n$  donde  $m$  y  $n$  son el número de imágenes positivas y negativas, respectivamente.
- Normalizar los pesos.
- Para cada característica, se evalúa el error con respecto a una ventana de  $24 \times 24$  píxeles de las imágenes positivas.
- De esta forma se escoge un clasificador, con el menor error.
- Se actualizan los pesos.
- Después de algunas iteraciones, se obtiene un clasificador fuerte, la cual este separa de manera eficiente las imágenes positivas de las imágenes negativas.

#### 1.4.1.4 Cascada de decisión

El algoritmo de Viola-Jones utiliza un árbol binario de clasificadores para decidir si una región se trata de una cara o no. Un clasificador no es suficiente para detectar rostros de manera eficiente, por eso se implementa una cascada de clasificadores entrenados y ajustados. Además está formada por nodos donde se analizan las diferentes características de las diferentes regiones de la imagen. Cada nodo representa un clasificador obtenido con el algoritmo AdaBoost (Delgado Rodríguez, 2012).

### 1.5 Hardware del prototipo

Para procesar la adquisición de imágenes por el dispositivo de captura es necesario utilizar una tarjeta de desarrollo que sea capaz de controlar dichas señales, para ello en la tabla 1-1 se analizan las características de las posibles tarjetas que puedan ser usadas.

**Tabla 1-1:** Comparación de tarjetas de desarrollo

Características	Raspberry Pi	Cubieboard	Arduino
<b>Procesador</b>	Broadcom BCM2837B0, Cortex-A53 (armv8) 64-bit SoC	AllWinner A20 SoC (dual-core ARM Cortex-A7 @ 1 GHz CPU	Atmega 2560
<b>RAM</b>	1GB LPDDR2 SDRAM	1GB DDR3	8Kb
<b>SD</b>	MicroSD	MicroSD	
<b>Conectividad inalámbrica</b>	2.4 GHz / 5 GHz IEEE 802.11.b/g/n/ac Bluetooth 4.2, BLE	-	-
<b>Conectividad de red</b>	Gigabit Ethernet over USB 2.0	10/100M Ethernet	-
<b>Puertos</b>	GPIO 40 terminales HDMI 4 x USB 2.0 CSI (cámara Raspberry Pi) DSI (pantalla táctil) Toma auriculares / vídeo compuesto Micro SD Micro USB	GPIO 96 terminales HDMI 2 x USB Micro USB	14 Puertos analógicos 54 Puertos Digitales Mini USB Micro SD
<b>Alimentación</b>	5 V	5V	5V
<b>Corriente</b>	2.5 A	2 A	400mA
<b>Precio</b>	\$60	\$225	\$15

Fuente: (Raspberry.org, 2019), (Arduino.org) y (ElectroniLab)

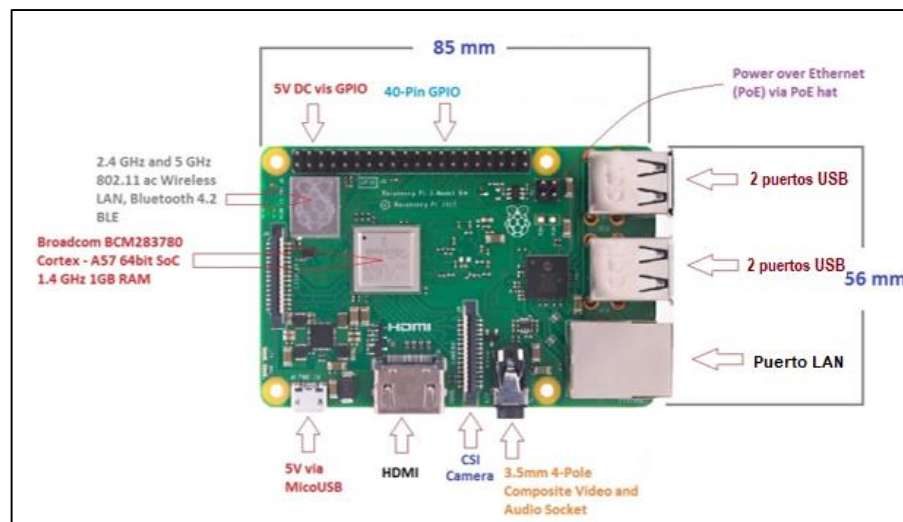
Realizado: Bertha Arévalo, 2019

Después de revisar cada una de las especificaciones técnicas se escoge la tarjeta Raspberry por su amplia gama de características, es de bajo costo, tiene un puerto CSI para la cámara y es apto para el procesamiento de imágenes.

### 1.5.1 Raspberry Pi

Son sistemas computacionales de uso general, bajo costo y completamente portables, promovidas por la fundación Raspberry, originaria del Reino Unido. La política de la fundación, es promover el uso de este tipo de tecnología, para aplicaciones principalmente enfocadas en la educación e investigación. Este comportamiento se verifica con un alto número de proyectos en donde se emplea la tecnología desarrollada por Raspberry e inclusive, algunos investigadores defienden su uso (Xataka.com, 2019).

Es una pequeña placa de computadora que viene con CPU, GPU, puertos USB, terminales de E / S, WiFi, Bluetooth, puertos USB y de red, donde es capaz de realizar algunas funciones como una computadora normal. Las dimensiones mecánicas es de 85 mm x 56 mm (alto x ancho) (Syed, Z. N.; Muhammad, A. A.; et al., 2019). En la figura 9-1 se observa la tarjeta Raspberry Pi con sus respectivos puertos donde se indica cada uno de ellos.



**Figura 9-1:** Tarjeta Raspberry Pi e identificación de puertos

Fuente: (Syed, Z. N.; Muhammad, A. A.; et al., 2019)

#### 1.5.1.1 Terminales de entrada y salida

Existen 40 terminales GPIO que se utiliza para desarrollar una conexión externa con el dispositivo electrónico a demás es compatible con todos los dispositivos, en donde este se divide de la siguiente manera:

- De los 40 terminales, 26 se utilizan como entradas y salidas digitales y 9 de los 14 restantes se denominan terminales de E / S dedicados que indican que no vienen con una función alternativa.
- Los terminales 3 y 5 vienen con una resistencia de extracción incorporada que 1,8 k $\Omega$  y los 27 y 28 están dedicados a la ID EEPROM. En el modelo B +, el cabezal GPIO se reposiciona ligeramente para permitir más espacio para el orificio de montaje adicional. Los dispositivos que son compatibles con el modelo B pueden funcionar con la versión B + (Syed, Z. N.; Muhammad, A. A.; et al., 2019).

#### 1.5.1.2 Definiciones básicas de las partes

- *CPU*: Es el cerebro que ayuda a llevar a cabo una serie de instrucciones basadas en las fórmulas matemáticas y lógicas. Viene con una capacidad de 64 bit.
- *Velocidad de reloj y RAM*: viene con una velocidad de reloj de 1.4 GHz Broadcom BCM2837B0 que contiene ARM Cortex-A53 de cuatro núcleos y la memoria RAM es de aproximadamente 1 GB.
- *GPU*: significa unidad de procesamiento de gráficos, utilizada para realizar el cálculo de imágenes.
- *Conector USB*: tiene 4 puertos USB útil para agregar nuevos componentes.
- *Conector de fuente de alimentación MicroUSB*: este conector se utiliza para proporcionar 5V de alimentación a la placa.
- *Conexión compuesta*: este tiene el conector de salida de audio como el compuesto de video, en donde solamente tiene un solo conector de 3,5 mm de 4 polos.
- *Unidad de disco duro USB*: está disponible en la placa que se utiliza para iniciar el dispositivo.
- *Power over Ethernet (PoE)*: tiene una característica donde la utilización de cable de datos permite pasar la corriente eléctrica necesaria (Raspberry.org, 2019).

#### 1.5.2 Dispositivo de Captura

En la tabla 2-1 se observa las características técnicas de las posibles cámaras a utilizarse en el proyecto, pues con estos detalles se procede a elegir la Cámara Pi NoIR por su resolución y compatibilidad con la tarjeta de desarrollo.

**Tabla 2-1:** Comparación de las cámaras

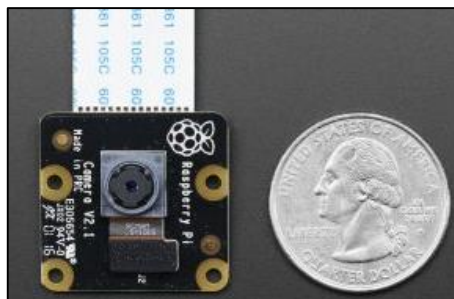
Características	Cámara Pi NoiR	Logitech
Dimensión	20 x 25 x 9 mm	
Resolución Nativa	8 Megapíxeles	3 Megapíxeles
Resolución Imágenes Estáticas	3280 x 2464 pixeles	640 x 480 pixeles
Tipo de sensor	Omnivisión	
Tamaño Óptico	1/4 "	
Voltaje de salida	3.3 V	5 V
Compatibilidad	A, B y B+ de Raspberry	-
Compatibilidad sensor de la cámara	Admite la conexión de infrarrojos LED flash de relleno y/o LED.	-

Fuente: (Electronilab.com, 2019)

Realizado: Bertha Arévalo, 2019

### 1.5.2.1 Cámara de Raspberry

Posee un sensor de imagen de 8 megapíxeles que otorga una alta calidad de imagen, lo que le convierte en la herramienta perfecta para la captura de imágenes en condiciones de poca luz. Es compatible con el modelo A, B y B+ de Raspberry Pi. Su sensor es capaz de tomar fotos fijas con una resolución de 3280 x 2464 pixeles, además de grabar vídeos en alta definición de 1080p30, 720p60 y 640x480p60/90. El módulo de la cámara ofrece contaminación de imagen reducida como ruido de patrón fijo y borrones. Dispone también de funciones de control automático como el control de exposición, el balance de blancos y la detección de luminancia lo que nos servirá para poder identificar los procesos a seguir ya sea en la detección del teléfono en el oído, la detección de somnolencia, y el asentamiento de cabeza, porque según eso se indicara el alarma (MCI-Electronics, 2019). En la figura 10-1 se observa la cámara Pi donde su tamaño es similar a la de una moneda.



**Figura 10-1:** Cámara Pi NoiR

Fuente: (RS-online.com, 2019)

El módulo de cámara Raspberry es un complemento diseñado a medida para el procesador del mismo nombre. Se adhiere a esta por medio de una de las dos pequeñas tomas en la superficie



superior del tablero. Esta utiliza la interfaz CSI, esta fue diseñada especialmente para interactuar con cámaras. El bus CSI es capaz de enviar datos a velocidades extremadamente altas, y transmite exclusivamente en forma de píxeles (RS-online.com, 2019).

El módulo en sí es pequeño, alrededor de 25 mm x 20 mm x 9 mm, pesa un poco más de 3g, esto lo hace perfecto para aplicaciones móviles u otras aplicaciones en las cuales su tamaño y el peso son importantes para su aplicación ya que es adecuado en todo sentido y por sus características. También se conecta al procesador de Raspberry mediante un cable plano corto o también llamado bus de datos. Para que la cámara está conectada al procesador BCM2835 en la Pi a través del CSI, tiene un enlace de mayor ancho de banda donde transporta datos de píxeles de la cámara al procesador (Electronilab.com, 2019).

#### 1.5.2.2 Conexión de la cámara a la Raspberry

Un cable plano de 15 cm fijado a las ranuras del módulo va directamente en el puerto de interfaz serie de su cámara Pi (CSI). Una vez conectado, puede acceder a la placa de cámara a través de la capa de abstracción multimedia (MMAL) o el vídeo. También hay bibliotecas como Picamera Python y muchas otras, los que pueden ser utilizados en aplicaciones futuras según el requerimiento que se tenga o se quiera aplicar en otro tipo de trabajo por que como base para próximos trabajos se tiene los elementos lo que es más importante. Este bus CSI puede variar su medida según los requerimientos necesitados (RS-online.com, 2019).

## 1.6 Software libre

Es un programa informático, donde prevalece la libertad para los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y modificar el *software*. Un *software* llega a ser libre si los usuarios tienen las siguientes libertades (Pérez Carvajal, 2014).

- Puede usar el programa, con cualquiera que sea su propósito.
- Puede estudiar cómo funciona el programa, y adaptarlo a lo que se requiera, se deberá tener acceso a código fuente, dado que sin él la tarea de incorporar cambios en un programa es extremadamente difícil.
- Puede redistribuir copias, ya sea de forma gratuita o pagada.
- Puede distribuir versiones modificadas del programa, de tal forma que la comunidad pueda aprovechar las mejoras introducidas.

El sistema operativo oficial de la fundación Raspberry Pi es Raspbian, sin embargo se tienen algunas opciones de sistemas como Noobs, Ubuntu Mate, Snappy Ubuntu Core, Windows 10 IOT Core, OSMC, Librelec, Pinet y Risc OS, pero en este proyecto de titulación se utiliza Windows 10 en donde se instalara el programa Python junto con la librería OpenCV, para la implementación del prototipo en donde este se ejecute de la manera que se requiere para los conductores con somnolencia y distracción (Xataka.com, 2019).

### **1.6.1 Python**

Es un lenguaje de programación de alto nivel y orientado a objetos preparado para realizar cualquier tipo de programa, además ofrece un entorno interactivo para la realización de pruebas. Es un lenguaje interpretado, donde no es necesario compilar el código fuente para poder ejecutarlo, ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. La sintaxis de Python es muy elegante y permite la escritura de programas cuya lectura resulta más fácil (Carrión Chamba, 2017).

### **1.6.2 Open CV, librería para el procesamiento digital de imágenes**

Open Source Computer Vision Library (OpenCV) es una librería de funciones de programación dirigida principalmente a la visión por computadora siendo una librería basada en aplicaciones en tiempo real para el tratamiento de imágenes. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real. Una de las ventajas son las diversas interfaces para otros lenguajes y entornos de aplicación con las que cuenta (Fernández Martí, 2015).

#### **1.6.2.1 Estructura modular de Open CV**

Tiene una estructura modular compuesto por paquetes con varias bibliotecas compartidas o estáticas. Los siguientes módulos están disponibles (OpenCv.org, 2019):

- *Funcionalidad básica (core)*: Módulo compacto capaz de definir las estructuras de datos básicas, incluyendo funciones básicas utilizadas por todos los demás módulos.
- *El procesamiento de imágenes (imgproc)*: Módulo de procesamiento de imágenes compuesto por el filtrado de imágenes no lineal y lineal, transformaciones geométricas, la conversión de espacio de color, histogramas, etc.
- *Video*: Módulo de análisis de vídeo donde incorpora la estimación de movimiento, la sustracción del fondo, y los algoritmos de seguimiento de objetos.

- *Calib3d*: Consta de múltiples algoritmos de visión geometría, la calibración individual de la cámara, y los elementos de la reconstrucción 3D.
- *Features2d*: Tiene algoritmos detectores de características sobresalientes, descriptores y comparadores de descriptores.
- *Objdetect*: Incluye algoritmos de detección y reconocimiento de objetos para objetos estándar.
- *Highgui*: Tiene una interfaz fácil de usar para cubrir las necesidades de interfaz de usuario sencilla.
- *Videoio*: Tiene una interfaz fácil de usar para la captura de vídeo.

### 1.6.2.2 OpenCV-Python

OpenCV-Python es la API de Python para OpenCV, que combina las mejores cualidades de la API de OpenCV C++ y el lenguaje de Python. Es una biblioteca de enlaces de Python diseñada para resolver problemas de visión de computadora (Mordvintsev, A. & Abid K., 2019).

Comparado con lenguajes como C/C++, Python es más lento. Por lo tanto, este se puede extender fácilmente con C/C++, lo que nos permite escribir código de computación intensiva en C/C++ y crear envoltorios de Python que se pueden usar como módulos de Python. Por lo que tiene dos ventajas (OpenCv.org, 2019):

- Primero, el código es tan rápido como el código C/C++ original, es decir el código C++ real que funciona en segundo plano.
- Segundo, es más fácil codificar en Python que en C/C++. OpenCV-Python es un contenedor de Python para la implementación original de OpenCV C++.

### 1.6.2.3 Paquetes adicionales en OpenCV-Python

Para utilizar OpenCV se necesitara de ciertos paquetes de Python los cuales serán de mucha utilidad al momento de desarrollar aplicaciones en visión artificial como son (OpenCv.org, 2019):

- *NumPy*: es una biblioteca de código abierto que da soporte a vectores y arrays para Python.
- *SciPy*: es una biblioteca de código abierto que contiene herramientas y algoritmos matemáticos para Python.
- *Matplotlib*: es una biblioteca de código abierto para la generación de gráficos a partir de vectores y arrays.
- *Pip*: gestor de paquetes para Python.

OpenCV-Python utiliza Numpy, que es una biblioteca altamente optimizada para operaciones numéricas con una sintaxis de estilo MATLAB. Esto también facilita la integración con otras bibliotecas que usan Numpy, como SciPy y Matplotlib (OpenCv.org, 2019).

### **1.6.3 PyCharm**

PyCharm es un IDE o entorno de desarrollo integrado multiplataforma utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django, entre otras bondades. PyCharm es desarrollado por la empresa JetBrains y debido a la naturaleza de sus licencias tiene dos versiones, la Community que es gratuita y orientada a la educación y al desarrollo puro en Python y la Professional, que incluye más características como el soporte desarrollo web con varios precios. Además, PyCharm es un IDE multiplataforma que funciona en Windows, macOS y Linux (Jetbrains.com, 2019).

#### *1.6.3.1 Características de PyCharm*

La nueva versión de PyCharm viene con grandes cambios en el IDE. JetBrains quiere estar a la par con las tecnologías de vanguardia del mundo Python, principalmente con las últimas versiones de Python y Django. Las características principales de PyCharm son las siguientes (Jetbrains.com, 2019):

- Asistencia y análisis de codificación, con finalización de código, sintaxis y resaltado de errores, integración de la impresora y soluciones rápidas
- Navegación de proyectos y códigos: vistas de proyectos especializados, vistas de estructura de archivos y salto rápido entre archivos, clases, métodos y usos
- Refactorización de Python incluye renombrar, extraer método, introducir variable, introducir constante, arrastrar hacia arriba, empujar hacia abajo y otros
- Soporte para frameworks web: Django, web2py y Flask.
- Depurador de Python integrado
- Prueba de unidad integrada, con cobertura de código línea por línea
- Desarrollo de Google App Engine Python
- Integración de control de versiones: interfaz de usuario unificada para Mercurial, Git , Subversion , Perforce y CVS con listas de cambios y combinación

PyCharm compite principalmente con una serie de otros IDE orientados a Python, incluido el PyDev de Eclipse y el IDE de Komodo con un enfoque más amplio (Jetbrains.com, 2019).

## CAPÍTULO II

### 2. MARCO METODOLÓGICO

En el presente capítulo se detallan las etapas que conforman el sistema y los elementos que son útiles para llevar a cabo el trabajo de titulación además se especifican las características y el diseño del *hardware* y *software* del prototipo.

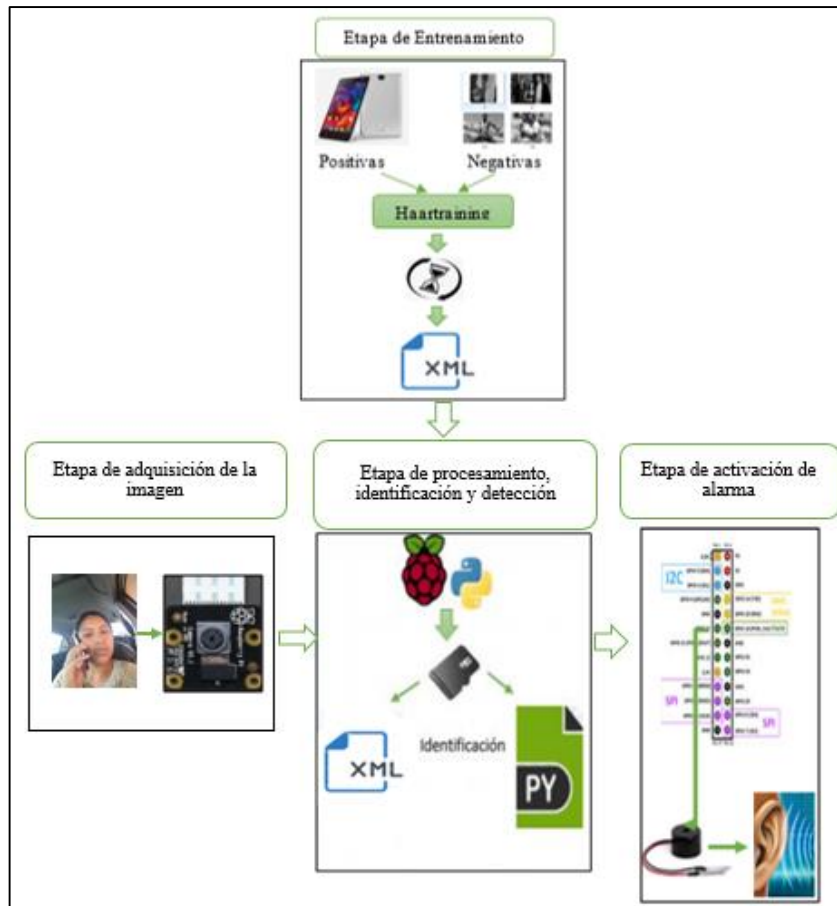
#### 2.1 Requerimientos del sistema

Por medio del estudio realizado en el capítulo anterior se pueden definir los requerimientos de diseño para satisfacer el prototipo de Sistema de Alerta para conductores y así tratar de evitar los accidentes de tránsito:

- Seleccionar una tarjeta desarrollo que sea capaz del procesamiento de imágenes.
- Seleccionar un dispositivo de captura que cuente con buena resolución.
- Seleccionar un dispositivo de alarma capaz de alertar al conductor.
- Seleccionar una fuente de alimentación para que se adapte en el vehículo.

#### 2.2 Concepción general del diseño de *hardware* del prototipo

En la figura 1-2 se observa la concepción general del prototipo, el diseño consta de una tarjeta de desarrollo el cual recibe la información del dispositivo de captura, es decir esta adquiere la escena donde a continuación se usa en la etapa de procesamiento digital de imágenes e identificación. Para el entrenamiento de un nuevo clasificador se toman fotos del teléfono celular en distintos ángulos, posiciones e iluminación y también aquellas imágenes que no tengan nada que ver con el objeto en estudio, después se procede a generar los archivos índices de la imágenes positivas y negativas para después crear un archivo XML (lenguaje de marca extensible), consecutivamente la cámara grabará en tiempo real y el *software* determinará si los objetos que aparecen en la escena tienen relación con las características obtenidas en los clasificadores, de ser así el ordenador enviará datos a los terminales 12 y 16 respectivamente , para que este envíe señales al buzzer el cual emitirá una alarma sonora, avisando que el conductor se encuentra distraído o somnoliento y así este no pierda su concentración al momento de conducir.



**Figura 1-2:** Concepción general del prototipo  
 Realizado por: Bertha Arévalo, 2019

### 2.3 Descripción de los elementos *hardware* del prototipo

A continuación, se detallan los componentes *hardware* que conforma el sistema con las características más importantes. En los anexos A, B, C, D y E se adjunta las respectivas hojas de datos.

#### 2.3.1 Selección de la tarjeta *Raspberry Pi*

En la tabla 1-2 se presentan las características de los últimos modelos lanzados por la fundación Raspberry de las cuales se realizará una comparación para posteriormente seleccionar el *hardware* adecuado para la implementación del prototipo.

**Tabla 1-2:** Características de los modelos de Raspberry

Características	Raspberry Pi 2 B+	Raspberry Pi 3 B	Raspberry Pi 3 B+
<b>Procesador</b>	BCM 2836 ARM v7 32bits	BCM 2438 64bits	BCM 2837B0 ARM v8 64bits
<b>Frecuencia</b>	900 Mhz	1.2 Ghz	1.4 Ghz
<b>Memoria RAM</b>	1 GB	1 GB	1 GB
<b>Conectividad inalámbrica</b>	IEE 802.11 B	IEE 802.11 b, Bluetooth 4.1	IEE 802.11 b, Bluetooth 4.2
<b>Conectividad de red</b>	Ethernet 100 Mbps	Gigabit Ethernet USB2.0	Gigabit Ethernet USB2.0 (300 Mbps)
<b>Puertos USB 2.0</b>	4	4	4
<b>Salidas de video</b>	DSI (Pantalla táctil) HDMI	DSI (Pantalla táctil) HDMI	DSI (Pantalla táctil) HDMI
<b>Almacenamiento</b>	Tarjeta SD	Tarjeta SD	Tarjeta SD
<b>Periféricos de bajo nivel</b>	40 x GPIO	40 x GPIO	40 x GPIO
<b>Corriente</b>	1.5 A	2.5 A	2.5 A
<b>Alimentación</b>	5 V	5 V	5 V

Fuente: (Raspberry.org, 2019)

Realizado: Bertha Arévalo, 2019

Con la tabla 1-2 se determinó que los últimos modelos de Raspberry Pi presentan similares características, excepto en la frecuencia de trabajo ya que el primer modelo tiene una frecuencia de trabajo de 1.2 GHz, mientras que en el segundo modelo existe una leve diferencia pues éste trabaja a 1.4GHz de frecuencia. Por lo mencionado anteriormente se concluye que los dos tipos de Raspberry resultan ser efectivas ya que la frecuencia de trabajo de los dos dispositivos es suficiente para cumplir con los requerimientos establecidos. En la figura 4-2 se visualiza a la tarjeta Raspberry Pi 3 B.



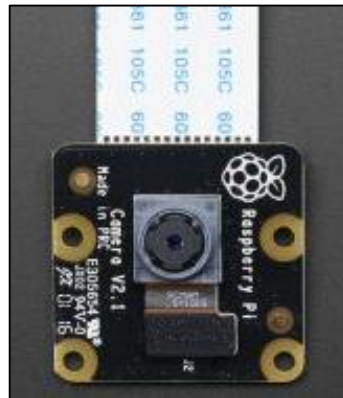
**Figura 2-2:** Raspberry Pi 3 B

Realizado por: Bertha Arévalo, 2019.

### 2.3.2 Cámara Pi NoiR

Los FPS (frame por segundo) que tiene la cámara y la velocidad de procesamiento hacen al elemento adecuado para la aplicación del prototipo, ya que con esas capacidades pueden realizar

capturas y procesamientos de imágenes como se requiere en esta ocasión. En la figura 5-2 se presenta al dispositivo y en la tabla 2-2 se visualiza las características de la cámara.



**Figura 3-2:** Cámara Pi NoiR  
Realizado por: Bertha Arévalo, 2019.

**Tabla 2-2:** Características técnicas de la cámara Pi NoiR

<b>Resolución</b>	8 megapíxeles
<b>Capacidad del Sensor</b>	2592 x 1944 píxeles
<b>Video</b>	1080p30 720p60 640 x 480 p60/90
<b>Dimensiones</b>	25mm x 20 mm x 90 mm
<b>Peso</b>	3g
<b>FPS</b>	15-120

Fuente: (Xataka.com, 2019)

Realizado por: Bertha Arévalo, 2019.

### 2.3.3 *Buzzer*

Es un dispositivo capaz de producir sonido o zumbido, ideal para emitir alarma como se observa en la figura 6-2. En la tabla 3-2 se observan las características técnicas de este dispositivo.



**Figura 4-2:** Buzzer  
Realizado por: Bertha Arévalo, 2019.



**Tabla 3-2:** Características técnicas del Buzzer

<b>Voltaje</b>	3-5V
<b>Corriente de consumo</b>	25mA
<b>Frecuencia</b>	2.7 kHz
<b>Impedancia</b>	16 $\Omega$

Fuente: (ABC COMPONENTS)

Realizado por: Bertha Arévalo, 2019.

### 2.3.4 Fuente de alimentación

Es una de las partes importantes para el funcionamiento de este proyecto porque a la tarjeta se debe alimentar con un voltaje adecuado, por esta razón se provee desde una fuente genérica con salida de 5V DC y una capacidad de corriente de al menos 2,5 A. Por lo tanto, se escoge el cargador propio de la raspberry de marca AUKURU donde este es un cargador adaptador. En la figura 7-2 se observa el cargador a utilizarse y en la tabla 4-2 se muestra las características de la misma.



**Figura 5-2:** Cardador de la Raspberry

Realizado por: Bertha Arévalo, 2019.

**Tabla 4-2:** Características técnicas de cargador adaptador

<b>Voltaje de entrada</b>	100 – 240 VAC
<b>Voltaje de salida de 5 V</b>	25mA
<b>Corriente de salida</b>	2.5 A
<b>Conexión</b>	Micro USB
<b>Longitud del cable</b>	1 m
<b>Dimensión</b>	77.4 x 4.2 x 2.7 cm
<b>Peso</b>	72.7 g

Fuente: (Raspberry.org, 2019)

Realizado por: Bertha Arévalo, 2019.

### 2.3.5 Inversor de corriente

Se elige el convertidor de corriente de marca BESTEK porque este dispositivo convierte la corriente directa de la batería del automóvil (12 VDC) a corriente alterna (110 VAC), donde se

puede conectar directamente a los bornes de la batería o desde el encendedor gracias a que incluye los cables necesarios. Tiene 2 contactos polarizados en los que puedes enchufar tus equipos, soportando una carga máxima de 600 Watts **Fuente:** (Raspberry.org, 2019). En la figura 8-2 se observa el inversor de corriente y en la tabla 5-2 se presenta las características de la misma.



**Figura 6-2:** Inversor de corriente  
**Realizado por:** Bertha Arévalo, 2019.

**Tabla 5-2:** Características técnicas del inversor de corriente

<b>Potencia de salida</b>	300 W
<b>Voltaje de entrada</b>	12 VDC
<b>Voltaje de salida</b>	110 V – 120 V
<b>Longitud del cable</b>	1 m
<b>Dimensión</b>	6.2 x 4.6 x 2.2 pulgadas
<b>Peso</b>	1.1 libras

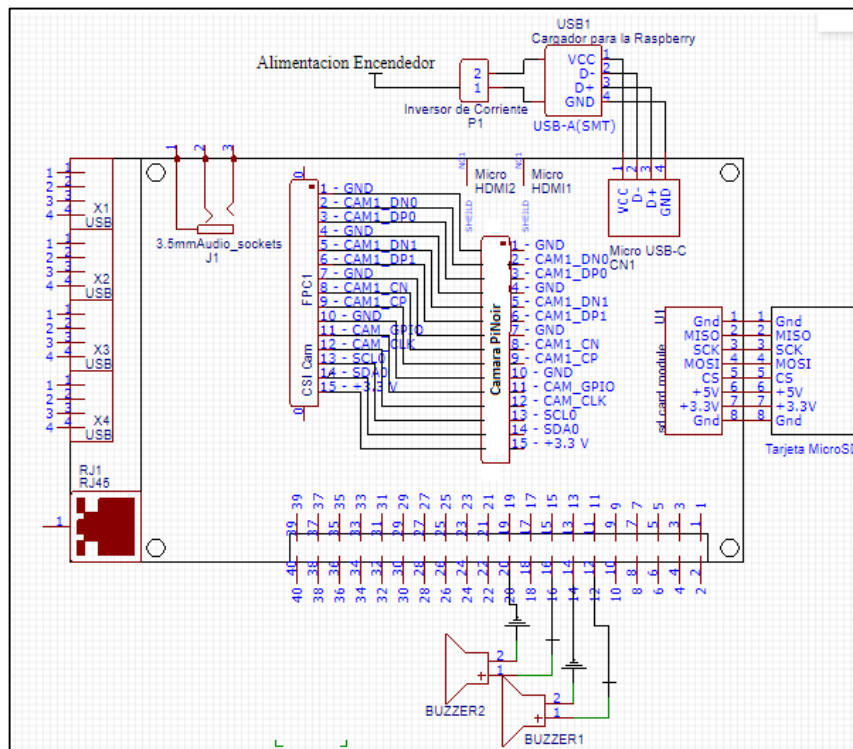
**Fuente:** (Raspberry.org, 2019)  
**Realizado por:** Bertha Arévalo, 2019.

## 2.4 Diagrama de conexiones del prototipo

En la figura 9-2 se observa la conexión esquemática de los dispositivos a utilizarse para el funcionamiento del proyecto, de los cuales se detalla a continuación:

- Se compone de un procesador Raspberry Pi donde su función será ejecutar el programa y realizar el procesamiento de imágenes.
- Se conecta dos Buzzer en los puertos 12 y 16, donde este se activará cuando se detecte somnolencia o distracción.
- La tarjeta de memoria micro SD se insertará en el puerto del mismo nombre para el almacenamiento del programa a realizarse.
- La cámara para la adquisición de imágenes estará conectada a través del conector CSI, el cual es propio de la cámara misma.
- La fuente de alimentación ira conectada al puerto micro USB el cual alimentará a la tarjeta y los demás dispositivos, además se enchufará el cargador a unos de los terminales del

convertidor de corriente que se acoplará al cenicero del vehículo, donde este convertirá el voltaje de salida de 12 V a uno de 110 V DC.



**Figura 7-2:** Diagrama de conexiones del prototipo  
Realizado por: Bertha Arévalo, 2019.

## 2.5 Selección del *software* del dispositivo

En esta parte se presenta el *software* a utilizarse para el procesamiento de imágenes, detección y activación de la alarma.

## 2.6 Requerimiento del *software*

Uno de los puntos más importantes para la ejecución de presente proyecto es también el requerimiento del *software* ya que todo el sistema depende de la programación, instalación y montarlo en el dispositivo que va a realizar todo el trabajo que se requiere. Además de algunos puntos importantes para dicho proyecto, las cuales son:

- Sistema Operativo y librerías gratuitas.
- Compatibilidad de las librerías con el sistema operativo.
- Entrenar un nuevo clasificador capaz de detectar el teléfono celular.
- Realizar el algoritmo para la detección de distracción y somnolencia.
- Emitir una alarma cuando el sistema detecte distracción o somnolencia.

### 2.6.1 *Software para el procesamiento de imágenes*

El *software* usado para el procesamiento de imágenes es Python 3.7.3, es un lenguaje de programación de alto nivel y fácil de aprender, posee estructuras de datos eficientes, que permiten realizar tareas con pocas líneas de código y de forma clara, además este viene instalado por defecto en el sistema operativo de la Raspberry.

### 2.6.2 *Análisis de OpenCV*

Para realizar la programación dirigida a la visión por computadora en tiempo real se utiliza OpenCV 4.1.0, además es una librería de funciones. Para utilizar OpenCV se necesitara de ciertas librerías de Python que serán de mucha utilidad al momento de desarrollar aplicaciones en visión artificial de las cuales son (Python.org, 2019):

- NumPy: es una biblioteca de código abierto que da soporte a vectores y arrays para Python.
- Sys: es un módulo que permite acceder a las funciones y objetos que se relacionan con el intérprete de Python.
- Time: Es una librería propia de Python que sirve para contabilizar el tiempo.
- Cv2: es una librería de visión artificial, pues posee algoritmos que hace que sea más sencillo el análisis y tratamiento de imágenes.

## 2.7 **Entrenamiento del nuevo clasificador**

En OpenCV ya contiene muchos clasificadores pre-entrenados para cara, ojos, sonrisa, etc. pero para la detección de otros objetos no existe, por lo tanto se procede al entrenamiento del clasificador para el teléfono celular. Para crear una cascada basada en el método de Viola-Jones, para cualquier tipo de objeto, es necesario construir dos conjuntos de imágenes independientes y de los cuales de detallaran paso a paso como se crea este clasificador:

**Conjunto de imágenes positivas:** En la figura 2-2 se observa el conjunto de imágenes los cuales va a contener fotografías del objeto en estudio, además se debe recolectar las fotos en distintos ángulos, iluminación y posición.



**Figura 8-2:** Recolección de imágenes positivas  
**Realizado por:** Bertha Arévalo, 2019

Una vez adquirida las imágenes se procede a generar el archivo índice que sirve para las siguientes etapas.

**Conjunto de imágenes negativas:** En la figura 3-2 se muestra las imágenes en las cuales no aparece el objeto en estudio, además deben estar en blanco y negro. También en estas imágenes se debe generar un archivo índice.



**Figura 9-2:** Conjunto de imágenes negativas  
**Realizado por:** Bertha Arévalo, 2019.

**Crear muestras:** Utilizando el comando `createsamples` se genera muestras donde este contiene el índice de las imágenes positivas. Este conjunto de imágenes es necesario para diseñar el clasificador.

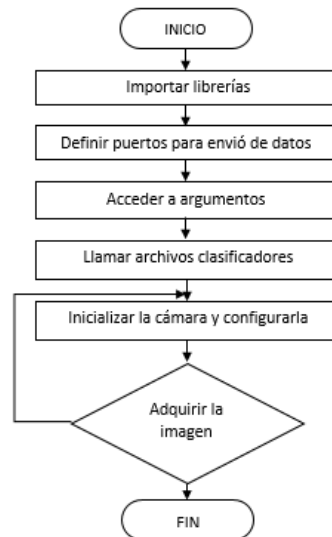
**Entrenamiento:** Para el entrenamiento del clasificador se utiliza el archivo índice de las imágenes negativas y el archivo de muestras generado en la etapa anterior. Como resultado de este proceso se obtiene el clasificador el cual será utilizado en el algoritmo para la detección del teléfono celular.

## 2.8 Programación en el IDE de Python

Una vez realizado el entrenamiento del clasificador para el teléfono celular se procede a la adquisición de la imagen, pre-procesamiento de imágenes e identificación y la activación de la alarma cuando se haya detectado somnolencia o distracción, donde el código completo en Python se encuentra en el anexo F y los pasos realizados se describen a continuación.

### 2.8.1 Adquisición de la imagen

Se adquiere la imagen mediante la cámara Pi NoiR, donde se obtiene la escena del conductor dormido (estado de los ojos), asentamiento de cabeza y utilización del teléfono celular. En la figura 10-2 se presenta el diagrama de flujo para la adquisición de la imagen.



**Figura 10-2:** Diagrama de flujo para la adquisición de la imagen  
Realizado por: Bertha Arévalo, 2019.

A continuación, se detalla el algoritmo del diagrama de flujo presentado en la figura 10-2.

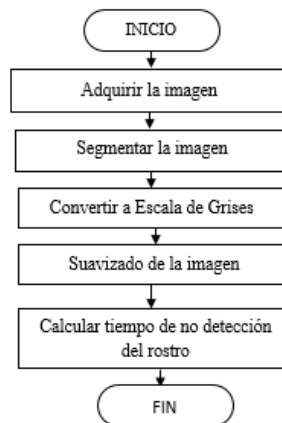
- *Importar las librerías:* como se menciona en el numeral 2.7.2.1 se usaron librerías para cumplir con el procedimiento establecido, esto se realiza implícitamente.
- *Definir puertos para envío de datos:* para la activación de la alarma se definen los puertos 12 y 16 para que una vez se detecte somnolencia o distracción, este suene respectivamente.
- *Acceder a los argumentos:* desde Python se puede acceder a los argumentos pasados a través de la consola, vía la lista `sys.argv`.
- *Llamar a archivos clasificadores:* se busca los archivos HAARs los cuales se ocupan de la detección de distracción o somnolencia. Esos archivos XML se almacenan en `C:\opencv-4.1.0\data\haarcascades / folder`.
- *Inicializar la cámara y configurarla:* se llama al objeto `PiCamera` con la línea de código de `“import picamera”` para limpiar los recursos. También se configura la resolución de la cámara,

esto solo se puede realizar cuando no se encuentre grabando. La resolución predeterminada será 1280x800, es decir si la pantalla ha sido desactivada.

- *Adquisición de la imagen:* una vez inicializada la cámara se procede a la captura de la imagen, donde se usa la siguiente línea de código “`camera.capture_continuous(rawCapture, format="bgr", use_video_port=True)`”, su función es gestionar la cámara en combinación con la Raspberry. Esta recibe las imágenes de vista previa y las envía tan rápido como sea posible para luego proceder con el procesamiento de la imagen.

### 2.8.2 Pre-procesamiento e identificación

Una vez adquirida la escena este tendrá canales almacenados en RGB (Azul, Verde y Rojo) por defecto, por lo tanto, se procede al procesamiento de imágenes, donde será necesario convertirlo a escala de grises para luego suavizarlo, es decir mejorar el contraste mediante una ecualización de histogramas donde este modifica las intensidades del pixel. Ya suavizada la imagen se segmenta para luego determinar características similares de los clasificadores, una vez ya identificadas las características se detecta somnolencia (estados de los ojos), asentamiento de cabeza y uso del teléfono celular. En la figura 11-2 se presenta el diagrama de flujo para el pre-procesamiento de la imagen.



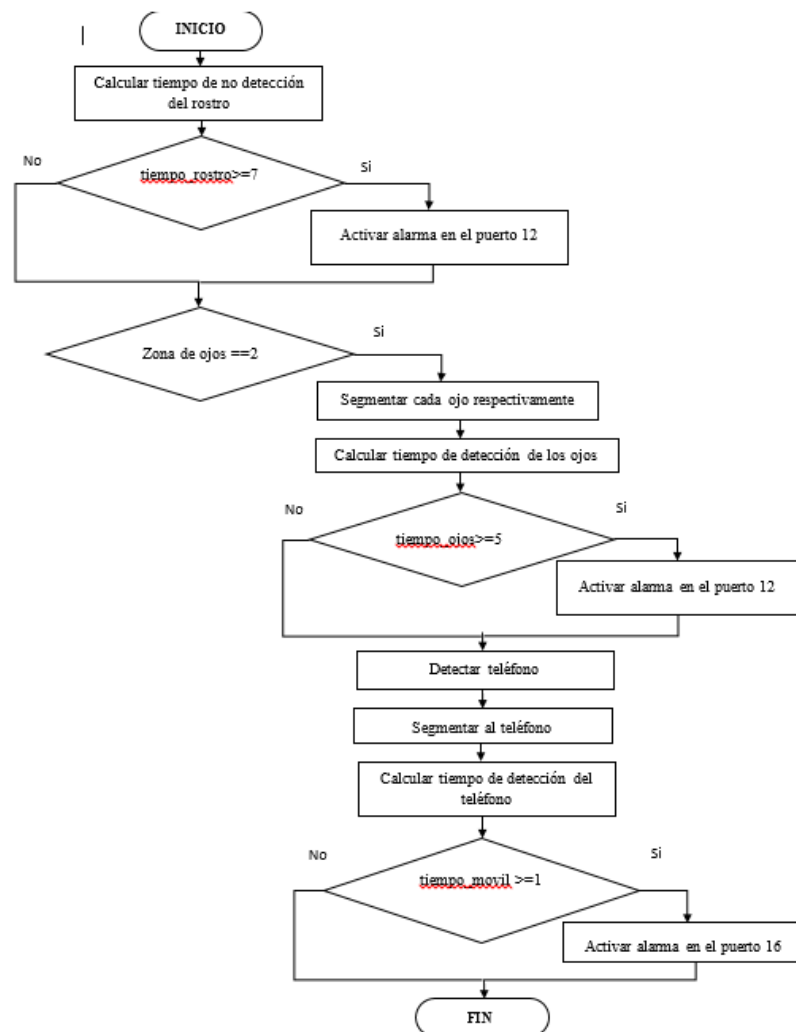
**Figura 11-2:** Diagrama de flujo del pre-procesamiento de imagen  
Realizado por: Bertha Arévalo, 2019.

- Una vez adquirida la imagen este tendrá canales almacenado en RGB (Azul, Verde y Rojo) por defecto, por lo tanto será necesario convertir la imagen a escala de grises. Para hacerlo se necesita llamar la función `cvtColor` el cual permite convertir la imagen de un espacio de color a otro, esta función recibe la imagen original y del código de conversión. Por lo tanto, como se necesita convertir la imagen original del espacio de color RGB a gris se usará el código `COLOR_BGR2GRAY`. A continuación de muestra la línea de código a utilizar “`gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)`”.

- Python tiene varias funciones para mejorar en contraste de la imagen, por esto es que la ecualización del histograma modifica las intensidades de pixel. En OpenCV tiene la función `equalizeHist()` en la cual se puede simplemente aplicar la ecualización de histogramas sobre una imagen. Además, es sencilla de aplicar en una imagen en escala de grises.
- Ya una vez suavizado la imagen se procede a calcular el tiempo donde no se detecta el rostro, esto se realiza para la detección de somnolencia (asentamiento de cabeza).

### 2.8.3 Activación de alarma

Una vez detectada la presencia de somnolencia o distracción se procede a enviar datos a los puertos 12 y 16 de la Raspberry respectivamente para así activar la alarma sonora donde este alertara al conductor de su falta de atención al momento de conducir, todo esto se realiza tratando de evitar accidentes de tránsito. En la figura 12-2 se presenta el diagrama de flujo para la activación de la alarma.



**Figura 12-2:** Diagrama de flujo para la activación de la alarma  
Realizado por: Bertha Arévalo, 2019.



- Una vez calculado el tiempo en donde no se detecta el rostro por más de 7 segundos este enviara un dato al puerto 12 de la Raspberry para emitir una alarma sonora.
- Cuando se haya detectado el rostro este procede a segmentar cada ojo respectivamente para posterior calcular el tiempo en que no se los detecta y así determinar si se encuentra somnoliento, es decir pasado un  $t \geq 5$  segundos este enviara una señal al puerto 12, para activar la alarma.
- Una vez finalizada con la detección de somnolencia se procede a la detección del teléfono celular una vez encontrado el objeto en estudio de la escena se segmenta para posteriormente calcular el tiempo en donde esta será mayor a 1 segundo y así enviará una señal al puerto 16 de la Raspberry para emitir una alarma sonora al instante.

Una vez finalizado con la realización del programa y cargado el programa en el sistema operativo de la Raspberry el dispositivo será ubicado con el objetivo de tener una respuesta idónea de detección.

## CAPÍTULO III

### 3. ANÁLISIS Y RESULTADOS

En el presente capítulo se exponen las pruebas realizadas al dispositivo, con el objetivo de validar el sistema a través de experimentos como: algoritmo de detección de distracción y somnolencia, detección con respecto a la iluminación y el tiempo de reconocimiento.

#### 3.1 Ubicación del dispositivo en el vehículo

Para la implementación del dispositivo en el vehículo se deben seguir algunos lineamientos básicos para la instalación del mismo, condiciones definidas para el diseño y bajo las cuales se ejecutará la calibración del algoritmo de reconocimiento de imágenes mediante OpenCV. Las principales consideraciones a seguir en la instalación del dispositivo son las siguientes:

- La distancia máxima a la que el dispositivo se instalara es de 60 cm. Distancias mayores disminuyen la efectividad en la detección de las características de distracción y somnolencia. Además, su ubicación será a 55 cm desde la ventana hacia el tablero y 30 cm desde el parabrisas hacia el centro, como se observa en figura 1-3.
- Para utilizar el sistema el conductor debe estar sentado, con la espalda erguida, cabeza recta y mirando al frente. La cámara debe ser capaz de adquirir la imagen sin dificultad, es decir que no existan objetos en la línea imaginaria entre la cámara y el rostro del conductor.

Las condiciones mencionadas ayudaran al uso adecuado del sistema y su correcta detección. En la figura 1-3 se visualiza la implementación final del dispositivo en el vehículo.



**Figura 1-3:** Ubicación del dispositivo en el vehículo  
Realizado por: Bertha Arévalo, 2019.

## **3.2 Metodología para las pruebas**

Las pruebas se realizaron con la cooperación de los integrantes de la cooperativa de taxis ejecutivo “Guanujotrans” de la ciudad de Guaranda provincia Bolívar, donde consta de 30 conductores de distintas edades. Cada uno ira acompañado por un “copiloto” que dirige los eventos controlados de somnolencia y distracción, donde el acompañante ordena al conductor que por un periodo de tiempo cierre los ojos, asiente la cabeza y utilice el teléfono celular, cuando la vía se preste para un movimiento rectilíneo, no existe presencia de vehículos y peatones en un amplio tramo. Sin embargo, algunos conductores no entendían como realizarlos a voluntad por lo tanto fue necesario desmostarles cómo debían hacerlo.

Para validar el sistema, el copiloto dirigirá estos eventos, haciendo que se repitan 10 veces. Se analizará el número de eventos detectados correctamente, los falsos positivos, falsos negativos y se determinará la eficiencia del sistema implementado.

### **3.2.1 Pruebas detección de somnolencia – Cierre de Ojos**

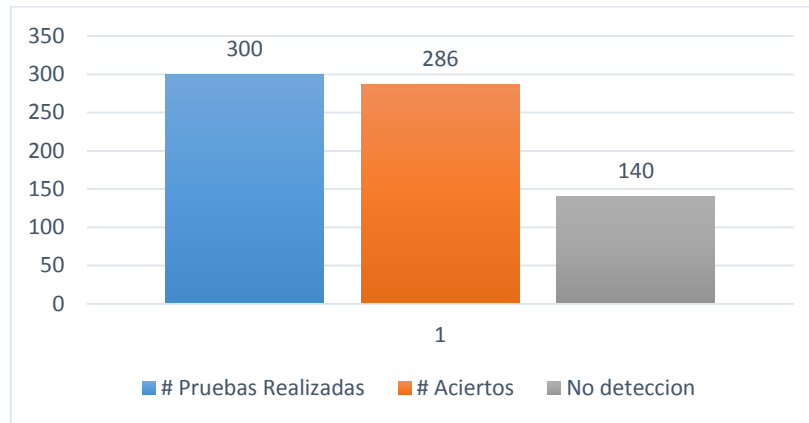
Para la detección de somnolencia, cada conductor cerró los ojos simulando estar dormido, todo se realizó bajo las condiciones mencionadas anteriormente. También se utilizó un computador para verificar la segmentación, determinación de características y detección de somnolencia – cierre de ojos. Además se valida el sistema comparando con un sistema patrón de detección en el estado de los ojos, donde se determina que tiene un porcentaje de error del 8 % (Charniya, N. N., & Nair, V. R., 2017).

**Tabla 1-3:** Detección de somnolencia - Cierre de ojos

# Conductor	# Pruebas Realizadas	# Aciertos	% Eficiencia	% Error
1	10	8	80 %	20 %
2	10	10	100 %	0 %
3	10	8	80 %	20 %
4	10	10	100 %	0 %
5	10	10	100 %	0 %
6	10	10	100 %	0 %
7	10	9	90 %	10 %
8	10	9	90 %	10 %
9	10	10	100 %	0 %
10	10	10	100 %	0 %
11	10	10	100 %	0 %
12	10	10	100 %	0 %
13	10	10	100 %	0 %
14	10	8	80 %	20 %
15	10	9	90 %	10 %
16	10	10	100 %	0 %
17	10	10	100 %	0 %
18	10	9	90 %	10 %
19	10	10	100 %	0 %
20	10	10	100 %	0 %
21	10	10	100 %	0 %
22	10	10	100 %	0 %
23	10	10	100 %	0 %
24	10	10	100 %	0 %
25	10	8	80 %	20 %
26	10	10	100 %	0 %
27	10	10	100 %	0 %
28	10	9	90 %	10 %
29	10	9	90 %	10 %
30	10	10	100 %	0 %
Total	300	286	2860%	140%
<b>Eficiencia en la Detección</b>			<b>95 %</b>	<b>5%</b>

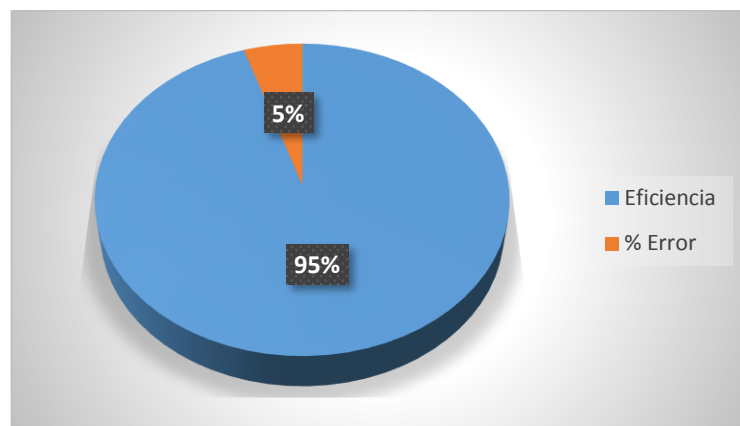
Realizado por: Bertha Arévalo, 2019

En la tabla 1-3 se exponen los resultados de la prueba, donde se observa que existió un total de 286 aciertos, mostrando así que el sistema posee un 95 % de eficiencia en la detección, con un error del 5%, por lo tanto en comparación con el sistema patrón se determina que el sistemas es eficiente en su detección, los resultados se muestran en el grafico 1-3.



**Gráfico 1-3:** Numero de aciertos en la detección de somnolencia - cierre de ojos  
**Realizado por:** Bertha Arévalo, 2019

El gráfico 2-3 muestra la eficiencia del sistema de alerta en la detección de somnolencia - cierre de ojos.



**Gráfico 2-3:** Eficiencia en la detección de somnolencia – cierre de ojos  
**Realizado por:** Bertha Arévalo, 2019

### 3.2.2 Pruebas detección de asentamiento de cabeza

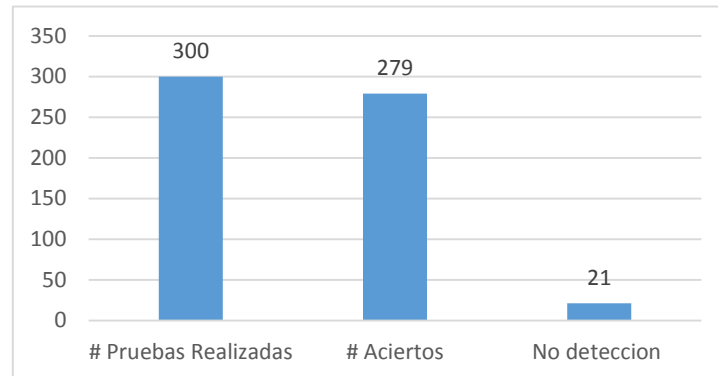
Para la detección de asentamiento de la cabeza, cada conductor inclino la cabeza simulando estar dormido o distraído (coger cosas del piso), todo se realizó bajo las condiciones mencionadas anteriormente. También se utilizó un computador para verificar la segmentación y detección. Además se valida el sistema comparando con un sistema patrón de detección del rostro con una eficiencia que varía del 86% a 96% (Fernandez A. , Usamentiaga R. & Casado R., 2017).

**Tabla 2-3:** Detección de Asentamiento de la cabeza

# Conductor	# Pruebas Realizadas	# Aciertos	% Eficiencia	% Error
1	10	10	100 %	0 %
2	10	10	100 %	0 %
3	10	8	80 %	20 %
4	10	8	80 %	20 %
5	10	10	100 %	0 %
6	10	10	100 %	0 %
7	10	9	90 %	10 %
8	10	9	90 %	10 %
9	10	10	100 %	0 %
10	10	10	100 %	0 %
11	10	10	100 %	0 %
12	10	9	90 %	10 %
13	10	9	90 %	10 %
14	10	8	80 %	20 %
15	10	8	80 %	20 %
16	10	10	100 %	0 %
17	10	10	100 %	0 %
18	10	9	90 %	10 %
19	10	10	100 %	0 %
20	10	10	100 %	0 %
21	10	8	80 %	20 %
22	10	8	80 %	20 %
23	10	10	100 %	0 %
24	10	10	100 %	0 %
25	10	8	80 %	20 %
26	10	10	100 %	0 %
27	10	10	100 %	0 %
28	10	9	90 %	10 %
29	10	9	90 %	10 %
30	10	10	100 %	0 %
Total	300	279	2790%	210%
<b>Eficiencia en la Detección</b>			<b>93 %</b>	<b>7%</b>

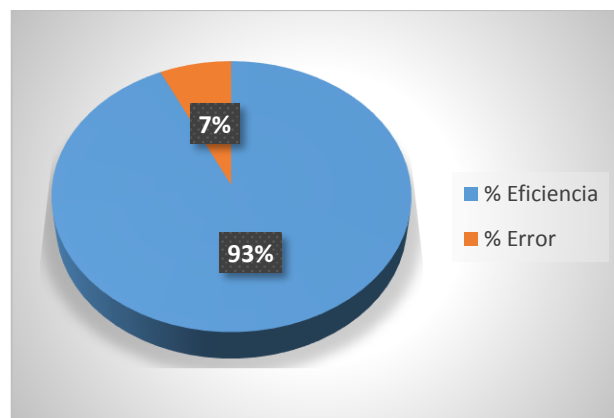
Realizado por: Bertha Arévalo, 2019

En la tabla 2-3 se exponen los resultados de la prueba, donde se observa que existió un total de 279 aciertos, mostrando así que el sistema posee un 93 % de eficiencia en la detección, con un error del 7%, por lo tanto en comparación con el sistema patrón se determina que el sistema es eficiente en su detección, los resultados se muestran en el grafico 3-3.



**Gráfico 3-3:** Numero de aciertos en la detección de asentamiento de cabeza  
Realizado por: Bertha Arévalo, 2019

El gráfico 4-3 muestra la eficiencia del sistema de alerta en la detección del asentamiento de cabeza.



**Gráfico 4-3:** Eficiencia en la detección de asentamiento de cabeza  
Realizado por: Bertha Arévalo, 2019

### 3.2.3 Pruebas detección del uso del teléfono celular

Para la detección del uso del celular, cada conductor utilizo el teléfono celular, todo se realizó bajo las condiciones mencionadas anteriormente. También se utilizó un computador para verificar la segmentación y detección. Además la validación del sistema se compara con un sistema patrón de detección de objetos con una eficiencia que varía del 86% a 96% (Fernandez A. , Usamentiaga R. & Casado R., 2017).

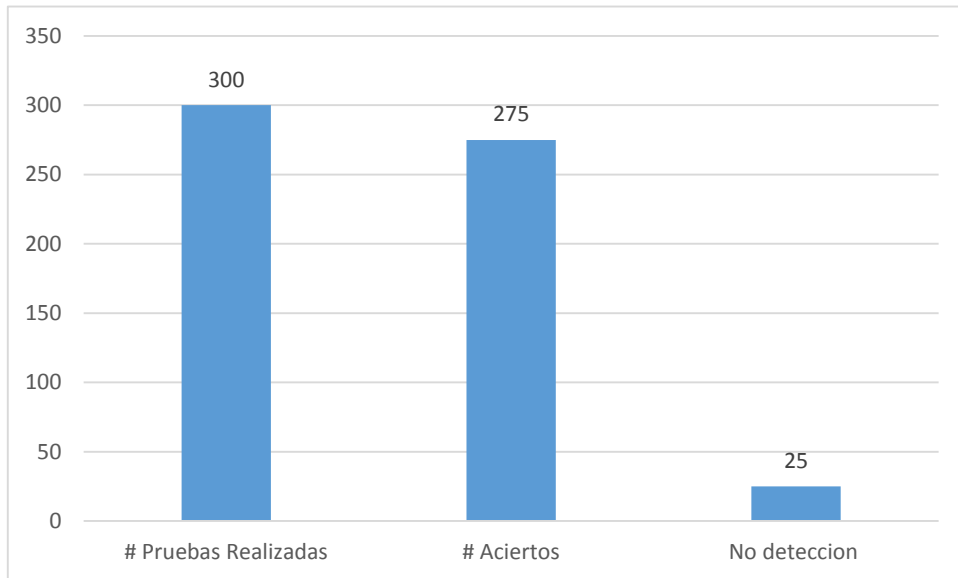
**Tabla 3-3:** Detección del uso del teléfono celular

# Conductor	# Pruebas Realizadas	# Aciertos	% Eficiencia	% Error
1	10	8	80 %	20 %
2	10	8	80 %	20 %
3	10	8	80 %	20 %
4	10	9	90 %	10 %
5	10	10	100 %	0 %
6	10	10	100 %	0 %
7	10	9	90 %	10 %
8	10	9	90 %	10 %
9	10	8	80 %	20 %
10	10	10	100 %	0 %
11	10	10	100 %	0 %
12	10	9	90 %	10 %
13	10	10	100 %	0 %
14	10	8	80 %	20 %
15	10	9	90 %	10 %
16	10	10	100 %	0 %
17	10	10	100 %	0 %
18	10	9	90 %	10 %
19	10	10	100 %	0 %
20	10	8	80 %	20 %
21	10	9	90 %	10 %
22	10	10	100 %	0 %
23	10	10	100 %	0 %
24	10	8	80 %	20 %
25	10	8	80 %	20 %
26	10	10	100 %	0 %
27	10	10	100 %	0 %
28	10	9	90 %	10 %
29	10	9	90 %	10 %
30	10	10	100 %	0 %
Total	300	275	2750%	250%
<b>Eficiencia en la Detección</b>			<b>92 %</b>	<b>8%</b>

Realizado por: Bertha Arévalo, 2019

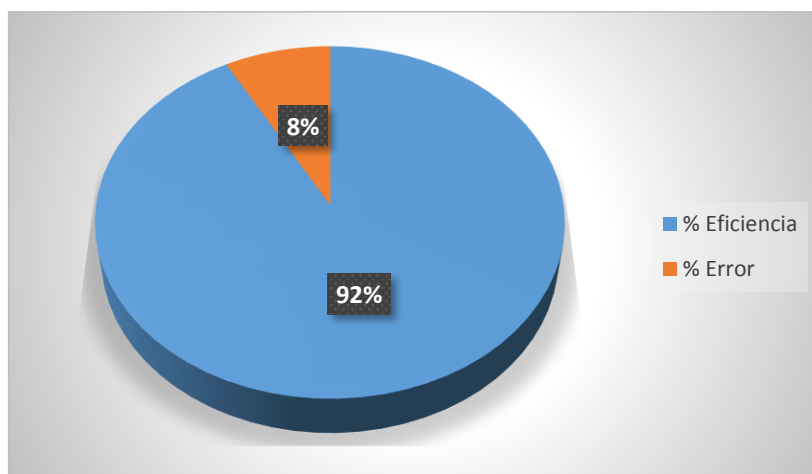


En la tabla 1-3 se exponen los resultados de la prueba, donde se observa que existió un total de 275 aciertos, mostrando así que el sistema posee un 92 % de eficiencia en la detección, con un error del 8%, por lo tanto en comparación con el sistema patrón se determina que el sistema es eficiente en su detección, los resultados se muestran en el grafico 5-3.



**Gráfico 5-3:** Numero de aciertos en la detección del uso del teléfono celular  
Realizado por: Bertha Arévalo, 2019

El gráfico 6-3 se muestra la eficiencia del sistema de alerta en la detección correcta del teléfono celular.



**Gráfico 6-3:** Eficiencia en la detección del teléfono celular  
Realizado por: Bertha Arévalo, 2019

### 3.2.4 Pruebas de detección con respecto a la iluminación

Se utilizó un multímetro con luxómetro de la marca AIIGSUN que se muestra en la figura 2-3 con el que se tomaron 5 muestras en diferentes horas del día con distintas condiciones climáticas, obteniéndose los datos a continuación representados que reflejan la no detección, falsos positivos y negativos.



**Figura 2-3:** Luxómetro de la marca AIIGSUN  
Realizado por: Bertha Arévalo, 2019

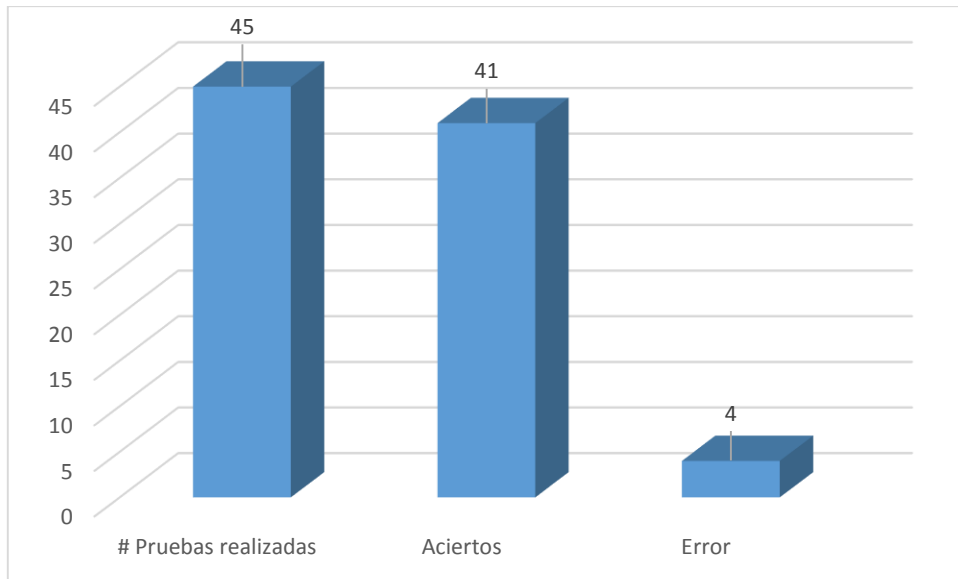
#### 3.2.4.1 Pruebas de iluminación para la detección de cierre de ojos y asentamiento de cabeza

En la tabla 4-3 se exponen los resultados de la prueba, donde se observa que existió un total de 41 aciertos con un porcentaje de error del 9%, esto es provocado por la falta de iluminación adecuada o porque la luz incide directamente sobre la cámara, los resultados son representados en el grafico 7-3.

**Tabla 4-3:** Pruebas de iluminación para la detección de cierre de ojos y asentamiento de cabeza

Hora / Tiempo	Iluminación natural (LUX)	# Pruebas realizadas	Aciertos	Error
7:00 – 9:00 soleado	< 20000 > 9000	5	5	0
9:00 – 14:00 soleado	< 40000 > 20000	5	5	0
14:00 – 17:00 soleado	< 20000 > 10000	5	5	0
7:00 – 9:00 parcialmente nublado	< 8000 > 5000	5	4	1
9:00 – 14:00 parcialmente nublado	< 10000 > 8000	5	5	0
14:00 – 17:00 parcialmente nublado	< 8000 > 6000	5	4	1
7:00 – 9:00 nublado	< 6000 > 4000	5	5	0
9:00 – 14:00 nublado	< 4000 > 2000	5	5	0
14:00 – 17:00 nublado	< 2000	5	3	2

Realizado por: Bertha Arévalo, 2019



**Gráfico 7-3:** Gráfico de resultados de las pruebas de iluminación  
Realizado por: Bertha Arévalo, 2019

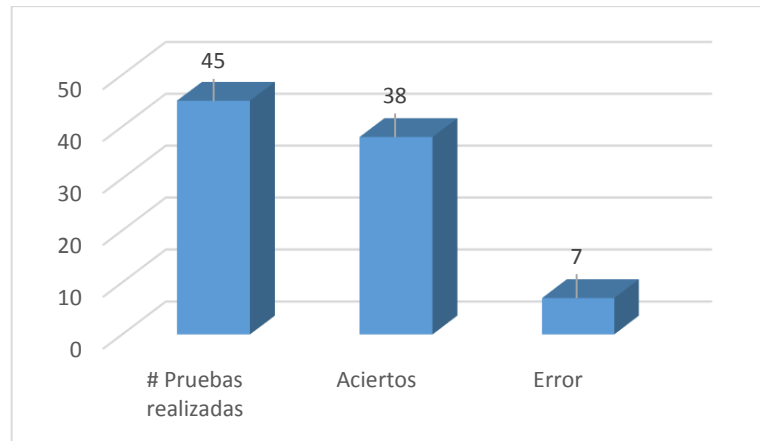
### 3.2.4.2 Pruebas de iluminación para la detección del teléfono celular

En la tabla 5-3 se exponen los resultados de la prueba, donde se observa que existió un total de 38 aciertos con un porcentaje de error de 16%, esto es provocado por la falta de iluminación adecuada o porque la luz incide directamente sobre la cámara, los resultados serán representados en el gráfico 8-3.

**Tabla 5-3:** Pruebas de iluminación en la detección del teléfono celular

Hora / Tiempo	Iluminación natural (LUX)	# Pruebas realizadas	Aciertos	Error
7:00 – 9:00 soleado	< 20000 > 9000	5	4	1
9:00 – 14:00 soleado	< 40000 > 20000	5	4	1
14:00 – 17:00 soleado	< 20000 > 10000	5	5	0
7:00 – 9:00 parcialmente nublado	< 8000 > 5000	5	4	1
9:00 – 14:00 parcialmente nublado	< 10000 > 8000	5	5	0
14:00 – 17:00 parcialmente nublado	< 8000 > 6000	5	5	0
7:00 – 9:00 nublado	< 6000 > 4000	5	4	1
9:00 – 14:00 nublado	< 4000 > 2000	5	5	0
14:00 – 17:00 nublado	< 2000	5	2	3

Realizado por: Bertha Arévalo, 2019



**Gráfico 8-3:** Gráfico de las pruebas de iluminación respecto al teléfono celular  
 Realizado por: Bertha Arévalo, 2019

### 3.2.5 Prueba de tiempo de reconocimiento

Para recolectar las pruebas para el tiempo de reconocimiento se utilizó la librería time, esta información se muestra en la consola de Python como se observa en la figura 3-3.

```

tiempo de reconocimiento 0.003097
tiempo de reconocimiento 0.003202
tiempo de reconocimiento 0.003017
tiempo de reconocimiento 0.003094
tiempo de reconocimiento 0.003086
tiempo de reconocimiento 0.003027
tiempo de reconocimiento 0.003039
tiempo de reconocimiento 0.010679
tiempo de reconocimiento 0.003194
tiempo de reconocimiento 0.003558
  
```

**Figura 3-3:** Datos del tiempo de reconocimiento mostrados en Python  
 Realizado por: Bertha Arévalo, 2019

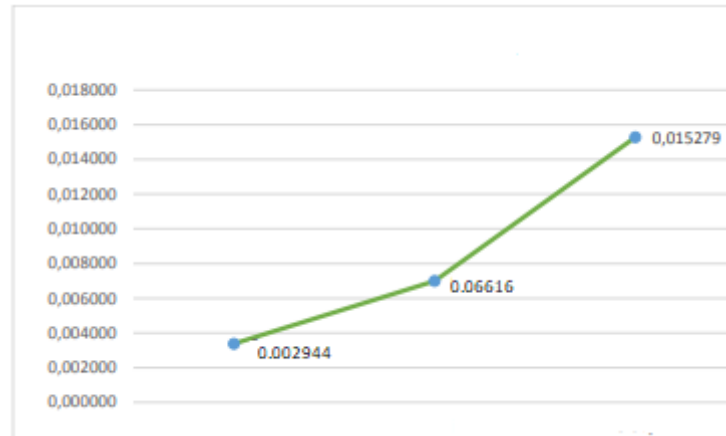
En la tabla se muestra los datos recolectados de los tiempos de reconocimiento del rostro, ojos y el teléfono celular.

**Tabla 6-3:** Datos obtenidos por el tiempo de reconocimiento de objetos.

# Pruebas	Tiempo (segundos)		
	Detección 1	Detección 2	Detección 3
1	0.003102	0.006565	0.01512
2	0.003233	0.006066	0.015623
3	0.003520	0.006401	0.014
4	0.003364	0.006894	0.010578
5	0.003097	0.006598	0.01154
6	0.003202	0.006581	0.02145
7	0.003130	0.007024	0.01528
8	0.003187	0.006565	0.015628
9	0.003358	0.006421	0.010398
10	0.003464	0.007041	0.012564
<b>Promedio</b>	0.002944	0.006616	0.015279

Realizado por: Bertha Arévalo, 2019

según los datos visualizados en la tabla 6-3 se determina que el tiempo de reconocimiento es de 3.47 ms y este aumenta a medida que detecta más objetos, esto se debe a que el *software* tarda en procesar las imágenes que aparecen en la escena, los resultados son mostrados en el gráfico 9-3.



**Gráfico 9-3:** Tiempo promedio del reconocimiento de rostro, ojos y teléfono celular  
**Realizado por:** Bertha Arévalo, 2019

### 3.3 Análisis de costos

En la tabla 7-3 se muestra el costo total del prototipo donde se determina que el sistema tiene un valor de 1.18% relacionado con el valor de un vehículo de trabajo, lo que demuestra que el prototipo implementado es de bajo costo. Cabe mencionar que los costos de un sistema de detección de somnolencia son muy elevados dentro del mercado, ya que existen solamente en algunos vehículos modernos.

**Tabla 7-3:** Costos del *Hardware*

Cantidad	Detalle	Valor Unitario	Valor Total
1	Kit Raspberry Pi 3 (Tarjeta Raspberry Pi 3 B+, carcasa, disipador y cargador)	120	\$ 120
1	Módulo de Cámara Raspberry Pi NoIRV2	60	\$ 60
1	Inversor de Voltaje	30	\$ 30
2	Buzzer	1.50	\$ 3.00
		<b>Total</b>	<b>\$ 213</b>

**Realizado por:** Bertha Arévalo, 2019

## CONCLUSIONES

- Se implementó un prototipo de sistema de alerta, usando visión artificial para disminuir accidentes de tránsito en el Ecuador, el cual consta de cuatro etapas: entrenamiento del clasificador para el teléfono celular, adquisición de imagen, procesamiento, detección y activación de la alarma.
- De las pruebas se determinó que el prototipo debe ser instalado sobre el tablero a una distancia de 55 cm desde la ventana izquierda y a 30 cm desde el parabrisas, además no debe existir objetos en la línea imaginaria entre la cámara y el rostro del conductor. A una distancia máxima de 60 cm con relación al conductor, concordando con lo establecido por la Norma IEC 62471.
- De las pruebas realizadas se determinó una eficiencia en la detección de somnolencia (estado de los ojos) de 95%, asentamiento de cabeza 93% y del teléfono celular es del 92%, evidenciándose un porcentaje de error menor al 10% ocasionado por la variación en la intensidad de la luz, cuando incide directamente sobre la cámara, ocasionando que la imagen no se pueda procesar correctamente.
- De las pruebas de iluminación se determinó que el sistema presenta más cantidad de errores en la detección cuando la iluminación del ambiente se mantiene por debajo de los 2000 luxes. Debido al decremento en la cantidad de luz normal y también porque la luz incide directamente sobre la cámara y por lo tanto se altera la detección del conductor.
- De las pruebas se determinó que el tiempo mínimo de procesamiento de imágenes que emplea el prototipo es de 3.38ms y este aumenta a medida que el algoritmo detecta más objetos. Por lo cual se concluye que aún el mayor tiempo de procesamiento no afecta el funcionamiento del prototipo manteniendo una visualización fluida de video.

## RECOMENDACIONES

- Ampliar el tiempo de pruebas del prototipo con la finalidad de identificar mejoras a su diseño.
- Estudiar la posibilidad de mejorar los algoritmos de procesamiento de imágenes con la finalidad de que tengan la capacidad de reconstruir y procesar imágenes captadas en ambientes de poca luz.
- Estudiar la posibilidad de incorporar un módulo electrónico basado en inteligencia artificial capaz de estabilizar, controlar y estacionar el vehículo cuando se detecte que su conductor presenta síntomas de somnolencia y distracción con la finalidad de precautelar su vida.
- Mejorar las etapas de procesamiento de imágenes mediante inteligencia artificial de tal manera que permita acelerar los tiempos de procesamiento.
- Según la Agencia Nacional de Tránsito existen otras formas de distracción relacionadas con el comer, beber, maquillarse, pantallas de video por lo tanto se recomienda mejorar el prototipo para incorporar procesos que permitan detectar este tipo de acciones.

## BIBLIOGRAFÍA

**AGENCIA NACIONAL DE TRANSITO (ANT).** *Estadísticas de transporte terrestre y seguridad vial.* [En línea] 15 de Septiembre de 2018. <https://www.ant.gob.ec/index.php/noticias/estadisticas>.

**BARRERO PARRA, E.** *Aceleración del algoritmo de Viola-Jones mediante rejillas de procesamiento masivamente paralelo en el plano focal.* 2015. pág. 5.

**BCNVISION.** *Sistemas de iluminación para aplicaciones de visión artificial.* [En línea] 28 de Diciembre de 2018. <http://www.bcnavision.es/blog-vision-artificial/iluminacion-vision-artificial2/>.

**CABANYE, Javier.** *Dormir y Soñar.* s.l. : EUNSA, 2015.

**CARRIÓN CHAMBA, Fernando Adrián.** *Desarrollo de un sistema de visión artificial para la detección de transeúntes con discapacidad y vehículos pequeños utilizando software libre.* 2017. págs. 29-33.

**CASTEJON, M. et al.** *Técnicas y algoritmos básicos de visión artificial.* [En línea] 2006. [Citado el: 02 de Enero de 2019.] <https://dialnet.unirioja.es/descarga/libro/338314.pdf>.

**CHARNIYA, N. N., & NAIR, V. R.** 2017. *Drunk driving and drowsiness detection.* 2017.

**ABC COMPONENTS.** *Buzzer.* [En línea] [http://www.farnell.com/datasheets/2032394.pdf?\\_ga=2.101101826.2090848696.15602272](http://www.farnell.com/datasheets/2032394.pdf?_ga=2.101101826.2090848696.15602272).

**CORKE, P.I.** *Robots, Vision y Control.* s.l. : Springer, 2011.

**DELGADO RODRÍGUEZ, M.** *Extracción automática de caras en imágenes captadas con móviles Android.* 2012. págs. 14, 15.

**ELECTRONILAB.COM.** *Cámara NoIR para Raspberry Pi v2 – 8 Megapíxeles.* [En línea] 8 de Febrero de 2019. <https://electronilab.co/tienda/camara-noir-raspberry-pi-v2-8-megapixeles/>.

**ESPINOZA OLGUÍN, E. & JORQUERA GUILLEN, P. I.** *Reconocimiento Facial.* [En línea] 2015. [http://opac.pucv.cl/pucv\\_txt/txt-1000/UCD1453\\_01.pdf](http://opac.pucv.cl/pucv_txt/txt-1000/UCD1453_01.pdf).

**GUEVARA, L. M., ECHEVERRY, J. D. & ARDILA, W.** *Faces Detection in Digital Images Using Cascade Classifiers.* 38, s.l. : Scientia et Technica, 2008, Vol. XIV, págs. 1-6.



**VIKRAM, K. & PADMAVATHI, S.** *Facial parts detection using Viola Jones algorithm..* 2017. 4th International Conference on Advanced Computing and Communication Systems (ICACCS). págs. 1-4.

**FERNANDEZ A. , USAMENTIAGA R. & CASADO R.** *Sistema Automatico para la deteccion de distraccion y somnolencia en conductores por medio de características visuales robustas.* España : Revista Iberoamericana de Automatica e Informatica Industrial, 2017. Vol. 14.

**FERNÁNDEZ MARTÍ, José Manuel.** *Desarrollo de una aplicación Android para la detección de señales de tráfico usando OpenCV* (Trabajo de Titulacion) (Ingenieria). 2015.

**FERRER, Patricia.** *La imagen digital.* [En línea] 30 de Diciembre de 2018. <https://es.scribd.com/document/343461168/La-Imagen-Digital>.

**FONTANELLAS, H. J.** *Manual de iluminación para cine y video.* [En línea] 2014. [Citado el: 29 de diciembre de 2018.] <https://ebookcentral.proquest.com/lib/epochsp/reader.action?docID=3226980&query=Iluminacion>.

**GÓMEZ RODRÍGUEZ, F. & DOMÍNGUEZ, M. J.** *Fundamentos de la Vision Artificial.* 2016.

**GUEVARA IPUSHIMA, K. & TORRES JARA, E.** *CALIDAD DE SUEÑO Y SOMNOLENCIA DIURNA EN CONDUCTORES* (Tesis Investigativo) . Lima, Peru : s.n., 2016. págs. 78, 79.

**JETBRAINS.COM.** *Pycharm The Python IDE for Professional Developers.* [En línea] 1 de Marzo de 2019. <https://www.jetbrains.com/pycharm/>.

**LEMKADDEM, A. et al.** *Multi-modal driver drowsiness detection: A feasibility study.* 2018.

**LÓPEZ MENA, D. M.** *Diseño e implementación de un sistema de Visión Artificial para determinar condiciones de fatiga en una persona mediante el índice Perclos, utilizando Open CV y una tarjeta Raspberry Pi 3* (Trabajo de Titulacion) (Ingenieria). 2017. págs. 30-35.

**MADUELL GARCÍA, E.** *Vision artificial.* 2016. pág. 18.

**MCI-Electronics.** *Camara Noir para raspberry Pi de 8 megapixeles.* [En línea] 5 de Febrero de 2019. <https://www.mcielectronics.cl/shop/product/camara-noir-para-raspberry-pi-de-8-megapixeles-21488>.

**MEDINA O., SÁNCHEZ N., et al.** *Alteraciones del sueño en los trastornos psiquiátricos*. 2015. Vol. 36, 4.

**MIRO, E.; CANO LOZANO, M.; & BUELA CASAL, G.** *Sueño y calidad de vida*. s.l. : Red Revista Colombiana de Psicología, 2009. pág. 17.

**MORDVINTSEV, A. & ABID K.** *Introducción a los tutoriales de OpenCV-Python*. [En línea] 25 de Febrero de 2019. [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_setup/py\\_intro/py\\_intro.html#intro](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_intro/py_intro.html#intro).

**SYED ZAIN, N., MUHAMMAD ADNAN, A., JUNAID, S & SYED DANIVAL.** *Introducción a la Raspberry* [En línea] 20 de 12 de 2018. <https://www.theengineeringprojects.com/2018/07/introduction-to-raspberry-pi-3-b-plus.html>.

**NAYLAMPMECHATRONICS.COM.** *Fuente de alimentación para Raspberry Pi 5VDC/2.5A*. [En línea] 20 de Marzo de 2019. <https://naylampmechatronics.com/fuentes-y-reguladores/342-fuente-de-alimentacion-para-raspberry-pi-5v-dc25a.html>.

**ORGANIZACION MUNDIAL DE LA SALUD (OMS).** *Uso del celular al volante: un problema creciente de distracción del conductor*. [En línea] 2015. [http://www.who.int/violence\\_injury\\_prevention/publications/road\\_traffic/distracted\\_driving\\_public/es](http://www.who.int/violence_injury_prevention/publications/road_traffic/distracted_driving_public/es).

**OpenCv.org.** *Introducción a OpenCv*. [En línea] 1 de Marzo de 2019. <https://docs.opencv.org/master/d1/dfb/intro.html>.

**OpenCV.org.** *OpenCV*. [En línea] 30 de Febrero de 2019. <https://opencv.org/>.

**Optalert.** *DriveTime | Optalert*. [En línea] 23 de Octubre de 2018. <http://drivetime.systems/optalert>.

**CARRILLO MORA, P., RAMÍREZ PERIS, J., & MAGAÑA VÁZQUEZ, K.** *Neurobiología del sueño y su importancia*. 2015.

**PÉREZ CARVAJAL, Rafael Jesús** *Mantenimiento del software (UF1894)*. s.l. : IC Editorial, 2014. pág. 11.

**Python.org.** *Python*. [En línea] 30 de Febrero de 2019. <https://www.python.org/>.

**Raspberry.org.** *raspberrypi Pi-3 Model B+*. [En línea] 1 de Febrero de 2019. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>.

**ROSALES MAYOR, E. & REY DE CASTRO MUJICA, J.** *Somnolencia: Qué es, qué la causa y cómo se mide.* [En línea] 10 de Noviembre de 2018. <http://www.scielo.org.pe/pdf/amp/v27n2/a10v27n2.pdf>.

**RS-online.com.** *Cámara Raspberry, Módulo de vídeo V2 PiNoir, Pack Raspberry Pi, 1 canal, CSI-2.* [En línea] 7 de Febrero de 2019. <https://es.rs-online.com/web/p/modulos-de-video/1115552/>.

**SADA, S. & SANZ DELGADO, J.** *Tratamiento de imágenes con ruido impulsivo mediante reglas difusas y algoritmos genéticos.* [En línea] 2015. <https://academica-e.unavarra.es/bitstream/handle/2454/19033/MemoriaTFG--SergioSada.pdf?sequence=1&isAllowed=y>.

**SÁNCHEZ CALLE, Ángel.** *Aplicaciones de la visión artificial y la biometría informática.* s.l. : Dykinson, 2005.

**SÁNCHEZ PÉREZ P.** *Distracciones al volante. PROCESO SELECTIVO PARA EL INGRESO EN LA ESPECIALIDAD DEL TRAFICO DEL CUERPO GENERAL ADMINISTRATIVO.* 2017, págs. 448,449.

**SÁNCHEZ PÉREZ, Perfecto.** *Distracciones al volante.* [En línea] 25 de Diciembre de 2018. [http://www.dgt.es/multimedia/educacion\\_vial/books/10/book.pdf](http://www.dgt.es/multimedia/educacion_vial/books/10/book.pdf).

**DACHSSOLUTIONS.** *Tipos y tecnicas de iluminacion.* [En línea] 28 de Diciembre de 2018. <https://dachssolutions.com/vision-artificial/tipos-de-iluminacion/>.

**ROSALES MAYOR, E., et al.** *Somnolence and sleep quality in medical students during hospital practices and holidays.* 2014, pág. 199.

**Sucar, E. L. and Gomez, G.** *Vision Computacional.* [En línea] 2005.

**Syed, Z. N.; Muhammad, A. A.; et al.** *Introduction to Raspberry Pi 3 B plus.* [En línea] 1 de Febrero de 2019. <https://www.theengineeringprojects.com/2018/07/introduction-to-raspberry-pi-3-b-plus.html>.

**TINAJERO, J. L.** *DISEÑO DE UN SISTEMA DE CONTROL VISUAL PARA NAVEGACIÓN DE INTERIORES USANDO FUSIÓN SENSORIAL PARA UNA PLATAFORMA MÓVIL DIFERENCIAL (Trabajo de Titulacion)(Maerstria).* Escuela Superior Politecnica de Chimborazo. 2018. pág. 30.

**CARRILLO MORA, P., BARAJAS MARTÍNEZ, K., et al.** *Trastornos del sueño: ¿qué son y cuáles son sus consecuencias?* 1, 2018, Vol. 61, pág. 17.

**Union de Mutuas, Mutua Colaboracion con la Seguridad.** *Fatiga y Sueño.* [En línea] 2017.  
<https://www.uniondemutuas.es/wp-content/uploads/2017/11/Distracciones-al-volante-SUEN%CC%83O-CAS.pdf>.

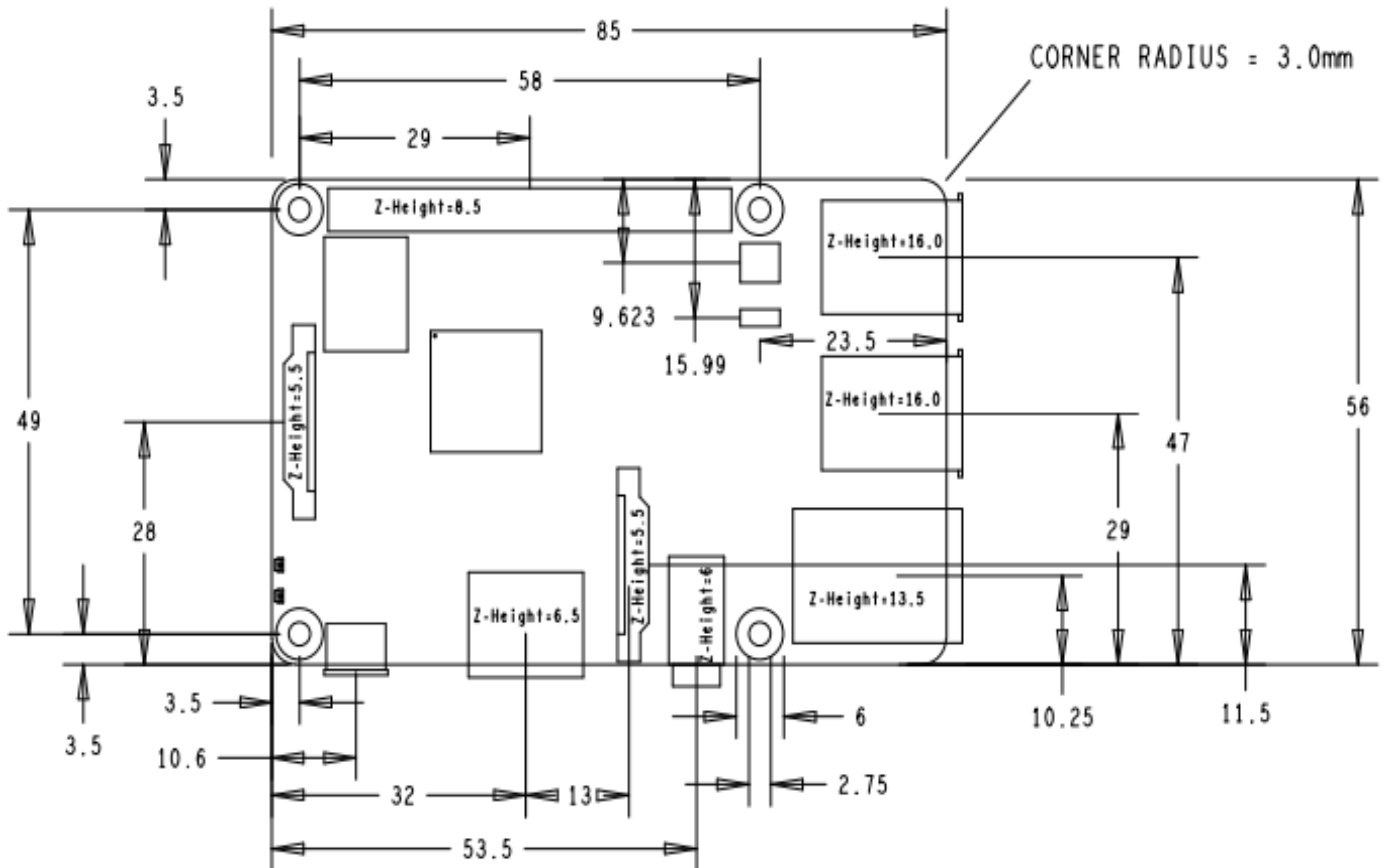
**VÉLEZ SERRANO, J.F., et al.** *Visión por computador.* 2003. págs. 87,88.

**Xataka.com.** *Raspberry Pi 3 Model B+, análisis: más potencia y mejor WiFi para un miniPC que sigue asombrando.* [En línea] 30 de Enero de 2019. .  
<https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando..>

**ZENON CUCHO, M. FRERI ORIHUELA, Q., et al.** *Microcontroladores.* 2013. págs. 3 - 10.

## ANEXOS

### Anexo A: Especificaciones físicas de uso de la Raspberry Pi 3 B



### Especificaciones Técnicas

<b>Procesador</b>	Broadcom BCM2837B0, Cortex-A53 (armv8) 64-bit SoC @ 1.4 GHz
<b>Memoria</b>	1GB LPDDR2 SDRAM
<b>Storage</b>	MicroSD
<b>Conectividad</b>	2.4 GHz / 5 GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet over USB 2.0 (300 Mbps de máximo teórico)
<b>Puertos</b>	GPIO 40 pines HDMI 4 x USB 2.0 CSI (cámara Raspberry Pi) DSI (pantalla táctil) Toma auriculares / vídeo compuesto Micro SD

**Alimentación** 5V/2.5<sup>a</sup> DC vía conector micro USB  
5V DC vía GPIO header  
Power-over-Ethernet (poe)

## **Advertencia**

- Este producto solo debe conectarse a una fuente de alimentación externa con una capacidad nominal de 5 V / 2,5 A CC. Cualquier fuente de alimentación externa utilizada con el Raspberry Pi 3 Modelo B + deberá cumplir con las regulaciones y normas pertinentes aplicables en el país de uso previsto.
- Este producto debe operarse en un ambiente bien ventilado y, si se usa dentro de un estuche, el estuche no debe estar cubierto.
- Mientras esté en uso, este producto debe colocarse sobre una superficie estable, plana, no conductora y no debe ser contactado por elementos conductores.
- La conexión de dispositivos incompatibles a la conexión GPIO puede afectar el cumplimiento, dañar la unidad e invalidar la garantía.
- Todos los periféricos utilizados con este producto deben cumplir con las normas pertinentes para el país de uso y estar marcados de manera correspondiente para garantizar que se cumplan los requisitos de seguridad y rendimiento. Estos artículos incluyen, entre otros, teclados, monitores y ratones cuando se usan junto con la Raspberry Pi.
- Los cables y conectores de todos los periféricos utilizados con este producto deben tener un aislamiento adecuado para que se cumplan los requisitos de seguridad pertinentes.

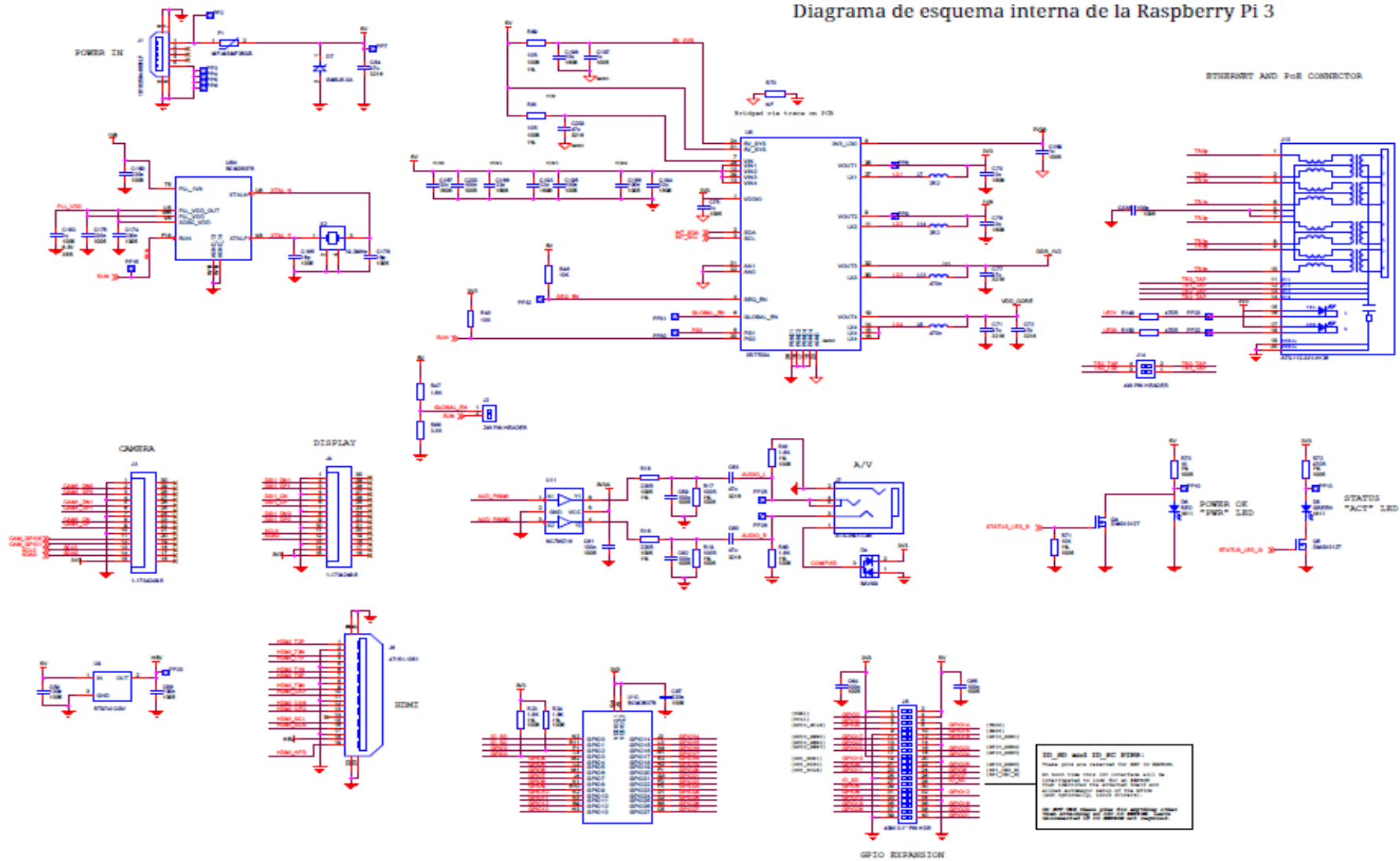
## **Instrucciones de seguridad**

***Para evitar el mal funcionamiento o daño de este producto, tenga en cuenta lo siguiente:***

- No lo exponga al agua ni a la humedad, ni lo coloque sobre una superficie conductora mientras esté en funcionamiento.
- No exponer al calor de ninguna fuente; La Raspberry Pi 3 Modelo B está diseñada para un funcionamiento confiable a temperaturas ambiente normales.
- Tenga cuidado al manipularlo para evitar daños mecánicos o eléctricos en la placa de circuito impreso y los conectores.
- Mientras esté encendido, evite manipular la placa de circuito impreso o solo por los bordes para minimizar el riesgo de daños por descarga electrostática.

Anexo B: Diagrama de esquema interna de la Raspberry Pi 3 B

Diagrama de esquema interna de la Raspberry Pi 3



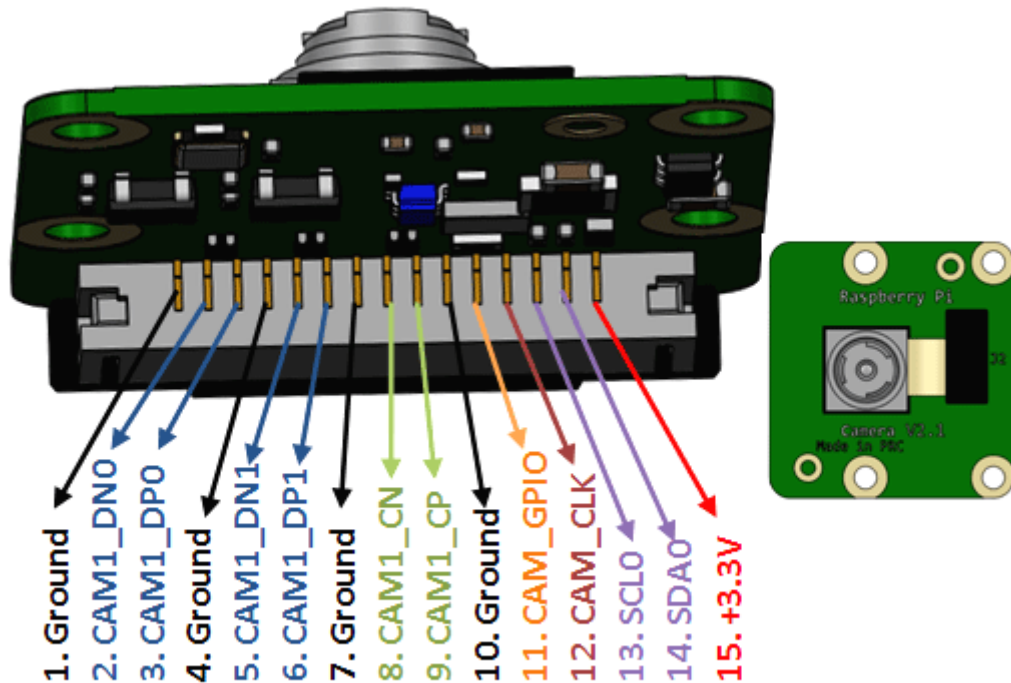
Anexo C: Datasheet del GPIO de las Raspberry Pi 3 B



+ 3,3 V	1	2	+ 5 V
(SDA) GPIO 2	3	4	+ 5 V
(SCL) GPIO 3	5	6	GND
(GPCLK0) GPIO 4	7	8	GPIO 14 (TXD)
GND	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
+ 3,3 V	17	18	GPIO 24
(MOSI) GPIO 10	19	20	GND
(MISO) GPIO 9	21	22	GPIO 25
(SCLK) GPIO 11	23	24	GPIO 8 (CE0)
GND	25	26	GPIO 7 (CE1)
ID_SD	27	28	ID_SC
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
GND	39	40	GPIO 21



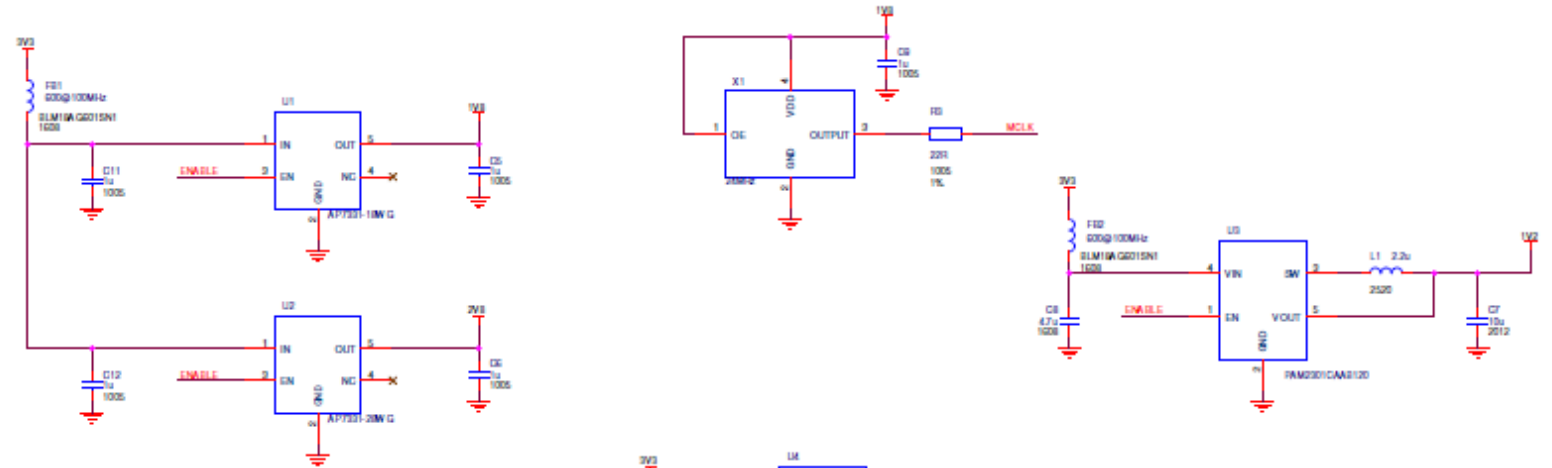
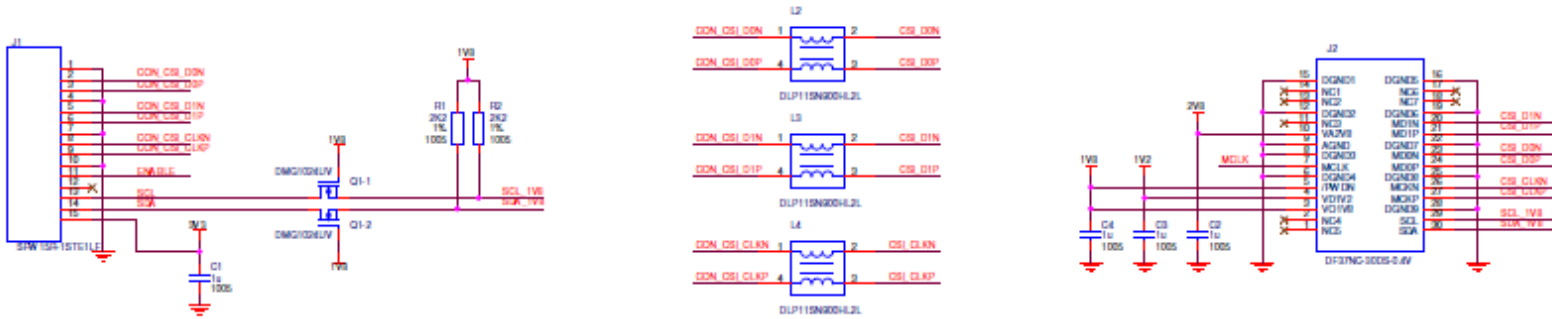
## Anexo D: Datasheet de los pines de la cámara PiNoIr V2.1




### Especificación Técnica

<b>Dimensión</b>	20 x 25 x 9 mm
<b>Resolución Nativa</b>	8 Megapíxeles
<b>Resolución Imágenes Estáticas</b>	3280 x 2464 pixeles
<b>Resolución de video</b>	1080p30, 720p60 y 640x480p60/90.
<b>Tipo de sensor</b>	Omnivisión
<b>Tamaño Óptico</b>	1/4 "
<b>Voltaje de salida</b>	3.3 V
<b>Compatibilidad</b>	A, B y B+ de Raspberry
<b>Compatibilidad sensor de la cámara</b>	Admite la conexión de infrarrojos LED flash de relleno y/o LED.

Anexo E: Diagrama de esquema interna de la cámara PiNoIR V2.1



 **Raspberry Pi** © Raspberry Pi 2018  
www.raspberrypi.org

Title: Raspberry Pi Camera V2.1		Drawn by: Mike Strass	
Rev: A3	Ref: RPI-CAM-V2_1	Rev: 2.1	
Date: Sunday, April 23, 2018	Sheet: 1 of 1		

## Anexo F: Líneas de Código en Python del sistema de alerta

```
# import the necessary packages
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
import RPi.GPIO as GPIO
import numpy as np
import sys, getopt

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

actuador_dormido = 12
actuador_movil = 16

GPIO.setup(actuador_dormido, GPIO.OUT)
GPIO.output(actuador_dormido, 0)

GPIO.setup(16, GPIO.OUT)
GPIO.output(actuador_movil, 0)

args, video_src = getopt.getopt(sys.argv[1:], '', ['cascade=',
'nested-cascade='])
args = dict(args)

cascade_fn = args.get('--cascade',
"/usr/local/share/OpenCV/haarcascades/haarcascade_frontalface_de
fault.xml")
nested_fn = args.get('--nested-cascade',
"/usr/local/share/OpenCV/haarcascades/haarcascade_eye.xml")
movil_fn = args.get('--movil-cascade',
"/usr/local/share/OpenCV/haarcascades/my_cascade_movil_5.xml")
face_cascade = cv2.CascadeClassifier(cascade_fn)
eye_cascade = cv2.CascadeClassifier(nested_fn)
movil_cascade = cv2.CascadeClassifier(movil_fn)

tiempo_rostro = 0
tiempo_movil = 0
tiempo_ojos = 0

# initialize the camera and grab a reference to the raw camera
capture
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 30
rawCapture = PiRGBArray(camera, size=(640, 480))

# allow the camera to warmup
time.sleep(0.1)

# capture frames from the camera
for frame in camera.capture_continuous(rawCapture, format="bgr",
use_video_port=True):
    # grab the raw NumPy array representing the image, then
```

```

initialize the timestamp
# and occupied/unoccupied text
image = frame.array
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
equ = cv2.equalizeHist(gray, 5)
##equ1 = cv2.equalizeHist(gray,25)
faces = face_cascade.detectMultiScale(equ, 1.8, 5)
if (len(faces) == 0):
    tiempo_rostro += 1
    if tiempo_rostro >= 7:
        GPIO.output(actuador_dormido, 1)
for (x, y, w, h) in faces:
    pos_eyes = 0
    tiempo_rostro = 0
    #### Graficar rectangulo alrededor de la zona de los
ojos
    cv2.rectangle(image, (x + int(0.13 * w), y + int(0.13 *
h)), (x + w - int(0.13 * w), y + h - int(0.45 * h)),
        (255, 0, 0), 4)
    ##for(x,y,w,h) in faces:
    #### Extraer zona de los ojos de imagen original
    roi_img = image[y + int(0.13 * h):y + h - int(0.45 * h),
x + int(0.13 * w):x + w - int(0.13 * w)]
    roi_gray = gray[y + int(0.13 * h):y + h - int(0.45 * h),
x + int(0.13 * w):x + w - int(0.13 * w)]
    #### Extraer zona de los ojos de imagen ecualizada
    roi_eq = equ[y + int(0.13 * h):y + h - int(0.45 * h), x
+ int(0.13 * w):x + w - int(0.13 * w)]
    eyes = eye_cascade.detectMultiScale(roi_eq, 1.1, 1)
if len(eyes) == 2:
    j = 0
    lado_der = None
    lado_izq = None
    for (ex, ey, ew, eh) in eyes:
        lado = None
        pixeles_b = []
        cv2.rectangle(roi_img, (ex + int(0 * ew), ey + int(0.15
* eh)),
            (ex + ew - int(0 * ew), ey + eh - int(0.15
* eh)), (0, 255, 0), 1)
        e_img = roi_img[ey + int(0.20 * eh):ey + eh - int(0.20 *
eh), ex + int(0.13 * ew):ex + ew - int(0.13 * ew)]
        e_gray = cv2.cvtColor(e_img, cv2.COLOR_BGR2GRAY)
        erb = cv2.equalizeHist(e_gray, e_gray)
        GPIO.output(actuador_dormido, 0)
        tiempo_ojos = 0
else:
    if len(eyes) < 2:
        tiempo_ojos += 1
        if tiempo_ojos >= 5:
            GPIO.output(actuador_dormido, 1)
    movil = movil_cascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=400, minSize=(72, 72), maxSize=(100, 100),

flags=cv2.CASCADE_SCALE_IMAGE)
if (len(movil) == 0):

```

```

        GPIO.output(actuador_movil, 0)
        tiempo_movil = 0
    else:
        tiempo_movil += 1
        if tiempo_movil >= 1:
            GPIO.output(16, 1)
            print("movil")
print(len(movil))
# movil = movil_cascade.detectMultiScale(equ,1.9,2,15)
for (x, y, w, h) in movil:
    cv2.rectangle(image, (x, y), (x + w, y + h), (10, 54, 200),
2)
    # show the frame
cv2.imshow("Rostro", image)
print(str(tiempo_ojos) + " " + str(tiempo_movil) + " " +
str(tiempo_rostro))
key = cv2.waitKey(1) & 0xFF

# clear the stream in preparation for the next frame
rawCapture.truncate(0)

# if the `q` key was pressed, break from the loop
if key == ord("q"):
    break
cv2.destroyAllWindows()

```