

**UNIVERSIDAD TÉCNICA DEL NORTE**



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

**DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID PARA LA DETECCIÓN Y  
ALERTA DE SOMNOLENCIA NOCTURNA EN TIEMPO REAL, MEDIANTE  
TÉCNICAS DE VISIÓN ARTIFICIAL**

Trabajo de grado presentado ante la Ilustre Universidad Técnica del Norte previo a la obtención del título de Ingeniera en Sistemas Computacionales.

Autora:

Adriana Lourdes Díaz Chimbo

Director:

MSc. Erick Patricio Herrera Granda

IBARRA – ECUADOR

ENERO 2020



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

**1. IDENTIFICACIÓN DE LA OBRA**

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

<b>DATOS DE CONTACTO</b>			
<b>CÉDULA DE IDENTIDAD:</b>	100396580-1		
<b>APELLIDOS Y NOMBRES:</b>	DÍAZ CHIMBO ADRIANA LOURDES		
<b>DIRECCIÓN:</b>	SANTO DOMINGO DE SAN ANTONIO DE IBARRA CALLE SIMÓN BOLIVAR Y ATAHUALPA		
<b>EMAIL:</b>	aldiazc@utn.edu.ec		
<b>TELÉFONO FIJO:</b>	062932775	<b>TELÉFONO MÓVIL:</b>	0982570309

<b>DATOS DE LA OBRA</b>	
<b>TÍTULO:</b>	DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID PARA LA DETECCIÓN Y ALERTA DE SOMNOLENCIA NOCTURNA EN TIEMPO REAL, MEDIANTE EL USO DE TÉCNICAS DE VISIÓN ARTIFICIAL
<b>AUTOR:</b>	DÍAZ CHIMBO ADRIANA LOURDES
<b>FECHA</b>	<b>27 enero del 2020</b>
<b>PROGRAMA</b>	<input checked="" type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSGRADO</b>
<b>TÍTULO POR EL QUE OPTA:</b>	INGENIERA EN SISTEMAS COMPUTACIONALES
<b>DIRECTOR:</b>	MSc. ERICK PATRICIO HERRERA GRANDA
<b>ASESOR:</b>	PHD. IVAN DANILO GARCÍA SANTILLAN
<b>ASESOR:</b>	MSc. MARCO REMIGIO PUSDA CHULDE

## 2. CONSTANCIAS

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

Ibarra a los 27 días del mes de enero del 2020



.....

Nombre: Adriana Lourdes Díaz Chimbo

Cédula: 1003965801

# CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE



UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS  
INGENIERÍA EN SISTEMAS COMPUTACIONALES

## CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, **DÍAZ CHIMBO ADRIANA LOURDES**, con cédula de identidad Nro. **1003965801**, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autora del proyecto de grado denominado: **“DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID PARA LA DETECCIÓN Y ALERTA DE SOMNOLENCIA NOCTURNA EN TIEMPO REAL, MEDIANTE EL USO DE TÉCNICAS DE VISIÓN ARTIFICIAL.”** que ha sido desarrollado para optar por el título de Ingeniera en Sistemas Computacionales, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autora me reservo los derechos morales de la obra antes citada.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

A handwritten signature in blue ink, appearing to read 'Adriana', is written over a horizontal dotted line.

Nombre: Adriana Lourdes Díaz Chimbo

Cédula: 1003965801

Fecha: 27 enero del 2020

## CERTIFICACIÓN DEL DIRECTOR



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

### CERTIFICACIÓN DEL DIRECTOR

Por medio del presente yo **MSc. ERICK PATRICIO HERRERA GRANDA**, certifico que la estudiante: **Adriana Lourdes Díaz Chimbo**, portadora de la cédula de identidad Nro. **100396580-1**, ha realizado en su totalidad el desarrollo del proyecto de grado "**DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID PARA LA DETECCIÓN Y ALERTA DE SOMNOLENCIA NOCTURNA EN TIEMPO REAL, MEDIANTE EL USO DE TÉCNICAS DE VISIÓN ARTIFICIAL.**", previo a la obtención del título de Ingeniera en Sistemas Computacionales y además, considero que el presente trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que se designe.

Es todo en cuanto puedo certificar en honor a la verdad.

Atentamente:

A handwritten signature in blue ink, appearing to be 'Erick Patricio Herrera Granda', written over a dashed horizontal line.

**MSc. Erick Patricio Herrera Granda**  
**DIRECTOR DE TESIS**

Las raíces de la educación son amargas, pero la fruta es dulce.

**Aristóteles.**

## **Dedicatorias**

El presente trabajo de titulación va principalmente dedicado **a Dios, a mi Papito y a mi Mamita, a mi Esposo y toda mi Familia**, quienes fueron la principal motivación durante el transcurso de la Carrera Universitaria, por la confianza que depositaron en mí y las palabras de aliento que nunca faltaban en el día a día, un gracias Infinito.

**Adriana Díaz**

## Agradecimientos

A **Dios y a la Virgencita**, por regalarme tantas bendiciones y ser siempre mis compañeros que guían mi camino.

A mis padres, **Severo Díaz** y **Delia Chimbo**, por darme la vida, por inculcarme tantos valores, por ser el pilar fundamental de mi formación académica, por brindarme tanto amor y sobre todo por enseñarme a luchar por mis sueños y no dejarme vencer fácilmente, los quiero muchísimo y gracias por todo.

A mi esposo, **Cristian Díaz**, por ser la persona a la cual amo inmensamente y a la vez también darle las gracias por siempre estar a mi lado dándome fuerza y aliento para seguir adelante, por siempre luchar juntos para cumplir nuestros sueños y finalmente decirle que **¡VAMOS POR MÁS MI AMOR! LE AMO.**

A mis Hermanas (Martica y Yolandita) y Hermanos (Fredy, Richy y Christian), por todo lo compartido y por ser unos excelentes hermanos un gracias inmenso y de igual forma un gracias por depositar toda la confianza en mí, los quiero mucho.

A mi tutor de tesis, **MsC. Erick Herrera** mi más profundo y sincero agradecimiento, por compartir sus conocimientos día a día durante todo el transcurso del desarrollo de la tesis y que gracias a su experticia me ha guiado para realizar de la mejor forma mi trabajo de Titulación y de esa manera he llegado a finalizarlo exitosamente.

A mis asesores de tesis **PhD. Ivan Garcia** y **MsC. Marco Pusedá** por apoyarme con su experiencia para que el presente trabajo salga de la mejor forma.

A mis amigos (**Josué, Kevin, Bryan, Cristo, Mary y Nelson**) por ser compañeros de esta interminable travesía y ser constructores de grandes momentos que quedarán impregnados en nuestros corazones.

Finalmente, un agradecimiento especial a la **Universidad Técnica del Norte**, profesionales y demás personas que han sido parte de esta formación académica por permitir que este sueño se haga realidad, en especial a los docentes de la Carrera de Ingeniería en Sistemas Computacionales por día a día compartir sus conocimientos.

**Adriana Diaz**

# Tabla de Contenidos

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	4
CERTIFICACIÓN DEL DIRECTOR .....	5
Dedicatorias .....	6
Agradecimientos .....	7
Tabla de Contenidos.....	8
Índice De Figuras .....	11
Índice de Tablas .....	12
RESUMEN.....	14
ABSTRACT .....	15
INTRODUCCIÓN .....	16
Antecedentes .....	16
Situación Actual.....	17
Prospectiva.....	18
Problema .....	18
Objetivos.....	18
Objetivo General.....	18
Objetivos Específicos .....	18
Alcance .....	19
Justificación.....	19
Objetivo número 3 “Salud y Bienestar” .....	20
Objetivo número 9 “Innovación e Infraestructura” .....	20
Justificación Social:.....	20
Justificación Tecnológica: .....	20
Justificación Económica: .....	20
Justificación Metodológica: .....	20
Contexto .....	21
CAPÍTULO 1 .....	1
Marco Teórico.....	1
1.1. Relación somnolencia-accidentes de tránsito .....	1
1.1.1. La somnolencia.....	1
1.1.2. Accidentes de tránsito .....	2
1.1.3. Tipos de accidentes de tránsito .....	3
1.1.4. Los accidentes de tránsito a nivel mundial.....	3
1.1.5. La somnolencia y los accidentes de tránsito .....	4
1.2. Inteligencia Artificial .....	5

1.2.1.	Visión Artificial.....	5
1.2.2.	Técnicas de detección de objetos .....	5
1.2.3.	Etapas de un sistema de visión Artificial .....	9
1.2.4.	Herramientas de un sistema de visión artificial en Android.....	9
1.3.	Sistemas de detección de somnolencia .....	10
1.3.1.	Análisis de patrones.....	11
1.3.2.	Análisis de señales fisiológicas y bioeléctricas .....	12
1.3.3.	Análisis de los ojos y expresiones faciales .....	13
1.4.	Sistemas para iluminación diurna.....	14
1.5.	Sistemas para iluminación nocturna .....	15
1.6.	Smartphone.....	16
1.7.	Sistema Operativo Móvil .....	17
1.7.1.	Sistema Operativo Móvil más usado.....	17
1.8.	Sistema Operativo Android.....	18
1.8.1.	Arquitectura de Android.....	19
1.8.2.	Principales Características de Android.....	21
CAPÍTULO 2.....		22
Desarrollo de la Propuesta .....		22
2.1	Introducción - Metodología XP .....	22
2.2	Fase De Planeación.....	22
2.2.1	Planificación del Proyecto.....	22
2.2.2	Participantes del Proyecto .....	24
2.2.3	Plan de Fases .....	24
2.3	Fase De Diseño.....	30
2.3.1	Arquitectura del sistema.....	30
2.3.2	Hardware y Software .....	30
2.3.3	Diagrama Work Break Down Structure para la realización del proyecto .....	31
2.3.4	Diagramas por bloques del proyecto.....	32
2.4	Fase de Codificación e Implementación.....	35
2.4.1	Configuración del Entorno Android para la creación de la aplicación móvil. ....	35
2.4.2	Obtención de imágenes a través de la cámara frontal del smartphone .....	36
2.4.3	Tratamiento de imágenes .....	36
2.4.4	Detección de Rostro .....	38
2.4.5	Detección de Ojos.....	43
2.4.6	Detección de pupila por medio de emparejamiento de plantillas .....	51
2.4.7	Detección de somnolencia.....	56
2.4.8	Emisión de alarma.....	57
2.4.9	Solución al problema de poca luminosidad en conducción nocturna .....	58

2.5	Fase de Pruebas .....	59
2.5.1	Resultados de Pruebas de Detección de rostros.....	61
2.5.2	Resultados de Pruebas de detección de Ojos.....	62
CAPÍTULO 3.....		64
Validación e Interpretación de Resultados.....		64
3.1	Muestreo.....	64
3.2	Gráfico de regresión .....	66
3.3	Matriz de Confusión de Rostros.....	67
3.4	Matriz Histograma de Densidad Correlación en Rostros.....	68
3.5	Resultados a partir de la Matriz de Densidad Correlación .....	69
3.6	Cálculo de Métricas de Desempeño en la Matriz de Rostros .....	72
3.6.1	Cálculo de la Precisión .....	72
3.6.2	Cálculo de la Exactitud.....	72
3.6.3	Cálculo de la Especificidad.....	73
3.6.4	Cálculo de la Medida Recall .....	73
3.6.5	Cálculo de la Medida F1.....	73
3.7	Matriz de Confusión de Ojos .....	76
3.8	Matriz de Densidad Correlación.....	77
3.9	Resultados a partir de la Matriz de Densidad Correlación de los Ojos .....	78
3.10	Cálculo de Métricas de Desempeño en la Matriz de Rostros .....	81
3.11	Curva ROC.....	83
3.12	Encuesta del prototipo de la aplicación móvil.....	84
3.13	Análisis e interpretación de resultados de la encuesta.....	96
CONCLUSIONES.....		100
RECOMENDACIONES .....		101
REFERENCIAS .....		102

# Índice De Figuras

Figura 1 Prototipo de la solución .....	19
Figura 2 Persona con estado de somnolencia.....	2
Figura 3 Accidente de Tránsito .....	2
Figura 4 Accidentes de Tránsito a Nivel Mundial.....	4
Figura 5: Algoritmo de Viola Jones.....	6
Figura 6: Histogramas de Gradientes Orientados .....	6
Figura 7 Aprendizaje Profundo.....	7
Figura 8 Cuadros delimitadores de YOLO.....	8
Figura 9 Resultado final de YOLO.....	8
Figura 10 Métodos de detección de somnolencia .....	11
Figura 11 Patrones líneas de carretera .....	12
Figura 12 Persona realizándose un electromiograma .....	12
Figura 13 Análisis del rostro.....	13
Figura 14 Algoritmo diurno .....	15
Figura 15 Algoritmo nocturno .....	16
Figura 16 Smartphone .....	17
Figura 17 Estadísticas de SO móviles más usados.....	18
Figura 18 Arquitectura de Android.....	19
Figura 19 Arquitectura del Sistema .....	30
Figura 20 Diagrama Work Break Down Structure para la realización del proyecto.....	32
Figura 21 Instalación de Android y de librerías para el desarrollo de la aplicación móvil.....	33
Figura 22 Obtención de la imagen .....	33
Figura 23 Tratamiento de imágenes.....	34
Figura 24 Detectar Rostros .....	34
Figura 25 Detectar Ojos .....	35
Figura 26 Detección de pupila.....	35
Figura 27 Imagen RGBA (izquierda) e Imagen transformada a escala a grises (derecha). .....	37
Figura 28 Proceso de HaarCascade .....	38
Figura 29 Imagen integral .....	38
Figura 30 Cálculo de la Imagen Integral.....	39
Figura 31 Características de Haar.....	40
Figura 32 Detección de Rostros en la aplicación móvil .....	42
Figura 33 Detección de Área de Ojo Derecho .....	45
Figura 34 Detección de Área de Ojo Izquierdo.....	46
Figura 35 Detección de Ojos.....	47
Figura 36 Detección de Ojos.....	51
Figura 37 Procesamiento óptico de imágenes. (a) imagen original, con líneas de barrido, tipo imagen de televisión; (b) transformada de Fourier del objeto; (c) transformada de Fourier modificada, después de filtrar y (d) imagen procesada, sin las líneas de barrido .....	53
Figura 38 Área de Interés y Solución .....	54
Figura 39 Plantilla de camión.....	55
Figura 40 ubicación del objetivo encontrado(izquierda) - determinación del valor máximo de intensidad(derecha) .....	56
Figura 41 Aplicación Móvil - Somnolencia.....	56
Figura 42 Iluminación de los leds del smartphone.....	59
Figura 43 Personas sin estado y con estado de somnolencia .....	60
Figura 44 Prototipo realizados para efectuar pruebas .....	60
Figura 45 Resultados de pruebas en consola .....	60
Figura 46 Resultado del cálculo del Muestreo.....	65
Figura 47 Gráfica de Regresión .....	66
Figura 48 Matriz Densidad Correlación .....	69
Figura 49 Si Rostros Si Detecta .....	69
Figura 50 Si Rostros No Detecta.....	70

Figura 51 No Rostros Si Detecta.....	70
Figura 52 No Rostros No Detecta .....	71
Figura 53 Matriz de Histograma Densidad Correlación en la detección de Ojos .....	78
Figura 54 Si Ojos Si Detecta .....	78
Figura 55 Si Ojos No Detecta.....	79
Figura 56 No Ojos Si Detecta.....	79
Figura 57 No Ojos No Detecta .....	79
Figura 58 Curva ROC .....	84
Figura 59 Encuesta a 33 Usuarios .....	85
Figura 60 Pregunta 1 - Informativa.....	87
Figura 61 Pregunta 2 – Informativa.....	87
Figura 62 Pregunta 1 – Precisión.....	88
Figura 63 Pregunta 2 - Precisión.....	88
Figura 64 Pregunta 3 - Precisión.....	89
Figura 65 Pregunta 4 - Precisión.....	90
Figura 66 Pregunta 5 - Precisión.....	90
Figura 67 Pregunta 1 - Solución al Problema.....	91
Figura 68 Pregunta 2 - Solución al Problema.....	91
Figura 69 Pregunta 3 - Solución al Problema.....	92
Figura 70 Pregunta 4 - Solución al Problema.....	93
Figura 71 Pregunta 5 - Solución al Problema.....	93
Figura 72 Pregunta 1 - Impacto.....	94
Figura 73 Pregunta 2 – Impacto.....	94
Figura 74 Pregunta 3 - Impacto.....	95
Figura 75 Pregunta 4 - Impacto.....	96
Figura 76 Pregunta 5 - Impacto.....	96
Figura 77 Análisis Factorial Combinatorio .....	97

## Índice de Tablas

TABLA 2.1 PLANIFICACIÓN DEL PROYECTO.....	22
TABLA 2.2 PLAN DE FASES.....	24
TABLA 2.3 HISTORIA DE USUARIO 1 .....	25
TABLA 2.4 HISTORIA DE USUARIO 2 .....	26
TABLA 2.5 HISTORIA DE USUARIO 3 .....	26
TABLA 2.6 HISTORIA DE USUARIO 4 .....	27
TABLA 2.7 HISTORIA DE USUARIO 5 .....	28
TABLA 2.8 HISTORIA DE USUARIO 6 .....	28
TABLA 2.9 HISTORIA DE USUARIO 7 .....	29
TABLA 2.10 HISTORIA DE USUARIO 8 .....	29
TABLA 2.11 HARDWARE Y SOFTWARE PARA EL PROYECTO .....	31
TABLA 2.12 CONVERSIÓN A ESCALA DE GRISES .....	37
TABLA 2.13 DETECCIÓN DEL ROSTRO .....	41
TABLA 2.14 DETECCIÓN DEL ÁREA DE LOS OJOS .....	47
TABLA 2.15 USUARIOS EN ESTADO DE SOMNOLENCIA .....	57
TABLA 2.16 EMISIÓN DE ALARMA.....	57
TABLA 3.1 MODELO DE REGRESIÓN MÚLTIPLE.....	66
TABLA 3.2 MATRIZ DE CONFUSIÓN DE ROSTROS.....	67

TABLA 3.3 DATOS ATÍPICOS - ROSTROS .....	68
TABLA 3.4 RESULTADOS DEL TEST DE SHAPIRO - ROSTROS.....	71
TABLA 3.5 CALCULO DE MÉTRICAS - ROSTROS.....	73
TABLA 3.6 RESULTADO DEL CALCULO DE LAS MÉTRICAS - ROSTROS.....	74
TABLA 3.7 CÁLCULO Y RESULTADO DE LA MEDIA DE LAS MÉTRICAS CALCULADAS - ROSTROS .....	75
TABLA 3.8 MATRIZ DE CONFUSIÓN - OJOS.....	76
TABLA 3.9 DATOS ATÍPICOS - OJOS .....	77
TABLA 3.10 RESULTADOS DEL TEST DE SHAPIRO - OJOS.....	80
TABLA 3.11 CALCULO DE MÉTRICAS - OJOS.....	81
TABLA 3.12 RESULTADO DE LAS MÉTRICAS CALCULADAS - OJOS.....	82
TABLA 3.13 CÁLCULO Y RESULTDOS DE LA MEDIA EN LAS MÉTRICAS - OJOS.....	83
TABLA 3.14 RESULTADOS DE LA ESCUESTA APLICADA .....	85
TABLA 3.15 PESOS DE CADA PREGUNTA PARA CADA SECCIÓN.....	97
TABLA 3.16 CÁLCULO DE PUNTAJE OBTENIDO EN CADA SECCIÓN SEGUN CADA USUARIO SOBRE EL 100%.....	98
TABLA 3.17 RESULTADOS DE LAS SECCIONES CALCULADAS .....	98
TABLA 3.18 CÁLCULO Y RESULTADO DE LA MEDIA OBTENIDA EN CADA SECCIÓN .....	99

## RESUMEN

Considerando el alto índice de accidentes tránsito que se registran anualmente en el país y la gran acogida que tienen los teléfonos inteligentes actualmente, surgen cuestionamientos sobre el apoyo de la academia al país para dar solución a estos problemas, consecuentemente nace la necesidad de buscar una alternativa para ayudar a reducir estos altos índices de accidentes viales, haciendo uso de inteligencia artificial específicamente visión por computador y herramientas tecnológicas que se encuentren en auge para garantizar la continuidad y funcionalidad del proyecto planteado.

Hoy en día el avance tecnológico hace posible lo imposible de esta manera permite desarrollar soluciones que ayuden en el vivir de las personas.

La finalidad del presente trabajo de titulación es desarrollar una aplicación móvil Android para la detección y alerta de somnolencia nocturna en tiempo real, mediante el uso de técnicas de visión artificial, lo cual se presume que una vez implementado ayudará a los conductores a darles una alerta cuando entren en un estado de somnolencia y de esta forma reducir el alto índice de accidentes. Este proyecto se lo realizó haciendo uso del IDE de Desarrollo Android Studio, la librería de OpenCv y algoritmos de Haar Cascade de Viola & Jones bajo la metodología de Programación Extrema conocida como metodología de desarrollo XP.

Como resultado del trabajo de Titulación, la aplicación móvil detecta rostros, segmenta la ubicación de los ojos, clasifica ojo derecho e izquierdo, detecta pupila por medio de técnicas de emparejamiento de plantillas, que es la técnica estudiada en este proyecto en la que los resultados de búsqueda de un objetivo indica si fue encontrado o no, por medio de cálculos de la Transformada Rápida de Fourier en la que se obtiene el mayor o menor intensidad de una imagen en escala a grises, esto se logró realizar haciendo uso de funciones de la biblioteca de OpenCv y finalmente la aplicación móvil emite una alarma sonora para alertar al conductor de su estado de somnolencia.

### **Palabras Claves:**

Somnolencia, OpenCv, Transformada Rápida de Fourier, algoritmos Haar Cascade, Metodología de Programación Extrema, Android Studio, emparejamiento de plantillas.

# ABSTRACT

Considering the high rate of traffic accidents that are recorded annually in the country and the great reception that smartphones currently have, questions arise about the academy's support to the country to solve these problems, consequently the need to seek an alternative is born to help reduce these high road accident rates, using artificial intelligence specifically computer vision and technological tools that are booming for the accuracy and functionality of the proposed project.

Nowadays, technological advancement makes the impossible possible in this way allows developing solutions that help people live.

The proposal of the present titling work is to develop an Android mobile application for the detection and alertness of nighttime sleepiness in real time, through the use of artificial vision techniques, which is presumed that once implemented help drivers give them an alert when it is in a state of drowsiness and in this way it reduces the high accident rate. This project was carried out with the Android Studio Development IDE, the OpenCv library and the Viola & Jones Haar Cascade algorithms under the Extreme Programming methodology known as XP development methodology.

As a result of the Titling work, the mobile application detects faces, segments the location of the eyes, classifies the right and left eye, detects pupils by means of template matching techniques, which is the technique studied in this project in which the results A target search indicates if it was found or not, by means of calculations of the Fast Fourier Transform in which the greater or lesser intensity of a grayscale image is obtained, this was achieved by using functions of the OpenCv library and finally the mobile application emits an audible alarm to alert the driver of his sleepiness.

## **Keywords:**

Drowsiness, OpenCv, Fast Fourier Transform, Haar Cascade algorithms, Extreme Programming Methodology, Android Studio, template matching.

# INTRODUCCIÓN

## Antecedentes

Según Garcés se conoce como somnolencia a la sensación de pesadez que principalmente está motivada por el sueño y el cansancio, según expertos esta sensación es el estado intermedio entre el estado de vigilia y el sueño profundo (Garcés, Salgado, Cruz, & Cañon, 2015).

En las páginas de la ONU registran que cada día alrededor de 3500 personas fallecen en las carreteras. Decenas de millones de personas sufren heridas o discapacidades cada año. Los niños, los peatones, los ciclistas y los ancianos son los usuarios más vulnerables de la vía pública (Organización Mundial de Salud, 2018).

En el sitio web de la Agencia Nacional de Tránsito existen estadísticas de somnolencia la cual representa el 1.7% de los del total de los accidentes suscitados (Agencia Nacional de Tránsito, 2018).

En un informe de la Fiscalía General del Estado, da a conocer que la Organización Mundial de la Salud ubica al Ecuador como el segundo país de Latinoamérica con el mayor índice de muertes causadas por accidentes de tránsito, estos datos concuerdan con la información estadística del INEC del 2012 (Agencia Nacional de Tránsito, 2016).

Según el diario el Universo y corroborando esta información las estadísticas de la Agencia Nacional de Tránsito, actualmente en Ecuador la tasa de mortalidad en accidentes de tránsito a aumentado considerablemente el primer semestre del año 2018 en base a los anteriores 5 años, en el año 2014 la tasa de fallecidos hasta junio era 6.34 por cada 100 accidentes de tránsito, en el año 2015 esta tasa bajo a 5.93, en el 2016 esta tasa volvió a subir a 6.28, en el año 2017 siguió aumentando hasta llegar a 7.06 y para el año 2018 es la tasa más alta de los cinco años alcanzando un 8.49. estos datos son almacenados en la Agencia Nacional de Tránsito de Ecuador (Agencia Nacional de Tránsito, 2018).

A nivel mundial una de las principales causas que ocasionan accidentes de tránsito es el estado de distracción y somnolencia trayendo consigo pérdidas humanas y económicas.

Ecuador no es la excepción ya que en el año 2016 se registraron 711 accidentes en las vías por conductores que se encontraban en estado de somnolencia, distracción, estado de fatiga, cansancio, malas condiciones físicas entre otras causas el resultado de aquello es 64 accidentes cada mes, en el año 2015 se reportaron un promedio de 73 hechos (Herrera Granda, y otros, 2018).

Según el Servicio de Investigación de Accidentes de Tránsito (SIAT) de Quito en el 2016 indagó 1632 siniestros viales los cuales fueron reportados a nivel del país y de ellos el 98% de

los casos son por fallas humanas, según el oficial se podría regularse de otra forma, además de regular por medio de los reglamentos de la Agencia Nacional de Tránsito (Agencia Nacional de Tránsito, 2018).

Se podría decir que los accidentes de tránsito son más comunes en personas que tienen trayectorias muy extensas de conducción, es decir, conductores que tienen altas jornadas de trabajo. Puede existir varios motivos o circunstancias que se pueden presentar en las horas de conducción ya sean de tipo climáticas, ergonómicas, cansancio o fatiga a pesar de las regulaciones son condiciones involuntarias que se presentan en todo ser vivo.

### **Situación Actual**

En Imbabura en la ciudad de Ibarra aún no se ha implementado equipos tecnológicos o aplicaciones móviles que ayuden a identificar y prevenir la somnolencia nocturna, eso es un punto en contra con los conductores ya que no cuentan con una ayuda para evitar este tipo incidentes que son muy habituales hoy en día, y pueden ser causados por razones varias por ejemplo el clima, el cansancio o el sueño ya que son factores que se presentan en cualquier momento de la noche.

Una solución de somnolencia nocturna aún no se encuentra desarrollado en su totalidad; existen varios estudios e investigaciones que fueron desarrolladas en la ciudad, uno de ellos se realizó en la Universidad Técnica del Norte con colaboración de otras Universidades, los estudios que se han realizado están enfocados en somnolencia diurna (Herrera Granda, y otros, 2018).

A pesar de que en la en la provincia de Imbabura la Agencia Nacional de Tránsito y la Empresa Pública de Movilidad MOVILDENOR EP son las encargadas de regular y controlar la gestión de transporte terrestre, tránsito y seguridad vial, hace falta la innovación e introducción en el tema de inteligencia artificial e inversión en equipos tecnológicos a más de los que ya poseen, equipos que estén enfocados a reducir estos incidentes.

A pesar de las regulaciones que se encuentran establecidas en el reglamento acerca de las jornadas máximas de conducción, que añaden que ningún conductor podrá sobrepasar las 12 horas acumuladas en un volante, los accidentes por cansancio siguen siendo una realidad en el medio (Ministerio de Relaciones Laborales, 2018).

## **Prospectiva**

Con la presente propuesta se pretende alertar a los conductores en caso de que estén entrando en estado de somnolencia mediante la emisión de una alarma o sonido en el celular inteligente y así ayudar a evitar pérdidas materiales y humanas.

Con la implementación futura de esta solución se pretende ayudar a minimizar muertes y accidentes de tránsito que registra la Agencia Nacional de Tránsito en la provincia de Imbabura.

## **Problema**

Los accidentes de tránsito son un fenómeno que se vienen presentando a lo largo del tiempo, con ello causando problemas sociales, económicos, institucionales y tecnológicos. Dicho problema se produce por diferentes causas las más comunes son: fatiga o somnolencia, la embriaguez, estacionarse mal, malas maniobras al rebasar y salida inesperada de animales en las vías; no respetar las señales de tránsito; superar los límites máximos de velocidad; conducir bajo influencia del alcohol o sustancias estupefacientes que alteran el equilibrio del cuerpo, mientras más rápido se controle esta situación será mejor (Garcés, Salgado, Cruz, & Cañon, 2015). La Organización Mundial de la Salud (OMS), en su informe de seguridad vial 2013, ubica al Ecuador como el segundo país de Sudamérica con el mayor índice de muertes causadas por accidentes de tránsito.

## **Objetivos**

### **Objetivo General**

Desarrollar una aplicación móvil Android para la detección y alerta de somnolencia nocturna en tiempo real, mediante el uso de técnicas de visión artificial.

### **Objetivos Específicos**

- ✓ Elaborar un marco teórico que sustente las técnicas y algoritmos de reconocimiento facial y detección de somnolencia.
- ✓ Implementar una alternativa que permita brindar la iluminación suficiente para que la aplicación funcione en condiciones nocturnas.
- ✓ Desarrollar la aplicación móvil e implementar el algoritmo que permita detectar la somnolencia nocturna.
- ✓ Validación de los resultados.

## Alcance

La presente solución tiene como finalidad desarrollar una aplicación móvil en Android Studio que permita detectar y alertar cuando un conductor este entrando en estado de somnolencia nocturna, haciendo uso de la luz que emite la pantalla del celular.

Cabe mencionar que la aplicación funcionará en celulares inteligentes y que cumplan ciertas características de procesamiento.

La solución consiste en implementar un algoritmo eficiente para la detección de somnolencia nocturna, que con su implementación futura permitirá la reducción de número de accidentes de tránsito y fallecidos por causas de estos, ofreciendo al conductor seguridad, tranquilidad y prevención de accidentes en las vías.

En la Figura 1 se muestra la arquitectura del prototipo. Luego de realizar la aplicación se validará los resultados obtenidos.



Figura 1 Prototipo de la solución

Fuente: Propia

## Justificación

Este proyecto se justifica por la falta de estudio en el área de inteligencia artificial y la inexistencia de aplicaciones móviles en Android que ayuden a reducir los incidentes por somnolencia nocturna.

También será de gran utilidad para los organismos encargados de controlar la movilidad terrestre de la ciudad de Ibarra.

Además, el presente proyecto tiene un enfoque hacia los objetivos de desarrollo sostenible ODS:

### **Objetivo número 3 “Salud y Bienestar”**

Para alcanzar este objetivo es fundamental garantizar una vida saludable y promover el bienestar universal. Se menciona en el punto 3.6 que “Para 2020, reducir a la mitad el número de muertes y lesiones causadas por accidentes de tráfico en el mundo.” (Naciones Unidas, 2015)

### **Objetivo número 9 “Innovación e Infraestructura”**

Se menciona en el punto 9.b que “Apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas.”; por otra parte en el punto 9.c de igual forma se menciona que “Aumentar significativamente el acceso a la tecnología de la información y las comunicaciones y esforzarse por proporcionar acceso universal y asequible a Internet en los países menos adelantados de aquí a 2020.” (Naciones Unidas, 2015).

**Justificación Social:** La presente solución permitirá a los ciudadanos de la localidad contar con una aplicación capaz de ayudar a minimizar el número de accidentes de tránsito con su implementación futura y así evitar problemas sociales o psicológicos en las familias de los afectados.

**Justificación Tecnológica:** Esto implica ayudar al avance en la innovación de la ciudad de Ibarra permitiendo a la ciudadanía conocer nuevas tecnologías y nuevos campos de estudio en este caso el campo de Inteligencia Artificial y Computación Móvil.

**Justificación Económica:** Cuando se presenta un accidente de tránsito se presentan varios gastos voluntarios e involuntarios de la parte afectada o no afectada esto implica afección a la economía de las personas involucradas en estos incidentes, con la presente solución se trata de alertar al conductor para evitar la somnolencia nocturna con lo que ayudaremos a la economía del país.

**Justificación Metodológica:** En la actualidad existen estudios sobre somnolencia diurna pero aún no existen estudio enfocados a dar una solución a la somnolencia nocturna y más aún una aplicación que permita simular el funcionamiento de la aplicación.

## Contexto

La epistemología de la inteligencia artificial, la visión artificial y las aplicaciones móviles dedicadas a la detección de somnolencia nocturna es la siguiente.

Se puede mencionar otros trabajos realizados en la Universidad Técnica del Norte con colaboración de otras instituciones que se enfocan en el estudio de detección de somnolencia diurna y además son estudios que se encuentran en repositorios prestigiosos uno de ellos es el tema de "Drowsiness Detection in Drivers through Real-time Image Processing of the Human Eye" (Herrera Granda, y otros, 2018).

La inteligencia artificial nace con el matemático Alan Turing, que fue un visionario en este campo ya que predecía posibles desarrollos de la Inteligencia Artificial y el aprendizaje automático. Por otro lado, en 1955 ya se presentaban trabajos sobre reconocimiento de patrones y máquinas para jugar ajedrez (González & López de Mántaras, 2017).

Por otra parte, debemos introducir el tema de visión artificial, es un área que pretende de cierta forma reproducir artificialmente el sentido de la vista mediante el procesamiento de imágenes que pueden ser captadas con equipos tecnológicos; comúnmente cámaras. (Sánchez Calle, 2005). Hoy en día con el avance de la tecnología de la mano con la inteligencia artificial y específicamente la visión artificial, se trabaja en varios proyectos que son de ayuda para la sociedad uno de ellos es la detección de somnolencia mediante diferentes técnicas de visión artificial.

Los estudios más actuales en este ámbito se enfocan a estudios y pruebas de diferentes técnicas orientadas a la visión artificial como menciona Egas Fabricio en su tesis de grado que los métodos de detección se dividen en dos grupos uno dedicado al desempeño y otro dedicado al estado del conductor, además menciona que el desarrollo de una aplicación móvil permitiría tranquilidad, seguridad y confianza al momento de conducir un vehículo (EGAS CUNALATA, 2017).

# **CAPÍTULO 1**

## **Marco Teórico**

### **1.1. Relación somnolencia-accidentes de tránsito**

Se conoce que varios de los accidentes que se suscitan a diario en el país y a nivel mundial son ocasionados por la somnolencia o distracciones. Para comprender esta problemática a continuación se esclarece algunos conceptos:

#### **1.1.1. La somnolencia**

La somnolencia puede aparecer en situaciones de estrés o fatiga de una manera inesperada o inoportuna. También, se la conoce como un estado intermedio entre la vigilia y el sueño profundo como se muestra en la Figura 2; se dice que puede ser producida por varios trastornos, incluso ciertos tipos de medicamento, aburrimiento, conducir por largos períodos de tiempo o cuando los asientos propios de los vehículos son demasiados cómodos y dan una sensación de confort, trayendo consigo la sensación de sueño; con ello se reduce el nivel de vigilia y produce situaciones de peligro y aumenta la probabilidad de que ocurra un accidente (Flores et al., 2010).

Según Ji (2002) expresa que la somnolencia produce la disminución de la capacidad de vigilia afectando negativamente a los sentidos, por ejemplo, mermando la capacidad de percepción, reconocimiento y control del vehículo y con ello aumentando la probabilidad de sufrir algún accidente o un sobresalto en la carretera, trayendo consigo pérdidas humanas, materiales y económicas. Además, cabe recalcar que este estado puede presentarse tanto en el día como en la noche.

Por otro lado, se considera que la somnolencia es un estado intermedio entre la vigilia y el sueño profundo, y por esta razón puede ser interrumpido fácilmente por estímulos visuales, auditivos o sensoriales, con lo que la persona puede retornar a un aparente estado de vigilia por un determinado tiempo que dependen de la fatiga y el estado de salud de la persona (Garcés, Salgado, Cruz, & Cañón, 2015).



Figura 2 Persona con estado de somnolencia

Fuente: (Jiménez M., 2017)

### 1.1.2. Accidentes de tránsito

Se conoce como accidente de tránsito a todo suceso eventual o acción involuntaria, que como efecto de una o más causas y con independencia del grado de estas, ocurre en vías o lugares destinados al uso público o privado, ocasionando personas muertas, individuos con lesiones de diversa gravedad o naturaleza y daños materiales en vehículos, vías o infraestructura, con la participación de los usuarios de la vía, vehículo, vía y/o entorno como se muestra en la Figura 3. (Guzmán, 2014).



Figura 3 Accidente de Tránsito

Fuente: (Mundo Hispano Digital Network, 2018)

### 1.1.3. Tipos de accidentes de tránsito

Los accidentes de tránsito se pueden producir de diferentes maneras, a continuación, se analizan cada una de ellas. Las definiciones fueron tomadas del Reglamento General para la Aplicación de la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial, Año 2012 (De, 2014).

**Atropello:** Impacto de un vehículo en movimiento a un peatón o animal.

**Arrollamiento:** Acción por la cual un vehículo pasa con su rueda o ruedas por encima del cuerpo de una persona o animal.

**Caída de pasajero:** Es la pérdida de equilibrio del pasajero que produce su descenso violento desde el estribo o del interior del vehículo hacia la calzada.

**Choque:** Es el impacto entre dos vehículos en movimiento.

**Colisión** Impacto de más de dos vehículos.

**Estrellamiento:** Impacto de un vehículo en movimiento contra otro estacionado o contra un objeto fijo.

**Rozamiento:** Es la fricción de la parte lateral de la carrocería de un vehículo en movimiento con un vehículo estacionado o un objeto fijo.

### 1.1.4. Los accidentes de tránsito a nivel mundial

El informe sobre el estado mundial de la seguridad vial 2018, lanzado por la Organización Mundial de la Salud (OMS) en diciembre de 2018, destaca que el número de muertes anuales por accidentes de tráfico ha alcanzado los 1,35 millones. En la Figura 4 se puede observar señalado los países con más accidentes de tránsito.

Las lesiones por accidentes de tránsito son ahora la principal causa de muerte entre las personas de 5 a 29 años. La carga la asumen en la mayoría los peatones, ciclistas y motociclistas, en particular los que viven en países en desarrollo, como el Ecuador.

El informe sugiere que el precio pagado por la movilidad es demasiado alto, especialmente porque existen medidas probadas. Se necesita una acción drástica para implementar estas medidas para cumplir con cualquier objetivo global futuro que pueda establecerse y salvar vidas.

Ecuador al igual que varios países presenta un número elevado de accidentes de tránsito anualmente.



Figura 4 Accidentes de Tránsito a Nivel Mundial

Fuente: Mapa: (RT, 2014)

### **1.1.5. La somnolencia y los accidentes de tránsito**

Los conductores en estado de fatiga son propensos a sufrir estados de micro sueños más conocidos como pequeños lapsos de sueño que son breves siestas de 4 a 5 segundos. Estas personas piensan que han estado despiertos todo el tiempo o que solamente han perdido la visión momentáneamente es un estado que puede sufrir cualquier persona inconscientemente, las mismas, corren un alto riesgo de sufrir varios tipos de accidentes de tránsito, salida del carril, o a su vez pueden atropellar a personas o sufrir un estallamiento (Castro & Rosales-Mayor, 2010).

Los accidentes de tránsito son un problema para todos los países del mundo y es que además de ser una de las causas principales de muerte a nivel mundial, también generan gastos a todos los gobiernos y sin dejar a un lado las secuelas graves que ocasionan a los involucrados desde el punto de vista físico, psicológico y económico (Guzmán, 2014).

## 1.2. Inteligencia Artificial

La inteligencia artificial, presenta muchas ramas que pueden servir de mucha ayuda en el cotidiano vivir del ser humano. Una de ellas es la visión artificial, que en los últimos años ha sido usada en algunos trabajos para detectar la somnolencia en conductores (Torres & Palma, 2017).

### 1.2.1. Visión Artificial

En la actualidad la visión artificial juega un papel fundamental en el desarrollo de sistemas inteligentes ya que encuentra aplicación en distintas áreas tales como: control de procesos, navegación de vehículos, supervisión y vigilancia, interacción usuario-máquina o búsqueda y organización de información en bases de datos. Con el rápido avance de la tecnología y el aumento en la capacidad de cómputo de los microprocesadores han permitido un gran desarrollo de los sistemas de visión artificial en las últimas décadas (Barrero et al., 2015).

La visión por computador actualmente es el método más usado para detectar somnolencia en conductores y operarios de maquinaria pesada. Existen aplicaciones comerciales en sistemas embebidos por parte de algunas empresas y es preferido por ser un método no invasivo para el operario y por la sencillez de su implementación. Toyota comenzó a hacer pruebas del sistema en simulaciones cotidianas de conducción para aplicarlo finalmente en 2006 como un sistema embebido en uno de sus modelos Lexus. Mercedes en el 2009 y Ford en el 2010 fueron los siguientes fabricantes en implementar este sistema en algunos de sus modelos (Guzmán, 2014)

### 1.2.2. Técnicas de detección de objetos

A continuación, se presenta una breve historia del (2001 al 2017) de la precisión de la detección de objetos ya que nos dará una idea de cómo cada arquitectura de aprendizaje profundo funciona en el campo de la visión artificial (Kaggle, 2018):

**Viola-Jones Algorithm, 2001:** el primer detector eficiente de rostros: Sus autores Paul Viola y Michael Jones crearon el algoritmo eficiente de detección de rostros en tiempo real; el proceso del algoritmo se muestra en la Figura 5; y demostraron su funcionamiento en una transmisión de webcam; en ese momento fue una demostración sorprendente de la visión por computador en ese tiempo, seguido se implementó en OpenCV(es una librería de visión por computador de código abierto); también tuvo algunas desventajas ya que no

podía reconocer rostros en ciertas posiciones de la cara (inclinada, boja abajo, usando una máscara, etc.).

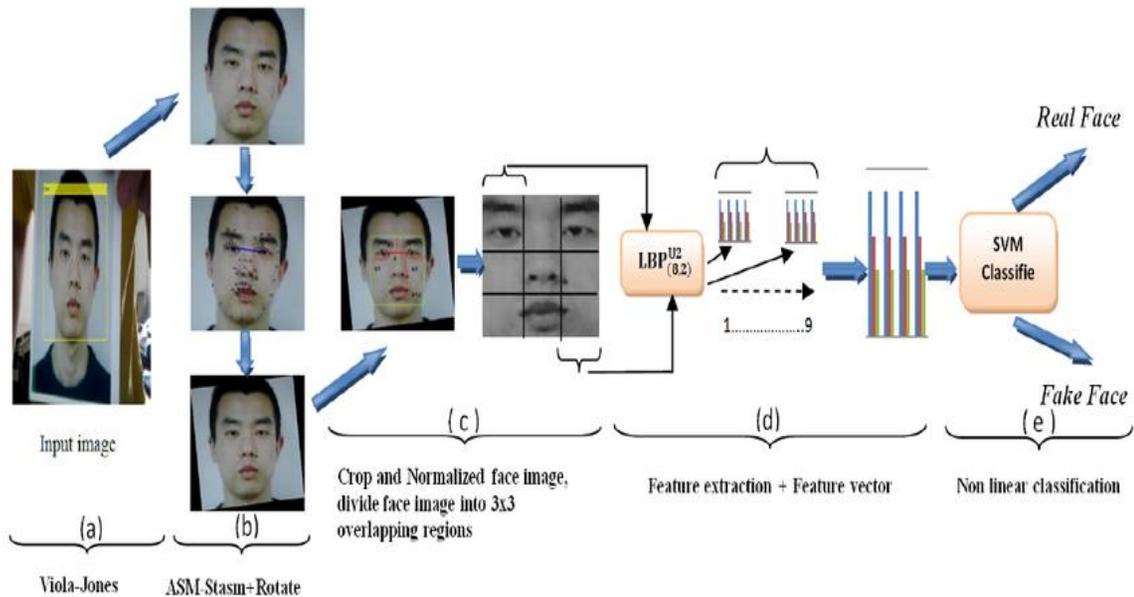


Figura 5: Algoritmo de Viola Jones

Fuente: (Kaggle, 2018)

**Técnica de detección mucho más eficientes: Histogramas de Gradientes Orientados, 2005:** Navneet Dalal y Bill Triggs en el 2005 inventaron HOG (Histogramas de gradientes orientados) para la detección de peatones, esta técnica supero considerablemente los algoritmos existentes en esta tarea ya que la técnica consistía en tomar histogramas de orientación de intensidad de borde en una región local. La Figura 6 muestra como es el proceso de los Histogramas de Gradientes Orientados.

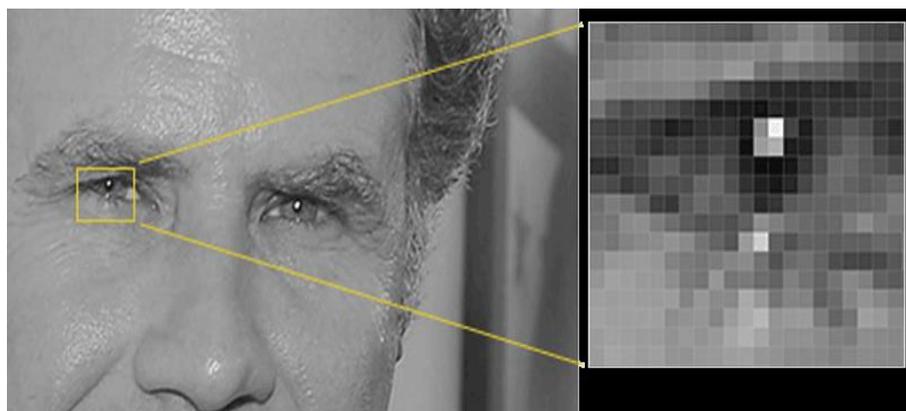


Figura 6: Histogramas de Gradientes Orientados

Fuente: (Kaggle, 2018)

**Comienza la era del aprendizaje profundo, 2012:** Las redes neuronales convolucionales se convirtieron en el estándar de oro para la clasificación de imágenes después del desempeño de Krizhevsky en CNN durante ImageNet. La Figura 7 muestra como es el proceso de las redes neuronales.

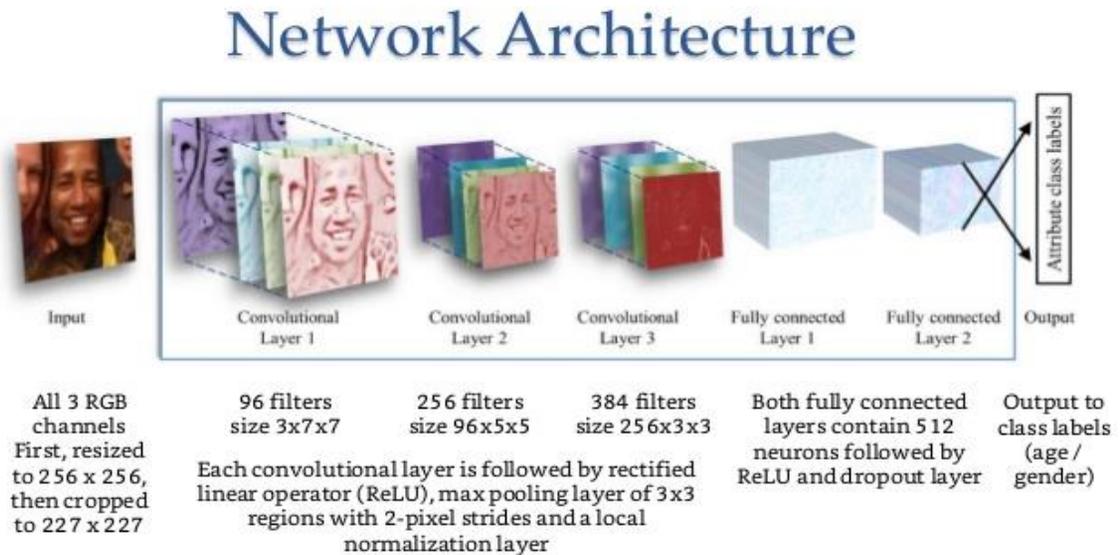


Figura 7 Aprendizaje Profundo

Fuente: (Kaggle, 2018)

En la clasificación, generalmente hay una imagen con un solo objeto como foco y la tarea es decir qué es esa imagen. Pero cuando se observa el mundo que nos rodea, se realiza tareas mucho más complejas. Se observa vistas complicadas con múltiples objetos superpuestos y diferentes fondos, y no solo clasifican estos diferentes objetos, sino que también identifican sus límites, diferencias y relaciones entre sí.

**YOLO adopta un enfoque diferente, 2017:** YOLO tiene un enfoque completamente diferente ya que no es un clasificador tradicional que se reutiliza para ser un detector de objetos porque en realidad mira la imagen solo una vez y de ahí proviene su nombre: You Only Look Once, pero de una manera inteligente (Kaggle, 2018).

El proceso de YOLO es que divide la imagen en una cuadrícula de 13 por 13 celdas y cada una de estas celdas es responsable de predecir 5 cuadros delimitadores y a su vez YOLO también genera un puntaje de confianza que nos dice qué tan seguro es que el cuadro delimitador previsto encierra algún objeto.

Los cuadros delimitadores predichos pueden tener un aspecto similar a la Figura 8, ya que cuanto más alto es el puntaje de confianza, el cuadrado tiene más grosor:

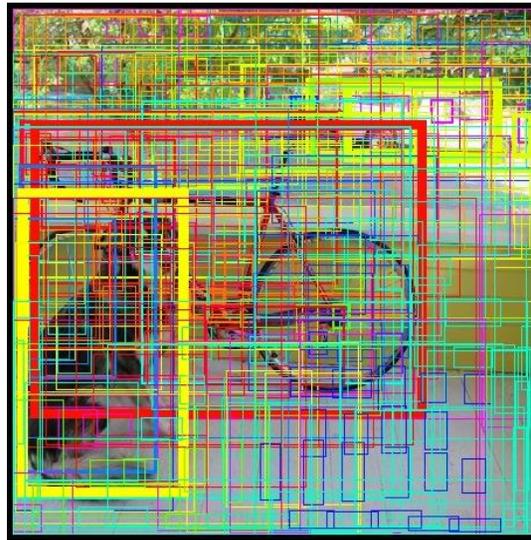


Figura 8 Cuadros delimitadores de YOLO

Fuente: (Kaggle, 2018)

Esto funciona igual que un clasificador: proporciona una distribución de, probabilidad sobre todas las clases posibles y puede detectar 20 clases diferentes (carro, bote, perro, gato, persona, etc.).

Dando como resultado final tres clases precisas las cuales son: bicicleta, perro y carro como se muestra en la Figura 9:

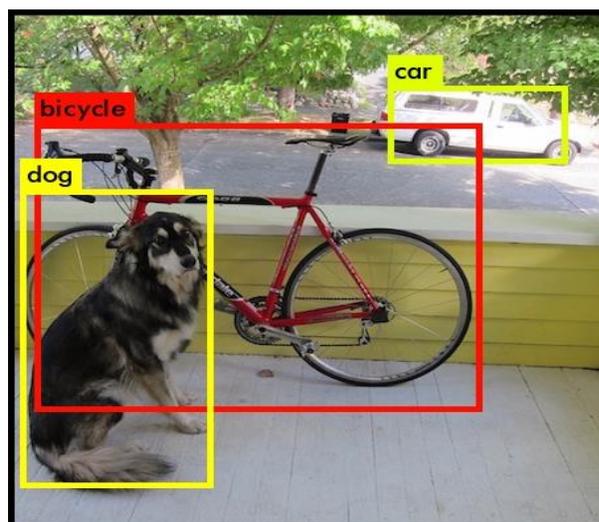


Figura 9 Resultado final de YOLO

Fuente: (Kaggle, 2018)

### 1.2.3. Etapas de un sistema de visión Artificial

Según López Romero (2016) un sistema de visión artificial consta de las siguientes etapas a continuación, se detallará cada etapa de un sistema de visión artificial (López Romero, 2016):

**Adquisición de imágenes:** Es el proceso que nos lleva a la obtención de una imagen mediante cámaras o tarjetas de adquisición de imágenes.

**Preprocesamiento:** En esta etapa se reduce características no deseadas en la imagen como el ruido y se realzan características importantes que permitan realizar de una mejor manera la siguiente etapa de segmentación.

**Segmentación:** En esta etapa se diferencian los objetos de interés de la imagen.

**Representación y descripción:** En esta etapa se extraen características que nos permitan identificar un objeto de otro. Las características son basadas en la textura, forma, color, área, perímetro, entre otros.

**Reconocimiento e interpretación:** En esta etapa se identifican los objetos y se les asigna un significado.

### 1.2.4. Herramientas de un sistema de visión artificial en Android

**OpenCV:** OpenCV (Open Source Computer Vision Library) es una librería de Fuente abierta de visión por ordenador. Es desarrollada por Intel y su primera versión apareció en 1991, se distribuye bajo licencia BSD lo que facilita a los usuarios emplear, modificar y distribuir el código con o sin modificaciones (Villanueva & Zavala, 2013).

La librería cuenta con más de 2500 algoritmos optimizados que se pueden utilizar para detectar y reconocer las caras, identificar objetos, clasificar las acciones humanas en los vídeos, seguimiento de objetos en movimiento, encontrar imágenes similares de una base de datos de imágenes, eliminar los ojos rojos de las imágenes tomadas con flash, seguir los movimientos de los ojos, etc. (OpenCV, 2016).

Esta librería es multiplataforma, dispone de versiones para Windows, Linux, Mac os y ha sido escrita en código C y C++ optimizados. Desde la versión 2.3.0 dispone de un SDK OpenCV para Android lo que permite añadir visión artificial a las aplicaciones desarrolladas para dispositivos móviles, lo cual resulta útil para esta investigación.

**FastCv:** Es una librería de visión por ordenador optimizada para dispositivos móviles, es desarrollada por Qualcomm, puede descargarse de forma gratuita y tiene una licencia de uso muy flexible. Además, ofrece dos versiones a los desarrolladores. Por un lado, una versión está optimizada para procesadores ARM (ARM es el procesador más extendido en dispositivos móviles), mientras que otra aprovecha las características del procesador Snapdragon. Ambas versiones comparten el mismo API.

Esta librería cuenta con algoritmos para reconocimiento de gestos, detección, reconocimiento y seguimiento de rostros, reconocimiento y seguimientos de textos, realidad aumentada, etc. (Qualcomm Developer Network, 2016).

**Google Mobile Visión API:** Google provee de un conjunto de interfaces de programación de visión artificial para dispositivos móviles (API de código abierto). Las API que ofrece son de detección y seguimientos de rostro, scanner de código de barras, reconocimiento de texto (Maxwell, 2015).

El código utilizado en los algoritmos de la librería fue creado tomando en cuenta las limitaciones que tienen los dispositivos móviles y su éxito se basa en que fueron diseñados y optimizados por la misma empresa propietaria de Android.

### **1.3. Sistemas de detección de somnolencia**

La detección de somnolencia es un tema que se trata a partir de los altos índices de accidentes de tránsito en las vías, en base a trabajos realizados previamente se puede clasificar algunos métodos existentes en tres categorías (Byung-Chan Chang et al., 2007): En la Figura 10 se muestra un gráfico de la clasificación de los métodos de detección de somnolencia.

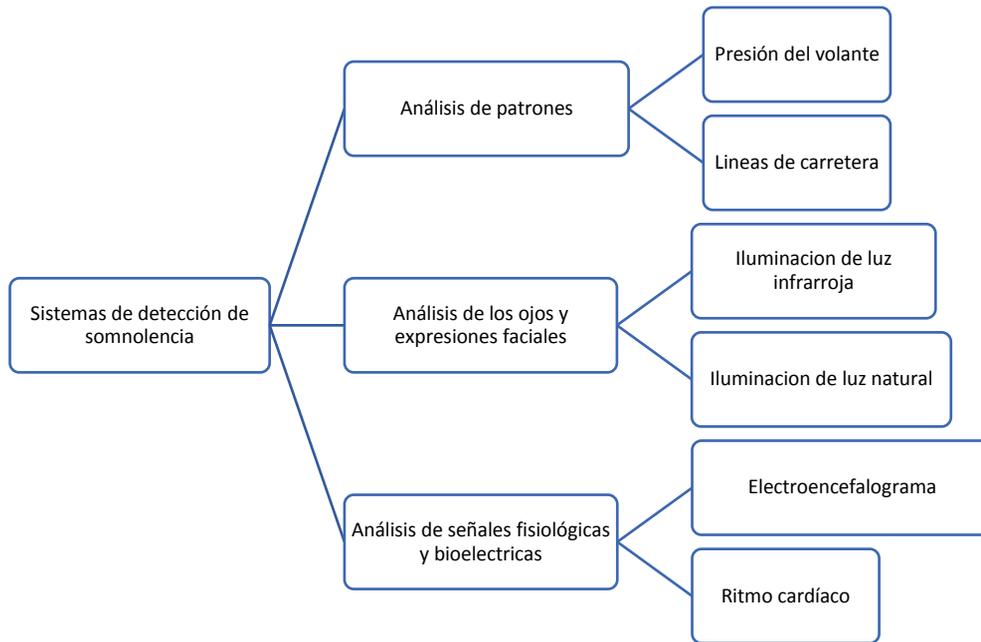


Figura 10 Métodos de detección de somnolencia

Fuente: (Flores, 2009)

A continuación, se presenta una explicación de los métodos existentes para detección de somnolencia:

### 1.3.1. Análisis de patrones

Se puede decir que este modelo de patrones es un método difícil de modelar y en la mayoría de los casos es necesario realizar un modelo distinto dependiendo del vehículo; ya que varía dependiendo de las características de estos y en ocasiones también depende del conductor. Para este modelo se considera la presión de las manos sobre el volante o las líneas de la carretera como se puede observar en la Figura 11. Otros parámetros, tomados sobre el vehículo, que se incluyen en esta clasificación, son la velocidad, la posición lateral, el ángulo de desviación, etc. que han demostrado ser buenos indicadores del nivel de vigilia del conductor, dentro de sus limitaciones (Byung-Chan Chang et al., 2007).

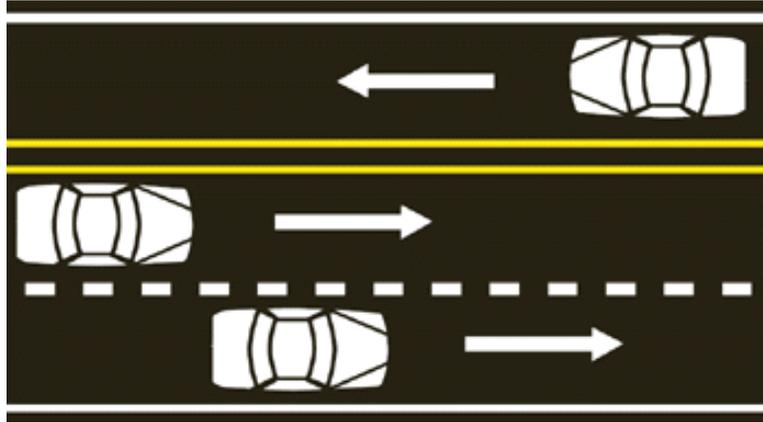


Figura 11 Patrones líneas de carretera

Fuente: (Lowest Price Traffic School, 2018)

### 1.3.2. Análisis de señales fisiológicas y bioeléctricas

Estos tipos de análisis están enfocados en medir los cambios fisiológicos del ser humano, tales como la variabilidad del ritmo cardiaco, el electromiograma y electroencefalograma del conductor, como un medio para detectar los estados cognitivos humanos; como se muestra en la Figura 12. Las ventajas que presentan es la precisión en el diagnóstico del estado. Uno de los principales inconvenientes está en el uso de sensores conectados al cuerpo del conductor ya que puede presentar molestias para el conductor y a la vez puede ser incomodo ya que va a tener todo tipo de materiales que se necesite para poder hacer los análisis.



Figura 12 Persona realizándose un electromiograma

Fuente: (Webconsultas,2018)

En cuanto a señales bioeléctricas se puede decir que el cuerpo genera diferentes tipos de señales de acuerdo con funciones o estados específicos en el que se encuentre el organismo.

Estas señales pueden ser debido a la temperatura, presión o cargas bioeléctricas que recorren el organismo. Las señales bioeléctricas se producen por el intercambio de iones entre células a través de su membrana, esto genera una diferencia de potencial medible, que puede ser procesada para brindar información de actividad del sistema nervioso central y el sistema muscular (Pal et al., 2001).

### 1.3.3. Análisis de los ojos y expresiones faciales

Para este método se utiliza técnicas de procesamiento y análisis de imágenes los cuales servirán para detectar cambios sobre los ojos y el rostro del conductor, dando énfasis en los ojos ya que a partir de ello se podrá detectar si la persona posee o no somnolencia durante el transcurso de la conducción. Se podría decir que estos métodos no son invasivos ya que a diferencia de los métodos de señales fisiológicas las personas no van a tener cables o cualquier equipo conectado o adherido a su cuerpo que causa molestias al conducir; como se muestra en la Figura 13.



Figura 13 Análisis del rostro

Fuente: (TN8,2018)

Algunas técnicas inmersas en este método son las siguientes:

**Técnica de emparejamiento de plantillas:** En este método se definen plantillas del conductor, tanto de sus ojos abiertos como cerrados, luego son comparadas con la imagen actual del conductor y se decide el estado de somnolencia (Reddy, Kim, Yun, Seo, & Jang, 2012).

**Técnica del comportamiento de los ojos:** En este método se calcula la frecuencia de parpadeo y la duración del cierre de los ojos para de esta manera determinar el índice de somnolencia. Uno de los índices más usados para calcular el nivel de somnolencia es

PERCLOS (PERcentage of the time Eyelids are CLOSeD), que mide el porcentaje de tiempo que los ojos de una persona se encuentran cerrados en un 80% al 100% en un periodo. Según un estudio realizado por Walter Wierwille y sus colegas consideran que PERCLOS está entre las más prometedoras medidas de alerta en tiempo real para sistemas de detección de somnolencia en vehículos (Liu et al., 2009). Esta técnica será posteriormente utilizada en esta investigación.

**Técnica basada en el bostezo:** Este método se basa en la frecuencia de bostezos del conductor. La apertura de la boca de una persona es mayor cuando bosteza que cuando habla normalmente. Se compara la apertura de la boca con un punto de referencia obtenido experimentalmente por el programador y se calcula las veces que el conductor ha bostezado para generar un índice de somnolencia (Cech, 2016). Esta técnica también será considerada en esta investigación.

#### **1.4. Sistemas para iluminación diurna**

Para analizar la somnolencia del conductor, se han construido varios sistemas en los últimos años. Por lo general, requieren que el problema se simplifique para funcionar parcialmente o para entornos específicos; por ejemplo, D'Orazio (D'Orazio et al., 2004) han propuesto un algoritmo de detección de ojos que buscan dentro de la imagen completa y han asumido que el iris siempre es más oscuro que la esclerótica. Usando la transformada de Hough para círculos y restricciones geométricas para ubicar los puntos candidatos para ser un ojo.

Seguido, pasaron a una red neuronal que clasifica entre ojos y no ojos. Este sistema es capaz de clasificar los ojos en estado abierto o cerrado. Las principales limitaciones de este algoritmo son: es aplicable solo cuando los ojos son visibles en la imagen y no es robusto al cambiar la iluminación.

Según, (Wen-Bing Horng et al., 2004) ha mostrado un sistema que utiliza un modelo de color de piel sobre el espacio HSI para la detección de rostros, información de bordes para la localización de los ojos y la correspondencia dinámica de plantillas para el seguimiento de los ojos. Usando la información del color del globo ocular, identifica el estado del ojo y calcula el estado del conductor, es decir, dormido o alerta; si los ojos están cerrados durante cinco cuadros consecutivos, el conductor está dormitando.

Brant T. ha mostrado un sistema que controla la somnolencia y la falta de atención del conductor (distracción) (Brandt et al., 2004), como se indica en la Figura 14 este es el algoritmo de detección de somnolencia diurna.

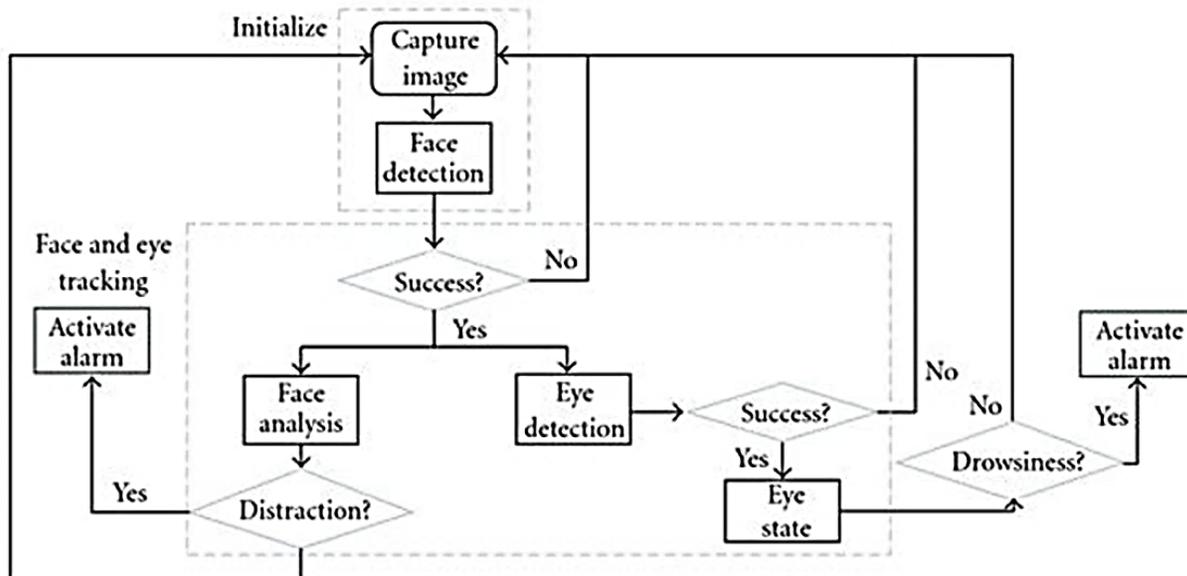


Figura 14 Algoritmo diurno

Fuente: (Flores et al., 2010)

## 1.5. Sistemas para iluminación nocturna

En este caso, debido a las condiciones de luz nocturna, Ji et al. y Ji y Yang en el 2002 y 2004 presentaron un sistema de detección de somnolencia basado en la iluminación NIR (near infrared) y la visión estéreo. Este sistema localiza la posición del ojo utilizando diferencias de imagen basadas en el efecto pupila brillante, característico en este tipo de imágenes (Ji, 2002), (Ji et al., 2004). Posteriormente, este sistema calcula la frecuencia del párpado ciego y la mirada para generar dos índices de somnolencia: PERCLOS (porcentaje de cierre del ojo a lo largo del tiempo) y AECS (velocidad promedio de cierre del ojo); como se muestra en la Figura 15.

Otro trabajo que utiliza este tipo de iluminación está representado por Grace (Grace & Steward, 2001) para medir el cierre lento del párpado. Los sistemas que usan iluminación NIR funcionan bien en condiciones de iluminación estables (Bergasa et al., 2006), sin embargo, esta es una deficiencia para aplicaciones en vehículos reales, donde la luz cambia todo el tiempo debido a las luces provocadas por otros vehículos. En este escenario, si las pupilas espectrales desaparecen será difícil detectar los ojos.

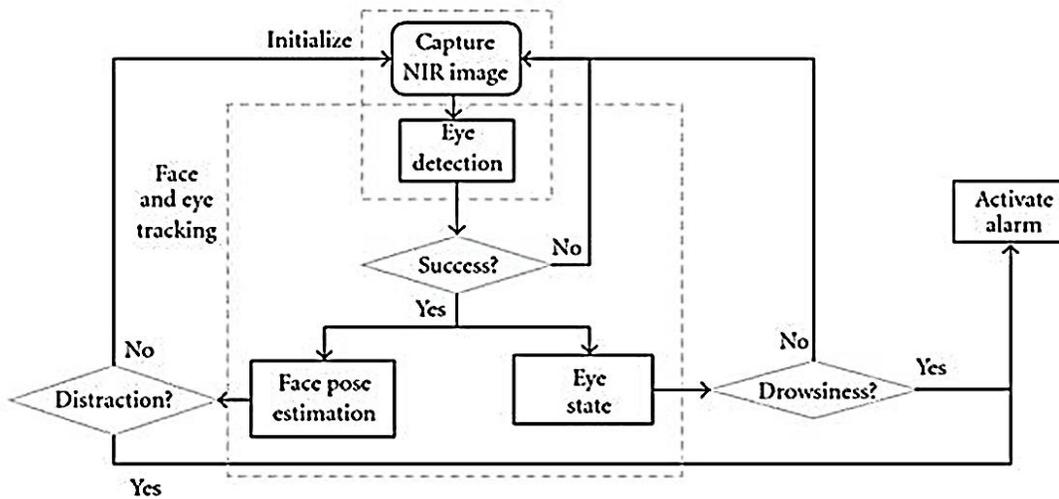


Figura 15 Algoritmo nocturno

Fuente: (Flores et al., 2010)

## 1.6. Smartphone

Un Smartphone es un dispositivo electrónico que permite a su usuario llevar consigo un objeto de dimensiones pequeñas, que cumpla funciones similares a las de un computador personal y cuya evolución no se limita a la transferencia de voz sino también a cierta cantidad de funciones que han atraído la atención de los usuarios, como por ejemplo: navegar por Internet, reproducir datos multimedia, realizar transacciones bancarias, orientarnos vía GPS, descargar juegos, conectarnos a las redes sociales, ver televisión vía streaming, etc. Para (Speckmann, 2018) una de las funciones más importantes que caracteriza a un dispositivo móvil es la conectividad e interacción con las redes de datos, tal es el caso de Internet el cual se ha convertido en un elemento fundamental para las comunicaciones.

Sin embargo, lo que hace atractivo a un Smartphone es el sistema operativo que junto con el hardware logran desarrollar operaciones que permiten correr cierta cantidad de procesos que son traducidos en importantes y llamativas aplicaciones, convirtiéndolos en dispositivos más competitivos que otros (Polanco & Taibo, 2011), un ejemplo de un smartphone se muestra en la Figura 16.



Figura 16 Smartphone

Fuente: Google smartphone

## 1.7. Sistema Operativo Móvil

Un sistema operativo móvil o SO móvil es el software o conjunto de programas que administran los recursos del dispositivo, tanto hardware (teclado, memoria, pantalla, cámara, etc.) como software (aplicaciones e instrucciones). Realiza la carga y ejecución de aplicaciones previamente instaladas. Todo sistema operativo ofrece una interfaz gráfica amigable compuesta por botones e imágenes que permiten al usuario comunicarse con el dispositivo.

El SO de un móvil, al igual que el SO de una computadora, viene cargado al momento de comprarlo, pero si el usuario decide y si es compatible puede actualizarlo y/o cambiarlo.

### 1.7.1. Sistema Operativo Móvil más usado

Según Acuña J. y el servicio de estadísticas NetMarketShare, la cuota de mercado de sistemas operativos móviles a principios de 2017 es el siguiente como se muestra en la Figura 17 (Acuña Morgado , 2018):

- ✓ Android 66,71 % (en países como España las diferencias son más significativas, donde Android tiene más del 90% de la cuota de mercado)
- ✓ iOS 29,55 %
- ✓ Windows Phone 1,41 %
- ✓ BlackBerry OS 0,37 %

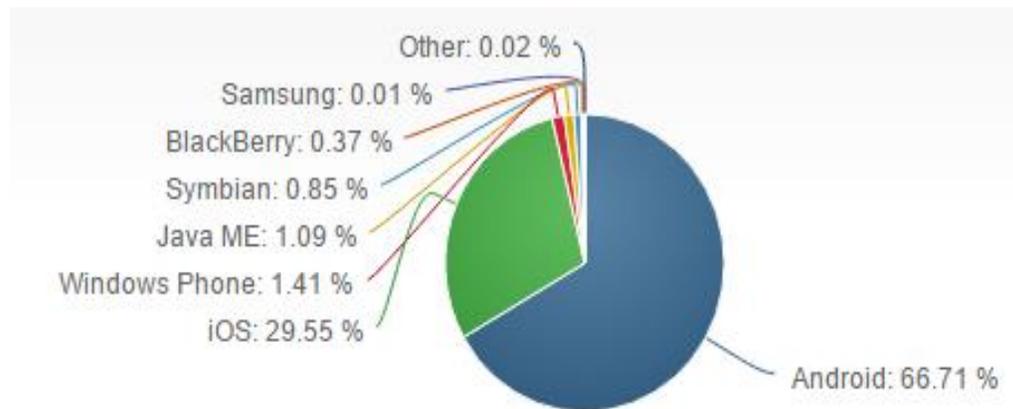


Figura 17 Estadísticas de SO móviles más usados

Fuente: (Acuña, 2018)

### 1.8. Sistema Operativo Android

Según la Figura 12 Android es el sistema operativo más utilizado y está basado en Linux para teléfonos móviles. Una de las claves de la popularidad de Android es que, como Linux, es una plataforma de código abierto, lo que permite a fabricantes, operadores y desarrolladores dar mayor funcionalidad a sus Smartphone. Además, Android es un sistema gratuito y multiplataforma; por multiplataforma se entiende que el sistema operativo puede ser usado en distintas plataformas, y por plataforma se comprende que es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos (Torres & Palma, 2017).

Según (Polanco & Taibo, 2011) un Sistema Operativo para dispositivos móviles es considerado el programa principal y éste es capaz de administrar todos sus recursos para ser utilizados de manera eficiente, cómoda y sin interrupciones, de tal manera que el usuario pueda mantener una comunicación sin problema haciendo uso de los recursos que el hardware le suministra.

Android es un sistema operativo con una plataforma abierta para dispositivos móviles adquirido por Google y la Open Handset Alliance. Su finalidad es satisfacer la necesidad de los operadores móviles y fabricantes de dispositivos, además de fomentar el desarrollo de aplicaciones, cualidad que ningún otro sistema operativo incluye en sus conceptos (Google, 2010).

### 1.8.1. Arquitectura de Android

La Arquitectura del sistema operativo Android está formada por las siguientes capas como se muestra en la Figura 18.

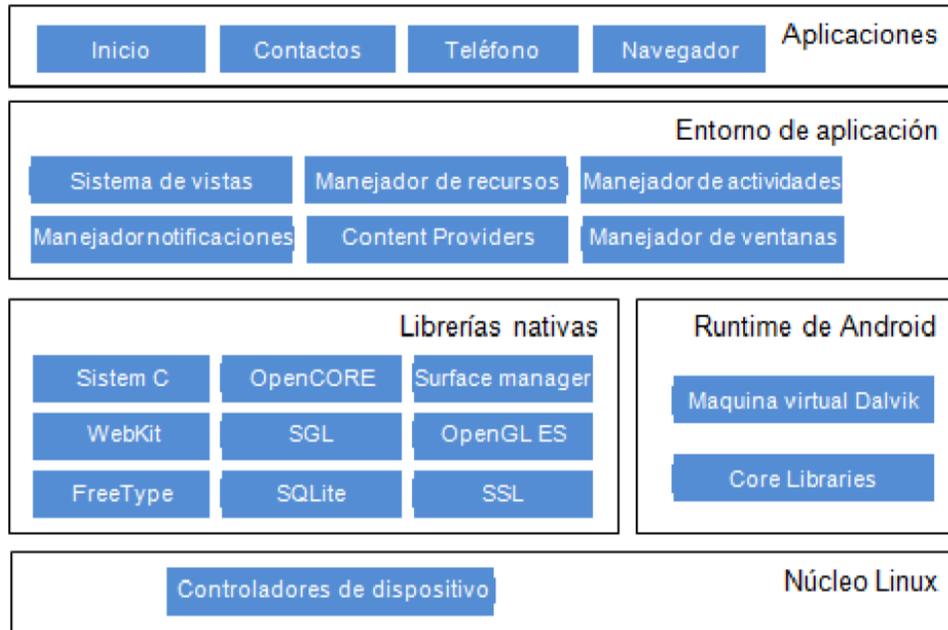


Figura 18 Arquitectura de Android

Fuente: (Girones,2012)

#### Kernel de Linux

Esta capa de la arquitectura actúa como una capa de abstracción de hardware HAL, en donde una aplicación no accede de manera directa a un componente hardware, sino que lo hace a través de la capa abstracta, la cual contiene los drivers necesarios. Por lo tanto, es la única capa que es dependiente del hardware.

El SO Android está basado en los servicios del núcleo de Linux versión 2.6 como son:

- ✓ Controladores hardware
- ✓ Gestión de energía
- ✓ Gestión de procesos
- ✓ Gestión de seguridad
- ✓ Gestión de memoria
- ✓ Pila de red.

## Librerías Nativas

Android incluye un conjunto de librerías escritas en código nativo del procesador (C/C++) usadas por varios componentes del sistema, algunas son:

- ✓ **libc:** Incluye cabeceras y funciones del lenguaje C.
- ✓ **Surface Manager:** Compone los diferentes elementos de navegación y ventanas en la pantalla.
- ✓ **SGL:** Soporte a gráficos 2D.
- ✓ **OpenGL ES:** Soporte a gráficos 3D.
- ✓ **Media Framework:** Proporciona códecs necesarios para múltiples formatos de audio, video e imagen.
- ✓ **SQLite:** Gestor de bases de datos.
- ✓ **WebKit:** Proporciona un motor para aplicaciones tipo navegador.
- ✓ **Free Type:** Trabaja con distintos tipos de fuentes.
- ✓ **SSL:** Soporte a comunicaciones seguras.

## Runtime de Android

Es un entorno de ejecución constituido por dos componentes:

- ✓ **Librerías del núcleo:** Son un conjunto de librerías escritas en lenguaje java que proveen la mayor parte de las funcionalidades de las librerías nativas.
- ✓ **Dalvik VM:** Es una máquina virtual optimizada para dispositivos móviles, tomando en consideración la poca memoria, las limitaciones del procesador y la alimentación por batería. Dalvik no trabaja directamente con el bytecode de java, sino que los transforma en un código más eficiente creado para procesadores pequeños. Esta transformación se lleva a cabo mediante la herramienta "dx" incluida en el SDK de Android que convierte los ficheros .class de java en ficheros .dex y luego los compila en un fichero de extensión .apk (Android package) que es el formato de archivo utilizado para la instalación de aplicaciones en los dispositivos móviles (Tudela et al., 2009).

## Entorno de aplicación

Es un conjunto de herramientas diseñadas para el desarrollo de aplicaciones. Esta capa ha sido diseñada para la reutilización de componentes, cuando una aplicación publica sus capacidades cualquier otra aplicación puede hacer uso de estas.

## **Aplicaciones Android**

Las aplicaciones Android pueden ser escritas en lenguaje Java usando el kit de desarrollo de software SDK y en lenguaje C/C++ usando el kit de desarrollo Nativo o en Python a través de la capa de software para Android SL4A.

### **1.8.2. Principales Características de Android**

Las principales características del sistema operativo Android se tornan realmente atractivas por diversas características, entre ellas se encuentran:

- ✓ Plataforma totalmente libre basado en Linux que permite desarrollar aplicaciones y/o modificar las ya existentes con lenguaje de Java.
- ✓ Es multitasking permitiendo mantener distintas aplicaciones corriendo al mismo tiempo.
- ✓ Compatible con una gran variedad de hardware en el mercado (tablets y dispositivos celulares de marcas como: Motorola, Samsung, ZTE, Huawei, Ericsson por nombrar algunas) permitiendo al usuario elegir el dispositivo que mejor se ajusta a sus necesidades.
- ✓ Posee un portal llamado Android Market donde se tiene acceso a muchas aplicaciones que pueden ser utilizadas.
- ✓ Permite realizar actualizaciones del sistema operativo en línea siempre y cuando el dispositivo soporte los requerimientos de este.
- ✓ Puede operar soluciones tecnológicas referentes al uso de redes sociales, mensajería instantánea, correo electrónico, modificación y lectura de procesadores de palabras, hojas de cálculo, presentaciones, lectura de formatos pdf, entre otros.
- ✓ Se puede conseguir mucha información a través de documentos web o libros.
- ✓ Como característica importante, cuenta con el gran apoyo y la capacidad tecnológica proporcionada por su principal socio Google.

# CAPÍTULO 2

## Desarrollo de la Propuesta

### 2.1 Introducción - Metodología XP

En el desarrollo del proyecto propuesto se utiliza la metodología ágil XP-Programación Extrema para la gestión de la propuesta. Como plataforma de desarrollo Android Studio para realizar la codificación e interfaz gráfica de la aplicación móvil la misma que abarca el módulo de desarrollo de una aplicación móvil Android para la detección y alerta de somnolencia nocturna en tiempo real, mediante técnicas de visión artificial, además se detalla todas las fases de desarrollo del proyecto y se facilita la información necesaria para demostrar el funcionamiento del aplicativo.

### 2.2 Fase De Planeación

En la siguiente sección se presenta cómo está planificado el proyecto, los involucrados y así mismo el plan de fases.

#### 2.2.1 Planificación del Proyecto

En la Tabla 2.1 se detalla el cronograma de actividades planificadas para el desarrollo del presente proyecto.

TABLA 2.1  
PLANIFICACIÓN DEL PROYECTO

TEMA	ACTIVIDADES	PERIODO	
		INICIO	FIN
Introducción	Antecedentes Situación Actual Prospectiva Problema Objetivos Alcance Justificación Contexto	Enero	Febrero

<b>Marco Teórico</b>	Relación Somnolencia y Accidentes de Tránsito Inteligencia Artificial Sistemas de detección de somnolencia Sistemas para iluminación diurna Sistemas para iluminación nocturna Smartphone Sistema operativo Móvil Sistema Operativo Android	Mayo	Junio
<b>Desarrollo de la Propuesta</b>	Plan de desarrollo Diseño Codificación Pruebas	Julio	Septiembre
<b>Validación de Resultados</b>	Pruebas del aplicativo Muestreo Gráfico de Regresión Matriz de Confusión de Rostros Matriz de Histograma de Densidad Correlación en Rostros Cálculo de Métricas de Desempeño en Rostros Precisión - Rostros Exactitud - Rostros Especificidad - Rostros Medida Sensibilidad - Rostros Medida F1 - Rostros Matriz de Confusión de Ojos Matriz de Histograma de Densidad Correlación en Ojos Cálculo de Métricas de Desempeño en Ojos Curva ROC Encuesta del Prototipo Análisis e interpretación de resultados de la encuesta	Octubre	Noviembre
<b>Conclusiones y Recomendaciones</b>	Conclusiones Recomendaciones	Noviembre	Diciembre

Fuente: Propia

## 2.2.2 Participantes del Proyecto

Según la metodología XP existen algunos roles propuestos, cabe mencionar que algunos roles fueron omitidos ya que no se ajustan a las necesidades del proyecto:

- ✓ **Programadores:** La encargada del desarrollo de la aplicación móvil fue Adriana Diaz.
- ✓ **Tester o Encargado de Pruebas:** Quien realizó las pruebas de la aplicación móvil fue Adriana Diaz.
- ✓ **Tracker o Encargado de Seguimiento:** En este caso, se asigna la presente función al MsC. Erick Herrera como director del proyecto. Además, el seguimiento del avance de este proyecto estuvieron a cargo el Phd. Ivan Garcia y MsC. Marco Pusdá.

## 2.2.3 Plan de Fases

El desarrollo de la aplicación Móvil se lo realizó en 4 fases de acuerdo con la metodología XP, como se detalla en la Tabla 2.2.

TABLA 2.1  
PLAN DE FASES

FASE	DURACIÓN
Planeación	2 semanas
Diseño	4 semanas
Desarrollo	12 semanas
Pruebas	8 semanas

Fuente: Propia

### ✓ **Historias de Usuario**

En la siguiente sección se detalla cada una de las Historias de Usuario del proyecto “Desarrollo de una aplicación móvil Android para la detección y alerta de somnolencia nocturna en tiempo real, mediante técnicas de visión artificial”.

1. Configuración de librerías necesarias para la elaboración de la aplicación móvil.
2. Desarrollo de la aplicación “Detección de Rostros”.
3. Desarrollo de la aplicación “Detección de ojo derecho e izquierdo”.
4. Desarrollo de la aplicación “Detección de pupila por medio de emparejamiento de plantillas”.
5. Desarrollo de la aplicación “Detectar somnolencia”.
6. Desarrollo de la aplicación “Emisión de alarma”.
7. Desarrollo de la aplicación “Solución al problema de poca luminosidad en conducción nocturna”.
8. Pruebas de funcionamiento

En la historia de usuario número 1 mostrada en la Tabla 2.3 se detalla la actividad planificada en cuanto a la configuración de librerías utilizadas en el desarrollo de la aplicación móvil.

TABLA 2.2  
HISTORIA DE USUARIO 1

<b>Historia de Usuario</b>	
<b>Número:</b> 1	<b>Usuario:</b> Usuario
<b>Nombre de Historia:</b> Configuración de librerías necesarias para la elaboración de la aplicación móvil	
<b>Prioridad en Negocio:</b> (Alta / Media / Baja)	<b>Riesgo en el Desarrollo:</b> (Alto / Media / Baja)
<b>Puntos estimados:</b> 1	<b>Interacción Asignada:</b> 1
<b>Programadores Responsables:</b>	Adriana Díaz
<b>Descripción:</b>	Garantizará el correcto funcionamiento de las librerías y la fusión con el IDE de Desarrollo en este caso Android Studio 3.2.1.
<b>Observaciones:</b>	

Fuente: Propia

En la historia de usuario número 2 mostrada en la Tabla 2.4 se detalla la actividad planificada, en este caso, la detección de rostros usando algoritmos de Haar Cascade.

TABLA 2.3

HISTORIA DE USUARIO 2

Historia de Usuario	
Número:	<b>2</b> <b>Usuario:</b> Usuario
Nombre de Historia: Desarrollo de la aplicación “Detección de Rostros”.	
Prioridad en Negocio:	<b>Riesgo en el Desarrollo:</b> <b>(Alta / Media / Baja)</b> <b>(Alto / Media / Baja)</b>
Puntos estimados:	<b>Interacción Asignada:</b> <b>1</b> <b>1</b>
Programadores Responsables:	<b>Adriana Díaz</b>
Descripción:	En base al algoritmo de Viola & Jones se pretende lograr focalizar el rostro del conductor para continuar con el procesamiento de la imagen o fotogramas.
Observaciones:	

Fuente: Propia

En la historia de usuario número 3 mostrada en la Tabla 2.5 se detalla la actividad planificada, en lo relativo a la detección de ojos tanto derecho como izquierdo con los clasificadores de Haar Cascade: para el ojo derecho haarcascade\_righteye\_2splits y para el ojo izquierdo haarcascade\_lefteye\_2splits.

TABLA 2.4

HISTORIA DE USUARIO 3

Historia de Usuario	
Número:	<b>3</b> <b>Usuario:</b> Usuario
Nombre de Historia: Desarrollo de la aplicación “Detección de ojo derecho e izquierdo”.	
Prioridad en Negocio:	<b>Riesgo en el Desarrollo:</b> <b>(Alta / Media / Baja)</b> <b>(Alto / Media / Baja)</b>
Puntos estimados:	<b>Interacción Asignada:</b> <b>1</b> <b>1</b>
Programadores Responsables:	<b>Adriana Díaz</b>

Descripción:	Obtener las posiciones del ojo derecho e izquierdo en la zona de interés en este caso el rostro detectado.
Observaciones:	

Fuente: Propia

En la historia de usuario número 4 mostrada en la Tabla 2.6 se detalla la actividad planificada, en este caso la detección de pupila, para ello se utilizó la técnica de emparejamiento de plantillas y a su vez el clasificador de Haar Cascade haarcascade\_eye\_tree\_eyeglasses para hacer una combinación de los dos resultados y optimizar a detección de la pupila.

TABLA 2.5

HISTORIA DE USUARIO 4

Historia de Usuario	
Número:	<b>4</b>
Nombre de Historia:	<b>Usuario: Usuario</b>
Nombre de Historia:	Desarrollo de la aplicación “Detección de pupila por medio del método de emparejamiento de plantillas”.
Prioridad	en Negocio: <b>Riesgo</b> en el <b>Desarrollo:</b>
(Alta / Media / Baja)	(Alto / Media / Baja)
Puntos	estimados: <b>Interacción Asignada:</b>
<b>1</b>	<b>1</b>
Programadores Responsables:	<b>Adriana Díaz</b>
Descripción:	Luego de realizar el procesamiento de los fotogramas se pretende detectar la pupila del ojo por medio de la técnica de emparejamiento de plantillas.
Observaciones:	

Fuente: Propia

En la historia de usuario número 5 mostrada en la Tabla 2.7 se detalla la actividad planificada, en cuanto a la detección somnolencia que una vez detectada la pupila se procede a emitir una etiqueta que señale la somnolencia.

TABLA 2.6

HISTORIA DE USUARIO 5

Historia de Usuario	
Número:	<b>5</b> <b>Usuario:</b> Usuario
Nombre de Historia:	Desarrollo de la aplicación “Detectar somnolencia”.
Prioridad	en Negocio: <b>Riesgo</b> en el <b>Desarrollo:</b>
<b>(Alta / Media / Baja)</b>	<b>(Alto / Media / Baja)</b>
Puntos	estimados: <b>Interacción Asignada:</b>
<b>1</b>	<b>1</b>
Programadores Responsables:	<b>Adriana Díaz</b>
Descripción:	Una vez detectada la pupila del ojo se pretende trabajar en la detección de somnolencia con el procesamiento de los frames.
Observaciones:	

Fuente: Propia

En la historia de usuario número 6 mostrada en la Tabla 2.8 se detalla la actividad planificada con referencia a la emisión de la alarma, se procede a reproducir un audio mp3 que se active cuando haya detectado somnolencia.

TABLA 2.7

HISTORIA DE USUARIO 6

Historia de Usuario	
Número:	<b>6</b> <b>Usuario:</b> Usuario
Nombre de Historia:	Desarrollo de la aplicación “Emisión de alarma”.
Prioridad	en Negocio: <b>Riesgo</b> en el <b>Desarrollo:</b>
<b>(Alta / Media / Baja)</b>	<b>(Alto / Media / Baja)</b>
Puntos	estimados: <b>Interacción Asignada:</b>
<b>1</b>	<b>1</b>
Programadores Responsables:	<b>Adriana Díaz</b>
Descripción:	Se procederá a emitir la alarma sonora de la aplicación móvil.
Observaciones:	

Fuente: Propia

En la historia de usuario número 7 mostrada en la Tabla 2.9 se detalla la actividad planificada, en este caso una solución al problema de poca luminosidad en conducción nocturna.

TABLA 2.8

HISTORIA DE USUARIO 7

Historia de Usuario	
Número:	<b>7</b> <b>Usuario:</b> Usuario
Nombre de Historia: Desarrollo de la aplicación “solución al problema de poca luminosidad en conducción nocturna”.	
Prioridad en	Negocio: <b>Riesgo</b> en el <b>Desarrollo:</b> (Alta / <b>Media</b> / Baja) (Alto / Media / Baja)
Puntos	estimados: <b>Interacción Asignada:</b>
<b>1</b>	<b>1</b>
Programadores Responsables:	<b>Adriana Díaz</b>
Descripción:	Dar una solución al problema de poca luminosidad en conducción nocturna.
Observaciones:	

Fuente: Propia

En la historia de usuario número 8 mostrada en la Tabla 2.10 se detalla la actividad planificada, en esta parte se trata de las pruebas de funcionamiento de la aplicación móvil la parte del funcionamiento y la detección en ambientes reales.

TABLA 2.9

HISTORIA DE USUARIO 8

Historia de Usuario	
Número:	<b>8</b> <b>Usuario:</b> Usuario
Nombre de Historia: Pruebas de Funcionamiento	
Prioridad en	Negocio: <b>Riesgo</b> en el <b>Desarrollo:</b> (Alta / <b>Media</b> / Baja) (Alto / Media / Baja)
Puntos	estimados: <b>Interacción Asignada:</b>
<b>1</b>	<b>1</b>
Programadores Responsables:	<b>Adriana Díaz</b>

---

---

Descripción: Realizar las pruebas de funcionamiento de la aplicación móvil.

Observaciones:

---

---

Fuente: Propia

## 2.3 Fase De Diseño

En esta fase se describe el proceso de desarrollo de la aplicación móvil haciendo uso de diagramas de flujo.

### 2.3.1 Arquitectura del sistema

En la Figura 19 se puede apreciar la arquitectura del sistema de la aplicación móvil para la detección de somnolencia nocturna, la cámara frontal del smartphone realiza la captura de las imágenes para realizar el procesamiento de los frames, detección de somnolencia y la emisión de la alarma haciendo uso de los parlantes del celular.

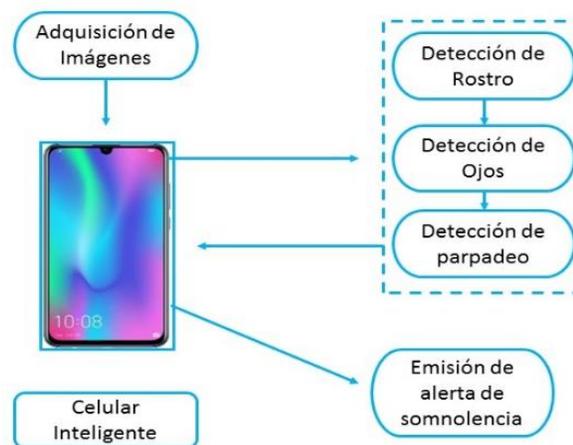


Figura 19 Arquitectura del Sistema

Fuente: Propia

### 2.3.2 Hardware y Software

Una vez determinados los requerimientos del proyecto, se eligió el software y hardware necesario que requiere el sistema. A continuación, en la Tabla 2.11 se presenta el software utilizado para la realización de la propuesta.

TABLA 2.10

HARDWARE Y SOFTWARE PARA EL PROYECTO

HARDWARE / SOFTWARE	IMAGEN	ESPECIFICACIÓN
IDE DE DESARROLLO		Android Studio 3.2.1 es el IDE seleccionado para realizar la codificación del algoritmo y la interfaz gráfica de usuario.
TRATAMIENTO DE IMÁGENES		OPEN CV 3.3.1 es el encargado del procesamiento de imágenes para el desarrollo del sistema.
CELULAR		Cámara frontal del smartphone para recibir las imágenes y después procesarlas. Para la instalación de la apk es necesario tener un celular inteligente con una versión de Android igual o superior a 8.0.0 para el correcto funcionamiento de la aplicación.

Fuente: Propia

### 2.3.3 Diagrama Work Break Down Structure para la realización del proyecto

En el siguiente diagrama de bloques en la Figura 20 se muestra el proceso y las etapas ejecutadas para el diseño del algoritmo, consta de 6 bloques; el primero dedicado a la configuración de librerías, el segundo obtener la imagen, el tercero tratamiento de la imagen, el cuarto dedicado a la detección de rostros, el quinto dedicado a la detección de ojos, el sexto dedicado a la detección de somnolencia.

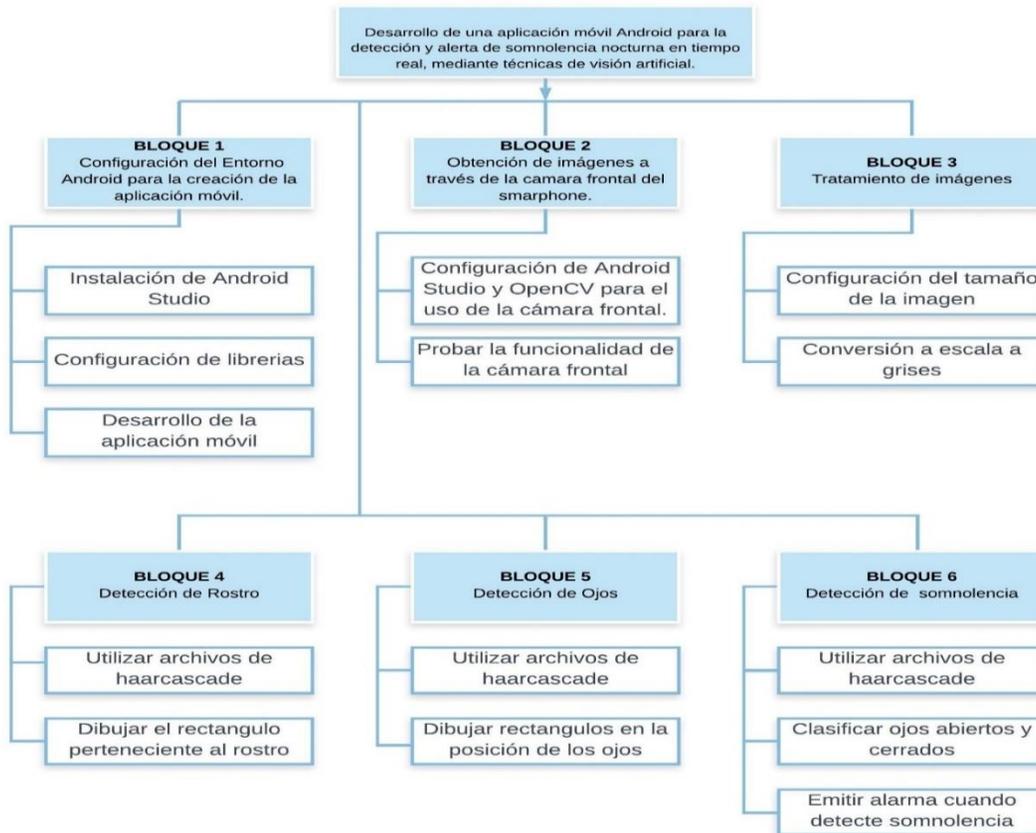


Figura 20 Diagrama Work Break Down Structure para la realización del proyecto

Fuente: Propia

### 2.3.4 Diagramas por bloques del proyecto

En la siguiente sección se muestra cada uno de los diagramas por bloques que se usó para el desarrollo del prototipo de la aplicación móvil.

#### ✓ **Configuración del Entorno Android para la creación de la aplicación móvil.**

Para la configuración del entorno de desarrollo se siguió el proceso de la Figura 21.

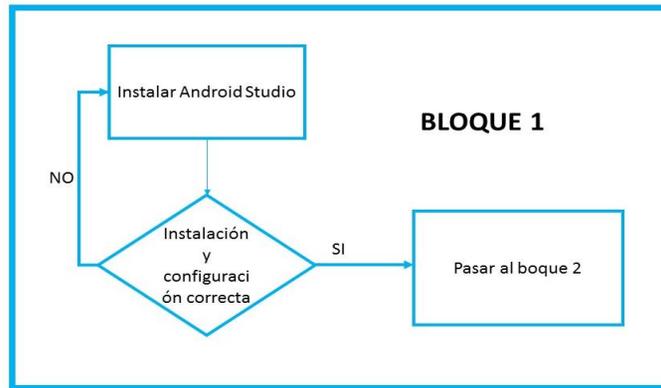


Figura 21 Instalación de Android y de librerías para el desarrollo de la aplicación móvil.

Fuente: Propia

✓ **Obtención de imágenes a través de la cámara frontal del smartphone.**

Como se muestra en la Figura 22, la imagen es adquirida de la cámara frontal del smartphone para continuar con el proceso siguiente del proyecto en este caso el procesamiento de las imágenes adquiridas.

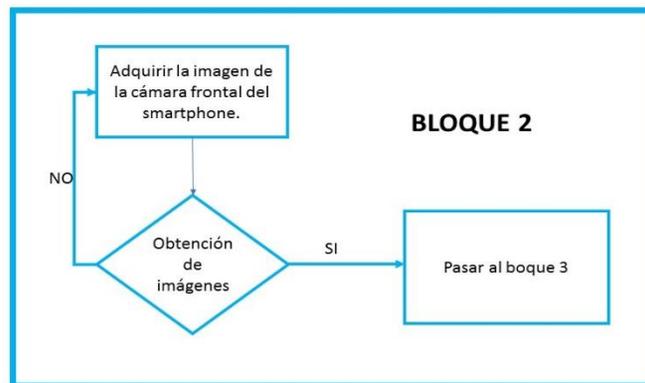


Figura 22 Obtención de la imagen

Fuente: Propia

✓ **Tratamiento de imágenes.**

Como se muestra en la Figura 23 del tratamiento de imágenes; una vez adquirida la imagen a través de la cámara frontal se procede a tratar la imagen, es decir, la conversión a escala a grises usando la librería de OpenCV.

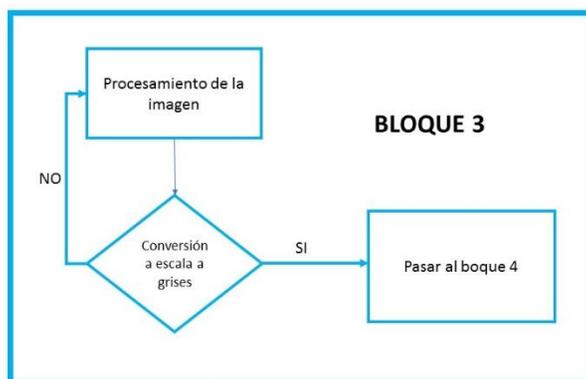


Figura 23 Tratamiento de imágenes

Fuente: Propia

✓ **Detección de Rostro.**

Una vez procesada la imagen se procede a detectar el rostro haciendo uso de OpenCv y sus librerías, para ello **haarcascade\_frontalface\_default.xml** se sigue el diagrama de la Figura 24.

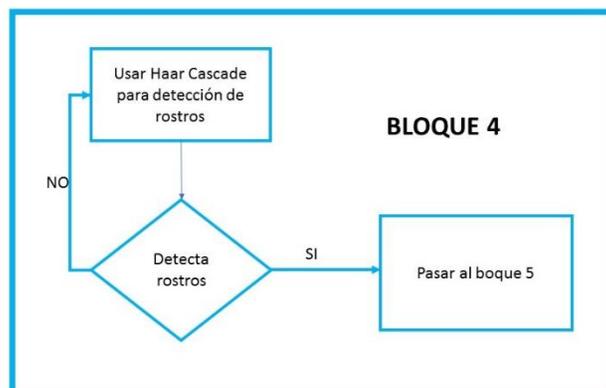


Figura 24 Detectar Rostros

Fuente: Propia

✓ **Detección de Ojos.**

Para la detección de ojos se utiliza 3 archivos haarcascade: **haarcascade\_eye\_tree\_eyeglasses.xml**, **haarcascade\_lefteye\_2splits.xml** y **haarcascade\_righteye\_2splits.xml** para ello se sigue el diagrama de la Figura 25.

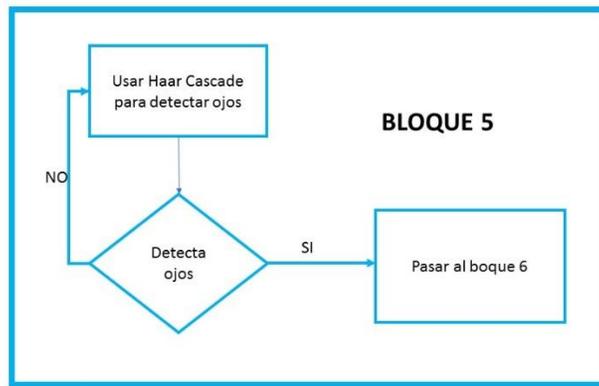


Figura 25 Detectar Ojos

Fuente: Propia

✓ **Detección de somnolencia mediante emparejamiento de plantilla.**

Una vez cumplida las fases anteriores se procesa a detectar la somnolencia siguiendo el proceso que muestra la Figura 26.

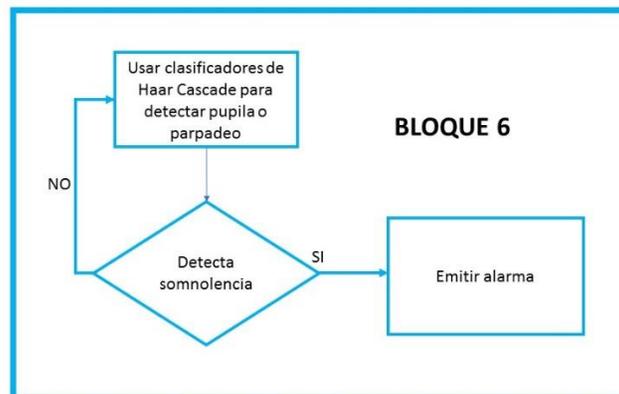


Figura 26 Detección de pupila

Fuente: Propia

## 2.4 Fase de Codificación e Implementación

En la presente fase se da a conocer toda la información que se realizó en la codificación del prototipo de la aplicación móvil. A continuación, se presenta los pasos que se siguió para el desarrollo de la aplicación:

### 2.4.1 Configuración del Entorno Android para la creación de la aplicación móvil.

Se descargó el instalador de Android Studio la versión más estable en lo concerniente se utilizó la versión 3.2.1.

Al momento de crear la aplicación es necesario seleccionar la opción de include C++ support porque OpenCV está desarrollado en lenguaje C ya que se podría necesitar posteriormente soporte para OpenCv.

Otro punto importante al momento de crear la aplicación es la selección de la API en la cual estará optimizada la aplicación para este caso se seleccionó el API 21 que abarca casi el 85% de los dispositivos que utilizan Android.

Finalmente se agregó la librería de OpenCV 3.3.1 para hacer uso de las herramientas de visión por computador, además de agregar openCv es necesario tener instalado CMake y NDK para poder reutilizar el código que está escrito en lenguaje C o C++ de OpenCv.

#### **2.4.2 Obtención de imágenes a través de la cámara frontal del smartphone**

OpenCv posee una clase específica llamada CameraBridgeViewBase, que se dedica a la iteración con las cámaras del smartphone y la librería de OpenCV, la misma posee métodos dedicados al procesamiento de los frames obtenidos.

Para ello se implementó dicha clase en el proyecto y se procede a acceder a la cámara frontal con el siguiente comando desde el código de la aplicación:

```
camera.setCameraIndex(1);
```

Donde camera es una variable de tipo CameraBridgeViewBase y un método llamado setCameraIndex que recibe el parámetro de la cámara a utilizar, respecto a esto se utilizó 1 ya que se necesitaba interactuar con la cámara frontal en caso de querer utilizar la cámara posterior solamente se cambiaría el parámetro de entrada al valor 0.

#### **2.4.3 Tratamiento de imágenes**

En Android estudio una vez integrado las librerías necesarias, se utilizó: Librería Mat para almacenar y para convertir la imagen a escala a grises como muestra la Figura 27.

Cuando inicia el aplicativo se debe inicializar la variable imgGris, así como imgColor.

```
imgGris = new Mat();  
imgColor = new Mat();
```

A la variable `imgColor` se le debe asignar lo que tenga el `inputFrame` convertido a una imagen de color RGBA y a la `imgGris` y se le asigna lo que tenga el `InputFrame` convertido a escala de grises, es necesario aclarar que estos medios son de OpenCV.

```
imgColor = inputFrame.cvtColor();
imgGris = inputFrame.cvtColor();
```



Figura 27 Imagen RGBA (izquierda) e Imagen transformada a escala a grises (derecha).

Fuente: Propia

Para ello se utilizó las siguientes líneas de código como se muestra en la Tabla 2.12 que indica cada una de las funciones utilizadas:

TABLA 2.11

CONVERSIÓN A ESCALA DE GRISES

<b>Algoritmo</b>	
<b>Input:</b>	<code>inputFrame</code> // <i>captura las imágenes que vienen desde la cámara frontal del smartphone.</i>
<b>Load:</b>	<code>CvCameraViewFrame</code> // <i>inicia la cámara con funciones de OpenCV.</i>
<b>Set:</b>	<code>rgba = inputFrame.cvtColor();</code> // <i>asigna a <code>imgColor</code> lo que tiene en el frame.</i>
<b>Train:</b>	<code>gray();</code> // <i>intenta convertir a escala de grises lo que tiene en el frame.</i>
<b>Output:</b>	<code>imgGris = inputFrame.cvtColor();</code> // <i>asigna el valor final del tratamiento de la imagen.</i>

Fuente: Propia

#### 2.4.4 Detección de Rostro

En lo que se refiere al reconocimiento facial OpenCv emplea clasificadores en cascada los cuales se basan en propiedades de Haar Like Features. El clasificador para el rostro es: lbpcascade\_frontalface.xml

Para poder comprender lo que realiza haarcascade se presenta el proceso que realiza como muestra en la Figura 28:

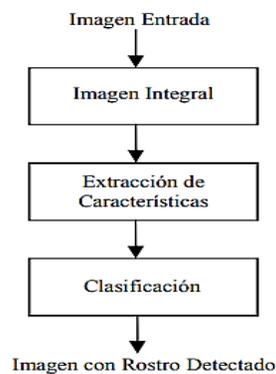


Figura 28 Proceso de HaarCascade

#### ✓ *Imagen Integral*

Para la detección de rostros se utiliza la metodología de Viola and Jones que se basa principalmente en tres etapas: cálculo de la imagen integral, extracción de características y la clasificación. El cálculo de la imagen integral permitirá extraer de forma rápida características a diferente escala ya que trabaja con imágenes acumulativas que se construyen a partir de operaciones básicas.

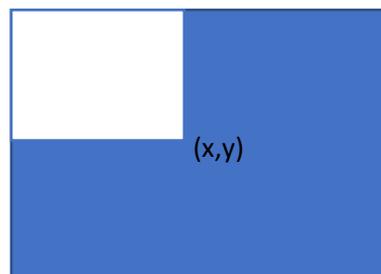


Figura 29 Imagen integral

Fuente: Propia

La imagen integral que se muestra en la figura 29 en la posición (x,y), contiene la suma de todos los pixeles de la parte superior izquierda de la imagen. La suma de todos los pixeles de cualquier rectángulo en la imagen como se representa a continuación:

Ec. [2.4.1]

Donde:

$II(x, y)$ : es la imagen integral.

$IM(x', y')$ : es la imagen original.

$$II(x, y) = \sum_{x' \leq x, y' \leq y} IM(x', y')$$

Donde  $II(x, y)$  es la imagen integral y  $IM(x', y')$  es la imagen original, con el cálculo de la anterior fórmula se puede obtener el valor de cualquier área de la imagen mediante la suma y resta de los valores situados en cada esquina, como se muestra en la Figura 30.

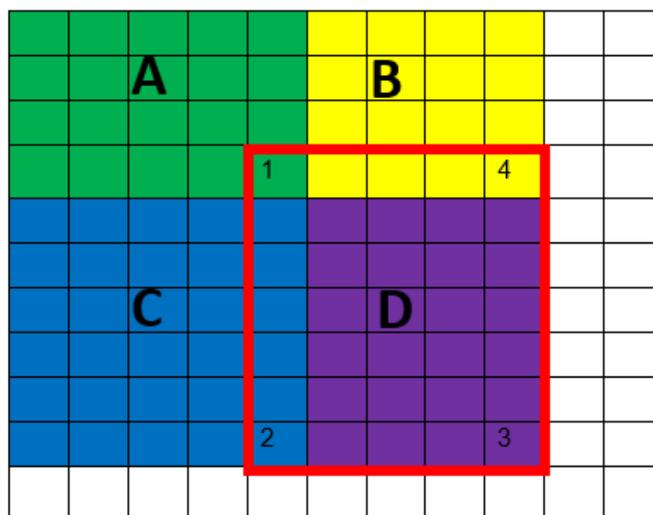


Figura 30 Cálculo de la Imagen Integral

Fuente: Propia

En la Figura 30 se muestra cómo se comporta la suma de los pixeles, en este caso para obtener la suma del elemento 1 sería todos los pixeles del recuadro verde (elemento A), para el elemento 4 sería todos los pixeles de los recuadros amarillo y verde (elemento A y B), para el elemento 2 sería todos los pixeles del recuadro azul y verde (elemento C y A) y

para elemento 3 sería todos los píxeles del recuadro morado, amarillo, verde y azul (elemento A, B, C y D). Entonces si se desea el elemento  $3+1-(2+4)$ .

#### ✓ *Extracción de Características*

Inicialmente, el algoritmo necesita muchas imágenes de rostros positivas e imágenes de rostros negativas para entrenar al clasificador. Entonces se necesita extraer características de él. Para esto, se utilizan las características de haar que se muestran en la Figura 31 a continuación.

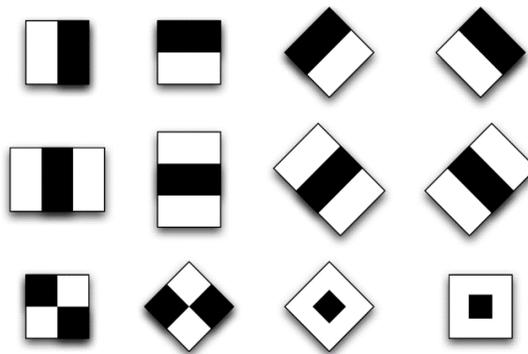


Figura 31 Características de Haar

### **Clasificación**

El objeto de la cara es una matriz general con varias matrices secundarias que se componen de cuatro números: x, y, ancho y alto de la cara detectada, por ejemplo:

```
[[256 97 115 115]
```

```
[ 80 68 107 107]]
```

Se puede explicar de la siguiente manera, significa que existen dos caras en la matriz o en la imagen con tamaños de (115,115) y con coordenadas (256,97) del primer rostro y (107,107) en el tamaño y coordenadas (80,68) del siguiente rostro.

Para comprobar su funcionamiento se dibujó un cuadrado en el área detectada del rostro con la siguiente línea de código:

Se utiliza funciones de OpenCv como es `Imgproc.rectangle` para dibujar líneas dando los puntos `tl()` y `br()` con un color en el espacio RGB con una línea de un grosor de 3 píxeles. En la

Figura 32 se puede apreciar la detección de rostros con el uso de *lbpcascade\_frontalface.xml*, y en la Tabla 2.13 se presenta el algoritmo que se sigue para realizar el prototipo.

TABLA 2.12  
DETECCIÓN DEL ROSTRO

<b>Algoritmo</b>	
<b>Input:</b>	<i>mGray // toma como referencia la imagen previamente obtenida con el algoritmo 1.</i>
<b>Load:</b>	<i>lbpcascade_frontalface.xml. // inicia el archivo xml de OpenCV entrenado para detectar rostros de frente.</i>
<b>Set:</b>	<i>MatOfRect faces = new MatOfRect();// crea una matriz para guardar la información del rostro detectado.</i>
<b>Train:</b>	<i>gray(); // intenta convertir a escala de grises lo que tiene en el frame</i>
<b>Output:</b>	<i>rectangle facesArray[i].tl(), facesArray[i].br(), // dibuja un rectángulo de color verde en el rostro detectado.</i>

Fuente: Propia

Para verificar el funcionamiento del código se dibujó un recuadro en el rostro detectado además del código que carga los archivos .xml y el código de tratamiento de la imagen en cuanto a escalas de reducción, tamaño máximo y mínimo que se considera para la detección del rostro, etc.

Código para cargar y leer el archivo entrenado para la detección de rostros.

```
InputStream is=getResources().openRawResource
(R.raw.lbpcascade_frontalface);
File cascadeDir = getDir("cascade", Context.MODE_PRIVATE);
mCascadeFile = new File(cascadeDir, "lbpcascade_frontalface.xml");
FileOutputStream os = new FileOutputStream(mCascadeFile);
```

Imágenes o frames procesados a escala de grises y a color que es el que se muestra en la pantalla del usuario.

```
imgColor = inputFrame.rgba();
imgGris = inputFrame.gray();
```

En esta parte del código se utiliza DetectMultiScale que la misma es una función de OpenCv y que sirve para detectar objetos de varios tamaños, en este caso se utilizó para detectar rostros, recibiendo algunos parámetros como por ejemplo la imagen de entrada que en este caso es imgGris, el vector que contiene el resultado de los rostros detectados con **lbpcascade\_frontalface.xml**, la escala que define el tamaño de reducción de la imagen de entrada este caso es el 10%, el número mínimo de vecinos para que se conserve un rostro detectado en este caso se puso como mínimo 2 vecinos, el tamaño mínimo en este caso se está utilizando los valores Viola and Jones.

```
detectMultiScale(mGray, rostros, 1.1, 2
//TODO:objdetect.CV_HAAR_SCALE_IMAGE      new Size(384, 288), new
Size());
```

Se almacena en la variable ArrayRostro, todos los rostros detectados en el orden adecuado haciendo uso del método toArray().

```
Rect[] ArrayRostro = rostros.toArray();
```

Se dibuja un recuadro de color blanco haciendo uso de funciones de OpenCv en este caso Imgproc.Rectangle que recibe como parámetros la imagen a color en este caso la que se visualiza por el usuario, los puntos de X y Y, el color de la línea y finalmente el ancho de la línea.

```
rectangle(imgColor, ArrayRostro[i].tl(), ArrayRostro[i].br(), Color, 3);
```



Figura 32 Detección de Rostros en la aplicación móvil

Fuente: Propia

## 2.4.5 Detección de Ojos

Una vez detectada la zona de interés en este caso el rostro y una vez confirmado que funciona correctamente, procedemos a detectar los ojos de la misma forma, pero en el marco del rostro detectado no en toda la imagen.

Se puede detectar la zona de la ubicación de los ojos para ello OpenCV proporciona varios archivos .xml que son archivos ya entrenados precisamente para esta función como por ejemplo: clasificadores para detectar ojos abiertos: **haarcascade\_eye.xml**, **haarcascade\_eye\_tree\_eyeglasses.xml**.

Se debe cargar los clasificadores o archivos entrenados en este caso son los clasificadores para el ojo izquierdo, derecho y abierto de la siguiente forma:

Código para cargar y leer el archivo entrenado para la detección del ojo derecho.

```
InputStream isod = getResources ().openRawResource
    (haarcascade_righteye_2splits);
File cascadeDirOD = getDir("cascadeOD", Context.MODE_PRIVATE);
cascadeFileOD = new File(cascadeDirOD, "haarcascade_eye_right.xml");
FileOutputStream oser = new FileOutputStream(cascadeFileOD);
```

Código para cargar y leer el archivo entrenado para la detección del ojo izquierdo.

```
InputStream isoi = getResources ().openRawResource
    (R.raw.haarcascade_lefteye_2splits);
File cascadeDirOI = getDir("cascadeOI", Context.MODE_PRIVATE);
cascadeFileOI = new File(cascadeDirOI, "haarcascade_eye_left.xml");
FileOutputStream osoi = new FileOutputStream(cascadeFileOI);
```

Código para cargar y leer el archivo entrenado para la detección de ojos abiertos.

```
InputStream isoa = getResources ().openRawResource
    (haarcascade_eye_tree_eyeglasses);
File cascadeDirOjoAbierto = getDir("cascadeOjoAbierto",
    Context.MODE_PRIVATE);
cascadeFileOjoAbierto = new File(cascadeDirOjoAbierto,
    "haarcascade_eye_tree_eyeglasses.xml");
FileOutputStream osoa = new FileOutputStream(cascadeFileOjoAbierto);
```

El primer paso para detectar la zona de la ubicación de los ojos es extraer la nueva matriz del rostro en el que se realizó lo siguiente:

```
Rect RectadelRostro = ArrayRostro[i];
```

A medida que se encuentra el rostro, el ROI (Región de Interés) disminuye ahora se debe detectar la zona de los ojos, para ello se debe calcular la medida del rostro ya que los ojos están siempre ubicados en la parte superior del rostro es decir de la mitad hacia arriba.

Cada vez que se encuentre el ROI es necesario hacer una nueva matriz para realizar cálculos correspondientes y ahorrar procesamiento computacional o ahorro de recursos de procesamiento.

Para determinar el rectángulo del ojo derecho se realizó el cálculo de la siguiente fórmula matemática, en el que se debe calcular los puntos de x, y, ancho y alto de la siguiente forma:

Donde:

RectadelRostro.x: Es el punto x que se utilizó para dibujar el recuadro del rostro detectado.

RectadelRostro.width: Es el ancho que se utilizó para dibujar el recuadro del rostro detectado.

RectadelRostro.y: Es el punto y que se utilizó para dibujar el recuadro del rostro detectado.

RectadelRostro.height: Es el alto que se utilizó para dibujar el recuadro del rostro detectado.

Para el cálculo del punto **X** de la nueva Recta llamada areaOjo\_Der se encontró los puntos x, y, ancho y alto, para ello se procede a encontrar primeramente el punto x de la siguiente manera:

Ec. [2.1.2]

$$punto X = RectadelRostro.x + \frac{RectadelRostro.width}{16}$$

Para el cálculo del punto **Y** se realizó la siguiente operación matemática.

Ec. [2.1.3]

$$punto Y = RectadelRostro.y + \frac{RectadelRostro.height}{4.5}$$

Para el cálculo del **ANCHO** se realizó la siguiente operación matemática.

Ec. [2.1.4]

$$Ancho = \frac{\frac{RectadelRostro.width - 2 * RectadelRostro.width}{16}}{2}$$

Para el cálculo del **ALTO** se realizó la siguiente operación matemática.

Ec. [2.1.5]

$$Alto = \frac{RectadelRostro.height}{3}$$

En la Figura 33 se muestra cómo debería quedar la solución de dibujar el área de la posición del ojo derecho:

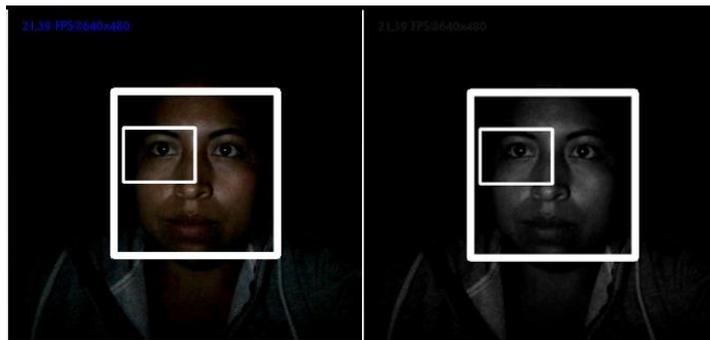


Figura 33 Detección de Área de Ojo Derecho

Fuente: Propia

Una vez calculado los puntos del ojo derecho se continúa con los puntos del ojo izquierdo, para ello se sigue la misma lógica de los cálculos que se hizo previamente:

Para el cálculo del punto **X** en el ojo izquierdo se realizó la siguiente operación matemática.

Ec. [2.1.6]

$$punto X = RectadelRostro.x + \frac{RectadelRostro.width}{16} + \frac{\left( RectadelRostro.width - 2 * \frac{RectadelRostro.width}{16} \right)}{2}$$

Para el cálculo del punto **Y** se realizó la siguiente operación matemática.

Ec. [2.1.7]

$$punto Y = RectadelRostro.y + \frac{RectadelRostro.height}{4.5}$$

Para el cálculo del **ANCHO** se realizó la siguiente operación matemática.

Ec. [2.1.8]

$$Ancho = \frac{\left( RectadelRostro.width - 2 * \frac{RectadelRostro.width}{16} \right)}{2}$$

Para el cálculo del **ALTO** se realizó la siguiente operación matemática.

Ec. [2.1.9]

$$Alto = \frac{RectadelRostro.height}{3}$$

En la Figura 34 se muestra cómo debería quedar la solución de dibujar el área de la posición del ojo izquierdo:

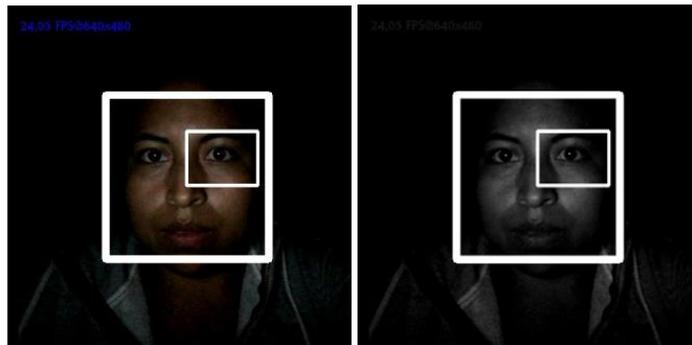


Figura 34 Detección de Área de Ojo Izquierdo

Fuente: Propia

Para ver el funcionamiento del cálculo de las fórmulas se procede a dibujar los rectángulos con los puntos calculados previamente, el resultado se muestra en la Figura 35.

```
Imgproc.rectangle(framesalida, areaOjo_Izq.tl (), areaOjo_Izq.br (), new  
color, 2);  
Imgproc.rectangle(framesalida, areaOjo_Der.tl (), areaOjo_Der.br (), new  
color, 2);
```

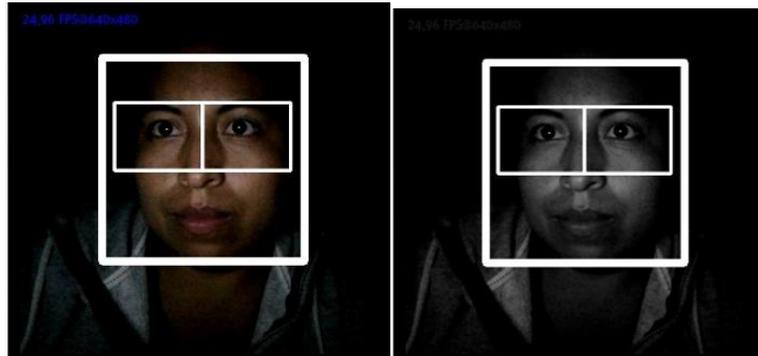


Figura 35 Detección de Ojos

Fuente: Propia

El pseudocódigo que se muestra en la Tabla 2.14 que permite la detección de la región donde se encuentran los ojos, cabe mencionar que estas regiones se obtuvieron haciendo los cálculos descritos previamente.

TABLA 2.13

DETECCIÓN DEL ÁREA DE LOS OJOS

**PSEUDOCÓDIGO**

---

**Input:** `Rect RectadelRostro = ArrayRostro[i];` // toma como referencia la matriz del algoritmo 2 de detección de rostros y empieza a trabajar en la zona objetivo u zona de interés.

**Load:** `haarcascade_eye_tree_eyeglasses.xml,`  
`haarcascade_eye_left.xml,`  
`haarcascade_eye_right.xml` // inicia los archivos xml de OpenCV entrenado para detectar ojo derecho, izquierdo y ojos abiertos.

**Set:** `areaOjo_Der` y `areaOjo_Izq` // es el resultado de hacer cálculos, los mismos que contienen posiciones x,y, ancho y alto de la ubicación de los ojos.

**Output:** `Rectangle (framesalida, areaOjo_Izq.tl (), areaOjo_Izq.br ());`  
`rectangle(framesalida, areaOjo_Der.tl(), areaOjo_Der.br())`  
 dibuja un rectángulo de color rojo tanto para el ojo derecho como el izquierdo.

---

Fuente: Propia

Como ya se localizó el área de la ubicación de los ojos entonces el siguiente paso es utilizar los clasificadores entrenados para esta función.

Para ello se creó un nuevo método con parámetros como el clasificador del ojo derecho, así como también del ojo izquierdo y además el área de interés, es decir, en este caso la ubicación de la zona de los dos ojos.

```
obtener_plantilla(CascadeClassifier clasificador, Rect RectAreaInteres)
```

Se creó la nueva región de interés realizando una extracción de una matriz pequeña a partir de la matriz general de la zona de interés (área del ojo derecho o izquierdo) con la siguiente línea de código.

```
Mat ROIojo = imgGris.submat(RectAreaInteres);
```

Para almacenar los ojos detectados en la zona de interés, se creó una nueva matriz de Rectas llamada ojos y se hace uso de la función detectMultiScale con los parámetros que se ajusten a la necesidad del desarrollador.

```
MatOfRect ojos = new MatOfRect();  
detectMultiScale(ROIojo, ojos, 1.1, 2, new Size(), new Size());
```

Para convertirle en una recta a la matriz de rectas se hizo un ciclo for para que recorra todo el ArrayOjos y el resultado se almacenó en la recta de Ojo Detectado para el cálculo de los puntos X y Y se hizo los siguientes cálculos matemáticos, el ojo detectado será almacenado en ROIojo, es decir, extrayendo una matriz pequeña con el valor del ojo detectado.

```
Rect[] ArrayOjos = ojos.toArray();  
for (int i = 0; i < ArrayOjos.length; ) {  
Rect OjoDetectado = ArrayOjos[i];  
OjoDetectado.x = RectAreaInteres.x + OjoDetectado.x;  
OjoDetectado.y = RectAreaInteres.y + OjoDetectado.y;
```

Donde:

*RectAreaInteres.x* = Es el punto x del área del ojo derecho o izquierdo.

*RectAreaInteres.y* = Es el punto y del área del ojo derecho o izquierdo.

Para el cálculo del punto **X** en el ojo detectado se realiza la siguiente operación.

Ec. [2.1.9]

$$OjoDetectado.x = RectAreaInteres.x + OjoDetectado.x;$$

Para el cálculo del punto **Y** en el ojo detectado se realiza la siguiente operación.

Ec. [2.1.9]

$$OjoDetectado.y = RectAreaInteres.y + OjoDetectado.y;$$

Ahora para guardar el recuadro del ojo detectado se hizo la operación de submat y se lo almacena en la variable ROlojo.

```
ROIojo = imgGris.submat(OjoDetectado);
```

Ahora el siguiente paso es detectar el iris del ojo en la región de interés que se almacenó en ROlojo, para ello se utiliza algunas funciones que nos ofrece OpenCV, una de ellas es minmaxLoc (que encuentra los valores mínimos y máximos de una matriz en este caso ROlojo).

```
minmaxLoc= Core.minMaxLoc(ROIojo);
```

Para calcular el punto **X** del IRIS se realiza los siguientes cálculos:

Donde:

**minmaxLoc.minLoc.x**: lo que hace es obtener el valor mínimo en el punto x ya que es el más optimo en este proyecto.

**minmaxLoc.minLoc.y**: lo que hace es obtener el valor mínimo en el punto y ya que se hizo las pruebas respectivas y es el más adecuado para la realización del proyecto.

Para el cálculo del punto **x** en el iris se realiza la siguiente operación.

$$iris.x = minmaxLoc.minLoc.x + OjoDetectado.x;$$

Ec. [2.1.9]

Para el cálculo del punto **y** en el iris se realiza la siguiente operación.

$$iris.y = minmaxLoc.minLoc.y + OjoDetectado.y;$$

Para el cálculo de la plantilla final del ojo se lo realizo de la siguiente forma:

Donde:

OjoDetectado.width: es el valor del ancho del ojo detectado.

OjoDetectado.height: es el valor del alto del ojo detectado.

iris.x: es el punto x de la posición del iris.

iris.y: es el punto y de la posición del iris.

Fórmula general para la plantilla final de ojo:

Ec. [2.1.9]

$$plantilla\_ojo = iris.x - \frac{OjoDetectado.width}{2}, \\ iris.y - \frac{OjoDetectado.height}{2}, OjoDetectado.width, OjoDetectado.height)$$

Cálculo del punto **x** de la plantilla del ojo:

Ec. [2.1.9]

$$iris.x = \frac{OjoDetectado.width}{2}$$

Cálculo del punto **y** de la plantilla del ojo:

Ec. [2.1.9]

$$iris.y = \frac{OjoDetectado.height}{2}$$

Para el **ancho** de la plantilla del ojo solo se debe tomar los valores del ojo detectado con el ancho correspondiente:

$$OjoDetectado.width$$

Para el **alto** de la plantilla del ojo solo se debe tomar los valores del ojo detectado con el alto correspondiente:

$$OjoDetectado.height$$

Para visualizar si el método está funcionando correctamente, se creó una variable de tipo Rect con el nombre de rectOder y de la misma forma para el ojo izquierdo, se pasa los parámetros necesarios (el clasificador entrenado y el área del ojo derecho), el código quedaría de la siguiente forma:

```
rectOder = obtener_plantilla(DetectorOjoD, OjoD);  
rectOizq = obtener_plantilla(DetectorOjoI, OjoI);
```

En la obtención de la plantilla del ojo abierto derecho se hizo una sobrecarga al método que obtiene la plantilla cambiando los parámetros que necesita el método, en este caso necesita el clasificador (ojos abiertos), área de interés (recta del ojo derecho o izquierdo), y el tamaño mínimo y máximo de la siguiente forma:

```
obtener_plantilla(CascadeClassifier clasificador, Rect  
RectAreaInteres)
```

Para verificar si funciona correctamente se dibujó un recuadro donde está siendo detectado el ojo tanto derecho como izquierdo, como se muestra en la Figura 36:

```
rectangle(imgColor, Oder.tl(),Oder.br(), color, 2);  
rectangle(imgColor, Oizq.tl(),Oizq.br(), color), 2);
```

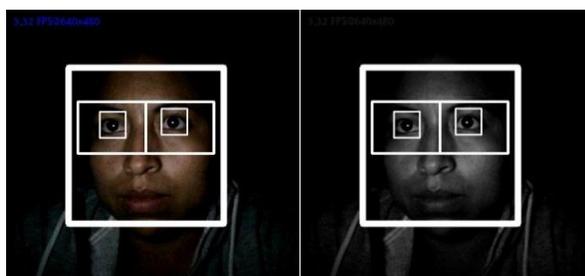


Figura 36 Detección de Ojos

Fuente: Propia

#### 2.4.6 Detección de pupila por medio de emparejamiento de plantillas

Una vez cumplidas las fases de: detección de la zonas de interés en este caso el rectángulo del ojo derecho e izquierdo, detectar el ojo derecho e izquierdo haciendo uso de los clasificadores Haar Cascade, realizar la plantillas de ojos abiertos (derecho e izquierdo), lo último que hace falta es verificar si los ojos se encuentran abiertos o cerrados, referente a esto se crea un método encargado de realizar la coincidencia de plantillas, su función es buscar y encontrar la ubicación de una imagen plantilla en una más grande, esta es una técnica de OpenCV que se utilizó en este proyecto.

Para ello se desarrolla un método que realiza lo mismo que el método obtener plantilla de los ojos derecho e izquierdo, solamente se cambia los parámetros de entrada en este caso pide el clasificador de ojos abiertos de haar cascade, el área de interés, el tamaño máximo y mínimo de la plantilla.

```
clasificador.detectMultiScale(ROIojo, ojos, 1.01, 2, mínimo, máximo);
```

Este método posteriormente fue usado para extraer las plantillas de los ojos solamente el ojo derecho e izquierdo que estén abiertos. Para ello se utilizó el siguiente código:

```
Derecho = obtener_plantilla(clasificador, OjoDer, min_size, max_size);  
plantillaD_abierto = Gris.submat(Derecho);  
  
Izquierdo= obtener_plantilla(clasificador, OjoIzq, min_size, max_size);  
plantillaI_abierto = Gris.submat(Izquierdo);
```

El siguiente paso es realizar un método para la coincidencia de plantillas, es decir, para comparar las plantillas que se ha conseguido con los anteriores métodos, con el objetivo de determinar si el ojo derecho e izquierdo están abiertos o cerrados y se lo realizó de la siguiente forma:

Método para coincidencia de plantillas de tipo booleano para que retorne true or false:

```
coincidencia_ojo(Mat plantillaODI)
```

Para este proyecto se planteó usar el método de emparejamiento de plantillas para realizar una búsqueda de objetos en la imagen, para este caso la búsqueda de la pupila en la región de interés que es la matriz del ojo derecho e izquierdo, la operación que realiza esta técnica es establecer un coste que mide cuanto es la similitud entre la plantilla y la imagen de prueba, como se mencionó previamente para el uso de esta técnica se utiliza la librería de OpenCV, el proceso de la técnica es realizar un barrido de un pixel a la vez de la siguiente forma: de izquierda a derecha y de arriba hacia abajo, en cada uno de estos procesos realiza el cálculo de una métrica estableciendo si la comparación resulta o eficiente.

El algoritmo de Template Matching. se desliza a través de la imagen comparando el Template (patrón) usando el método específico, almacenando el resultado disponible el cual se puede encontrar como mínimos globales cuando se utiliza el método TM\_SQDIFF o máximos cuando se manejan los métodos de TM\_CCORRO TM\_CCOEFF(Ponce et al., 2018).

Aquí se utilizó el método TM\_SQDIFF para encontrar los mínimos globales de la siguiente forma, para poder encontrar el punto más oscuro en la región de interés:

```
minmaxLoc = Core.minMaxLoc(ROIojo);
```

Es necesario mencionar que los cálculos operacionales que realizan internamente las mencionadas técnicas son: cálculo de la Transformada Rápida de Fourier, es una técnica muy utilizada en el procesamiento de imágenes, en este caso se utiliza para encontrar el dominio de la frecuencia ya que cada imagen en este caso la plantilla y la imagen donde se va a buscar son

almacenados en arreglos y cada uno de sus elementos se le conoce como píxel y el siguiente paso es asignar un valor numérico a la luminosidad promedio de cada píxel con la finalidad de mejorar la calidad de la misma como se muestra en la Figura 37:

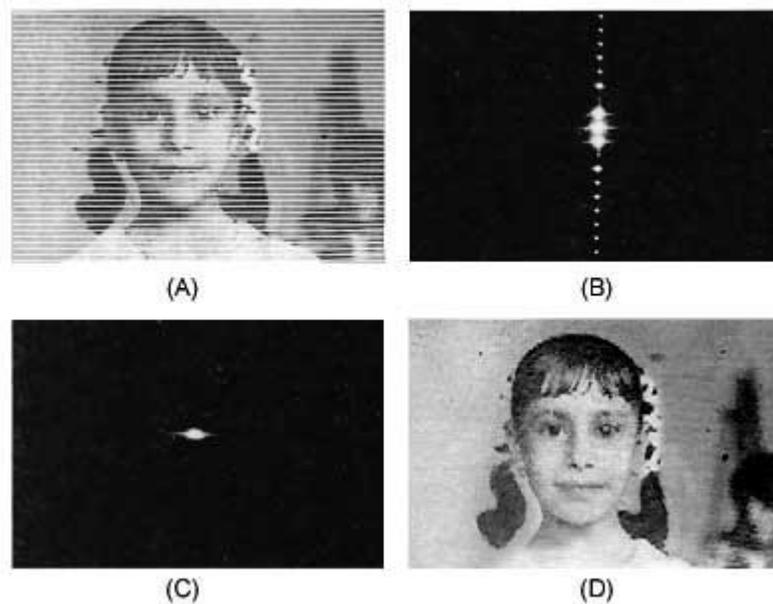


Figura 37 Procesamiento óptico de imágenes. (a) imagen original, con líneas de barrido, tipo imagen de televisión; (b) transformada de Fourier del objeto; (c) transformada de Fourier modificada, después de filtrar y (d) imagen procesada, sin las líneas de barrido

El siguiente paso es alterar los valores de la luminosidad de los píxeles mediante las operaciones o transformaciones matemáticas necesarias, a fin de resaltar los detalles de la imagen que sean convenientes.

#### ✓ Fórmula de la Transformada Rápida de Fourier

La Transformada Rápida de Fourier es muy utilizada en el análisis de algoritmos ya que ofrece mayor eficiencia tanto en tiempo como en recursos y por ello que las funciones usadas para el prototipo hacen uso internamente de esta fórmula por su precisión y rapidez (Lucía, 2013)

$$x[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{kn} = \sum_{r=0}^{N/2-1} x[2r] \cdot W_N^{rk} + \sum_{r=0}^{N/2-1} x[2r+1] \cdot W_N^{(2r+1)k} \quad r = 0, 1, \dots, \frac{N}{2} - 1$$

$$x[k] = \sum_{r=0}^{N/2-1} x[2r] \cdot (W_N^2)^{rk} + W_N^k \sum_{r=0}^{N/2-1} x[2r+1] \cdot (W_N^2)^{rk}$$

Además del uso de la Transformada Rápida de Fourier también se hace uso de la ecualización de histogramas con la finalidad de obtener para una imagen un histograma con una distribución uniforme, es decir, que exista el mismo número de píxeles para cada nivel de gris del histograma de una imagen monocroma.

La función de la ecualización de histogramas es:

$$v(i, j) = \frac{Fu(u(i, j) - Fv(a))}{1 - Fu(a)}(L - 1) + 0.5$$

Lo que quiere decir que cada píxel toma un valor entero entre 0 y 255 para asignarle un color, generando una escala entre estos valores para cada canal. De esta manera, las zonas más oscuras de una imagen estarán representadas por píxeles con valores cercanos a 0, mientras que las zonas más claras con píxeles cercanos a 255.

Por ende, en el prototipo, la aplicación de estas funciones da como resultado el punto más oscuro del ojo, es decir, la pupila ya que luego de hacer internamente todas las operaciones matemáticas dan como resultado la ubicación de la pupila en el ROI deseado.

Entonces el uso de la función minmaxLoc facilita el cálculo operacional y disminución de recursos computacionales, dando como resultado la ubicación del objetivo como se muestra en la Figura 38.

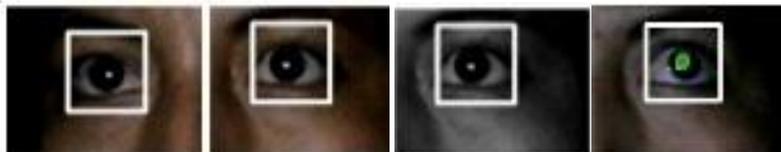


Figura 38 Área de Interés y Solución

Fuente: Propia

Finalmente se hace una operación condicional para saber si existe valores de 0 tanto en las columnas o en las filas y se asigna el valor de retorno de la operación.

```
if (plantillaODI.cols == 0 || plantillaODI.rows == 0)
    return false;
else return true;
```

Se hace uso del método con los parámetros asignados para poder hacer las preguntas si el usuario esta con los ojos abiertos o cerrados.

```
OjoAbierto_D = coincidencia_ojo(plantillaD);  
OjoAbierto_I = coincidencia_ojo(plantillaI);
```

### **Emparejamiento de plantillas**

La principal función del trabajo es realizar la detección de pupila por emparejamiento de plantillas es decir buscar una imagen pequeña en una imagen más grande, según (Stone & Li, 1996) expresa que existen dos tipos de búsqueda para clasificar si una imagen se parece a la otra o no, la primera es el emparejamiento de plantillas y el otro es la clasificación mediante un aprendizaje supervisado o no.

La técnica utilizada es la primera ya que el resultado de la búsqueda mediante operaciones matemáticas indicase si existe coincidencia o no y se puede decir que esta técnica es aplicable a cualquier tipo de imagen. El trabajo realizado busca una imagen de acuerdo con la intensidad que poseen los pixeles, pero siempre y cuando la imagen este convertida a escala a grises, se puede buscar ya sea por la mayor o menor intensidad de cada uno de los pixeles de la imagen.

Lo que se realizó en este proyecto es hacer coincidir una plantilla con una imagen de igual tamaño o más grande mediante el criterio de búsqueda de intensidad, se dice que una o cualquier región de la imagen es coincidente si su intensidad promedio es cercana a la intensidad de la plantilla, como se muestra a continuación en las siguientes figuras:

Supongamos que la figura 39 es la plantilla para buscar en las imágenes:



Figura 39 Plantilla de camión

La figura 40 muestran el resultado de búsqueda en la imagen:

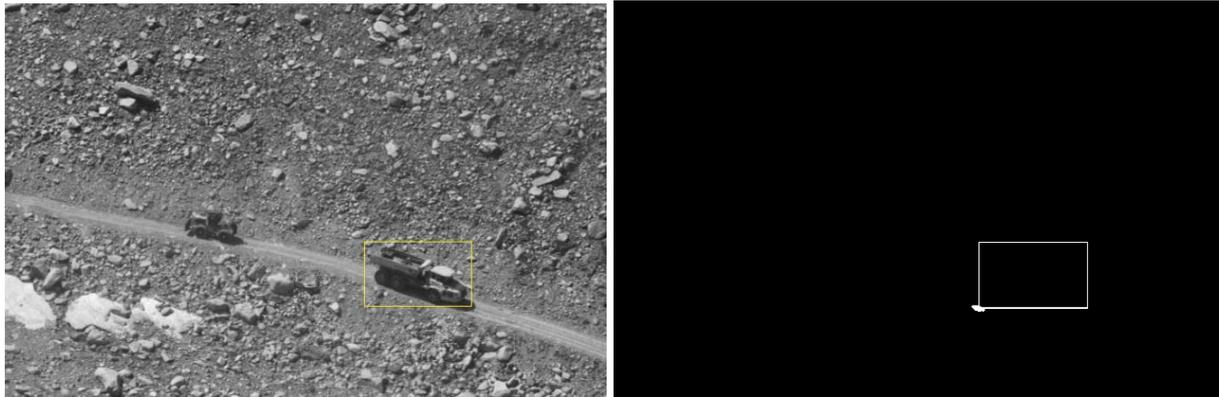


Figura 40 ubicación del objetivo encontrado(izquierda) - determinación del valor máximo de intensidad(derecha)

### 2.4.7 Detección de somnolencia

Ahora finalmente se tiene la plantilla del ojo derecho abierto y del ojo izquierdo abierto, si ambas se cumplen emitirá una etiqueta que señale que los ojos están abiertos y caso contrario indicará que los ojos cerrados y la variable de FrameSomnolencia se aumenta el contador en 1, esta variable se utilizó posteriormente para verificar si existe somnolencia o no.

Para verificar que un usuario este en estado de somnolencia se determinó que, si cuenta dos frames consecutivos de ojos cerrados, emita la etiqueta que diga alerta y eso nos indica que el usuario está dormido como se muestra en la Figura 41.

```
FramesSomnolencia++;
if (FramesSomnolencia >= 1)
{Imgproc.putText( "Alerta");}
```



Figura 41 Aplicación Móvil - Somnolencia

Fuente: Propia

A continuación, en la Tabla 2.15, se muestra algunos resultados en la tabla con personas que están en estado de somnolencia:

TABLA 2.14

USUARIOS EN ESTADO DE SOMNOLENCIA



Fuente: Propia

### 2.4.8 Emisión de alarma

Para la emisión de la alarma se consideró el número de frames seguidos con el estado del ojo cerrado, en este caso se contabilizó 2 frames consecutivos con los ojos cerrados para que emita la alarma sonora, como se detalla en la Tabla 2.16.

TABLA 2.15

EMISIÓN DE ALARMA

<b>PSEUDOCÓDIGO</b>	
<b>Input:</b>	FrameOjosCerrados >2 // entra la variable frame ojos cerrados con la cantidad de frames consecutivos con el

---

*estado de los ojos cerrados y con la condición de que si la variable es mayor a 2.*

**Load:** *mediaPlayer // inicia el reproductor de audio de Android Studio.*

**Set:** *MediaPlayer.create(this,R.raw.sos);// se busca el audio .mp3 en la carpeta raw de nuestro proyecto y lo cargamos a la aplicación.*

**Output:** *mediaPlayer.start(); // si se ha cumplido la condición inicial el sonido se activará con la función start() de Media Player. Log.e(TAG, "Fallo en reproducción del sonido"+e);// en caso de que el sonido presente un fallo se enviara un etiqueta en log out de nuestra aplicación que el sonido no se pudo reproducir*

---

Fuente: Propia

Esto se lo logro con el siguiente código, además el contador se lo debe reiniciar en 0:

```
final MediaPlayer = MediaPlayer.create(this, sonido);
try {
    @Override
    public void onPrepared(MediaPlayer arg0) {
        mediaPlayer.start();
    }
} catch (Exception e) {
    Log.e("Fallo la reproducción del audio " + e);
}
FramesSomnolencia = 0;
```

#### **2.4.9 Solución al problema de poca luminosidad en conducción nocturna**

Para solucionar los problemas de luminosidad, se llevaron a cabo pruebas empleando varios métodos. Sin embargo, el que mejor funcionó fue el apoyo de luminosidad con luz de los leds de la pantalla dejando una franja de color blanco en el lado derecho, izquierdo y la parte inferior de la pantalla, de la siguiente forma, como se muestra en la Figura 42:

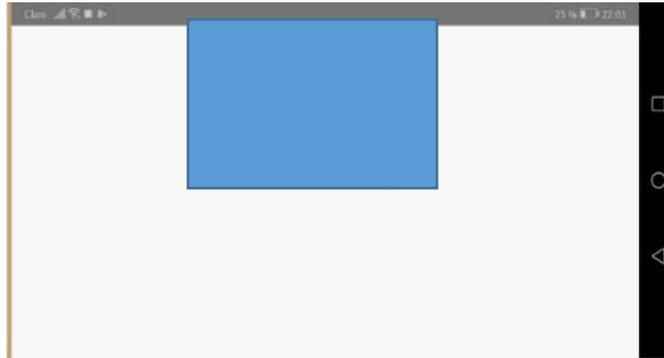


Figura 42 Iluminación de los leds del smartphone

Fuente: Propia

Lo que permite estas franjas de color blanco es la iluminación en una parte del rostro para que sea más fácil la detección del rostro y los ojos que es el principal objetivo.

En el aplicativo la dimensión que se dejó como medidas estándares son las siguientes: Para la medida de inicio desde donde se va a empezar a visualizar los frames en la pantalla y hasta donde va a ocupar la pantalla que muestra los frames al usuario:

```
android:layout_marginStart="170dp"
```

```
android:layout_height="145dp"
```

## 2.5 Fase de Pruebas

En esta fase se probó el prototipo de la aplicación móvil de detección de somnolencia nocturna en ambientes reales de conducción y en ambientes controlados, para ello se tomó una muestra de 56 personas entre ellos estudiantes de la Universidad Técnica del Norte, a continuación, en la Figura 43 se presenta algunas imágenes de las pruebas realizadas.



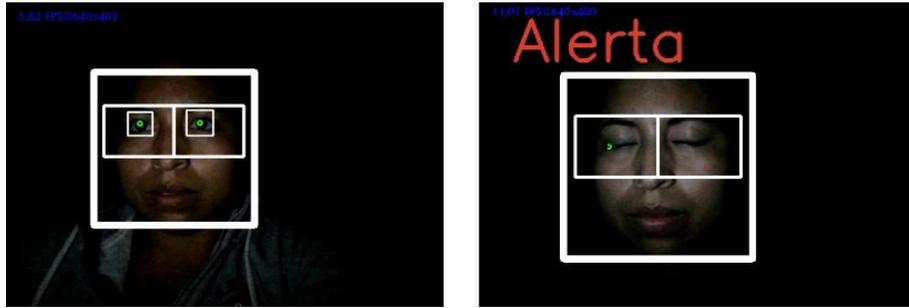


Figura 43 Personas sin estado y con estado de somnolencia

Fuente: Propia

Para las pruebas de funcionamiento se desarrolló dos prototipos de la aplicación una específicamente para la detección de rostros y la otra para detección de ojos como se muestra en la Figura 44.



Figura 44 Prototipo realizados para efectuar pruebas

Fuente: Propia

Para la prueba se consideró el número de frames que deberían contabilizarse para efectos de las pruebas, concluyendo que la aplicación dedicada a la detección de ojos procesaría un numero de 300 frames y se cerraría la aplicación. Para la prueba de detección de ojos se consideró el número de frames que deberían procesarse en este caso se condicionó que la aplicación debería estar en funcionamiento hasta que procese 300 frames donde detecte rostros, con la finalidad de obtener una mayor suficiencia de muestreo debido a que la aplicación se cerraba rápidamente, finalmente los resultados se imprimirán como se muestra en la Figura 45.

```
I/OCVSample::Activity: Numero de Rostros: 297    I/OCVSample::Activity: Numero de Frames: 301
Numero Ojos Abiertos: 134
Numero Ojos Cerrados: 96
Frame Total: 300
Numero Rostros: 300
```

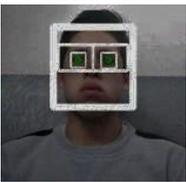
Figura 45 Resultados de pruebas en consola

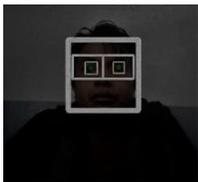
Fuente: Propia

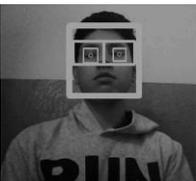
Cabe mencionar que la recopilación de datos para evaluar la aplicación móvil adecuadamente es un reto, ya que no se tiene los ambientes necesarios y las personas indicadas para evaluar la fatiga del conductor en condiciones reales, cabe mencionar que son pruebas que se las realiza con una diversidad de personas y con criterios que se asemejen a la realidad de la conducción en ambientes nocturnos y ambientes controlados.

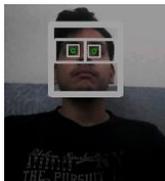
### 2.5.1 Resultados de Pruebas de Detección de rostros

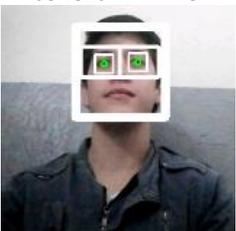
A continuación, se presenta diferentes pruebas de la aplicación móvil, específicamente en la detección de rostros, a varios usuarios.

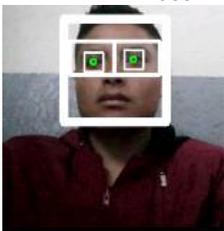
		Predicción de Rostros	
		Rostros	No Rostros
		299	1
Valores Actuales De Rostros	Rostros		
	No Rostros	0	300

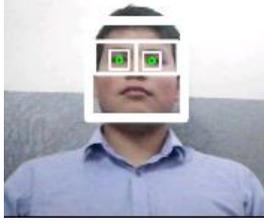
		Predicción de Rostros	
		Rostros	No Rostros
		299	1
Valores Actuales De Rostros	Rostros		
	No Rostros	0	300

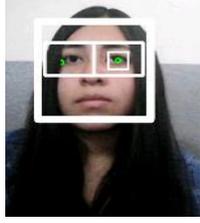
		Predicción de Rostros	
		Rostros	No Rostros
		299	1
Valores Actuales De Rostros	Rostros		
	No Rostros	2	298

		Predicción de Rostros	
		Rostros	No Rostros
		296	4
Valores Actuales De Rostros	Rostros		
	No Rostros	1	299

		Predicción de Somnolencia	
		Rostros	No Rostros
		298	2
Valores Actuales De Rostros	Rostros		
	No Rostros	12	288

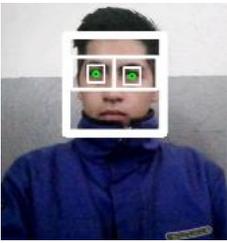
		Predicción de Rostros	
		Rostros	No Rostros
		300	0
Valores Actuales De Somnolencia	Rostros		
	No Rostros		3000

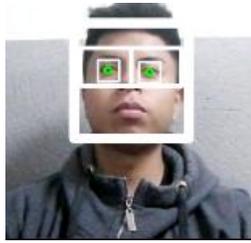
<b>ULICES VALENCIA</b>		<b>Predicción de Rostros</b>	
		Rostros	No Rostros
<b>Valores Actuales De Rostros</b>	Rostros	299	1
	No Rostros	3	297

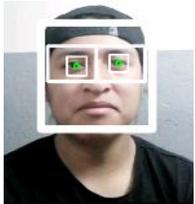
<b>JOHANNA PUERCHAMBUD</b>		<b>Predicción de Rostros</b>	
		Rostros	No Rostros
<b>Valores Actuales De Rostros</b>	Rostros	300	0
	No Rostros	37	263

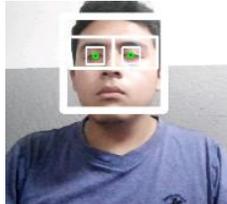
### 2.5.2 Resultados de Pruebas de detección de Ojos

A continuación, se presenta diferentes pruebas de la aplicación móvil, específicamente en la detección de ojos, a varios usuarios en diferentes ambientes de luminosidad.

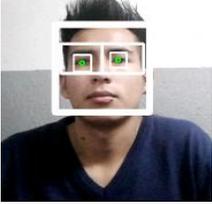
<b>ERICK FLORES</b>		<b>Predicción de Ojos</b>	
		Ojos	No Ojos
<b>Valores Actuales De Ojos</b>	Ojos	327	73
	No Ojos	0	400

<b>CARLOS IPIALES</b>		<b>Predicción de Ojos</b>	
		Ojos	No Ojos
<b>Valores Actuales De Ojos</b>	Ojos	344	56
	No Ojos	14	386

<b>KEVIN LOPEZ</b>		<b>Predicción de Ojos</b>	
		Ojos	No Ojos
<b>Valores Actuales De Ojos</b>	Ojos	349	51
	No Ojos	7	393

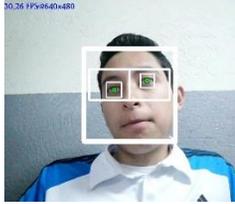
<b>KEVIN VINUEZA</b>		<b>Predicción de Ojos</b>	
		Ojos	No Ojos
<b>Valores Actuales De Ojos</b>	Ojos	397	3
	No Ojos	0	400

**HENRY MEZA** **Predicción de Ojos**



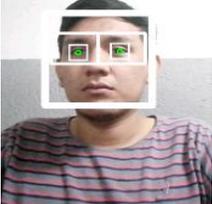
	Ojos	No Ojos
<b>Valores</b>	<b>390</b>	<b>10</b>
<b>Actuales</b>	<b>0</b>	<b>400</b>
<b>De</b>	<b>Ojos</b>	<b>No Ojos</b>
<b>Somnolencia</b>		

**KEVIN MENDEZ** **Predicción de Ojos**



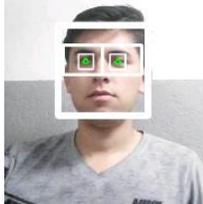
	Ojos	No Ojos
<b>Valores</b>	<b>398</b>	<b>2</b>
<b>Actuales</b>	<b>0</b>	<b>400</b>
<b>De</b>	<b>Ojos</b>	<b>No Ojos</b>
<b>Somnolencia</b>		

**OLIVER ZAMORA** **Predicción de Ojos**



	Ojos	No Ojos
<b>Valores</b>	<b>398</b>	<b>2</b>
<b>Actuales</b>	<b>0</b>	<b>400</b>
<b>De Ojos</b>	<b>Ojos</b>	<b>No Ojos</b>

**EDWIN HERRERA** **Predicción de Ojos**



	Ojos	No Ojos
<b>Valores</b>	<b>349</b>	<b>51</b>
<b>Actuales</b>	<b>13</b>	<b>387</b>
<b>De Ojos</b>	<b>Ojos</b>	<b>No Ojos</b>

# CAPÍTULO 3

## Validación e Interpretación de Resultados

### 3.1 Muestreo

Para evaluar si el prototipo de la aplicación móvil está funcionando adecuadamente se hizo varias pruebas con diferentes personas entre ellas estudiantes de la Universidad Técnica del Norte de la carrera de Ingeniería de Software.

Para llevar a cabo el muestreo se hizo uso de una prueba de la familia T, que se basa en la distribución de probabilidad T-student y es la recomendada para 1 o 2 grupos de la población que pueden estar relacionados de cierta forma. Para este caso se tiene un solo grupo de usuarios ya que no se los clasifico por alguna característica específica.

En cuanto al tipo de prueba estadística se optó por una regresión lineal multivariada porque en las pruebas de funcionamiento constan 4 variables explicativas o también conocidas como regresoras que en este caso son Verdaderos Positivos, Falsos Negativos, Falsos Positivos y Verdaderos Negativos, además va a existir una relación entre la variable de interés y las variables explicativas.

Para el presente trabajo se utilizó el software G\*Power de la Universidad de California en los Ángeles que lo utilizan los estudiantes de postgrado para realizar investigaciones de primer nivel, con el mayor énfasis en el análisis de datos relacionados con subvenciones y publicaciones, pero también incluyendo la investigación de tesis y además es un software utilizado en casi todo el mundo, para el desarrollo del proyecto se utilizó para calcular un tamaño total de la muestra que permita obtener resultados representativos. En la Figura 46, se muestra la gráfica y los datos utilizados en el software:

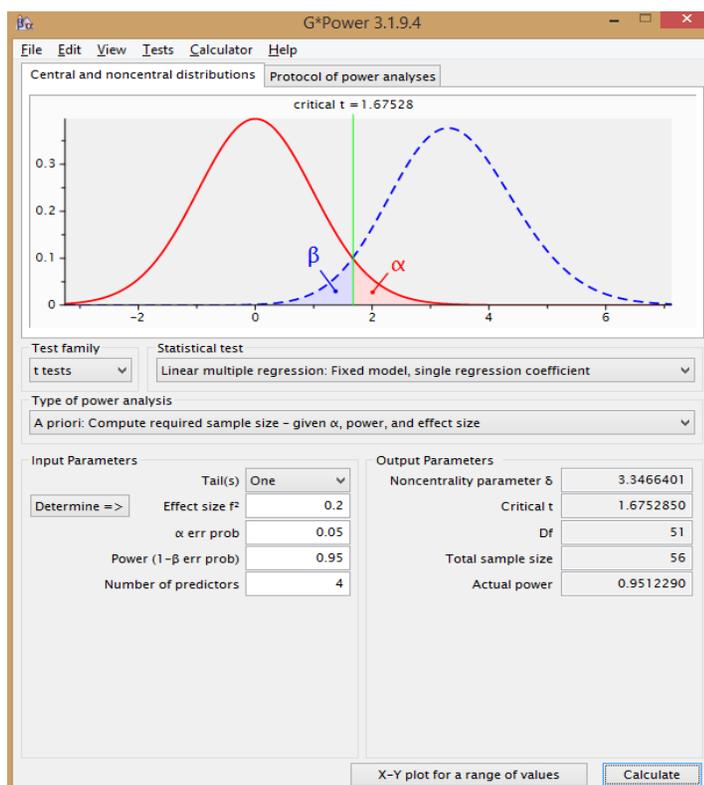


Figura 46 Resultado del cálculo del Muestreo

Fuente: Propia

Donde:

El tamaño de efecto  $F^2$  o conocida como  $F^2$  de Cohen: es utilizada para calcular el tamaño del efecto dentro de un modelo de regresión múltiple, a continuación, se presenta la fórmula que utiliza:

$$F^2 = \frac{R^2}{1 - R^2}$$

$\alpha$ : es la probabilidad de error que casi siempre es 0,05.

Power (1- $\beta$  error de probabilidad): en este caso es la potencia de la prueba que se configuró en 0.95 que corresponde a una potencia del 95% en la prueba.

Además, es importante mencionar que en el valor de  $F^2$  se consideró el 0.20 porque es un valor aceptado por las pautas de Cohen que explica de la siguiente forma:

Si el valor de  $F^2 \geq 0.02$  representan tamaños de efecto bajos, Si  $F^2 \geq 0.15$  representan tamaños de efectos medios y  $F^2 \geq 0.35$  representan tamaños de efectos grandes. En el presente proyecto se utilizó un valor de 0,2 lo que representa un tamaño de efecto mediano(Cárdenas Castro, 2014).

La siguiente Tabla 3.1 muestra un resumen del proceso realizado anteriormente tipo pseudocódigo con sus respectivos resultados:

TABLA 3.1  
MODELO DE REGRESIÓN MÚLTIPLE

<b>t tests - Linear multiple regression: Fixed model, single regression coefficient</b>			
Analysis:	A priori: Compute required sample size		
Input:	Tail(s)	=	One
	Effect size $f^2$	=	0.2
	$\alpha$ err prob	=	0.05
	Power (1- $\beta$ err prob)	=	0.95
	Number of predictors	=	4
Output:	Noncentrality parameter $\delta$	=	3.3466401
	Critical t	=	1.675285
	Df	=	51
	Total, sample size	=	56
	Actual power	=	0.951229

Fuente: Propia

### 3.2 Gráfico de regresión

En la Figura 47 sugiere que se necesita menos de 20 usuarios para tener un efecto grande, aproximadamente 35 – 40 para detectar un efecto mediano y más de 56 para detectar un efecto pequeño

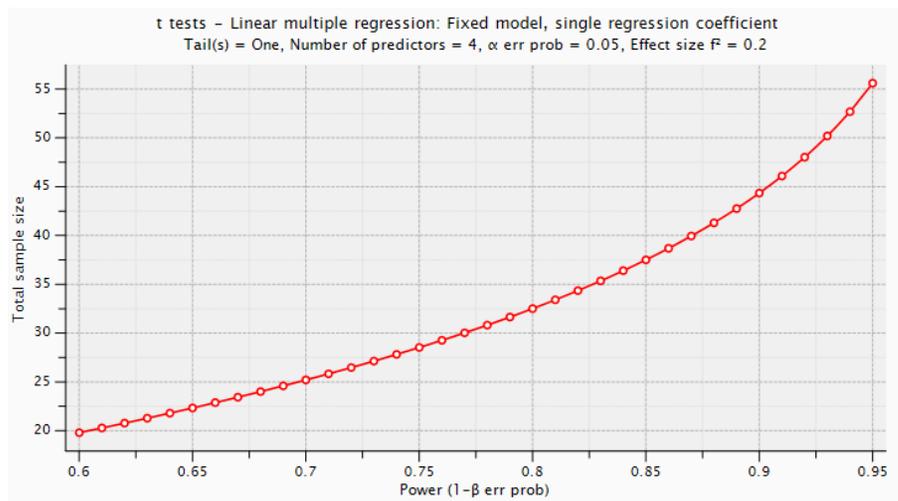


Figura 47 Gráfica de Regresión

Fuente: Propia

Una vez llenos los campos se obtiene un valor de 56 muestras, lo que significa que se debe probar la aplicación móvil con 56 personas, pero para efectos de las pruebas se realizó 58 pruebas a diferentes personas:

Se considero 2 parámetros para evaluar el algoritmo: Prueba de detección de rostros y prueba de detección de ojos

### 3.3 Matriz de Confusión de Rostros

Se realizó una matriz de confusión, como se muestra en la Tabla 3.2, para la evaluación de la aplicación con los cuatro componentes que son: verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, para poder comprender esta matriz de confusión es necesario explicar lo que significa cada uno de estos términos y a que se refieren cada uno de ellos.

TABLA 3.2  
MATRIZ DE CONFUSIÓN DE ROSTROS

		Predicción Detección de Rostros	
		Positivo (Si Rostros)	Negativo (No Rostros)
Valores Actuales Detección de Rostros	Matriz de Confusión Positivo (Si Rostros)	Verdadero Positivo (VP)	Falso Negativo (FN)
	Negativo (No Rostros)	Falso Positivo (FP)	Verdadero Negativo (VN)

Fuente: Propia

**VP:** Se pronosticaron como positivos y en realidad son positivos, para este caso sería así (SiRostros\_SiDetecta).

**VN:** Se pronosticaron como negativos y en realidad son negativos, para este caso sería así (NoRostros\_NoDetecta).

**FP:** Se pronosticaron como positivos, pero en realidad son negativos, para este caso sería así (NoRostros\_SiDetecta).

**FN:** Se pronosticaron como negativos, pero en realidad son positivos, para este caso sería así (SiRostros\_NoDetecta).

Para la primera prueba se consideró cuatro ítems a los cuales se les nombro de la siguiente manera: SiRostros\_SiDetecta, SiRostros\_NoDetecta, NoRostros\_SiDetecta y NoRostros\_NoDetecta, haciendo referencia a los valores que posee la matriz de confusión. Para poder realizar las pruebas primero se debe verificar si todos los datos obtenidos son válidos o no, para ello se calculó los datos atípicos en la matriz de difusión, y se realizó la siguiente codificación en R Studio:

```
## calcular datos atípicos
Cutoff = qchisq(1-.001, ncol(replaced))
```

Al realizar el cálculo respectivo da como resultado un punto de corte de:

```
##puntaje de corte
20.51501
```

Por otro lado, también se calculó la distancia de Mahalanobis que consiste en determinar la similitud entre variables

```
mahal = mahalanobis(replaced, colMeans(replaced), cov(replaced))
```

Si el resultado de la operación del cálculo de Mahal es menor al resultado del cálculo de qchisq o el resultado del punto de corte esos valores se conservarán caso contrario se eliminarán, en este caso ha encontrado 4 valores atípicos los mismos que fueron removidos de la base de datos, lo que quiere decir que se trabajara con 54 muestras como se muestra en la Tabla 3.3:

TABLA 3.3

DATOS ATÍPICOS - ROSTROS

MODE	FALSE	TRUE
logical	4	54

Fuente: Propia

### 3.4 Matriz Histograma de Densidad Correlación en Rostros

Para mostrar la relación entre los cuatro componentes que estas siendo evaluados se presenta la codificación que se realizó en R Studio y su respectivo resultado mostrado en la Figura 48:

```

library(foreign)
library(ggplot2)
library(psych)
plot(noout)
pairs.panels(noout, pch= 21,main="Matriz histograma densidad correlacion")

```

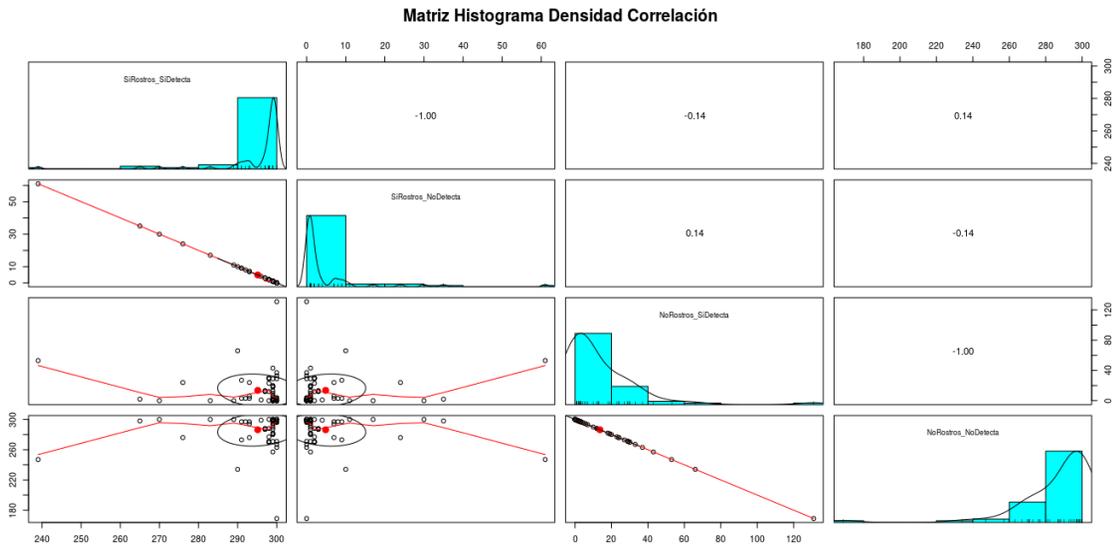


Figura 48 Matriz Densidad Correlación

Fuente: Propia

### 3.5 Resultados a partir de la Matriz de Densidad Correlación

A continuación, en la Figura 49 se presenta los resultados de la prueba de rostros con la variable SiRostros\_SiDetecta, es decir, los Verdaderos Positivos:

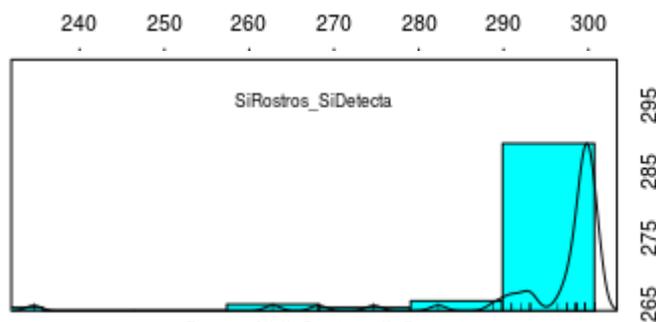


Figura 49 Si Rostros Si Detecta

Fuente: Propia

Se puede apreciar que cuando si existe rostros en la aplicación móvil existe una alta cantidad de aciertos de detección de rostros, verdaderos positivos.

En la Figura 50 se presenta la prueba con la variable SiRostros\_NoDetecta, es decir, los Falsos Negativos:

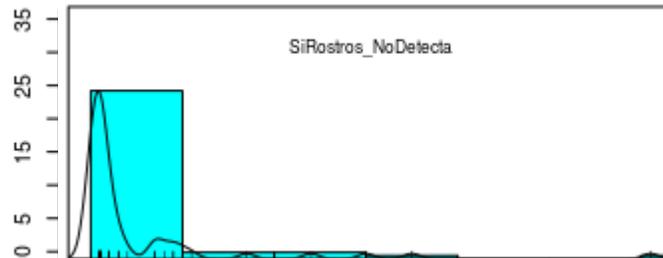


Figura 50 Si Rostros No Detecta

Fuente: Propia

Se puede observar que el número de rostro que no fueron detectados cuando si existía rostros es muy bajo y que llega máximo a 25 rostros de 300 que fueron procesados.

En la Figura 51 se presenta la prueba con la variable NoRostros\_SiDetecta en decir los Falsos Positivos:

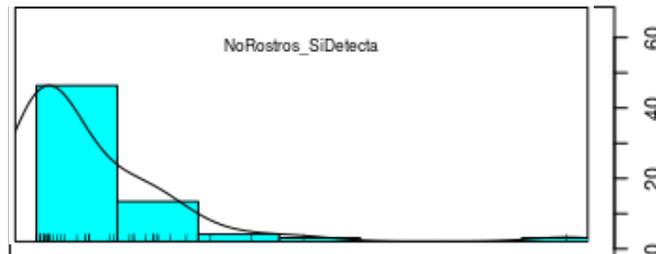


Figura 51 No Rostros Si Detecta

Fuente: Propia

Los resultados muestran datos bajos ya que el número máximo de rostros detectados cuando no existía rostro es muy bajo ya que representa el 20% del total de rostros.

En la Figura 52 se presenta la prueba con la variable NoRostros\_NoDetecta, es decir, los Verdaderos Negativos:

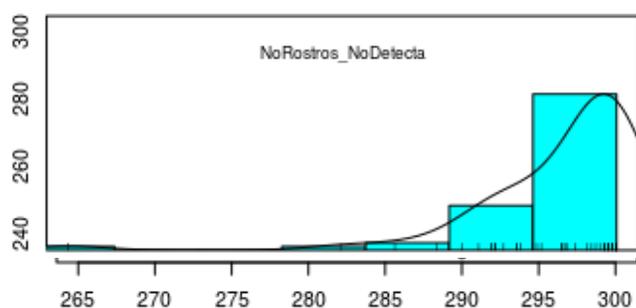


Figura 52 No Rostros No Detecta

Fuente: Propia

Los resultados son muy alentadores ya que falla en un 7 % aproximadamente cuando no existe rostros y no debe detectar ningun rostro.

Seguido se aplica la prueba de normalidad univariada de cada componente si p-value es menor a 0.05 se acepta la hipotesis alternativa, en otras palabras, los datos son normales como se muestra en la Tabla 3.4:

TABLA 3.4

RESULTADOS DEL TEST DE SHAPIRO - ROSTROS

Prueba	Código	Resultado
<b>SiRostros_SiDetecta(VP)</b>	shapiro.test(noout\$SiRostros_SiDetecta)	7.403e-13
<b>SiRostros_NoDetecta(FN)</b>	shapiro.test(noout\$SiRostros_NoDetecta)	7.403e-13
<b>NoRostros_SiDetecta(FP)</b>	shapiro.test(noout\$NoRostros_SiDetecta)	1.213e-10
<b>NoRostros_NoDetecta(VN)</b>	shapiro.test(noout\$NoRostros_NoDetecta)	1.213e-10
<b>Precisión</b>	shapiro.test(noout\$Precisión)	3.351e-09
<b>Exactitud</b>	shapiro.test(noout\$Exactitud)	6.897e-10
<b>Especificidad</b>	shapiro.test(noout\$Especificidad)	1.213e-10
<b>Sensibilidad</b>	shapiro.test(noout\$Sensibilidad)	7.403e-13
<b>Medida F1</b>	shapiro.test(noout\$`Medida F1`)	1.989e-09

Fuente: Propia

Seguido se procede a verificar la fiabilidad de la escala de medida o la correlación de entre las variables para ello se procede a calcular mediante el alpha de cronbach, que dará como resultado la suficiencia de muestreo, por lo general se pretende que las respuestas estén correlacionadas pero no perfectas, es decir, el resultado es aceptable a partir de 0,70, a continuación se muestra el procedimiento:

```
medidas=noout[,c(5,6)]
correl=cor(medidas, use = "pairwise.complete.obs")
symnum(correl)
correl
cortest.bartlett(correl, n=nrow(noout))
```

Y el resultado de la suficiencia de correlación es que el valor de p-value es de 2.54864e-26 lo que significa que si existe suficiencia. Finalmente el resultado del alpha de cronbach es del 0.87 lo que significa que existe fiabilidad en las medidas.

### 3.6 Cálculo de Métricas de Desempeño en la Matriz de Rostros

A partir de la matriz de difusión se puede calcular algunas métricas de desempeño que son muy conocidas en estadística: precisión, exactitud, especificidad, sensibilidad, medida F1:

#### 3.6.1 Cálculo de la Precisión

**Precisión:** se le conoce como valor predictivo positivo ya que calcula la relación entre las predicciones positivas correctas y los pronósticos positivos totales:

$$precisión = \frac{VP}{VP + FP}$$

#### 3.6.2 Cálculo de la Exactitud

**Exactitud:** calcula la tasa de aciertos totales y da como resultado la eficacia del algoritmo.

$$exactitud = \frac{VP + VN}{VP + FN + VN + FP}$$

### 3.6.3 Cálculo de la Especificidad

**Especificidad:** se le conoce como tasa de negativos verdaderos, es decir, la proporción de negativos reales que se identifican correctamente ya que, cuanto más negativos verdaderos tienen los datos mayor es la especificidad.

$$especificidad = \frac{VN}{VN + FP}$$

### 3.6.4 Cálculo de la Medida Recall

**Recall o Sensibilidad:** calcula la tasa de positivos verdaderos porque identifica todas las instancias verdaderas.

$$sensibilidad = \frac{VP}{VP + FN}$$

### 3.6.5 Cálculo de la Medida F1

**Medida F1:** conocida como media armónica de precisión y recuperación, calcula el equilibrio entre la precisión y la recuperación, que varía de 0 a 1.

$$F1 = \frac{2 * precisión * sensibilidad}{precisión + sensibilidad}$$

En la Tabla 3.5 se muestra el proceso para calcular la precisión, la exactitud y especificidad, sensibilidad y medida F1.

TABLA 3.5  
CALCULO DE MÉTRICAS - ROSTROS

Medida	Fórmula	Cambio de nombre a la columna
Precisión	$master[,5]=master[,1]/(master[,1]+master[,3])$	<code>colnames(master)=c(colnames(master)[-5], "Precisión")</code>

<b>Exactitud</b>	$master[6] = (master[1] + master[4]) / (master[1] + master[2] + master[3] + master[4])$	$colnames(master) = c(colnames(master)[-6], "Exactitud")$
<b>Especificidad</b>	$master[7] = master[4] / (master[4] + master[3])$	$colnames(master) = c(colnames(master)[-7], "Especificidad")$
<b>Sensibilidad</b>	$master[8] = master[1] / (master[1] + master[2])$	$colnames(master) = c(colnames(master)[-8], "Sensibilidad")$
<b>Medida F1</b>	$master[9] = (2 * master[5] * master[8]) / (master[5] + master[8])$	$colnames(master) = c(colnames(master)[-9], "Medida F1")$

Fuente: Propia

Los resultados de los cálculos se presentan a continuación en la Tabla 3.6:

TABLA 3.6  
RESULTADO DEL CÁLCULO DE LAS MÉTRICAS - ROSTROS

N°	Usuarios	VP	FN	FP	VN	Precisión	Exactitud	Especificidad	Sensibilidad	Medida F1
1	MAURICIO RUIZ	300	0	29	271	0.9118541	0.9516667	0.903333	1	0.9538951
2	BRAYAN PULAMARIN	300	0	0	300	1	1	1	1	1
3	KARLA HERRERA	299	1	1	299	0.9966667	0.9966667	0.996667	0.9966667	0.9966667
4	ALEXIS SOLANO	299	1	0	300	1	0.9983333	1	0.9966667	0.9983306
5	ALEXIS VACA	299	1	29	271	0.9115854	0.95	0.903333	0.9966667	0.9522293
6	KEVIN ULCUANGO	293	7	3	297	0.9898649	0.9833333	0.99	0.9766667	0.9832215
7	VANESSA CADENA	291	9	3	297	0.9897959	0.98	0.99	0.97	0.979798
8	EDUARDO OLIVO	297	3	12	288	0.961165	0.975	0.96	0.99	0.9753695
9	DIEGO SOLANO	299	1	13	287	0.9583333	0.9766667	0.956667	0.9966667	0.9771242
10	DE LA TORRE WILLIAM	297	3	13	287	0.9580645	0.9733333	0.956667	0.99	0.9737705
11	RAMIREZ GISSELA	283	17	0	300	1	0.9716667	1	0.9433333	0.9708405
12	ESTEBAN NARVAEZ	298	2	30	270	0.9085366	0.9466667	0.9	0.9933333	0.9490446
13	DILAN RAMIREZ	299	1	0	300	1	0.9983333	1	0.9966667	0.9983306
14	KEVIN LLIGUISUPA	300	0	2	298	0.9933775	0.9966667	0.993333	1	0.9966777
15	EDISON GUAICHICO	299	1	0	300	1	0.9983333	1	0.9966667	0.9983306
16	MATTHEW GARCIA	299	1	2	298	0.9933555	0.995	0.993333	0.9966667	0.9950083
17	GABRIEL LIMA	296	4	1	299	0.996633	0.9916667	0.996667	0.9866667	0.9916248
18	JORGE JARAMILLO	298	2	12	288	0.9612903	0.9766667	0.96	0.9933333	0.9770492
19	BRAYAN VELASCO	300	0	0	300	1	1	1	1	1
20	ULICES VALENCIA	299	1	3	297	0.9900662	0.9933333	0.99	0.9966667	0.9933555
21	JOHANNA PUERCHAMBUD	300	0	37	263	0.8902077	0.9383333	0.876667	1	0.9419152
22	ERICK FLORES	300	0	0	300	1	1	1	1	1
23	CARLOS IPIALES	299	1	7	293	0.9771242	0.9866667	0.976667	0.9966667	0.9867987
24	KEVIN LOPEZ	297	3	13	287	0.9580645	0.9733333	0.956667	0.99	0.9737705

25	KLEVER VILLALBA	298	2	29	271	0.911315	0.9483333	0.903333	0.9933333	0.9505582
26	SEBASTIAN TIRIRA	299	1	0	300	1	0.9983333	1	0.9966667	0.9983306
27	KEVIN VINUEZA	299	1	1	299	0.9966667	0.9966667	0.996667	0.9966667	0.9966667
28	HENRY MEZA	299	1	43	257	0.874269	0.9266667	0.856667	0.9966667	0.9314642
29	ANGEL IMBAQUINGO	298	2	23	277	0.9283489	0.9583333	0.923333	0.9933333	0.9597424
30	KEVIN MENDEZ	298	2	0	300	1	0.9966667	1	0.9933333	0.9966555
31	OLIVER ZAMORA	299	1	1	299	0.9966667	0.9966667	0.996667	0.9966667	0.9966667
32	EDWIN HERRERA	299	1	18	282	0.9432177	0.9683333	0.94	0.9966667	0.9692058
33	JHOAN PASTILLA	300	0	0	300	1	1	1	1	1
34	ADRIAN MARTINEZ	299	1	19	281	0.9402516	0.9666667	0.936667	0.9966667	0.9676375
35	STALIN PALMA	300	0	4	296	0.9868421	0.9933333	0.986667	1	0.9933775
36	DYLAN ALVAREZ	299	1	5	295	0.9835526	0.99	0.983333	0.9966667	0.9900662
37	EMERSON MONTESDEOCA	300	0	2	298	0.9933775	0.9966667	0.993333	1	0.9966777
38	LUIS CATUQUAMBA	299	1	0	300	1	0.9983333	1	0.9966667	0.9983306
39	JOSTIN ROLDAN	299	1	10	290	0.9676375	0.9816667	0.966667	0.9966667	0.9819376
40	BRAYAN ORTEGA	292	8	3	297	0.9898305	0.9816667	0.99	0.9733333	0.9815126
41	ANDY CUASCOTA	299	1	2	298	0.9933555	0.995	0.993333	0.9966667	0.9950083
42	ELVIS TUQUERREZ	291	9	27	273	0.9150943	0.94	0.91	0.97	0.9417476
43	DANIEL ALPALA	293	7	24	276	0.9242902	0.9483333	0.92	0.9766667	0.9497569
44	WILSON SEVILLA	298	2	30	270	0.9085366	0.9466667	0.9	0.9933333	0.9490446
45	ALEXIS VACA	299	1	1	299	0.9966667	0.9966667	0.996667	0.9966667	0.9966667
46	FRANCISCO RIVERA	300	0	0	300	1	1	1	1	1
47	SANTIAGO AGUIRRE	293	7	6	294	0.9799331	0.9783333	0.98	0.9766667	0.9782972
48	WELINTON VALENZUELA	300	0	3	297	0.990099	0.995	0.99	1	0.9950249
49	CRISTIAN TONTAQUIMBA	300	0	33	267	0.9009009	0.945	0.89	1	0.9478673
50	JHON RAMIREZ	289	11	0	300	1	0.9816667	1	0.9633333	0.9813243
51	EDISON GUAICHICO	300	0	131	169	0.6960557	0.7816667	0.563333	1	0.8207934
52	JUAN DIEGO IZA	299	1	20	280	0.9373041	0.965	0.933333	0.9966667	0.9660743
53	KATHERIN BURGOS	290	10	66	234	0.8146067	0.8733333	0.78	0.9666667	0.8841463
54	MATTHEW GARCIA	299	1	2	298	0.9933555	0.995	0.993333	0.9966667	0.9950083

Fuente: Propia

Una vez calculadas las medidas se procede a calcular la media de las nuevas columnas de la tabla, en la siguiente Tabla 3.7 se presenta el resultado de la media de cada columna presentada anteriormente.

TABLA 3.7  
CÁLCULO Y RESULTADO DE LA MEDIA DE LAS MÉTRICAS CALCULADAS - ROSTROS

Medida	Código	Resultado
Precisión	Mean(master\$presision)	0.9592959

<b>Exactitud</b>	Mean(master\$exactitud)	0.9691379
<b>Especificidad</b>	Mean(master\$especificidad)	0.9544828
<b>Sensibilidad</b>	Mean(master\$sensibilidad)	0.9837931
<b>Medida F1</b>	Mean(master\$ ' Medida F1')	0.9703842

Fuente: Propia

### 3.7 Matriz de Confusión de Ojos

Para la segunda prueba se consideró cuatro ítems a los cuales se les nombró de la siguiente manera: SiOjos\_SiDetecta, SiOjos\_NoDetecta, NoOjos\_SiDetecta y NoOjos\_NoDetecta, haciendo referencia a los valores que posee la matriz de confusión, para ello se realizó una matriz de confusión para la evaluación de la aplicación con los cuatro componentes que son: verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, para poder comprender esta matriz de confusión es necesario explicar lo que significa cada uno de estos términos y a que se refieren cada uno de ellos. En la Tabla 3.8 se muestra la Matriz e Confusión de los Ojos.

TABLA 3.8

MATRIZ DE CONFUSIÓN - OJOS

		<b>Predicción Detección de Ojos</b>	
		Positivo (Si Ojos)	Negativo (No Ojos)
<b>Valores Actuales</b>	<b>Matriz de Confusión</b> Positivo (Si Ojos)	Verdadero Positivo (VP)	Falso Negativo (FN)
	Negativo (No Ojos)	Falso Positivo (FP)	Verdadero Negativo (VN)

Fuente: Propia

**VP:** Se pronosticaron como positivos y en realidad son positivos, para este caso sería así (SiOjos\_SiDetecta).

**VN:** Se pronosticaron como negativos y en realidad son negativos, para este caso sería así (NoOjos\_NoDetecta).

**FP:** Se pronosticaron como positivos, pero en realidad son negativos, para este caso sería así (NoOjos\_SiDetecta).

**FN:** Se pronosticaron como negativos, pero en realidad son positivos, para este caso sería así (SiOjos\_NoDetecta).

Para poder realizar las pruebas primero se debe verificar si todos los datos obtenidos son válidos o no, para ello se calculó los datos atípicos en la matriz de difusión, y se realizó la siguiente codificación en R Studio:

```
## calcular datos atípicos
Cutoff = qchisq(1-.001, ncol(replaced))
```

Al realizar el cálculo respectivo da como resultado un punto de corte de:

```
##puntaje de corte
20.51501
```

Por otro lado, también se calculó la distancia de Mahalanobis que consiste en determinar la similitud entre variables:

```
mahal = mahalanobis(replaced, colMeans(replaced), cov(replaced))
```

Si el resultado de la operación del cálculo de Mahal es menor al resultado del cálculo de qchisq o el resultado del punto de corte esos valores se conservarán caso contrario se eliminarán, para este caso ha encontrado 4 valores atípicos, lo que quiere decir que se trabajara con 54 muestras como se muestra en la Tabla 3.9:

TABLA 3.9

DATOS ATÍPICOS - OJOS

MODE	FALSE	TRUE
logical	4	54

Fuente: Propia

### 3.8 Matriz de Densidad Correlación

Para mostrar la relación entre los cuatro componentes que estas siendo evaluados se presenta la codificación que se realizó en R Studio y su respectivo resultado mostrado en la Figura 53:

```
library(foreign)
library(ggplot2)
```

```

library(psych)
plot(noout)
pairs.panels(noout, pch= 21,main="Matriz histograma densidad correlacion")

```

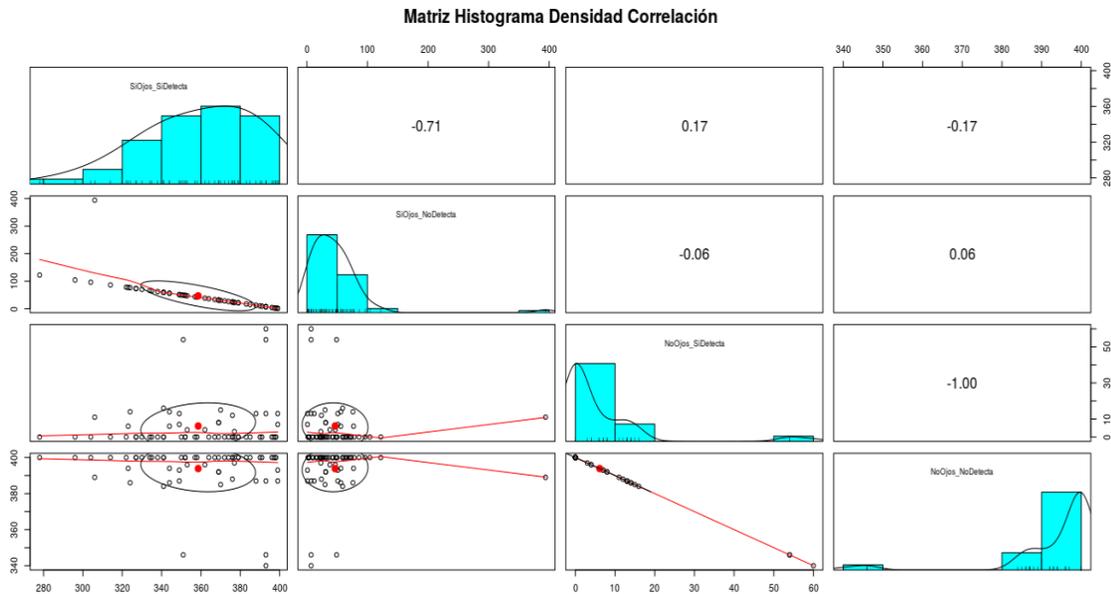


Figura 53 Matriz de Histograma Densidad Correlación en la detección de Ojos

Fuente: Propia

### 3.9 Resultados a partir de la Matriz de Densidad Correlación de los Ojos

A continuación, en la Figura 54 se presenta los resultados de la prueba de ojos con la variable SiOjos\_SiDetecta es decir los Verdaderos Positivos:

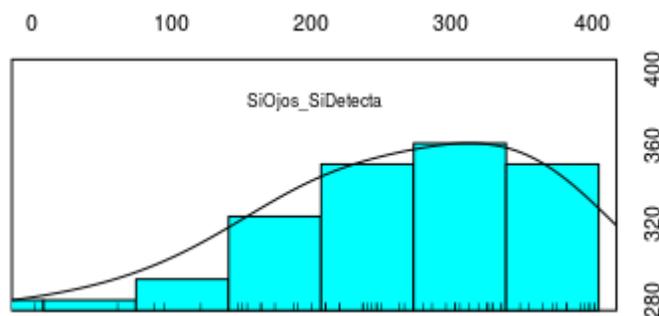


Figura 54 Si Ojos Si Detecta

Fuente: Propia

Se puede observar que existe un alto índice de aciertos al momento de detectar ojos cuando previamente si existe un rostro.

En la Figura 55 se presenta la prueba con la variable SiOjos\_NoDetecta en decir los Falsos Negativos:

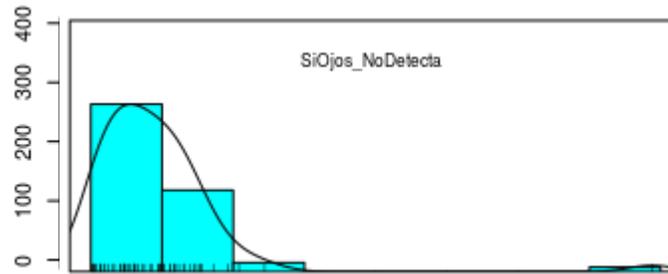


Figura 55 Si Ojos No Detecta

Fuente: Propia

En la gráfica se muestra que el detector está funcionando adecuadamente y que en ocasiones muy leves si detecta ojos cuando no debería.

En la Figura 56 se presenta la prueba con la variable NoOjos\_SiDetecta en decir los Falsos Positivos:



Figura 56 No Ojos Si Detecta

Fuente: Propia

Cuando no existe ojos la aplicación está detectando ojos en pocas ocasiones.

En la Figura 57 se presenta la prueba con la variable NoOjos\_NoDetecta, es decir, los Verdaderos Negativos:

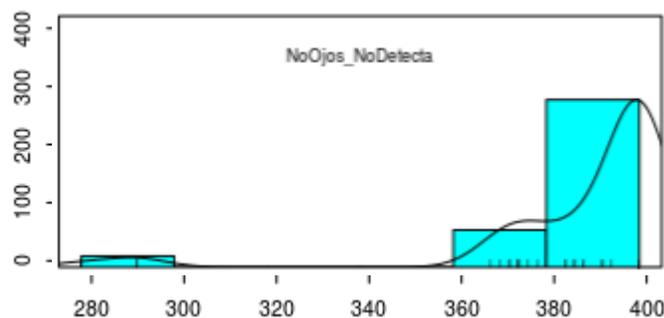


Figura 57 No Ojos No Detecta

Fuente: Propia

El índice de la gráfica muestra que el detector está en óptimas condiciones y que en la mayoría de las ocasiones no detecta ojos cuando no existe ojos.

Al igual que en la detección de rostros se procede a calcular el la detección de ojos la prueba de normalidad univariada de cada componente si p-value es menor o igual a 0.05 los datos son normales como se muestra en la Tabla 3.10:

TABLA 3.10  
RESULTADOS DEL TEST DE SHAPIRO - OJOS

<b>Prueba</b>	<b>Código</b>	<b>Resultado</b>
<b>SiOjos_SiDetecta(VP)</b>	shapiro.test(noout\$SiOjos_SiDetecta)	0.03146
<b>SiOjos_NoDetecta(FN)</b>	shapiro.test(noout\$SiOjos_NoDetecta)	0.03146
<b>NoOjos_SiDetecta(FP)</b>	shapiro.test(noout\$NoOjos_SiDetecta)	1.608e-12
<b>NoOjos_NoDetecta(VN)</b>	shapiro.test(noout\$NoOjos_NoDetecta)	1.608e-12
<b>Precisión</b>	shapiro.test(noout\$Precisión)	6.035e-12
<b>Especificidad</b>	shapiro.test(noout\$Especificidad)	1.608e-12
<b>Sensibilidad</b>	shapiro.test(noout\$Sensibilidad)	0.03146
<b>Medida F1</b>	shapiro.test(noout\$`Medida F1`)	0.05

Fuente: Propia

Seguido se procede a verificar la fiabilidad de la escala de medida o la correlación de entre las variables para ello se procede a calcular mediante el alpha de cronbach, que dará como resultado la suficiencia de muestreo, por lo general se pretende que las respuestas estén correlacionadas pero no perfectas es decir el resultado es aceptable a partir de 0,70, a continuación se muestra el procedimiento:

```
medidas=noout[,c(5,6)]
correl=cor(medidas, use = "pairwise.complete.obs")
symnum(correl)
correl
```

correst.bartlett(correl, n=nrow(noout))

Y el resultado de la suficiencia de correlación es que el valor de p-value es de 0.027 lo que significa que si existe suficiencia. Finalmente, el resultado del alpha de cronbach es del 0.85 lo que significa que existe fiabilidad en las medidas.

A continuación, de la misma forma que se calculó las medidas en la prueba de los rostros al igual que realizó en los rostros, a continuación la codificación el R Studio:

### 3.10 Cálculo de Métricas de Desempeño en la Matriz de Rostros

Como ya se realizó el cálculo de las medidas propuestas en la prueba de detección de rostros, se procede a realizar los mismos cálculos en la detección de ojos, de la siguiente forma como se muestra en la Tabla 3.11:

TABLA 3.11  
CALCULO DE MÉTRICAS - OJOS

Medida	Fórmula	Cambio de nombre a la columna
<b>Precisión</b>	$master[,5]=master[,1]/(master[,1]+master[,3])$	colnames(master)=c(colnames(master)[-5], "Precisión")
<b>Exactitud</b>	$master[,6]=(master[,1]+master[,4])/(master[,1]+master[,2]+master[,3]+master[,4])$	colnames(master)=c(colnames(master)[-6], "Exactitud")
<b>Especificidad</b>	$master[,7]=master[,4]/(master[,4]+master[,3])$	colnames(master)=c(colnames(master)[-7], "Especificidad")
<b>Sensibilidad</b>	$master[,8]=master[,1]/(master[,1]+master[,2])$	colnames(master)=c(colnames(master)[-8], "Sensibilidad")
<b>Medida F1</b>	$master[,9]=(2 * master[,5] * master[,8]) / (master[,5] + master[,8])$	colnames(master)=c(colnames(master)[-9], "Medida F1")

Fuente: Propia

A continuación, en la Tabla 3.12, se presenta los resultados del cálculo de las 5 medidas previamente mostradas:

TABLA 3.12

RESULTADO DE LAS MÉTRICAS CALCULADAS - OJOS

Nº	Usuarios	VP	FN	FP	VN	Precisión	Exactitud	Especificidad	Sensibilidad	Medida F1
1	MAURICIO RUIZ	379	21	0	400	1	0.97375	1	0.9475	0.9730424
2	BRAYAN PULAMARIN	369	31	0	400	1	0.96125	1	0.9225	0.9596879
3	KARLA HERRERA	327	73	0	400	1	0.90875	1	0.8175	0.8995873
4	ALEXIS SOLANO	393	7	54	346	0.8791946	0.92375	0.865	0.9825	0.9279811
5	ALEXIS VACA	367	33	0	400	1	0.95875	1	0.9175	0.9569752
6	KEVIN ULCUANGO	393	7	13	387	0.9679803	0.975	0.9675	0.9825	0.9751861
7	VANESSA CADENA	399	1	7	393	0.9827586	0.99	0.9825	0.9975	0.9900744
8	EDUARDO OLIVO	399	1	13	387	0.9684466	0.9825	0.9675	0.9975	0.9827586
9	DIEGO SOLANO	344	56	6	394	0.9828571	0.9225	0.985	0.86	0.9173333
10	DE LA TORRE WILLIAM	330	70	0	400	1	0.9125	1	0.825	0.9041096
11	RAMIREZ GISSELA	338	62	0	400	1	0.9225	1	0.845	0.9159892
12	ESTEBAN NARVAEZ	352	48	0	400	1	0.94	1	0.88	0.9361702
13	DILAN RAMIREZ	393	7	60	340	0.8675497	0.91625	0.85	0.9825	0.9214537
14	KEVIN LLIGUISUPA	376	24	0	400	1	0.97	1	0.94	0.9690722
15	EDISON GUAICHICO	379	21	0	400	1	0.97375	1	0.9475	0.9730424
16	MATTHEW GARCIA	369	31	8	392	0.9787798	0.95125	0.98	0.9225	0.9498069
17	GABRIEL LIMA	362	38	4	396	0.989071	0.9475	0.99	0.905	0.9451697
18	JORGE JARAMILLO	374	26	0	400	1	0.9675	1	0.935	0.9664083
19	BRAYAN VELASCO	314	86	0	400	1	0.8925	1	0.785	0.8795518
20	ULICES VALENCIA	377	23	3	397	0.9921053	0.9675	0.9925	0.9425	0.9666667
21	JOHANNA PUERCHAMBUD	335	65	0	400	1	0.91875	1	0.8375	0.9115646
22	ERICK FLORES	327	73	0	400	1	0.90875	1	0.8175	0.8995873
23	CARLOS IPIALES	344	56	14	386	0.9608939	0.9125	0.965	0.86	0.9076517
24	KEVIN LOPEZ	349	51	7	393	0.9803371	0.9275	0.9825	0.8725	0.9232804
25	KLEVER VILLALBA	385	15	0	400	1	0.98125	1	0.9625	0.9808917
26	SEBASTIAN TIRIRA	391	9	0	400	1	0.98875	1	0.9775	0.988622
27	KEVIN VINUEZA	397	3	0	400	1	0.99625	1	0.9925	0.9962359
28	HENRY MEZA	390	10	0	400	1	0.9875	1	0.975	0.9873418
29	ANGEL IMBAQUINGO	324	76	14	386	0.9585799	0.8875	0.965	0.81	0.8780488
30	KEVIN MENDEZ	398	2	0	400	1	0.9975	1	0.995	0.9974937
31	OLIVER ZAMORA	398	2	0	400	1	0.9975	1	0.995	0.9974937
32	EDWIN HERRERA	349	51	13	387	0.9640884	0.92	0.9675	0.8725	0.9160105
33	JHOAN PASTILLA	341	59	0	400	1	0.92625	1	0.8525	0.9203779
34	STALIN PALMA	350	50	0	400	1	0.9375	1	0.875	0.9333333
35	DYLAN ALVAREZ	372	28	0	400	1	0.965	1	0.93	0.9637306
36	EMERSON MONTESDEOCA	322	78	0	400	1	0.9025	1	0.805	0.8919668
37	LUIS CATUQUAMBA	341	59	0	400	1	0.92625	1	0.8525	0.9203779
38	JOSTIN ROLDAN	396	4	0	400	1	0.995	1	0.99	0.9949749
39	BRAYAN ORTEGA	352	48	0	400	1	0.94	1	0.88	0.9361702
40	ANDY CUASCOTA	388	12	13	387	0.967581	0.96875	0.9675	0.97	0.968789
41	ELVIS TUQUERREZ	351	49	54	346	0.8666667	0.87125	0.865	0.8775	0.8720497
42	DANIEL ALPALA	377	23	0	400	1	0.97125	1	0.9425	0.970399
43	EDISON PINANJOTA	358	42	0	400	1	0.9475	1	0.895	0.944591
44	WILSON SEVILLA	353	47	4	396	0.9887955	0.93625	0.99	0.8825	0.9326288

45	ALEXIS VACA	357	43	0	400	1	0.94625	1	0.8925	0.9431968
46	KEVIN ULCUANGO	364	36	0	400	1	0.955	1	0.91	0.9528796
47	STEVEN MANZANO	334	66	0	400	1	0.9175	1	0.835	0.9100817
48	SANTIAGO AGUIRRE	376	24	12	388	0.9690722	0.955	0.97	0.94	0.9543147
49	WELINTON VALENZUELA	323	77	6	394	0.9817629	0.89625	0.985	0.8075	0.8861454
50	CRISTIAN TONTAQUIMBA	341	59	16	384	0.9551821	0.90625	0.96	0.8525	0.9009247
51	JHON RAMIREZ	376	24	0	400	1	0.97	1	0.94	0.9690722
52	EDISON GUAICHICO	370	30	15	385	0.961039	0.94375	0.9625	0.925	0.9426752
53	KATHERIN BURGOS	383	17	0	400	1	0.97875	1	0.9575	0.9782886
54	MATTHEW GARCIA	369	31	8	392	0.9787798	0.95125	0.98	0.9225	0.9498069

Fuente: Propia

Una vez calculadas las medidas se procede a calcular el promedio y la media de las nuevas columnas que se tiene en la tabla:

Seguido en la Tabla 3.13 se presenta el cálculo de las medianas en las 5 medidas que se calculó anteriormente:

TABLA 3.13  
CÁLCULO Y RESULTDOS DE LA MEDIA EN LAS MÉTRICAS - OJOS

Medida	Código	Resultado
<b>Precisión</b>	Mean(master\$presision)	0.9846004
<b>Exactitud</b>	Mean(master\$exactitud)	0.9365184
<b>Especificidad</b>	Mean(master\$especificidad)	0.9846983
<b>Sensibilidad</b>	Mean(master\$sensibilidad)	0.8908559
<b>Medida F1</b>	Mean(master\$ ' Medida F1')	0.9321915

Fuente: Propia

### 3.11 Curva ROC

Finalmente se procede a graficar la curva ROC o curva característica de funcionamiento del receptor, con la finalidad de mostrar el rendimiento del modelo clasificación en todos los umbrales de clasificación.

Con la siguiente fórmula se calculó la regresión:

$$f(x) = 0.04088 + 5.266725x - 14.4728x^2 + 15.15215x^3 - 5.703825x^4$$

$$AUC = \int_0^1 f(x)dx$$

Como resultado se obtuvo un 0.84 lo cual es muy óptimo ya que mientras el valor se aproxime mas a 1 significa que el prototipo tiene una precisión optima, la Figura 58 muestra el resultado obtenido.

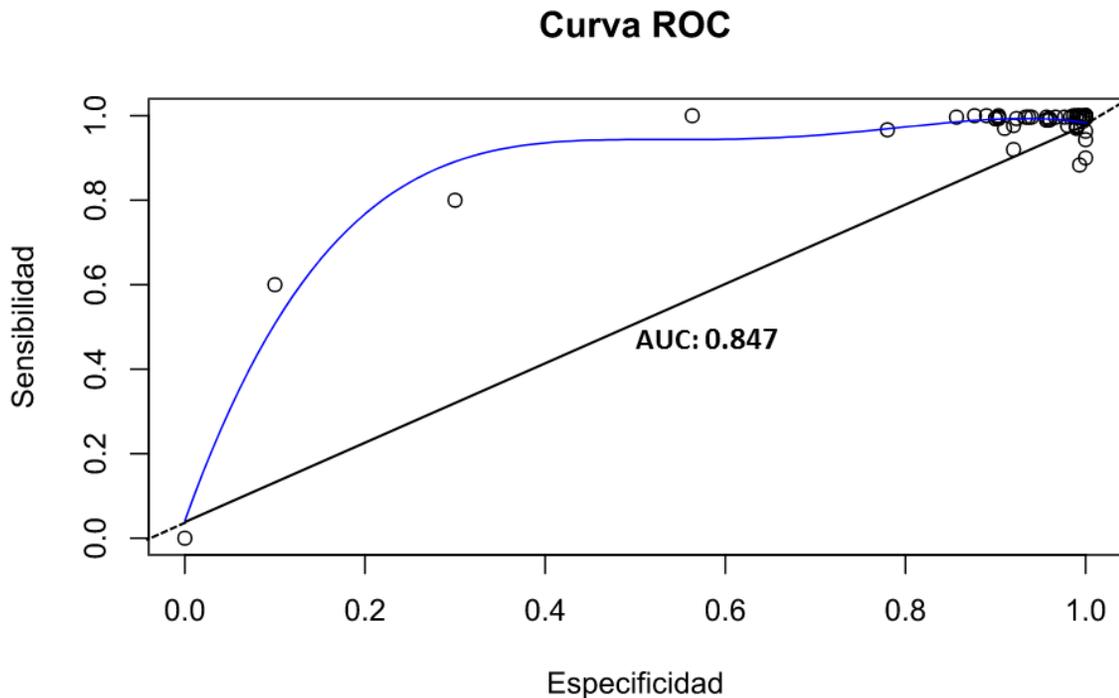


Figura 58 Curva ROC

Fuente: Propia

### 3.12 Encuesta del prototipo de la aplicación móvil

En este capítulo se presenta la validación del prototipo de la aplicación móvil para detección de somnolencia nocturna según la perspectiva de los usuarios que probaron el prototipo, para ello se diseñó una encuesta que consta de 18 preguntas y 15 de ellas fueron evaluadas según la escala de Likert y se consideró la siguiente valoración: **Totalmente de acuerdo = 7, Medianamente de acuerdo = 6, Parcialmente de acuerdo = 5, Ni de acuerdo ni en desacuerdo = 4, Parcialmente en desacuerdo = 3, Medianamente en desacuerdo = 2, Totalmente en desacuerdo = 1.** Es necesario mencionar que 7 es el valor más alto que puede tener una pregunta y 1 el más bajo.

Los eventos o acciones evaluadas están divididas en 4 secciones, la primera es una sección informativa y las siguientes secciones cuentan con 5 preguntas respectivamente:

Es necesario mencionar que se obtuvo el resultado de 33 respuestas a la encuesta como se muestra en la Figura 59:

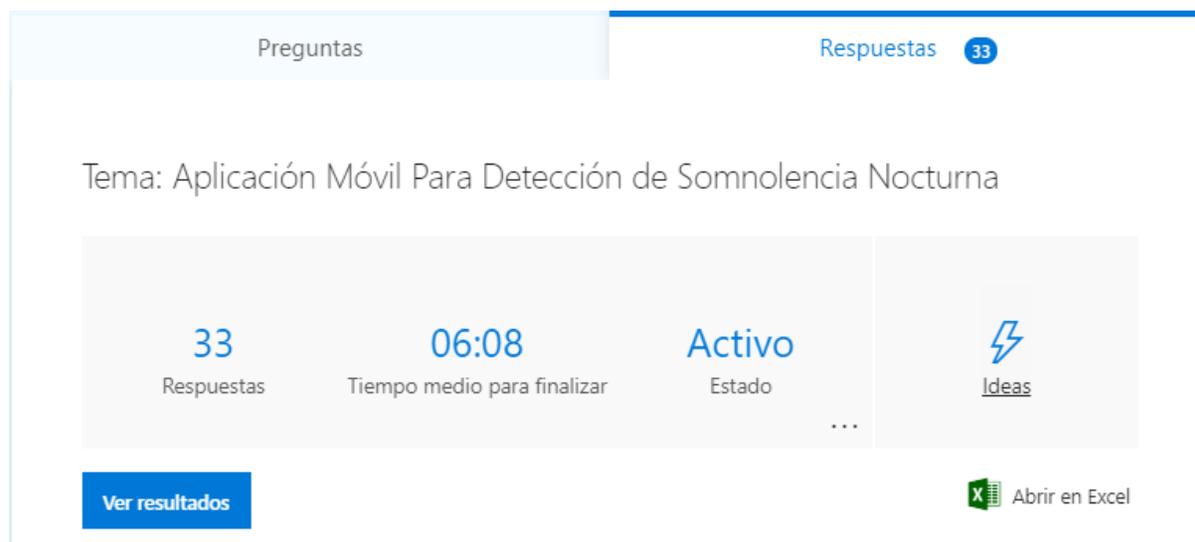


Figura 59 Encuesta a 33 Usuarios

Fuente: Propia

A continuación, en la Tabla 3.14, se presenta los resultados de la encuesta aplicada a los usuarios que probaron el prototipo de la aplicación.

TABLA 3.14

RESULTADOS DE LA ESCUESTA APLICADA

P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11	P 12	P 13	P 14	P 15	P 16	P 17	P 18
ERICK HERRERA	M	3	6	7	5	5	7	7	7	5	6	5	6	5	7	6	7
JUAN DIEGO IZA	M	3	5	5	5	7	6	6	5	5	4	6	4	4	5	4	5
KARLA HERRERA	F	2	5	6	5	6	4	5	5	6	4	5	5	5	5	5	5
RICHARD AGUIRRE	M	2	7	7	7	7	7	7	7	7	7	7	6	7	7	7	7
WILSON SEVILLA	M	3	6	7	6	7	7	7	7	7	7	7	5	5	7	7	7
DEISON MARTINEZ	M	1	6	6	6	7	6	6	7	7	6	6	7	6	7	6	6
JOSHUA ALMEIDA	M	2	6	5	5	6	6	7	7	7	6	7	7	6	7	7	7
ELVIS TUQUERREZ	M	3	7	6	5	5	6	6	6	6	5	6	7	6	7	6	7
EMERSON MONTESDEOCA	M	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
JOSTIN ROLDAN	M	2	7	7	6	7	6	7	7	7	7	7	7	7	7	7	7
KEVIN OLIVO	M	2	6	6	5	5	6	7	7	7	7	6	5	4	6	7	7
JOHANNA PUERCHAMBUD	F	2	7	6	6	7	7	7	7	7	5	7	6	7	7	7	7
EDWIN HERRERA	M	2	6	7	6	6	6	7	7	7	6	7	7	7	7	7	7
HENRY IMBAQUINGO	M	2	7	7	7	7	7	6	7	7	7	7	7	6	7	7	7
STEVEN SOLANO	F	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
SEBASTIAN TIRIRA	M	3	6	6	6	6	5	7	7	6	5	6	6	5	7	7	6

JOHNN RAMIREZ	M	3	6	6	6	7	7	7	7	7	7	7	6	5	7	7	7
KEVIN LLIGUISUPA	M	1	7	7	7	7	7	7	7	7	7	6	6	6	7	7	7
BRAYAN PULAMARIN	M	2	6	6	7	6	5	4	7	7	4	6	5	5	7	7	6
DILAN RAMIREZ	M	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
ULISES VALENCIA	M	2	6	6	5	7	6	7	7	7	7	7	4	7	7	7	7
JORGE JARAMILLO	M	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
ALEXIS VACA	M	2	4	5	5	5	6	6	5	5	3	6	1	1	6	1	7
EDISON GUAICHICO	M	4	6	7	7	7	7	7	7	7	6	7	7	7	7	7	7
MATTHEW GARCIA	M	1	6	6	5	7	6	7	7	7	6	7	7	7	7	7	7
GISSELA RAMIREZ	F	2	7	7	7	7	7	7	6	6	6	6	6	6	6	6	6
VANESSA CADENA	F	2	4	5	5	5	5	5	5	5	4	5	4	4	5	5	7
OLIVER ZAMORA	M	2	7	7	7	7	7	6	6	5	6	6	7	6	5	6	7
KATERINE BURGOS	F	3	4	7	7	6	7	6	6	6	6	7	6	5	6	6	7
CRISTIAN TONTAQUIMBA	M	2	6	6	6	6	6	7	6	6	5	6	7	3	6	6	7
KEVIN MENDEZ	M	3	6	7	6	7	6	7	7	7	5	7	6	6	6	7	7
ERICK FLORES	M	1	7	7	7	7	7	7	7	7	5	7	6	7	7	7	7
EDISON PINANJOTA	M	3	6	6	6	5	5	6	6	6	5	5	6	5	5	6	7

Fuente: Propia

A continuación, se presenta los resultados de las respuestas en porcentajes según cada sección:

### Sección 1 Informativa:

#### 1. Seleccione su género.

- 1) Masculino
- 2) Femenino

El 85% de los usuarios pertenecen al género Masculino y el 15% restante son de género Femenino, como se puede apreciar en la figura, lo que significa que el mayor porcentaje de los encuestados es de género masculino, como se muestra en la Figura 60.

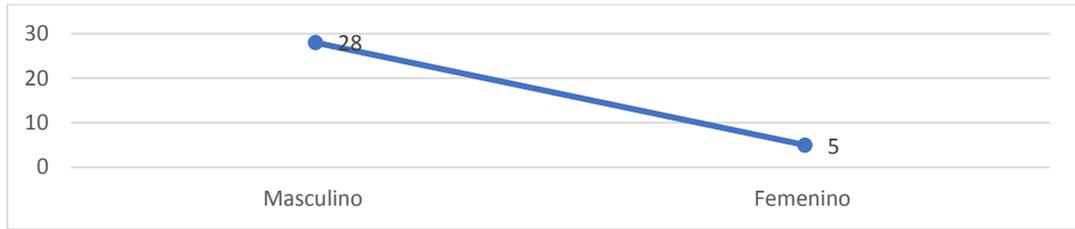


Figura 60 Pregunta 1 - Informativa

Fuente: Propia

2. Seleccione la opción que mejor se ajuste a su color de piel.

- 1) Piel clara
- 2) Piel clara Intermedia
- 3) Piel morena o media
- 4) Piel Oscura o Marrón

El 55 % de los encuestados tienen un color de piel clara intermedia, el siguiente 24% de los usuarios tienen el color de piel morena y media, el 18% tiene un color de piel clara y el 3% tiene piel oscura, lo que significa que el mayor porcentaje de los encuestados tienen piel clara intermedia como se muestra en la Figura 61.

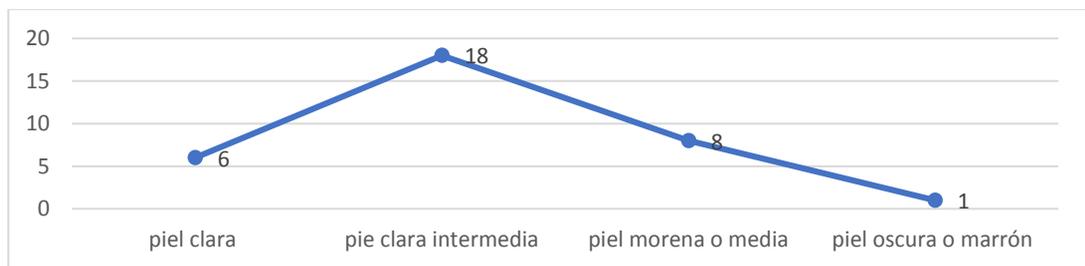


Figura 61 Pregunta 2 – Informativa

Fuente: Propia

## Sección 2 Precisión:

1: ¿Considera usted que el tiempo de respuesta de la aplicación al momento de la ejecución es adecuado?

El 39% de los usuarios respondieron que están totalmente de acuerdo con que el tiempo de respuesta del prototipo de la aplicación móvil es adecuada, mientras que el 46% están medianamente de acuerdo, el 6% dice que está parcialmente de acuerdo y finalmente el 9% no está ni en acuerdo ni en desacuerdo, llegando a la conclusión que el 46% es el porcentaje más alto, lo que significa que los usuarios están medianamente de acuerdo con la precisión al momento de iniciar el prototipo en el tiempo adecuado, los resultados se pueden apreciar en la Figura 62.

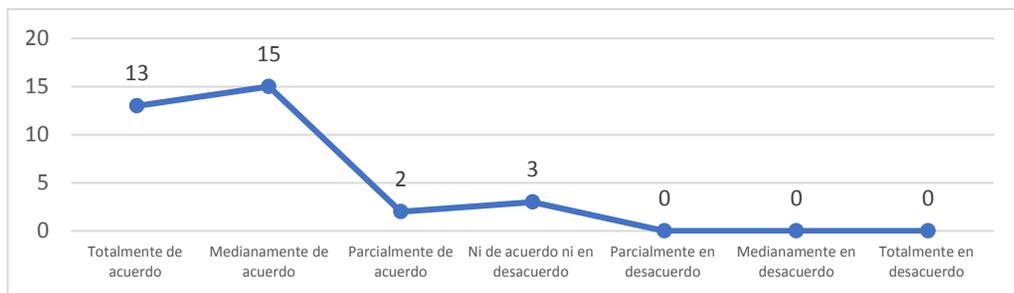


Figura 62 Pregunta 1 – Precisión

Fuente: Propia

2: ¿Considera usted que la precisión de la aplicación al momento de detectar rostros es adecuada?

El 52% de los usuarios respondieron que están totalmente de acuerdo con la precisión de detección de rostros en el prototipo, mientras que el 36% están medianamente de acuerdo y finalmente el 12% está parcialmente de acuerdo, llegando a la conclusión que el 52% es el porcentaje más alto, lo que significa que los usuarios están medianamente de acuerdo con la precisión al momento de iniciar el prototipo en el tiempo adecuado, los resultados se pueden apreciar en la Figura 63.



Figura 63 Pregunta 2 - Precisión

Fuente Propia

3: ¿Considera usted que la precisión de la aplicación al momento de detectar ojos es adecuada?

El 40% de los usuarios respondieron que están totalmente de acuerdo con la precisión de detección de ojos en el prototipo, mientras que el 30% están medianamente de acuerdo y finalmente el 30% está parcialmente de acuerdo, llegando a la conclusión que el 40% es el porcentaje más alto, lo que significa que la mayoría de los usuarios están totalmente de acuerdo con la precisión al momento de detectar ojos, los resultados se pueden apreciar en la Figura 64.



Figura 64 Pregunta 3 - Precisión

Fuente Propia

4: ¿Considera usted que la alarma sonora se activa en el momento adecuado, es decir cuando la persona ya está en estado de somnolencia?

El 61% de los usuarios respondieron que están totalmente de acuerdo con la precisión de la emisión de la alarma en el momento preciso, mientras que el 21% están medianamente de acuerdo y finalmente el 18% está parcialmente de acuerdo, llegando a la conclusión que el 61% es el porcentaje más alto, lo que significa que la mayoría de los usuarios están totalmente de acuerdo con la precisión de la emisión de la alarma, los resultados se pueden apreciar en la Figura 65.

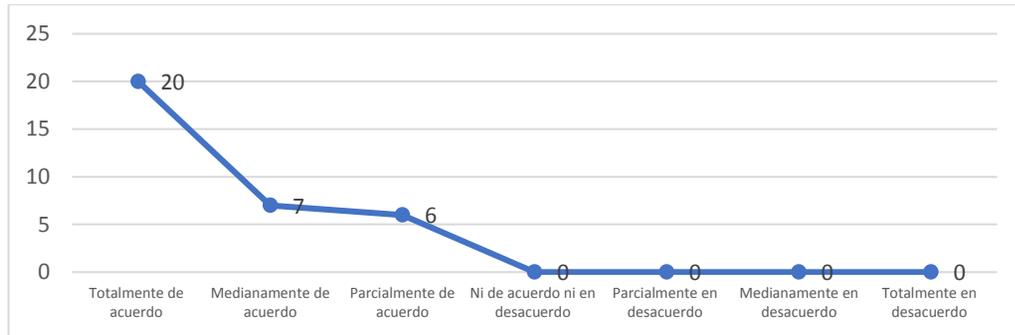


Figura 65 Pregunta 4 - Precisión

Fuente Propia

5: ¿Considera usted que el tiempo de ejecución de la alarma sonora es adecuado?

El 49% de los usuarios respondieron que están totalmente de acuerdo con la precisión del tiempo de ejecución de la alarma, mientras que el 36% están medianamente de acuerdo y finalmente el 12% está parcialmente de acuerdo, llegando a la conclusión que el 49% es el porcentaje más alto, lo que significa que la mayoría de los usuarios están totalmente de acuerdo con la precisión del tiempo de ejecución de la alarma, los resultados se pueden apreciar en la Figura 66.

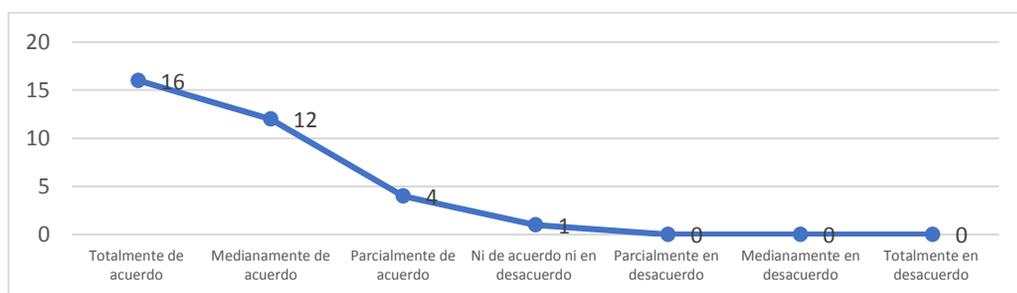


Figura 66 Pregunta 5 - Precisión

Fuente Propia

### Sección 3 Solución al Problema:

1: ¿Considera usted que el uso de la aplicación ayudaría a reducir pérdidas materiales?

El 67% de los usuarios respondieron que están totalmente de acuerdo que el prototipo ayudaría a solucionar el problema planteado en cuanto a reducción de pérdidas materiales, mientras que el 24% están medianamente de acuerdo y finalmente el 6% está parcialmente de acuerdo, llegando a la conclusión que el 67% es el porcentaje más alto,

lo que significa que la mayoría de los usuarios están totalmente de acuerdo con que el prototipo ayudaría a reducir las pérdidas materiales, los resultados se pueden apreciar en la Figura 67.

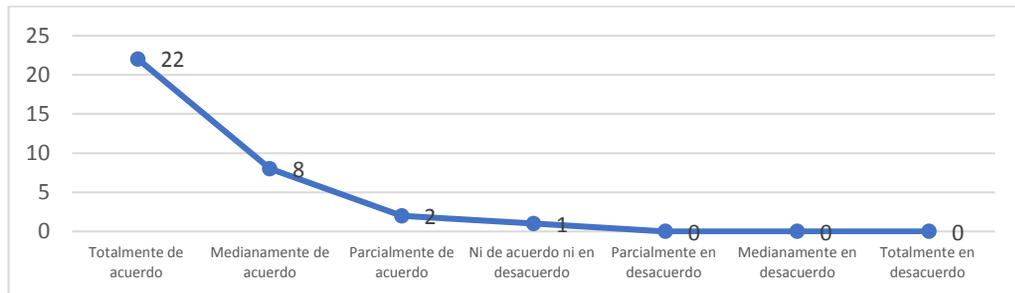


Figura 67 Pregunta 1 - Solución al Problema

Fuente Propia

2: ¿Considera usted que el uso de la aplicación ayudaría a reducir el número de pérdidas humanas en nuestro País?

El 70% de los usuarios respondieron que están totalmente de acuerdo con que el prototipo ayudaría a solucionar el problema planteado en cuanto a reducción de pérdidas humanas, mientras que el 18% están medianamente de acuerdo y finalmente el 12% está parcialmente de acuerdo, llegando a la conclusión que el 70% es el porcentaje más alto, lo que significa que la mayoría de los usuarios están totalmente de acuerdo con que el prototipo ayudaría a reducir las pérdidas humanas, los resultados se pueden apreciar en la Figura 68.



Figura 68 Pregunta 2 - Solución al Problema

Fuente Propia

3: ¿Considera usted que la aplicación ayudaría a reducir el número de accidentes de tránsito que se registran periódicamente por somnolencia?

El 64% de los usuarios respondieron que están totalmente de acuerdo con que el prototipo ayudaría a solucionar el problema planteado en cuanto a reducción del número de accidentes de tránsito, mientras que el 21% están medianamente de acuerdo y finalmente el 15% está parcialmente de acuerdo, llegando a la conclusión que el 64% es el porcentaje más alto, lo que significa que la mayoría de los usuarios están totalmente de acuerdo con que el prototipo ayudaría a reducir el número de accidentes de tránsito, los resultados se pueden apreciar en la Figura 69.

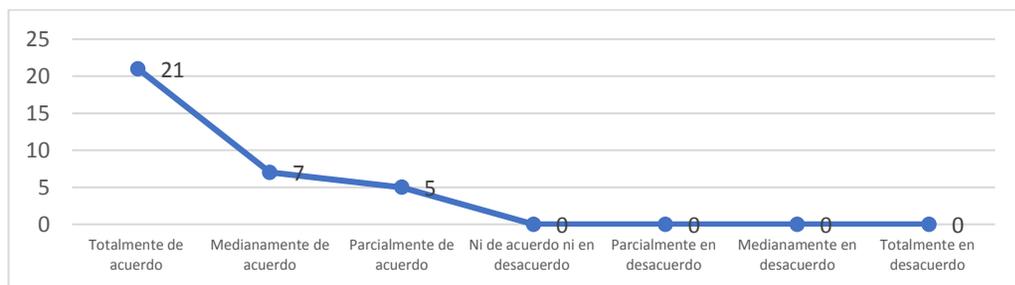


Figura 69 Pregunta 3 - Solución al Problema

Fuente Propia

4: ¿Considera usted que el uso adecuado de la aplicación móvil ayudaría a reducir problemas sociales y psicológicos de las personas?

El 37% de los usuarios respondieron que están totalmente de acuerdo con que el prototipo ayudaría a reducir problemas sociales y psicológicos de las personas, mientras que el 30% están medianamente de acuerdo, 18% está parcialmente de acuerdo, 12% no está ni en acuerdo ni en desacuerdo y finalmente el 3% está parcialmente en desacuerdo, llegando a la conclusión que el 37% es el porcentaje más alto, lo que significa que en su mayoría los usuarios están totalmente de acuerdo con que el prototipo ayudaría a reducir problemas sociales y psicológicos de las personas, los resultados se pueden apreciar en la Figura 70.



Figura 70 Pregunta 4 - Solución al Problema

Fuente Propia

5: ¿Considera usted que la aplicación es adecuada para alertar al conductor?

El 55% de los usuarios respondieron que están totalmente de acuerdo con que el prototipo es adecuado para alertar al conductor, mientras que el 33% están medianamente de acuerdo y finalmente el 12% está parcialmente de acuerdo, llegando a la conclusión que el 55% es el porcentaje más alto, lo que significa que la mayoría de los usuarios están totalmente de acuerdo con que el prototipo es adecuado para alertar al conductor cuando este es estado de somnolencia, los resultados se pueden apreciar en la Figura 71.



Figura 71 Pregunta 5 - Solución al Problema

Fuente Propia

#### Sección 4 Impacto:

1: ¿Considera usted que la puesta en producción de la aplicación móvil generaría fuentes de trabajo?

El 43% de los usuarios respondieron que están totalmente de acuerdo con que la puesta en producción del prototipo ayudaría a generar fuentes de trabajo, mientras que el 33%

están medianamente de acuerdo, 12% está parcialmente de acuerdo, 9% no está ni en acuerdo ni en desacuerdo y finalmente el 3% está totalmente en desacuerdo, llegando a la conclusión que el 43% es el porcentaje más alto, lo que significa que en su mayoría los usuarios están totalmente de acuerdo con que la puesta en producción de la aplicación móvil generaría fuentes de trabajo, los resultados se pueden apreciar en la Figura 72.



Figura 72 Pregunta 1 - Impacto

Fuente Propia

2: ¿Considera usted que la aplicación móvil ayudaría al desarrollo económico del país?

El 37% de los usuarios respondieron que están totalmente de acuerdo con que la puesta en producción de la aplicación móvil ayudaría al desarrollo económico del país, mientras que el 24% están medianamente de acuerdo, 21% está parcialmente de acuerdo, 9% no está ni en acuerdo ni en desacuerdo, 6% están parcialmente en desacuerdo y finalmente el 3% está totalmente en desacuerdo, llegando a la conclusión que el 37% es el porcentaje más alto, lo que significa que en su mayoría los usuarios están totalmente de acuerdo con que la puesta en producción de la aplicación móvil ayudaría al desarrollo económico del país, los resultados se pueden apreciar en la Figura 73.



Figura 73 Pregunta 2 – Impacto

Fuente Propia

3: ¿Considera usted que el desarrollo de la aplicación ayuda a la innovación tecnológica?

El 67% de los usuarios respondieron que están totalmente de acuerdo con que el prototipo ayuda a la innovación tecnológica, mientras que el 18% están medianamente de acuerdo y finalmente el 15% está parcialmente de acuerdo, llegando a la conclusión que el 67% es el porcentaje más alto, lo que significa que la mayoría de los usuarios están totalmente de acuerdo con que el prototipo ayuda a la innovación tecnológica, los resultados se pueden apreciar en la Figura 74.



Figura 74 Pregunta 3 - Impacto

Fuente Propia

4: ¿Considera usted que el uso de la aplicación ayudaría a garantizar el bienestar y la seguridad de las personas?

El 64% de los usuarios respondieron que están totalmente de acuerdo con que la puesta en producción del prototipo ayudaría a garantizar el bienestar de las personas, mientras que el 24% están medianamente de acuerdo, 64% está parcialmente de acuerdo, 3% no está ni en acuerdo ni en desacuerdo y finalmente el 3% está totalmente en desacuerdo, llegando a la conclusión que el 64% es el porcentaje más alto, lo que significa que en su mayoría los usuarios están totalmente de acuerdo con que la puesta en producción de la aplicación móvil ayudaría a garantizar el bienestar y la seguridad de las personas los resultados se pueden apreciar en la Figura 75.

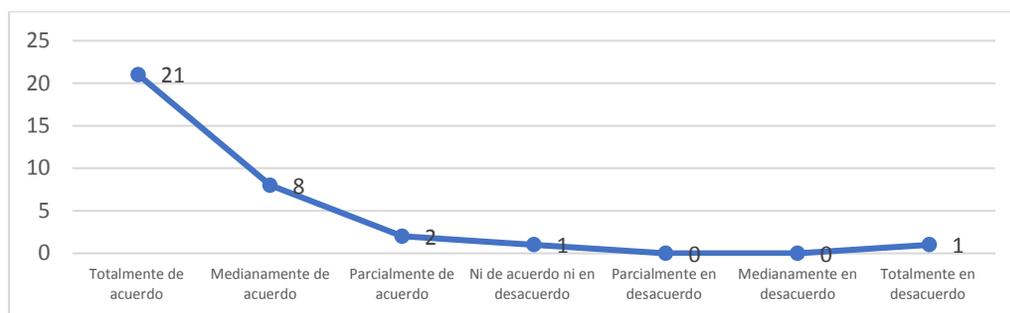


Figura 75 Pregunta 4 - Impacto

Fuente Propia

5: ¿Considera usted que es necesario seguir impartiendo materias sobre Inteligencia Artificial y Visión por Computador?

El 82% de los usuarios respondieron que están totalmente de acuerdo con que se siga impartiendo materias sobre Inteligencia Artificial y Visión por Computador, mientras que el 12% están medianamente de acuerdo y finalmente el 6% está parcialmente de acuerdo, llegando a la conclusión que el 82% es el porcentaje más alto, lo que significa que la mayoría de los usuarios están totalmente de acuerdo con que se siga impartiendo materias sobre Inteligencia Artificial y Visión por Computador, los resultados se pueden apreciar en la Figura 76.

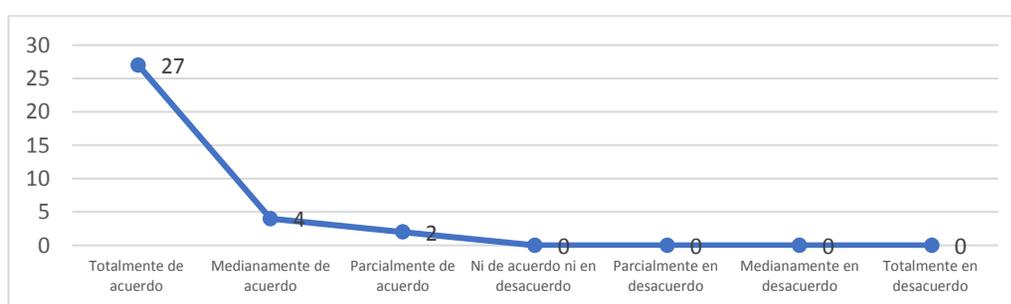


Figura 76 Pregunta 5 - Impacto

Fuente Propia

### 3.13 Análisis e interpretación de resultados de la encuesta

Para efectuar la validación de la encuesta y garantizar la validez y la fiabilidad, se procede a realizar un Análisis Factorial Confirmatorio (ACF) para validar la estructura factorial, para ello se hizo uso del software R Studio Online, donde se obtuvo un modelo de 3 estados para que se

ajuste a las 3 secciones: precisión, solución al problema e impacto, el mismo que generó saturaciones por encima de 0.4 por cada pregunta, asignando los siguientes pesos para cada pregunta de cada sección como muestra en la Tabla 3.15 o en la Figura 77:

TABLA 3.15  
PESOS DE CADA PREGUNTA PARA CADA SECCIÓN

Sección	P 1	P 2	P 3	P 4	P 5
Precisión	0.76	0.83	0.77	0.67	0.70
Solución al Problema	0.59	0.94	0.85	0.77	0.70
Impacto	0.67	0.78	0.78	0.87	0.44

Fuente: Propia

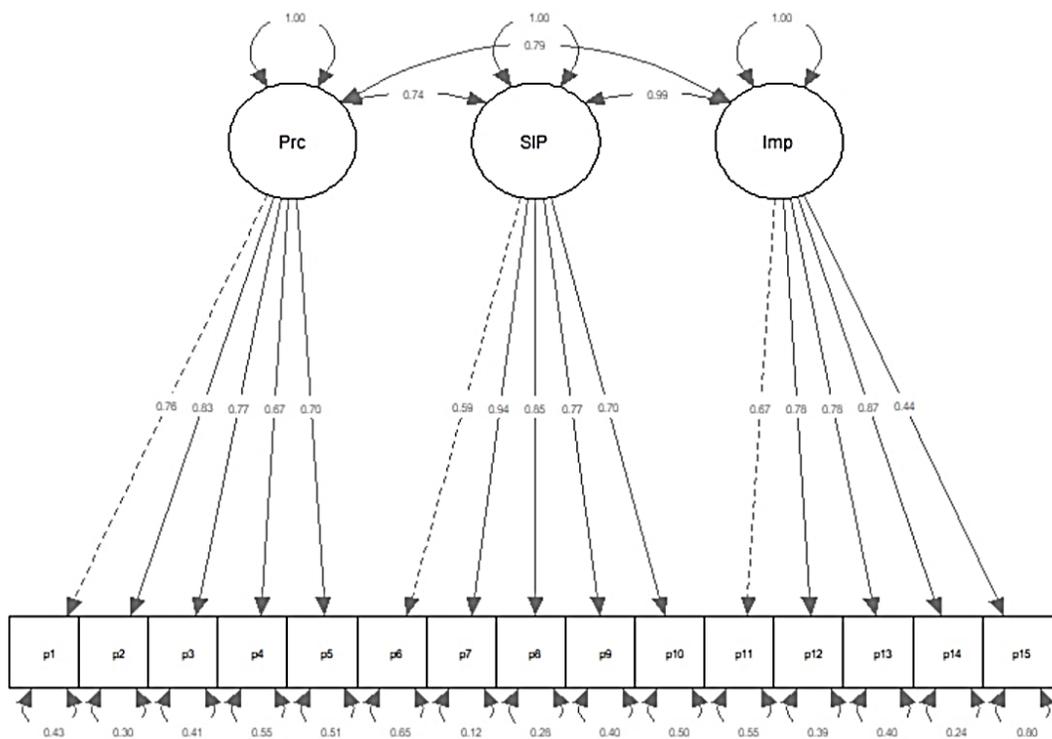


Figura 77 Análisis Factorial Combinatorio

Fuente: Propia

El siguiente paso que se realizó es calcular el resultado de las 3 secciones con los pesos correspondientes, en cada uno de los usuarios que probaron el prototipo y finalmente la media general de cada sección, usando las siguientes fórmulas, como se muestra en la Tabla 3.16:

TABLA 3.16

CÁLCULO DE PUNTAJE OBTENIDO EN CADA SECCIÓN SEGUN CADA USUARIO SOBRE EL 100%

Sección	Fórmula	Resultado
<b>Precisión</b>	$master[,16]=100*(0.76*master[,1]+0.83*master[,2]+0.77*master[,3]+0.67*m aster[,4]+0.70*master[,5])/(7*(0.76+0.83+0.77+0.67+0.70))$	colnames(master)=c (colnames(master[,16]),"Precision")
<b>Solución al Problema</b>	$master[,17]=100*(0.59*master[,6]+0.94*master[,7]+0.85*master[,8]+0.77*m aster[,9]+0.70*master[,10])/(7*(0.59+0.94+0.85+0.77+0.70))$	colnames(master)=c (colnames(master[,17]),"SolucionProblema")
<b>Impacto</b>	$master[,18]=100*(0.67*master[,11]+0.78*master[,12]+0.78*master[,13]+0.87*master[,14]+0.44*master[,15])/(7*(0.67+0.78+0.78+0.87+0.44))$	colnames(master)=c (colnames(master[,18]),"Impacto")

Fuente: Propia

Los resultados se muestran a continuación en la Tabla 3.17:

TABLA 3.17

RESULTADOS DE LAS SECCIONES CALCULADAS

Usuario	Precisión %	Solución al Problema %	Impacto %
ERICK HERRERA	86.05898	85.64007	87.48991
JUAN DIEGO IZA	79.24167	73.35807	62.06618
KARLA HERRERA	74.49253	71.72542	71.42857
RICHARD AGUIRRE	100	100	97.29621
WILSON SEVILLA	94.14018	100	88.29701
DEISON MARTINEZ	88.28035	92.35622	91.56578
JOSHUA ALMEIDA	79.58637	97.14286	96.8523
ELVIS TUQUERREZ	83.10992	82.85714	93.3414
EMERSON MONTESDEOCA	100	100	100
JOSTIN ROLDAN	94.36997	100	100
KEVIN OLIVO	80.19916	97.4026	82.00161
JOHANNA PUERCHAMBUD	93.87208	94.28571	97.29621
EDWIN HERRERA	88.89314	97.14286	100
HENRY IMBAQUINGO	100	97.81076	96.8523
STEVEN SOLANO	100	100	100
SEBASTIAN TIRIRA	83.03332	88.53432	89.22518

<b>JOHNN RAMIREZ</b>	90.96132	100	91.00081
<b>KEVIN LLIGUISUPA</b>	100	97.4026	94.14851
<b>BRAYAN PULAMARIN</b>	85.98238	82.26345	86.52139
<b>DILAN RAMIREZ</b>	100	100	100
<b>ULISES VALENCIA</b>	85.33129	100	91.88862
<b>JORGE JARAMILLO</b>	100	100	100
<b>ALEXIS VACA</b>	71.19877	70.50093	40.67797
<b>EDISON GUAICHICO</b>	97.08924	97.14286	100
<b>MATTHEW GARCIA</b>	85.33129	97.14286	100
<b>GISELA RAMIREZ</b>	100	87.90353	85.71429
<b>VANESSA CADENA</b>	68.51781	68.57143	69.12833
<b>OLIVER ZAMORA</b>	100	82.5603	87.046
<b>KATERINE BURGOS</b>	88.70165	88.31169	84.34221
<b>CRISTIAN TONTAQUIMBA</b>	85.71429	85.04638	80.75061
<b>KEVIN MENDEZ</b>	91.45921	94.28571	91.00081
<b>ERICK FLORES</b>	100	94.28571	97.29621
<b>EDISON PINANJOTA</b>	80.46725	80.25974	81.19451

Fuente: Propia

Una vez calculadas las 3 secciones se procede a calcular la media de cada una de ellas dando como resultado los valores de la Tabla 3.18 :

TABLA 3.18

CÁLCULO Y RESULTADO DE LA MEDIA OBTENIDA EN CADA SECCIÓN

<b>Sección</b>	<b>Media</b>	<b>Resultado</b>
<b>Precisión</b>	mean(master\$Precision)	89.57%
<b>Solución al Problema</b>	mean(master\$SolucionProblema)	91.028%
<b>Impacto</b>	mean(master\$Impacto)	89.92%

Fuente: Propia

## CONCLUSIONES

- ✓ Toda la información indagada fue útil para realizar la parte del marco teórico correspondiente al objetivo número uno de este trabajo de investigación. En lo referente a herramientas, algoritmos y estudios realizados con visión artificial se logró definir el método a utilizar y las herramientas necesarias para realizar el desarrollo del prototipo de la aplicación móvil.
- ✓ Una vez determinada la técnica a usarse en el desarrollo del proyecto y después de haber realizado la integración del IDE de desarrollo y las librerías necesarias de OpenCV, se realizó la implementación de la técnica de detección de pupilas en la codificación de la aplicación móvil y dando como resultado el detector de somnolencia nocturna en una aplicación móvil para sistemas operativos móviles Android.
- ✓ Para el adecuado funcionamiento del prototipo de la aplicación móvil fue necesario encontrar una alternativa que provea la luminosidad suficiente para que el prototipo funcione en condiciones nocturnas, la mejor solución fue acortar el área de presentación del usuario con la finalidad de dejar espacios en blanco los mismos que ayudaron emitiendo luz blanca que proporciona la pantalla del celular y de esa forma la aplicación móvil funciona adecuadamente.
- ✓ Mediante la aplicación del modelo de regresión lineal multivariada se determinó el tamaño de la muestra para efectuar las pruebas del prototipo, seguido se aplicó el cálculo de Métricas en el software R Studio Online se realizó la evaluación del prototipo de la aplicación móvil permitiendo validar algunas de las más conocidas métricas como son precisión, exactitud, sensibilidad, especificidad, medida F1 y finalmente la curva ROC, dando como resultados puntajes muy buenos en cada una de ellas.

## RECOMENDACIONES

- ✓ Elegir cuidadosamente la versión del IDE de desarrollo y de las librerías que se encuentran en el mercado para que no presenten inconvenientes al momento de integrar las mismas, asimismo de inestabilidad y por ende de funcionamiento al momento de desarrollar.
- ✓ Para trabajos futuros de Tesis en el campo de Inteligencia Artificial específicamente Visión por Computador, es necesario verificar si existe documentación necesaria para abordar el campo de estudio elegido y realizar pruebas previamente para verificar si la propuesta es viable y desarrollable.
- ✓ Que se siga motivando a los estudiantes a seguir indagando y dando énfasis al campo de la investigación, con la finalidad de dar soluciones tecnológicas e innovadoras a los diversos problemas sociales que estamos expuestos en el día a día.
- ✓ Para el caso de detección de somnolencia nocturna, se sugiere continuar buscando opciones que ayuden a mejorar el rendimiento del algoritmo ya sea empleando nuevas técnicas, mejorando el algoritmo presentado en este documento o implementando hardware para el apoyo de luminosidad para obviar el uso de la iluminación de la pantalla del smartphone.
- ✓ Usar una metodología ágil para el desarrollo de los proyectos, como XP ya que se adapta fácilmente a cualquier proyecto y está dirigida a pequeños y medianos grupos de trabajo y además se dice que es muy utilizada para implementar nuevas tecnologías donde los requerimientos cambian con gran facilidad.
- ✓ Seguir haciendo uso de herramientas estadísticas ya que ayudan a validar los resultados obtenidos y de esa forma tener una validación de resultados con calidad.

## REFERENCIAS

- 4Mejores. (21 de 08 de 2017). *Las 4 mejores cámaras termográficas baratas y de calidad*. Obtenido de <https://www.4mejores.com/las-4-mejores-cameras-termograficas-baratas-y-de-calidad/>
- Acuña Morgado, J. M. (17 de 04 de 2018). *TecnoBlog*. Obtenido de <https://jmacuna.tecnoblog.guru/2017/03/sistemas-operativos-moviles.html>
- Agencia Nacional de Tránsito. (2016). *Visión Mision*. Obtenido de [www.ant.com](http://www.ant.com): <https://www.ant.gob.ec/index.php/noticias/estadisticas>
- Agencia Nacional de Tránsito. (Enero de 2018). *Estadísticas de transporte terrestre y seguridad vial*. Obtenido de <https://www.ant.gob.ec/index.php/noticias/estadisticas>
- ARQHYS. (2012). Cámaras infrarrojas. *ARQHYS*, 12.
- Barrero, E. P., Vázquez, Á. R., & Berni, J. F. (2015). *Aceleración del algoritmo de Viola-Jones mediante rejillas de procesamiento masivamente paralelo en el plano focal*. 68.
- Bergasa, L. M., Nuevo, J., Sotelo, M. A., Barea, R., & Lopez, M. E. (2006). Real-Time System for Monitoring Driver Vigilance. *IEEE Transactions on Intelligent Transportation Systems*, 7(1), 63-77. <https://doi.org/10.1109/TITS.2006.869598>
- Brandt, T., Stemmer, R., & Rakotonirainy, A. (2004). *Affordable Visual ]DriverMonitoring System for Fatigue and Monotony*. 6.
- Byung-Chan Chang, Jung-Eun Lim, Hae-Jin Kim, & Bo-Hyeok Seo. (2007). A study of classification of the level of sleepiness for the drowsy driving prevention. *SICE Annual Conference 2007*, 3084-3089. <https://doi.org/10.1109/SICE.2007.4421521>
- Cárdenas Castro, J. M. (2014). Potencia estadística y cálculo del tamaño del efecto en G\*Power: Complementos a las pruebas de significación estadística y su aplicación en psicología. *Salud & Sociedad*, 5(2), 210-244. <https://doi.org/10.22199/S07187475.2014.0002.00006>
- Castro, J. R., & Rosales-Mayor, E. (2010). CANSANCIO Y SOMNOLENCIA DURANTE EL DESEMPEÑO LABORAL DE LOS CONDUCTORES INTERPROVINCIALES: EXPERIENCIA PERUANA Y PLANTEAMIENTO DE PROPUESTAS. *Rev Peru Med Exp Salud Publica.*, 6.

- Cech, J. (2016). Real-Time Eye Blink Detection using Facial Landmarks. *Center for Machine Perception, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University in Prague*, 1-8. <https://doi.org/10.1017/CBO9781107415324.004>
- Cool Cosmos. (20 de 05 de 2019). Obtenido de [http://legacy.spitzer.caltech.edu/espanol/edu/ir\\_zoo/dog.html](http://legacy.spitzer.caltech.edu/espanol/edu/ir_zoo/dog.html)
- De, R. O. S. (2014). *LEY ORGANICA DE TRANSPORTE TERRESTRE TRANSITO Y SEGURIDAD VIAL*. 1-66.
- D’Orazio, T., Leo, M., & Distante, A. (2004). Eye detection in face images for a driver vigilance system. *IEEE Intelligent Vehicles Symposium, 2004*, 95-98. <https://doi.org/10.1109/IVS.2004.1336362>
- EGAS CUNALATA , F. D. (Agosto de 2017). Sistema de Vigilancia al conductor vehicular basado en técnicas de Visión Artificial e implementado en un smartphone para la detección y alerta de somnolencia. 2. Latacunga, Ecuador.
- Flores, M., Armingol, J., & de la Escalera, A. (2010). Driver Drowsiness Warning System Using Visual Information for Both Diurnal and Nocturnal Illumination Conditions. *EURASIP Journal on Advances in Signal Processing, 2010*(1). <https://doi.org/10.1155/2010/438205>
- Garcés, M. A., Salgado, J. D. J., Cruz, J. A., & Cañon, W. H. (2015). Sistemas de detección de somnolencia en conductores: Inicio, desarrollo y futuro. *Ingeniería y Región, 13*(1), 159. <https://doi.org/10.25054/22161325.717>
- Grace, R., & Steward, S. (2001). Drowsy Driver Monitor and Warning System. *Proceedings of the First International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design: Driving Assessment 2001*, 64-69. <https://doi.org/10.17077/drivingassessment.1010>
- González , P. M., & López de Mántaras, B. R. (2017). *Inteligencia artificial*. CSIC Consejo Superior de Investigaciones Científicas. Obtenido de <http://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=5307873>
- Guzmán, C. (2014). *Análisis de los indicadores de seguridad vial para la disminución de accidentes de tránsito en el Ecuador*. 143.

- Herrera Granda, E. P., Caraguay Procel, J., Granda Gudiño, P., Herrera Granda, I., Lorente Leyva Diego, L., & Peluffo Ordoñez, D. (6 de Agosto de 2018). Drowsiness Detection in Drivers through Real-time Image Processing of the Human Eye. *Drowsiness Detection in Drivers through Real-time Image Processing of the Human Eye*.
- Ji, Q. (2002). Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance. *Real-Time Imaging*, 8(5), 357-377. <https://doi.org/10.1006/rtim.2002.0279>
- Ji, Q., Zhu, Z., & Lan, P. (2004). Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue. *IEEE Transactions on Vehicular Technology*, 53(4), 1052-1068. <https://doi.org/10.1109/TVT.2004.830974>
- Kaggle. (2018). *Técnicas de detección de objetos*. Obtenido de <https://www.kaggle.com/infernop/object-detection-techniques>
- Liu, C. C., Hosking, S. G., & Lenné, M. G. (2009). Predicting driver drowsiness using vehicle measures: Recent insights and future challenges. *Journal of Safety Research*, 40(4), 239-245. <https://doi.org/10.1016/j.jsr.2009.04.005>
- López Romero, W. L. (2016). *Sistema de control del estado de somnolencia en conductores de vehículos*. 122.
- Lucía, A. (2013). *FFT: Transformada Rápida de Fourier*. 3.
- Maxwell, T. (13 de 08 de 2015). *9to5google*. Obtenido de <https://9to5google.com/2015/08/13/google-mobile-vision-api-android/>
- Ministerio de Relaciones Laborales. (12 de 12 de 2018). *Ministerio del Trabajo - Ecuador*. Obtenido de <http://www.trabajo.gob.ec/wp-content/uploads/downloads/2013/01/Normas-que-regulan-la-aplicacion-y-procedimiento-de-autorizaci%C3%B3n-de-horarios-especiales.pdf>
- Naciones Unidas. (2015). 17 objetivos para transformar nuestro mundo. *17 objetivos para transformar nuestro mundo*. Obtenido de <https://www.un.org/sustainabledevelopment/es/>
- Organización Mundial de Salud. (03 de Diciembre de 2018). *Prevención de la violencia y los traumatismos*. Obtenido de [https://www.who.int/violence\\_injury\\_prevention/road\\_traffic/es/](https://www.who.int/violence_injury_prevention/road_traffic/es/)

- Pal, S., Bhattacharya, K. F., Agapito, C., & Chaudhuri, K. R. (2001). *A study of excessive daytime sleepiness and its clinical significance in three groups of Parkinson 's disease patients taking pramipexole , cabergoline and levodopa mono and combination therapy.* 71-72.
- Polanco, K. M., & Taibo, J. L. B. (2011). «ANDROID» GOOGLE'S OPERATING SYSTEM FOR MOBILE DEVICES. *journal of Management*, 18.
- Ponce, L. A. E., García, Y. R., Alonso, J. M. M., & Gonzáles, A. L. (2018). *VEHÍCULO AÉREO NO TRIPULADO Y VISIÓN POR COMPUTADORA PARA EL MONITOREO DE FAUNA.* 40(130), 15.
- Qualcomm Developer Network. (2016). *FastCV Computer Vision SDK.* Obtenido de <https://developer.qualcomm.com/software/fastcv-sdk>
- Reddy, B., Kim, Y., Yun, S., Seo, C., & Jang, J. (s. f.). 2.1. *Conventional Approaches for Drowsiness Detection.*
- Sánchez Calle, Á. D. (2005). *Aplicaciones de la visión artificial y la biometría informática.* Calle Ángel Sánchez Dykinson,. Obtenido de <http://ebookcentral.proquest.com/lib/utnortesp/detail.action?docID=4498933>.
- Speckmann, B. (2018). *The Android mobile platform.* 131.
- Stone, H. S., & Li, C.-S. (1996). *Image matching by means of intensity and texture matching in the Fourier domain* (I. K. Sethi & R. C. Jain, Eds.; pp. 337-349). <https://doi.org/10.1117/12.234773>
- Torres, Y., & Palma, M. (2017). *Aplicación móvil para la detección de somnolencia de un conductor aplicando visión artificial.* 12.
- Tudela, J. A., Vázquez, C. C., & de, E. (2009). *DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES SOBRE LA PLATAFORMA ANDROID DE GOOGLE.* 199.
- Villanueva, M. G., & Zavala, S. R. (2013). *Integración de OpenCv en una Raspberry Pi: Sistema de Detección de Rostro vía WEB.* 7.
- Wen-Bing Horng, Chih-Yuan Chen, Yi Chang, & Chun-Hai Fan. (2004). Driver fatigue detection based on eye tracking and dynamic template matching. *IEEE International Conference on Networking, Sensing and Control, 2004*, 7-12. <https://doi.org/10.1109/ICNSC.2004.1297400>