



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

“DETECTOR DE ALCOHOLEMIA PARA CONDUCTORES QUE ANALIZA
VARIABLES FACIALES Y AMBIENTALES DEL AUTOMÓVIL MEDIANTE EL
APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA REDUCCIÓN DE
ACCIDENTES DE TRÁNSITO”.

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN

ELECTRÓNICA Y REDES DE COMUNICACIÓN

AUTORA: PORTILLA ARTEAGA LICETH GEOVANA

DIRECTOR: MSC. JAIME ROBERTO MICHILENA CALDERÓN

IBARRA - ECUADOR



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN.

MAGISTER JAIME MICHILENA, DIRECTOR DEL PRESENTE TRABAJO DE
TITULACIÓN CERTIFICA:

Que, el presente trabajo de Titulación “DETECTOR DE ALCOHOLEMIA PARA CONDUCTORES QUE ANALIZA VARIABLES FACIALES Y AMBIENTALES DEL AUTOMÓVIL MEDIANTE EL APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA REDUCCIÓN DE ACCIDENTES DE TRÁNSITO”. Ha sido desarrollado por la señorita Portilla Arteaga Liceth Geovana portadora de la cédula de identidad número 040172412-5, bajo mi supervisión.

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



.....

Ing. Jaime Michilena, MSc.

1002198438

DIRECTOR



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DECLARACIÓN.

Yo, Liceth Geovana Portilla Arteaga con cédula de identidad 040172412-5, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; y que éste no ha sido previamente presentado para ningún grado o calificación profesional.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo a la Universidad Técnica del Norte, según lo establecido en las leyes de Propiedad Intelectual, Reglamentos y Normatividad vigente en la Universidad Técnica del Norte.

Liceth Geovana Portilla Arteaga

040172412-5

Ibarra, mayo del 2018

Dedicatoria

Dedico este trabajo.

A mi padre José Luis Portilla quien me ha demostrado que se debe luchar día tras día para superar los obstáculos, por alentarme a culminar lo que un día inicié sin importar las dificultades que se presenten, por su paciencia, su compañía, su amor y su apoyo incondicional, pero sobre todo por forjar mi vida con valores de solidaridad, responsabilidad, honestidad, respeto y honradez.

A mi madre Yolanda Arteaga por el tiempo que estuvo a mi lado para guiarme, brindarme su apoyo y sus consejos.

A mis hermanos Vanessa y Brandon quienes me inspiran para demostrarles que con trabajo duro y esfuerzo se pueden lograr todas las metas propuestas y por estar a mi lado en todo el transcurso de mi carrera para darme su apoyo incondicional.

A mis familiares y amigos que siempre tuvieron confianza en que lograría llegar a cumplir esta meta, por su apoyo desinteresado y sus palabras de aliento.

A ti Jonathan, por tu apoyo permanente durante toda mi carrera universitaria, por tu paciencia y amor, por motivarme a ser mejor día a día y sobre todo por creer en mí y compartir tu tiempo conmigo.

Liceth Geovana Portilla

Agradecimiento

Agradezco infinitamente,

A Dios, a mis padres y hermanos por guiar mi camino, brindarme sabiduría y perseverancia, para lograr culminar esta meta.

A la familia Calderón Grijalva y a mis amigos: Edison Mauricio, Paola, Edison, Pamela, Luis, Paola Alexandra, Andrés, José Luis, Gabriela, Carla, Andrea, Darío, Galo Mauricio y Jonathan por su apoyo en el desarrollo de mi trabajo de titulación.

Al Ingeniero Jaime Michilena MSc. y la Ingeniera Sandra Narvárez, director y codirectora de este Trabajo de Grado por compartir su tiempo y conocimientos que contribuyeron para culminar mi carrera estudiantil de la mejor forma.

A los señores conductores de la Cooperativa de Taxis “Atahualpa” por su apoyo y sugerencias durante las pruebas de funcionamiento del sistema.

Al Servicio Integrado de seguridad ECU 911, en especial al Ingeniero Danny Ontaneda Supervisor de Transporte Seguro por su colaboración para realizar las pruebas de la activación del botón de pánico.

Finalmente, a las autoridades de la Universidad Técnica del Norte, al Coordinador de la Carrera de Electrónica y Redes de Comunicación y a todos mis docentes que durante mi estancia como estudiante supieron ayudarme a crecer como profesional y como persona.

Liceth Portilla

Índice

CERTIFICACIÓN.....	II
DECLARACIÓN.....	III
Dedicatoria.....	IV
Agradecimiento.....	V
Índice.....	VI
Índice de Figuras.....	XIII
Índice de Tablas.....	XXI
Índice de Ecuaciones.....	XXIII
Índice de los Anexos.....	XXIV
Resumen.....	XXVI
Abstract.....	XXVII
Presentación.....	XXVIII
Capítulo I. Antecedentes.....	1
1.1. Tema.....	1
1.2. Planteamiento del problema.....	1
1.3. Objetivos.....	3
1.3.1. Objetivo General.....	3
1.3.2. Objetivos Específicos.....	3
1.4. Alcance.....	4
1.5. Justificación.....	6
1.6. Contexto.....	8

Capítulo II. Revisión bibliográfica.....	9
2.1. Accidentes de tránsito en el Ecuador	9
2.1.1. Registro de Siniestros de tránsito a causa de conducir bajo los efectos de alcohol. .	12
2.2. Comportamiento del cuerpo humano al ingerir bebidas alcohólicas	13
2.2.1. Efectos del alcohol al conducir un vehículo.	15
2.2.2. Sanciones que se aplican a conductores en estado de ebriedad en el Ecuador.	17
2.3. Métodos para la detección de alcoholemia en conductores	18
2.3.1. Análisis de alcohol en sangre.	18
2.3.2. Análisis de alcohol en la orina.	20
2.3.3. Análisis de alcohol en la saliva.	21
2.3.4. Análisis de alcohol en el aliento.....	22
2.3.4.1. Dispositivos de prueba de aliento evidenciales.....	23
2.3.4.2. Dispositivos de desechables.....	24
2.3.4.3. Dispositivos de bloqueo en vehículos.....	24
2.3.4.4. Sensores de detección de gases.....	25
2.3.5. Análisis de los cambios térmicos en personas en estado etílico.	27
2.3.5.1. Detección de cambios térmicos en el rostro.....	27
2.3.6. Relación entre el consumo de alcohol y la dilatación de la Pupila.	29
2.4. Visión por computador	32
2.4.1. Etapas básicas de una aplicación en visión artificial.	33
2.4.2. Librería de Visión Computarizada de Código Abierto.	34

2.4.2.1. Estructura de OpenCv.	35
2.4.3. Detección de rostros.	37
2.4.3.1. Detección de rostros: Viola & Jones.	38
2.4.4. Reconocimiento facial.	41
2.4.4.1. Eigenfaces.	42
2.4.4.2. Fisherfaces.	42
2.4.4.3. Local binary pattern.	43
2.5. Aprendizaje automático	44
2.5.1. Aprendizaje automático supervisado.	45
2.6. Sistemas de bloqueo del automóvil.	46
2.6.1. Sistema de bloqueo mecánico a través de la llave de contacto.	47
2.6.2. Sistema de bloqueo a través de la bomba del combustible.	48
2.6.3. Sistema de bloqueo de encendido mediante la función anti-arranque o anti-encendido.	48
2.7. Metodologías de desarrollo.	50
2.7.1. Modelo lineal.	50
2.7.2. Modelo en cascada.	51
2.7.3. Modelo en v.	52
2.7.4. Modelo de riesgos o de espiral.	52
Capítulo III. Desarrollo Experimental.	54
3.1. Metodología.	54
3.2. Análisis	56

3.2.1. Análisis de las características del vehículo.	56
3.2.1.1. Dimensiones generales del vehículo.	56
3.2.1.2. Sistema de ventilación y acondicionador de aire.	62
3.3. Requerimientos del sistema	65
3.3.1. Requerimientos iniciales del sistema.	66
3.3.2. Requerimiento de arquitectura.	69
3.3.3. Requerimientos del stakeholder.	72
3.3.4. Selección de hardware y software.	74
3.3.4.1. Hardware.	74
3.3.4.2. Software.	82
3.4. Diseño del sistema.....	83
3.4.1. Diagrama de bloques general del sistema.	84
3.4.1.1. Diagrama de bloques de la primera etapa.	86
3.4.1.2. Diagrama de bloques de la segunda etapa.....	88
3.4.2. Diagrama de conexión del sistema.....	89
3.4.2.1. Diagrama de Pines y Puertos - Raspberry Pi 3.	89
3.4.2.2. Diagrama de Pines y Puertos – Arduino UNO R3.....	90
3.4.2.3. Diagrama de Pines – Módulo Sensor MQ-3.	91
3.4.2.4. Diagrama de Pines – Módulo Sensor MLX90621.	92
3.4.3. Diagrama de circuitual del sistema.	93
3.4.4. Diagrama de flujo del sistema.....	95

3.4.5. Desarrollo de software (programación).	104
3.4.5.1. Lectura del Sensor MQ-3.	104
3.4.5.2. Lectura del Sensor MLX90621.	106
3.4.5.3. Recibir Datos de la placa Arduino.	108
3.4.5.4. Habilidad de la cámara.	109
3.4.5.5. Adquisición de imágenes.	110
3.4.5.6. Preprocesamiento de imágenes.	112
3.4.5.7. Detección de Rostro.	113
3.4.5.8. Detección de Ojos.	114
3.4.5.9. Extraer Ojos.	115
3.4.5.10. Detectar la pupila del Ojo.	116
3.4.5.11. Alineación de la salida del sensor de Temperatura MLX90621 con la imagen óptica.	118
3.4.6. Cronograma de Recolección de Información.	124
3.4.7. Construcción del modelo de Aprendizaje Automático Supervisado.	127
3.4.7.1. Recolectar los datos.	127
3.4.7.2. Pre procesar los datos.	128
3.4.7.3. Entrenar el algoritmo.	132
3.4.8. Mensajes de Alerta Visuales y Auditivos.	150
3.4.8.1. Mensajes Visuales.	150
3.4.8.2. Mensajes Auditivos.	152
3.4.9. Reconocimiento facial del Conductor.	156

3.4.9.1. Aprender rostro del conductor.	156
3.4.9.2. Reconocimiento facial.....	157
3.4.10. Desarrollo de Hardware.	159
3.4.10.1. Conexión eléctrica para alimentación de la placa Raspberry Pi 3.	159
3.4.10.2. Alineación del sensor de temperatura MLX90621 con la cámara para que apunten al mismo objetivo.	162
3.4.10.3. Montaje del sensor de gas y humedad.....	163
3.4.10.4. Montaje del botón para iniciar el sistema.	164
3.4.10.5. Conexión eléctrica para activar y desactivar la bomba de combustible para controlar el encendido del vehículo.	165
3.4.10.6. Activación del botón de pánico del ECU 911.....	168
3.4.10.7. Implementación del Detector de Alcoholemia en el vehículo.	170
Capítulo IV. Pruebas de funcionamiento y resultados	178
4.1. Análisis de resultados obtenidos.	178
4.1.1. Encuesta.	179
4.1.1.1. Identificación de la población.	180
4.1.1.2. Análisis y conclusión de la encuesta.	183
Capítulo V. Análisis Económico.....	187
5.1. Presupuesto del Hardware.....	187
5.2. Presupuesto del Software	188
5.3. Análisis Costo Beneficio.....	189
5.3.1. Análisis de costos.	189

5.3.2. Análisis de los Beneficios.	191
5.3.2.1. Beneficios Sociales.	191
5.3.2.2. Beneficios monetarios.	192
5.3.3. Análisis de la relación Beneficio-Costo.	194
5.3.4. Conclusión del análisis Costo - Beneficio.	195
Capítulo VI. Conclusiones y Recomendaciones	197
6.1. Conclusiones	197
6.2. Recomendaciones.....	198
Bibliografía	200
Glosario.....	209

Índice de Figuras

Figura 1. Número de Siniestros de Tránsitos entre los años 2010 y 2016.	12
Figura 2. Porcentaje que representa conducir bajo la influencia de alcohol sustancias estupefacientes o psicotrópicas y/o medicamentos con relación a todos los accidentes ocasionados por las demás causas probables entre los años 2015 y 2017.	13
Figura 3. Alteración de las capacidades en la conducción a diferentes grados de alcoholemia.	16
Figura 4. Concentración de alcohol en la sangre de acuerdo con el peso, sexo y número de tragos ingerido al transcurrir una y dos horas.	20
Figura 5. Representación evolutiva de la concentración de alcohol en la sangre con relación a la concentración de alcohol en la orina.	21
Figura 6: Relación entre el aire espirado y el contenido de alcohol en la sangre con el transcurso del tiempo.	23
Figura 7. Dispositivo desechable Breath Scan para realizar una prueba de alcohol.	24
Figura 8. Dispositivo de bloqueo de vehículos Dräger Interlock XT.	25
Figura 9. Puntos para determinar los cambios de temperatura en el rostro.	28
Figura 10. Las imágenes térmicas del ojo izquierdo representan a personas sobrias, mientras que las del lado derecho corresponden a personas en estado etílico.	29
Figura 11. Diagrama de bloques con las etapas de Visión Artificial.	33
Figura 12. Características principales que comprueba el detector Viola & Jones.	39
Figura 13. Métodos de Reconocimiento Facial.	41
Figura 14. Creación de vector de algoritmo LBP.	43
Figura 15. Reconocimiento facial con los patrones binarios locales.	44

Figura 16. Etapas del Modelo Lineal.	50
Figura 17. Etapas del Modelo en Cascada.	51
Figura 18. Etapas del Modelo en V.....	52
Figura 19. Etapas del Modelo Espiral.	53
Figura 20. Modelo en V para el desarrollo de Proyectos.	55
Figura 21. Vista Frontal del vehículo.....	57
Figura 22. Vista lateral del vehículo.	58
Figura 23. Vista Interior del vehículo.	59
Figura 24. Equipamiento del tablero del vehículo.	60
Figura 25. Difusores de salidas de aire en el vehículo.....	62
Figura 26. Diagrama de Bloques General.	85
Figura 27. Diagrama de Bloques Primera Etapa.	86
Figura 28. Diagrama de Bloques Segunda Etapa.....	88
Figura 29. Pines y Puertos del Sistema Embebido Raspberry Pi 3.	90
Figura 30. Distribución de Pines Arduino UNO R3.	91
Figura 31. Distribución de Pines Modulo MQ-3.	92
Figura 32. Distribución de Pines Modulo MLX90621.	92
Figura 33. Diagrama Circuital Fuente de Alimentación.	93
Figura 34. Diagrama Circuital del Detector de Alcholemia.	94
Figura 35. Características de sensibilidad del sensor MQ-3.	96
Figura 36. Gráfico de Dispersión de la curva de características de Sensibilidad del sensor MQ-3.....	97

Figura 37. Línea de tendencia potencial y ecuación del gráfico resultante de la curva de características de Sensibilidad del sensor MQ-3.	98
Figura 38. Diagrama de Flujo del Sistema.....	103
Figura 39. Calculo de la Concentración de alcohol en el IDE de Arduino.....	104
Figura 40. Respuesta del Sensor MQ-3 ante la presencia y ausencia de Alcohol.	105
Figura 41. Librerías para el sensor MLX90621.	106
Figura 42. Temperatura Ambiente del sensor MLX90621.	107
Figura 43. Temperatura Objeto del sensor MLX90621.	108
Figura 44. Definir las librerías necesarias para Recibir los datos de la Placa Arduino.	109
Figura 45. Librerías para Recibir los datos de la Placa Arduino.	109
Figura 46. Menú para la habilitación del Módulo de la cámara en el sistema embebido.	110
Figura 47. Comandos empleados para adquirir una imagen con el módulo PiCamera.	111
Figura 48. Imagen en RGB.	111
Figura 49. Conversión de una imagen RGB a una imagen a escala de grises.	112
Figura 50. Imagen RBG aplicado escala de grises y ecualización del histograma.	113
Figura 51. Prueba de Detección de Rostro.....	114
Figura 52. Prueba de Detección de Rostro y Ojos.	115
Figura 53. Prueba de Almacenamiento de Ojos.....	116
Figura 54. Función Houghcircles.....	117
Figura 55. Detección de la Pupila	118
Figura 56. Módulos empleados en el script para medir la temperatura facial del conductor.	118

Figura 57. Tamaño y cadena que se reciben de la placa Arduino uno R3 de la Temperatura Objeto.....	120
Figura 58. Matriz 16x4 con los valores de la Temperatura Objeto.....	120
Figura 59. Rangos de Temperatura Facial.	121
Figura 60. Asignar el color a cada valor de Temperatura.	121
Figura 61. Dibujar rectángulos sobre la imagen capturada por la cámara con los valores de Temperatura del sensor MLX90621.	122
Figura 62. Vector con las Temperaturas faciales del conductor y valor de la temperatura máxima y mínima.	123
Figura 63. Alineación del sensor de temperatura MLX90621 con la imagen capturada por el módulo Pi Camera.....	124
Figura 64. Visualización del archivo con los datos almacenados en la recolección de datos.	128
Figura 65. Matrices con los nombres de los atributos y los atributos del conjunto de datos.	129
Figura 66. Vector con los nombres de las clases y las clases del conjunto de datos.	130
Figura 67. Matrices con los nombres de los atributos y atributos seleccionados para construir el algoritmo.	131
Figura 68. Atributos y clases del conjunto de entrenamiento y el conjunto de prueba.....	132
Figura 69. Representación de la distancia euclidiana en un plano en 3 dimensiones.	136
Figura 70. Exactitud del algoritmo de aprendizaje automático supervisado mediante el clasificador KNN cuando la base de datos es 360 y 435 datos.....	138
Figura 71. Calculo de Sobreajuste en el algoritmo de aprendizaje basado en KNN.	139
Figura 72. Predicción de la clase a la que pertenecen los atributos dados en el vector.	139

Figura 73. Nivel de Alcohol como atributo de mayor Importancia para seleccionar a que clase pertenecen.	142
Figura 74. Radio de la pupila y temperatura máxima como siguientes atributos con mayor Importancia para seleccionar a que clase pertenecen.	142
Figura 75. Exactitud del algoritmo de aprendizaje automático supervisado mediante el clasificador de árbol de decisión con 360 y 435 datos.....	143
Figura 76. Árbol de decisión con 390 datos en el conjunto de datos mediante el cual el algoritmo clasifica si el conductor esta sobrio o ebrio.	144
Figura 77. Árbol de decisión con 425 datos en el conjunto de datos mediante el cual el algoritmo clasifica si el conductor esta sobrio o ebrio.	145
Figura 78. Predicción de la clase a la que pertenecen los atributos con 390 datos.....	146
Figura 79. Predicción de la clase a la que pertenecen los atributos con 435 datos.....	146
Figura 80. Matrices con los nombres de los atributos y atributos seleccionados para construir el algoritmo.	147
Figura 81. Característica más importante del conjunto de datos al elegir como atributos la temperatura fácil máxima y el radio de la pupila.....	148
Figura 82. Exactitud del algoritmo de aprendizaje automático supervisado basado en el árbol de decisión usando como atributos solo la temperatura facial máxima y el radio de la pupila.	148
Figura 83. Exactitud del algoritmo de aprendizaje automático supervisado basado en el árbol de decisión usando como atributos solo la temperatura facial máxima y el radio de la pupila.	149

Figura 84. Predicción de un nuevo dato usando como atributos solo la temperatura facial máxima y el radio de la pupila.....	149
Figura 85. Mensajes de alerta visuales para el Conductor del vehículo.	151
Figura 86. Mensajes de alerta auditivos para el Conductor del vehículo.....	152
Figura 87. Rostros del conductor para el reconocimiento facial.....	156
Figura 88. Verificación del conductor designado a través del Reconocimiento facial.	158
Figura 89. Conductor detectado como desconocido a través del Reconocimiento Facial.	158
Figura 90. Empalme de los terminales positivo y tierra del Encendedor de cigarrillos.....	160
Figura 91. Cableado eléctrico.	160
Figura 92. Comprobación del voltaje.....	161
Figura 93. Conexión eléctrica para el sistema.	161
Figura 94. Montaje del sensor de temperatura, la cámara y la luz infrarroja.....	162
Figura 95. Montaje del sensor de Alcohol MQ3 y el sensor de Humedad DHT11.	164
Figura 96. Montaje del botón de activación del sistema.....	165
Figura 97. Circuito para Activar la Bomba de Combustible.....	166
Figura 98. Empalme de los terminales de la cámara de combustible.	166
Figura 99. Conexión del módulo relé y la bomba de combustible.....	167
Figura 100. Modulo Relé que se implementó en el vehículo.....	168
Figura 101. Diagrama de Conexión para el Botón de Pánico.....	169
Figura 102. Conexión del módulo relé al botón de pánico implementado por el Servicio Integrado de Seguridad ECU 911.	170
Figura 103. Cableado del Sistema.....	171

Figura 104. Alto ancho y profundidad para instalar la cámara y el sensor de temperatura. ..	172
Figura 105. Ubicación de la cámara y el sensor de temperatura en el interior del vehículo..	173
Figura 106. Ubicación de la pantalla LCD en el interior del vehículo.	174
Figura 107. diagrama de bloques de cómo está conectado el sistema en el interior del vehículo.	175
Figura 108. Elementos que intervienen en el Detector de alcoholemia.	176
Figura 109. Prueba de funcionamiento del Detector de Alcoholemia.	179
Figura 110. Conductor del grupo C de la Cooperativa de Taxis Atahualpa de la ciudad de Tulcán respondiendo la encuesta.	182
Figura 111. Primera pregunta de la encuesta.	330
Figura 112. Segunda pregunta de la encuesta.	331
Figura 113. Tercera pregunta de la encuesta.....	332
Figura 114. Cuarta pregunta de la encuesta.	333
Figura 115. Quinta pregunta de la encuesta.	334
Figura 116. Sexta pregunta de la encuesta.	335
Figura 117. Séptima pregunta de la encuesta.....	336
Figura 118. Justificación séptima pregunta de la encuesta en la que los usuarios están de acuerdo en bloquear el encendido del vehículo.	336
Figura 119. Justificación séptima pregunta de la encuesta con la opción de que están parcialmente de acuerdo.	337
Figura 120. Octava pregunta de la encuesta,.....	338

Figura 121. Justificación Octava pregunta de la encuesta por la que están de acuerdo con la activación del botón de pánico..... 339

Figura 122. Novena pregunta de la encuesta, 340

Figura 123. Justificación Novena pregunta de la encuesta con las razones porque están de acuerdo..... 340

Índice de Tablas

Tabla 1. Causas Probables a Nivel Nacional de Accidentes de Tránsito.....	10
Tabla 2. Sanciones por Conducir en estado de Ebriedad	17
Tabla 3. Incremento de temperatura en las regiones de nariz y frente del rostro en una persona ebria.....	28
Tabla 4. Valores pupilares en la sesión sin alcohol (pre-test) y en la sesión final (post-test)..	30
Tabla 5. Aplicaciones de la visión artificial.....	32
Tabla 6. Ventajas y desventajas de los sistemas de bloqueo del vehículo.....	49
Tabla 7. Dimensiones de la vista frontal del vehículo.	57
Tabla 8. Dimensiones de la vista lateral del vehículo.....	58
Tabla 9. Dimensiones de la vista Interior del vehículo.....	59
Tabla 10. Equipamiento del vehículo.....	61
Tabla 11. Dirección del aire para los difusores ajustables.....	63
Tabla 12. Distribución del flujo de aire para diferentes posiciones.....	63
Tabla 13. Definición de Acrónimos.....	65
Tabla 14. Prioridad de los Requerimientos del sistema.....	66
Tabla 15. Requerimientos Iniciales del Sistema.....	68
Tabla 16. Requerimientos de Arquitectura.....	70
Tabla 17. Lista de Stakeholders del sistema.....	72
Tabla 18. Requerimientos del Stakeholder.....	73
Tabla 19. Elección del sistema embebido.....	75

Tabla 20. Especificaciones técnicas de la placa Raspberry Pi 3.	76
Tabla 21. Elección de cámara.	77
Tabla 22. Especificaciones técnicas Raspberry Pi camera module v2 (Pi Noir).....	78
Tabla 23. Elección del sensor de gas.	79
Tabla 24. Especificaciones técnicas del Sensor MQ-3).	80
Tabla 25. Elección del sensor de temperatura.....	81
Tabla 26. Especificaciones técnicas del Sensor MLX90621.	82
Tabla 27. Elección del Software.	83
Tabla 28. Tabla de valores de la curva de las características de sensibilidad del sensor MQ3.	96
Tabla 29. Tabla de las constantes almacenadas en la EEPROM.	101
Tabla 30. Cronograma de la recolección de Información.	125
Tabla 31. Mensajes de Audio para el sistema.	153
Tabla 32. Parámetros del sensor DHT11.	163
Tabla 33. Partes del Detector de alcoholemia.....	177
Tabla 34. Presupuesto de componentes electrónicos.	187
Tabla 35. Presupuesto de materiales para la implementación del Detector de Alcoholemia en el vehículo.....	188
Tabla 36. Presupuesto de Software.	189
Tabla 37. Costos del proyecto	190
Tabla 38. Beneficios económicos	193
Tabla 39. Beneficio monetario	193

Índice de Ecuaciones

Ecuación 1. Ecuación de la Imagen Integral.....	39
Ecuación 2. Ecuación de la concentración de alcohol en función de la resistencia eléctrica. .	98
Ecuación 3. Ecuación para obtener el valor aproximado de R_o	99
Ecuación 4. Ecuación de la Temperatura Ambiente del Sensor.	100
Ecuación 5. Ecuación de la Temperatura vista del Sensor.....	101
Ecuación 6. Ecuación para obtener el tamaño de la cadena que contiene las temperaturas de objeto de la matriz 16×4	119
Ecuación 7. Distancia Euclidiana.....	134
Ecuación 8. Distancia de Manhattan.....	134
Ecuación 9. Distancia de Chebychev.	134
Ecuación 10. Distancia del coseno.....	134
Ecuación 11. Distancia de Mahalanobis.	135
Ecuación 12. Distancia usando la función delta.....	135
Ecuación 13. Normalización de los datos de Entrenamiento.	136
Ecuación 14. Función del vector que contiene los atributos del conjunto de datos.	140
Ecuación 15. Función del vector que contiene las clases del conjunto de datos.	140
Ecuación 16. Modelo matemático del algoritmo de aprendizaje automático basado en el árbol de decisión.	141
Ecuación 17. Medidas comunes de impureza son Gini.....	141
Ecuación 18. Clasificación errónea del modelo árbol de decisión.....	141
Ecuación 19. Cálculo de muestra, propuesto por Fisher y Navarro.	180

Índice de los Anexos

Anexo 1. Programación script IDE-Arduino para la lectura de los sensores y activación del módulo Relé.....	214
Anexo 2. Programación script para lectura del nivel de alcohol de la placa Arduino	224
Anexo 3. Programación script para lectura de la temperatura ambiente de la placa Arduino	225
Anexo 4. Programación script para lectura de la temperatura facial de la placa Arduino	226
Anexo 5. Programación script para adquirir imagen en formato RGB.....	235
Anexo 6. Programación script para preprocesamiento de imágenes.....	236
Anexo 7. Programación script para detectar rostro	237
Anexo 8. Programación script para detectar ojos.....	238
Anexo 9. Programación script para diámetro pupilas	240
Anexo 10. Programación script para colores de la temperatura facial.....	247
Anexo 11. Programación script para posicionesMatriz de la temperatura facial.....	253
Anexo 12. Datos de Entrenamiento	257
Anexo 13. Imágenes diámetro de pupilas de los datos de entrenamiento.....	264
Anexo 14. Imágenes temperatura facial de los datos de entrenamiento	277
Anexo 15. Programación script para iniciar el sistema.....	288
Anexo 16. Programación preprocesamiento de los datos de entrenamiento.....	296
Anexo 17. Programación clasificador k-vecinos más cercanos	298
Anexo 18. Programación clasificador árbol de decisión.....	301
Anexo 19. Programación del script de Captura del Conductor.....	304

Anexo 20. Programación del script de Reconocimiento del Conductor	307
Anexo 21. Programación del script recibir dato de Humedad de la placa Arduino.....	313
Anexo 22. Oficio con la Autorización para activar el botón de pánico	314
Anexo 23. Autorización para realizar las pruebas de funcionamiento del sistema.....	315
Anexo 24. Formato de encuesta aplicada a los señores conductores.....	316
Anexo 25: Señores conductores del grupo C de la Cooperativa de taxis Atahualpa de la ciudad de Tulcán que realizaron la encuesta	319
Anexo 26: Tabulación de resultados de encuestas aplicadas los 58 señores conductores del grupo C de la Cooperativa de taxis Atahualpa de la ciudad de Tulcán	330
Anexo 27. Certificado de las pruebas de funcionamiento del sistema.....	341

Resumen

En el presente trabajo de titulación se realizará un detector de alcoholemia para conductores que analiza variables faciales y ambientales del automóvil mediante el aprendizaje automático supervisado, con el cual se pretende obtener una herramienta diseñada para la reducción de accidentes de tránsito a causa de conductores que se encuentran en estado etílico, el proyecto se encuentra desarrollado en base a la metodología del modelo en V con el cual se establecen los parámetros y funciones que se deben cumplir en cada una de sus fases que dará como resultado un modelado eficiente del sistema. Dicho modelamiento se encuentra desarrollado en dos etapas, la primera tiene la finalidad recolectar la temperatura facial, diámetro de la pupila y nivel de alcohol en el aliento del conductor para crear un conjunto de datos que serán utilizados para el entrenamiento del algoritmo de aprendizaje automático supervisado basado en la clasificación del árbol de decisión, la segunda etapa del sistema verifica el estado del conductor haciendo uso del algoritmo mencionado cuyo resultado determinará si el vehículo se enciende o no; el sistema es sometido a pruebas de funcionabilidad con diferentes sujetos en diferentes estados etílicos y sobrios cuyos resultados demuestran la factibilidad del sistema y a su vez mediante un análisis de costo beneficio se demuestra la rentabilidad del proyecto.

Abstract

In the present degree work will be developed a breathalyzer detector for drivers, which analyzes facial variables and environmental variables of the automobile through supervised machine learning, with which it is intended to obtain a tool designed for the reduction of traffic accidents due to of drunk drivers, the project is developed in base on the methodology of the V model with which the parameters and functions that must be met in each of its phases are established that will result in an efficient modeling of the system. Before mentioned modeling is developed in two stages, the first one has the purpose of collecting the facial temperature, diameter of the pupil and level of alcohol in the driver's breath, to create a set of data that will be used for the training of the supervised machine learning algorithm based on the Decision Tree Classification, the second stage of the system verifies the driver's status using the mentioned algorithm whose result will determine if the vehicle turns on or not; the system is submitted to functionality tests with different subjects in different ethyl states and sober whose results demonstrate the feasibility of the system and in turn, through a cost-benefit analysis, the cost effectiveness of the project is demonstrated.

Victor Rodríguez
1715496129
R. B. B.



Presentación

El presente trabajo de titulación tiene como objetivo desarrollar un detector de alcoholemia para conductores que analiza variables faciales y ambientales del automóvil mediante el aprendizaje automático supervisado para la reducción de accidentes de tránsito

En el primer capítulo se describen las bases para el desarrollo del trabajo de titulación, en el cual se plantea el tema juntamente con los objetivos a cumplir en el transcurso del proyecto, el problema que originó la investigación, el alcance que tendrá y su justificación, con el propósito de manifestar la importancia de la implementación de un Detector de Alcoholemia para la reducción de accidentes de tránsito.

En el segundo capítulo se presenta la recopilación bibliográfica de la investigación realizada para el desarrollo del proyecto, se describen los efectos del consumo de alcohol y los registros de accidentes de tránsito en el Ecuador, así como el estudio de los sensores utilizados en la identificación de conductores en estado etílico. También se abordan los principios de la Visión Artificial, con los principales algoritmos de reconocimiento facial y la aplicación de OpenCV. Finalmente se trata los conceptos de aprendizaje automático supervisado, sus enfoques y algoritmos.

En el tercer capítulo se indica el desarrollo experimental del proyecto en base a la metodología de desarrollo del Modelo en V, con la finalidad de cumplir con las distintas herramientas de desarrollo de cada etapa que especifica el modelo.

En el cuarto capítulo se indica las pruebas de funcionamiento del software y hardware del detector de alcoholemia, se desarrollan los resultados obtenidos en el estudio y cuáles fueron las novedades y problemas en su realización para la recolección de resultados se procederá a emplear técnicas para recopilación de datos con la finalidad de determinar los resultados obtenidos y la aceptación de los conductores del proyecto.

En el capítulo cinco se realiza el análisis económico del Detector de Alcoholemia que analiza variables faciales y ambientales del automóvil mediante el aprendizaje automático supervisado para la reducción de accidentes de tránsito, en el cual se considera el monto referencial del presupuesto que se invirtió en el diseño y la implementación del proyecto, y las etapas que se realizan para evaluar el Costo-Beneficio de este.

El capítulo seis finaliza el presente trabajo de titulación, en el cual se presentan las conclusiones y recomendaciones del proyecto.

Capítulo I. Antecedentes

En el primer capítulo se describen las bases para el desarrollo del presente trabajo de titulación, en el cual se plantea el tema delimitado conjuntamente con los objetivos a cumplir en el transcurso del proyecto, el problema que origino esta investigación, el alcance que tendrá y su justificación, con el propósito de manifestar la importancia de la implementación de un Detector de Alcoholemia para conductores que analice variables faciales y ambientales del automóvil mediante el aprendizaje automático supervisado para la reducción de accidentes de tránsito.

1.1. Tema

DETECTOR DE ALCOHOLEMIA PARA CONDUCTORES QUE ANALIZA VARIABLES FACIALES Y AMBIENTALES DEL AUTOMÓVIL MEDIANTE EL APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA REDUCCIÓN DE ACCIDENTES DE TRÁNSITO.

1.2. Planteamiento del problema

A pesar de que dentro del Código Orgánico Integral Penal COIP existen sanciones para la persona que conduzca en estado de ebriedad, los accidentes de tránsito en el Ecuador han aumentado significativamente en los últimos años por conducir bajo la influencia de alcohol, debido a que el conductor ebrio presenta: un falso estado de euforia, un aumento del tiempo de reacción, la reducción del campo visual, la invasión de un carril sin motivo, lo que produce

un alto número de siniestralidad en término de pasajeros muertos, heridos y pérdidas materiales, ocasionando en la ciudadanía incertidumbre y preocupación a causa de personas imprudentes que conducen en estado de ebriedad. Datos estadísticos adquiridos de la página oficial de la Agencia Nacional de Tránsito del Ecuador “indican que en el periodo enero 2015-marzo 2016 se suscitaron 166 fallecidos y 2.057 lesionados, provocados por conductores en estado etílico.” (AGENCIA NACIONAL DE TRÁNSITO (ANT), 2015).

En la actualidad en el mercado existe un gran número de métodos para la detección del nivel de alcohol en un conductor, uno de las técnicas más populares es el “analizador de aliento” (SIDALC, BD NORTEC, 2013); sin embargo, este no es infalible, ya que para determinar el efecto que tiene el etanol ingerido sobre la capacidad de conducir el conductor debe exhalar a través del analizador para tomar la muestra del aliento y medir la concentración de etanol en el aire exhalado, el inconveniente más relevante en dicha prueba es la dificultad que tiene para determinar que la persona que realiza es en realidad quien va a conducir el vehículo y no un pasajero que se encuentre en estado de sobriedad.

Otra técnica empleada en Países desarrollados es el uso de tiras transductoras sobre la calzada para delimitar carriles, no obstante, este sistema es muy costoso y existe la posibilidad de daños en los transductores cuando hay mucha fricción con las ruedas debido al exceso de velocidad (Aravind, Karthick & Sakthivel).

También se proponen enfoques para la “discriminación de una persona con altos niveles de alcohol usando imágenes infrarrojas térmicas” (Koukiou, G., & Anasassopoulos, V., 2013) con el fin de vigilar los cambios de la temperatura en las regiones cerca de la boca, nariz y frente, además de visualizar el aumento de actividad en los vasos sanguíneos faciales, debido al incremento de la temperatura en personas que consumen alcohol, el inconveniente que presenta la implementación de este sistema es su elevado costo de instalación de las cámaras térmicas

para distinguir que una persona ha ingerido bebidas alcohólicas por medio de valores radiométricos en su rostro.

Los problemas detectados en los alcoholímetros “existentes en el país radica en la inexistencia en el mercado de sistemas electrónicos para automóviles” (IMPORVALIA S.A., 2011) que permitan analizar diferentes parámetros del conductor como: su temperatura, el comportamiento del ser humano, aliento, entre otros para determinar el estado en el que se encuentra y de esta manera dar el resultado analizando diferentes patrones y que no se limite a una sola muestra, además de que permita identificar la identidad del conductor sin la necesidad de la presencia de policías o agentes civiles de tránsito para asegurar que el sistema no pueda ser violado fácilmente por suplantación de persona.

1.3. Objetivos

1.3.1. Objetivo General.

Desarrollar un detector de alcoholemia para conductores de vehículos que analice el comportamiento facial, cambios de temperatura y la concentración de etanol en el aire del automóvil, mediante el aprendizaje automático supervisado, para disminuir los accidentes de tránsito.

1.3.2. Objetivos Específicos.

- Analizar la literatura disponible sobre métodos para la detección de alcoholemia en conductores y el comportamiento del cuerpo humano al ingerir bebidas alcohólicas.

- Diseñar el prototipo de un sistema de detección de alcoholemia en vehículos para la recolección de los datos de la temperatura, aliento y el comportamiento humano en conductores que han ingerido bebidas alcohólicas.
- Desarrollar el algoritmo de aprendizaje automático para el detector de alcoholemia, mediante el procesamiento digital de señales con las variables necesarias para identificar los cambios de temperatura y el estado del conductor.
- Realizar pruebas del detector de alcoholemia con un grupo de personas para verificar el comportamiento del sistema.
- Implementar las correcciones del detector de alcoholemia para adquirir los resultados esperados.

1.4. Alcance

Con la realización de este proyecto, se desarrollará un detector de alcoholemia para vehículos que analice el comportamiento de personas en estado etílico mediante el aprendizaje automático supervisado, partiendo de la revisión literaria de métodos de identificación de conductores en estado etílico y el comportamiento del cuerpo humano al ingerir bebidas alcohólicas, adquiridos de las bases de datos de información alojadas en las bibliotecas virtuales, libros y otras publicaciones.

Posteriormente al comprender las distintas técnicas de detección y el conocimiento de las tecnologías empleadas para determinar si una persona se encuentra en estado de embriaguez, se continuará con el diseño del prototipo de sistema de detección de alcoholemia para vehículos, considerando el cambio de temperatura del rostro con un sensor infrarrojo térmico, la variación en el comportamiento facial a través de librerías de software libre de visión artificial y la concentración de etanol en el aire empleando un sensor de gas, además de una unidad de control para el procesamiento de la información y software libre para la tabulación de datos.

A continuación, se realizará el desarrollo del algoritmo de aprendizaje automático supervisado del detector de alcoholemia, para ello se determinarán las variables a censar por el sistema, necesarias para realizar el análisis de los cambios de temperatura y comportamiento facial del ser humano cuando una persona se encuentra en estado etílico.

Una vez terminada la etapa de desarrollo del detector de alcoholemia se procede a realizar las pruebas de funcionamiento, con lo que se determinará si es necesario realizar cambios en el captador, en la unidad de control o en el algoritmo de detección para mejorar la efectividad del sistema e identificar si el conductor efectivamente se encuentra en estado etílico.

Si en el proceso de pruebas se ha determinado que existen falencias en el detector de alcoholemia se procederá a realizar las correcciones necesarias, con el objetivo de determinar el estado del conductor y emitir una señal de alerta.

Una vez detectado que el conductor se encuentra en estado etílico se procederá a emitir una señal de alerta, en las pruebas de funcionamiento del detector de alcoholemia el usuario

visualizará una alerta que indique el estado en que se encuentra. Para la implementación en vehículos reales se deberá implementar dos señales de alerta. Una señal que impida el encendido del vehículo y una señal de accionamiento automático del botón de pánico, instalado en el vehículo por el Servicio Integrado de seguridad ECU911.

Posteriormente se realizará el desarrollo del presupuesto, el análisis de costos y el análisis de costo-beneficio haciendo referencia a que en realidad se puede implementar en vehículos reales.

Al finalizar el proyecto se procederá a realizar un análisis de los resultados obtenidos para determinar el comportamiento del detector de alcoholemia con diferentes escenarios del vehículo y la redacción de las respectivas conclusiones y recomendaciones.

1.5. Justificación

En los últimos años el número de accidentes de tránsito ocurridos en el Ecuador han aumentado de un 6,31% en el año 2014 (Agencia Nacional de Tránsito, 2014) a un 6,72% en el año 2015 (Agencia Nacional de Tránsito, 2015) a causa de conductores irresponsables que no miden el riesgo que conlleva manejar en estado de ebriedad, como son las pérdidas materiales, personas heridas, la muerte del conductor o los tripulantes, además de arriesgar la vida de terceras personas. Con la implementación de un detector de alcoholemia en vehículos que analice el comportamiento humano mediante el aprendizaje automático se contribuirá a la disminución de los accidentes automovilísticos, en donde el conductor es el responsable de la vida de todas las personas que se encuentran dentro del automotor, aportará a reducir la mortalidad por accidentes de tránsito a 13 muertes por cada 100 000 habitantes que es la “meta

6.2 propuesta en el Plan Nacional del Buen Vivir 2013-2017” (Plan Nacional del Buen Vivir, 2013-2017).

El desarrollo de este proyecto permite el uso de nuevas herramientas y tecnologías para mejorar los sistemas de vigilancia e identificación de los niveles de estado étílico existentes en vehículos de gama media, posibilita el aumento de metodologías de investigación para la discriminación de una persona con niveles de alcohol, además de encontrar una solución que permita analizar diferentes variables para determinar si el conductor se encuentra en estado étílico e identificar que la persona examinada es el conductor y de esta manera mejorar la seguridad en las vías.

Además de fomentar el campo de investigación en las áreas de electrónica aplicada a los automóviles que mejoren la comodidad y seguridad de los seres humanos, a través del diseño de dispositivos de seguridad activos y pasivos con aprendizaje automático que garanticen la seguridad del conductor y los pasajeros. La Universidad Técnica del Norte cuenta con la Carrera de Ingeniería Electrónica y Redes de Comunicación, con la misión de formar ingenieros competentes, capaces de ejecutar procesos tecnológicos, de conocimientos científicos y de innovación en el sector de la electrónica; la realización de este proyecto contribuye a la investigación de nuevos proyectos como el detector de alcoholemia para conductores de vehículos, que permiten plasmar los conocimientos adquiridos a lo largo la carrera. Gracias a la inclusión de sistemas embebidos, la electrónica, módulos de control y adquisición de datos, procesamiento digital de señales y técnicas de visión artificial para identificar los cambios del comportamiento del cuerpo humano al ingerir bebidas alcohólicas e identificar patrones que facilitan el desarrollo de sistemas capaces de detectar la presencia de alcohol en el conductor del vehículo.

1.6. Contexto

En la Universidad Técnica del Norte se ha realizado el trabajo de titulación: “Implementación de un dispositivo de seguridad para el bloqueo de encendido de un vehículo mediante Alcocheck” (Cupuerán E, 2013) que detecta el porcentaje de alcohol en el aliento a través del sensor semiconductor de gas. Antes de encender el vehículo, el conductor debe soplar en una pequeña unidad para proporcionar una muestra de aliento posteriormente es analizada, si el dispositivo detecta una cantidad inadecuada de alcohol, el motor no se encenderá y se enviará un mensaje de texto a tres diferentes números de celular; el aporte de la investigación y futura implementación de un detector de alcoholemia para conductores de vehículos que analiza el comportamiento humano mediante el aprendizaje automático supervisado, será el estudio de la conducta del conductor en tiempo real, a través del censado de diferentes variables necesarias para realizar el análisis de los cambios de temperatura y comportamiento del cuerpo humano cuando una persona se encuentra en estado etílico, encontrar un algoritmo de aprendizaje automático que permita la implementación de detectores de alcoholemia en diferentes vehículos, además una solución de detección que no necesite hacer contacto con el conductor para realizar la prueba de alcoholemia y de esta manera identificar que la persona examinada es el conductor y no un tripulante del automóvil, se realizará un análisis de los resultados obtenidos para determinar el comportamiento del detector de alcoholemia con diferentes escenarios del vehículo, en las pruebas de funcionamiento del detector de alcoholemia el usuario visualizará una alerta que indique el estado en que se encuentra.

Capítulo II. Revisión bibliográfica

En este capítulo se presenta la recopilación bibliográfica de la investigación realizada para el desarrollo del proyecto, se describen los datos estadísticos de accidentes de tránsito en el Ecuador y las causas probables, el registro de siniestros de tránsito a causa de conducir bajo los efectos de alcohol, niveles de concentración de alcohol, comportamiento del cuerpo humano al ingerir bebidas alcohólicas, así como el estudio de los métodos utilizados en la detección de alcoholemia en conductores. También se abordan los principios de la Visión Artificial, con los principales algoritmos de detección y reconocimiento facial y la aplicación de OpenCV. Finalmente se trata los conceptos de aprendizaje automático supervisado y las librerías de machine learning con el objetivo de crear una función capaz de predecir que el conductor se encuentra en estado etílico.

2.1. Accidentes de tránsito en el Ecuador

Durante los últimos 2 años la Agencia Nacional de Tránsito, en las estadísticas de transporte terrestre y seguridad vial ha clasificado los accidentes en siniestros, lesionados y fallecidos; dentro de los cuales existen 28 causas probables de accidentes automovilísticos a Nivel Nacional con su respectivo código, se ha registrado mayor cantidad de accidentes en las vías causados por: Conducir desatento a las condiciones de tránsito, seguido de conducir el vehículo superando los límites máximos de velocidad, no respetar las señales reglamentarias de tránsito y conducir bajo la influencia de alcohol, sustancias estupefacientes o psicotrópicas y/o medicamentos. A continuación, en la Tabla 1 se presentan las causas probables de accidentes de Tránsito.

Tabla 1. Causas Probables a Nivel Nacional de Accidentes de Tránsito

Código	Causas	2015			2016			2017		
		S	L	F	S	L	F	S	L	F
C14	Conducir desatento a las condiciones de tránsito.	4505	3031	380	6455	4488	736	5115	3877	526
C09	Conducir vehículo superando los límites máximos de velocidad	3777	3320	282	3755	3421	259	4147	3680	387
C23	No respetar las señales reglamentarias de tránsito	4897	3263	174	3750	2896	107	3967	3008	74
C11	No mantener la distancia prudencial con respecto al vehículo que le antecede	3156	1685	122	2610	1355	89	2572	1520	167
C06	Conduce bajo la influencia de alcohol, sustancias estupefacientes o psicotrópicas y/o medicamentos	2400	1712	146	2096	1472	76	2094	1443	79
C25	No ceder el derecho de vía o preferencia de paso a vehículos	2806	1767	96	2074	1215	79	1497	1073	61
C26	No ceder el derecho de vía o preferencia de paso al peatón	1888	1918	140	1941	1689	148	1794	1684	174
C19	Realizar cambio brusco o indebido de carril	2173	1505	107	1726	1349	93	2016	1827	209
C12	No guardar la distancia lateral mínima de seguridad entre vehículos	2852	1628	107	1504	598	29	1895	918	70
C03	Conducir en estado de somnolencia o malas condiciones físicas	907	376	24	776	231	21	693	185	19
C16	No transitar por las aceras o zonas de seguridad destinadas para el efecto	1218	1098	181	744	604	128	798	656	164
C22	Adelantar o rebasar a otro vehículo en movimiento en adelantar o rebasar a otro vehículo en movimiento en zonas o sitios Peligrosos tales como	893	445	19	542	270	25	227	180	18
C01	Caso fortuito o fuerza mayor	600	451	73	452	275	28	254	154	31
C15	Dejar o recoger pasajeros en lugares no permitidos	378	399	23	323	286	5	225	191	11
C27	Peatón que cruza la calzada sin respetar la señalización existente	290	296	25	307	248	27	251	218	15
C18	Conducir en sentido contrario a la vía normal de circulación	1035	878	111	258	271	38	373	459	59
C05	Falla mecánica en los sistemas y/o neumáticos	269	184	20	188	153	19	142	124	29

C10	Condiciones ambientales y/o atmosféricas	458	315	28	174	135	9	199	182	17
C21	Malas condiciones de la vía y/o configuración	146	129	17	142	126	13	63	52	7
C17	Bajarse o subirse de vehículos en movimiento sin tomar las precauciones debidas	188	180	6	122	125	9	285	290	4
C02	Presencia de agentes externos en la vía	114	102	6	99	90	10	112	85	10
C07	Transita bajo influencia de alcohol, sustancias estupefacientes o psicotrópicas y/o medicamentos	90	111	10	68	50	11	108	110	13
C08	Peso y volumen-no cumplir con las normas de seguridad necesarias al transportar cargas	74	67	5	48	35	0	33	19	4
C24	No respetar las señales manuales del agente de transito	347	243	28	46	25	2	26	16	0
C04	Daños mecánicos previsibles	69	71	6	41	41	6	46	49	4
C20	El conductor que detenga o estacione vehículos en sitios o zonas que entrañen peligro, tales como zona de seguridad, curvas, puentes, túneles, pendientes	47	24	2	26	9	0	27	11	1
C28	Dispositivo regulador de tránsito en mal estado de funcionamiento (semáforo)	9	6	0	2	1	0	8	7	0
C13	Uso de celular-conducir desatento a las condiciones de transito	120	30	0	0	0	0	0	0	0
TOTAL			35706	25234	2138	21458	1967	28967	22018	2153

Fuente: (Agencia Nacional de Tránsito, 2017)

2.1.1. Registro de Siniestros de tránsito a causa de conducir bajo los efectos de alcohol.

En los datos oficiales proporcionados por la ANT, conducir bajo la influencia de alcohol es una de las principales causas de siniestros de tránsito a nivel Nacional, por esta razón se busca alternativas que permitan mejorar el transporte terrestre y la seguridad vial para los tripulantes del vehículo y peatones que sean víctimas de un conductor en estado de ebriedad. La variación de siniestros automovilísticos a causa de conducir en estado etílico desde el año 2010, así como también el registro de personas que resultaron lesionadas y fallecidas en los 3 últimos años se muestran en el Figura 1.

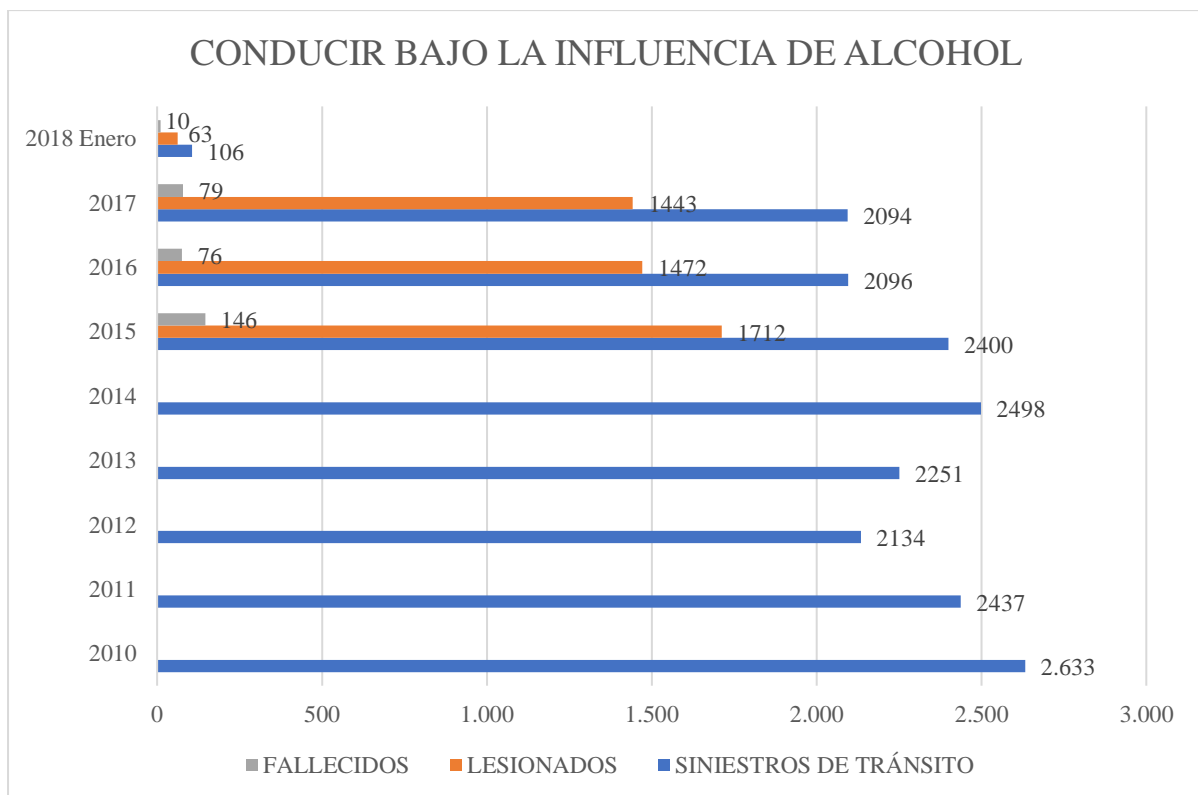


Figura 1. Número de Siniestros de Tránsitos entre los años 2010 y 2016.

Fuente: Adaptado de (Agencia Nacional de Tránsito, 2017)

Como se puede apreciar en la Figura 1 en los tres últimos años el número de siniestros, lesionados y fallecidos de tránsito a causa de conducir bajo la influencia de alcohol, sustancias estupefacientes o psicotrópicas y/o medicamentos ha reducido; sin embargo tomando en cuenta el porcentaje que representan con relación a todos los accidentes de tránsito que se han producido por todas las causas probables el porcentaje incrementa, en la Figura 2 se muestra el porcentaje que representa en los 3 últimos años.

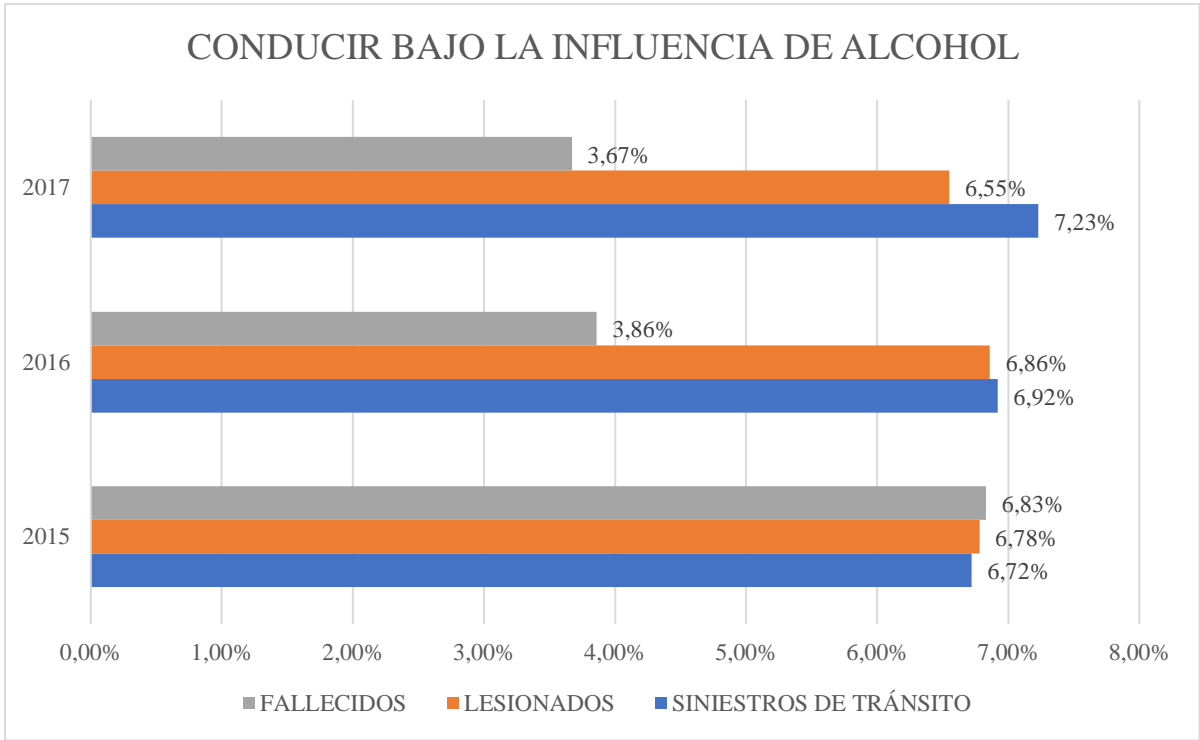


Figura 2. Porcentaje que representa conducir bajo la influencia de alcohol sustancias estupefacientes o psicotrópicas y/o medicamentos con relación a todos los accidentes ocasionados por las demás causas probables entre los años 2015 y 2017.

Fuente: Adaptado de (Agencia Nacional de Tránsito, 2017)

2.2. Comportamiento del cuerpo humano al ingerir bebidas alcohólicas

Conocer el comportamiento del cuerpo humano después de la ingesta de alcohol es el primer paso fundamental y necesario para emprender nuevos métodos que permitan verificar el estado del conductor designado, esto se debe a que el cuerpo humano muestra diversos

cambios provocados a causa de que aproximadamente solo el 10% de etanol presente en el organismo se elimina a través de los poros de la piel como sudoración, mediante los pulmones en forma de aliento y por medio de los riñones a través de la orina; sin embargo, el 90% restante es oxidado a través de un proceso químico que realiza el hígado (Loor López, 2012). La presencia del 90% de alcohol en el organismo afecta al cerebro, ya que es considerado uno de los órganos más sensible a la acción de etanol, esto se refleja apenas después de 30 segundos de haber ingerido el contenido de una copa de bebida alcohólica, iniciando con los efectos en varias de las funciones normales del cerebro que perturban la conducta verbal, corporal, psicológica e intelectual. El comportamiento de las personas que se encuentran tomando licor se ve afectado por diversos factores, incluyendo el comportamiento psicológico que se manifiesta según sea el estado de ánimo, algunos aspectos biológicos y el entorno en el cual se consume el alcohol etílico. Se puede generalizar de la siguiente manera los cambios que se originan en el comportamiento del cuerpo humano según el porcentaje de CAS¹.

- Menos de 0,03% a 0,05%, generalmente no producen cambios fácilmente reconocibles.
- Con una CAS entre “0,05 y 0,10%, se produce una sensación de relajación, sedación y /o euforia” (Arias, 2005, p.145). Debido a que el alcohol puede liberar ciertas inhibiciones y ocultar la fatiga, ocasionando que erróneamente el alcohol sea considerado como estimulante al sentir esa sensación.
- A niveles de CAS entre 0,10% y 0,20%, la mayoría de las personas en estado etílico muestran señales visibles de intoxicación: impedimento físico y mental que afecta la percepción y ejecución. El comportamiento corporal se deteriora ocasionando descoordinación muscular, la conducta psicológica se ve afectada y las reacciones a los

¹ Concentraciones de Alcohol en la Sangre

diferentes estímulos se retardan. Además de presentar variaciones en la discriminación visual y auditiva y la conducta verbal es, en ocasiones, muy confusa.

- En concentraciones mayores al 0,20%, la mayoría de las personas se ve muy “sedada”, es decir, no percibe bien lo que ocurre a su alrededor y tiene serias dificultades para mantenerse de pie y permanecer en estado alerta.
- Aun nivel de CAS de 0,40% el individuo puede quedar en estado vegetativo y a 0,60% en peligro de muerte.

2.2.1. Efectos del alcohol al conducir un vehículo.

Para evitar accidentes en las autopistas ocasionados por conductores irresponsables es necesario conocer los efectos que produce la ingesta de bebidas alcohólicas debido a que “afecta directamente el sistema nervioso central, provocando en los conductores alteración sobre la visión, la función psicomotora, cambios de activación-alerta, el comportamiento y la conducta” (Loor, 2012, p.25).

- Con relación a la conducta visual se destaca la disminución del campo visual, conocido como efecto túnel, que provoca en el individuo falta de visibilidad manifestada en la irrupción de un carril sin motivo o la realización de giros excesivamente amplios. “También se ve afectada la capacidad para seguir objetos y la recuperación de la visión después de la exposición al deslumbramiento” (Guanche, Martínez y Gutiérrez, 2007, p.7).
- Con respecto a las funciones psicomotoras el tiempo de reacción del conductor se extiende, esto se refleja principalmente cuando el conductor necesita “cambiar el pie del acelerador al freno que normalmente es de 0,75 segundos, mientras que para un

conductor en estado de embriaguez el tiempo de reacción puede llegar a ser de 2 o más segundos” (Delfino, 2012). El aumento significativo del aumento en el tiempo de reacción provoca que la distancia de detención del automóvil se prolongue, además de disminuir los reflejos cuando en la carretera se presente cualquier tipo de obstáculos y la imposibilidad de captar las señales de tránsito. También se debe mencionar que se ve afectada la coordinación bimanual y la capacidad para juzgar la velocidad y distancia.

- Con respecto a los cambios de activación y alerta el etanol hace que la fatiga muscular y sensorial sea mayor, disminuyendo considerablemente la alerta y vigilia ante algún incidente automovilístico, además de provocar que el cansancio se presente con rapidez, aumentando significativamente la tendencia al sueño y adormilamiento.
- Debido a la cantidad ingerida de alcohol se ve alterado el comportamiento y la conducta, ya que los individuos en estado etílico presentan un menor sentido de la responsabilidad y la prudencia.

En la Figura 3 se presentan las alteraciones en la conducción con diferentes grados de alcoholemia.

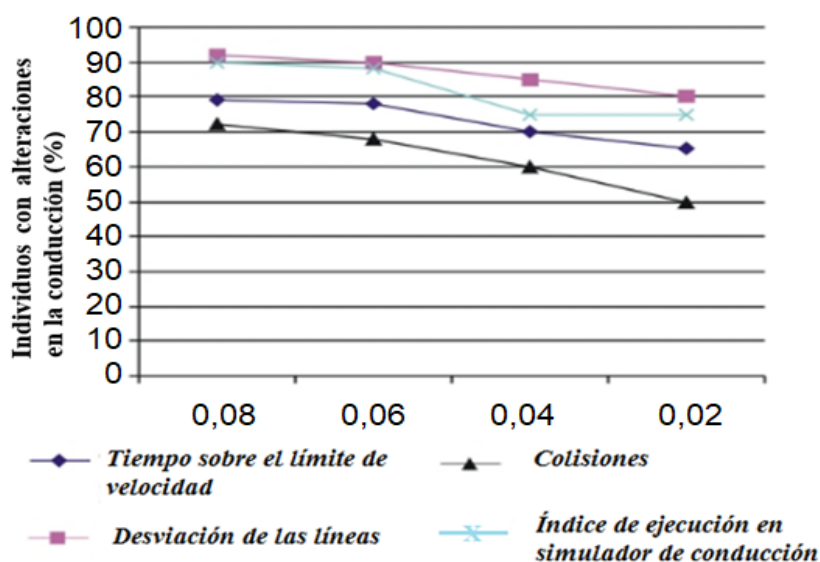


Figura 3. Alteración de las capacidades en la conducción a diferentes grados de alcoholemia.

Fuente: Adaptado de (Guanche Garcell, Martínez Quesada, & Gutiérrez García, 2007)

Existe una relación directamente proporcional debido a que cuando se presenta mayor concentración de etanol en la sangre mayor es el deterioro de las capacidades motoras y cognitivas para evitar accidentes automovilísticos.

2.2.2. Sanciones que se aplican a conductores en estado de ebriedad en el Ecuador.

Las sanciones o multas aplicadas para las personas que conduzcan en estado de ebriedad por infringir alguna contravención de tránsito en el Ecuador se establecen como detalla el Código Orgánico Integral Penal en el artículo 385, en el cual se establece que la persona que conduzca un vehículo en estado de embriaguez será sancionada de acuerdo con una escala de niveles (Zuñiga, et al,2014). A continuación, en la Tabla 2 se muestra las sanciones señaladas:

Tabla 2. Sanciones por Conducir en estado de Ebriedad

NIVEL	Nivel de alcohol por litro de sangre (gramos)	Multa en salario básico unificado del trabajador en general	Pérdida de puntos en la licencia de conducir	Privación de la libertad (Días)	Aprehensión del vehículo (horas)
1	0,3 a 0,8	1	5	5	24
2	0,8 a 1,2	2	10	15	24
3	Mayor 1,2	3	suspensión de la licencia por 60 días	30	24
Vehículos transporte público	Nivel máximo 0,1		30	90	24

Fuente: (Zuñiga, et al,2014)

2.3. Métodos para la detección de alcoholemia en conductores

Las primeras pruebas realizadas a conductores de vehículos para verificar que no se encuentren en estado etílico, se fundaban en la relación que existe entre las facultades psicológicas y físicas que posee una persona al momento de conducir y el volumen de etanol en el que se encuentra en el organismo. Las cuales radicaban en pruebas de equilibrio tomando como referencia una línea recta en la cual el conductor debía caminar, coordinación de las extremidades superiores e inferiores y la prueba de percepción espacial considerando el entorno en el cual se encuentra; estas pruebas eran realizadas por la policía de tránsito, a los individuos sospechosos de conducir bajo la influencia de bebidas alcohólicas; sin embargo, la relación existente no era cuantificable con facilidad, a menos que el examen sea realizado por personal especialista. Debido a la dificultad para calcular el porcentaje de alcohol presente en el cerebro se recurre a diferentes métodos que relacionen la presencia de etanol con la concentración en el interior de las células nerviosas (Andrade, 2009).

2.3.1. Análisis de alcohol en sangre.

Diversos estudios realizados a lo largo de la historia indican la relación que existe entre la concentración de alcohol en la sangre BAC² y la posibilidad de causar accidentes automovilísticos, por lo que se lo considerado como el método que presenta mayor exactitud cuándo se requiere medir el grado de etanol en una persona que se encuentra bajo los efectos de bebidas alcohólicas, esta prueba mide la concentración en la sangre y se estima la concentración que existe en el interior de las células. Para realizar este análisis se toma una

² Concentración de alcohol en la sangre

muestra de sangre y posteriormente llevar la muestra al laboratorio. “Los niveles de alcohol en la sangre pueden determinarse mediante: aireación, destilación, cromatografía gaseosa o análisis enzimático específico con alcohol deshidrogenasa” (Gennaro, 2003). Los resultados del análisis de la concentración de alcohol en sangre es el examen legal con mayor precisión, sin embargo, presenta inconvenientes al momento de verificar que el conductor se encuentre en estado etílico, esto se debe a que el procedimiento que se realiza es agresivo con el conductor, costoso, requiere de tiempo para obtener el resultado es decir no es inmediato, el examen solo puede ser realizado por agentes capacitados, si no existe personal adecuado se debe realizar el traslado de la muestra a un centro médico de análisis, lo que podría ocasionar una posible contaminación de la muestra durante el proceso de extracción, transporte o almacenamiento. En la Figura 4 se indica el número de bebidas en tragos, es decir una copa de cerveza o vino y los BAC correspondientes.

Cuando un conductor alcanza niveles $BAC=0,02$ mg/dl tiene una tolerancia cero, con niveles $BAC=0,05$ el riesgo de muerte por un accidente de tráfico es 6 a 17 veces mayor, mientras que cuando el nivel BAC es superior a 0,15 mg/dl tiene 244 veces más probabilidades de tener un accidente fatal, debido a su falta de reacción ante obstáculos que se le presenten en la carretera. (Ruiz, Marcías, Gómez, Rondón y Lozano, 2010, p.251)

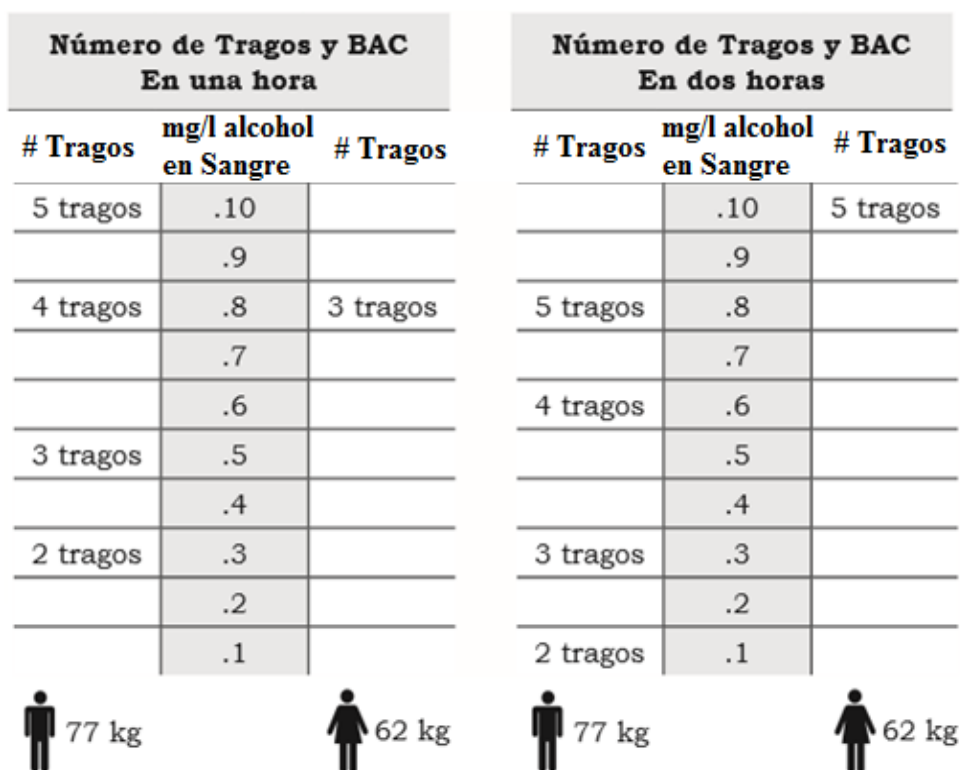


Figura 4. Concentración de alcohol en la sangre de acuerdo con el peso, sexo y número de tragos ingerido al transcurrir una y dos horas.

Fuente: Adaptado de (Ruiz, Marcías, Gómez, Rondón y Lozano, 2010, p.251)

2.3.2. Análisis de alcohol en la orina.

El resultado de una prueba de orina manifiesta que en el conductor ha ingerido alcohol, sin embargo, no especifica el estado actual en el que se encuentra, esto se debe a que el alcohol se incorpora a la sangre a través del estómago luego del transcurso de 15 minutos y para que el organismo lo metabolice y comience a aparecer el etanol en una muestra de orina debe transcurrir alrededor de 1:30 a 2 horas (Prada y Martínez, 2003). Entre las dificultades que presenta el test de orina se tiene: imposibilidad de cuantificar con exactitud el grado de alcohol ingerido por el conductor debido a que al menos un 10% de etanol ingerido se elimina sin metabolizar por orina, el resultado de la prueba se obtiene varias horas después por lo que no ofrece una imagen real del estado actual del conductor, otro inconveniente que se presenta es

que la concentración de alcohol en la orina se altera según el organismo y el tiempo en el que el cuerpo metabolice el etanol. A continuación, en la Figura 5 se puede realizar una comparación entre la concentración de etanol en sangre y la concentración de alcohol en orina, en función del tiempo; en donde si la prueba se realiza en un tiempo menor a 2 horas la posibilidad de que el resultado sea cero aumenta, mientras que una prueba de sangre indicara la presencia de 100 mg/dL si la muestra es adquirida en 2 horas (Cedilo,2010).

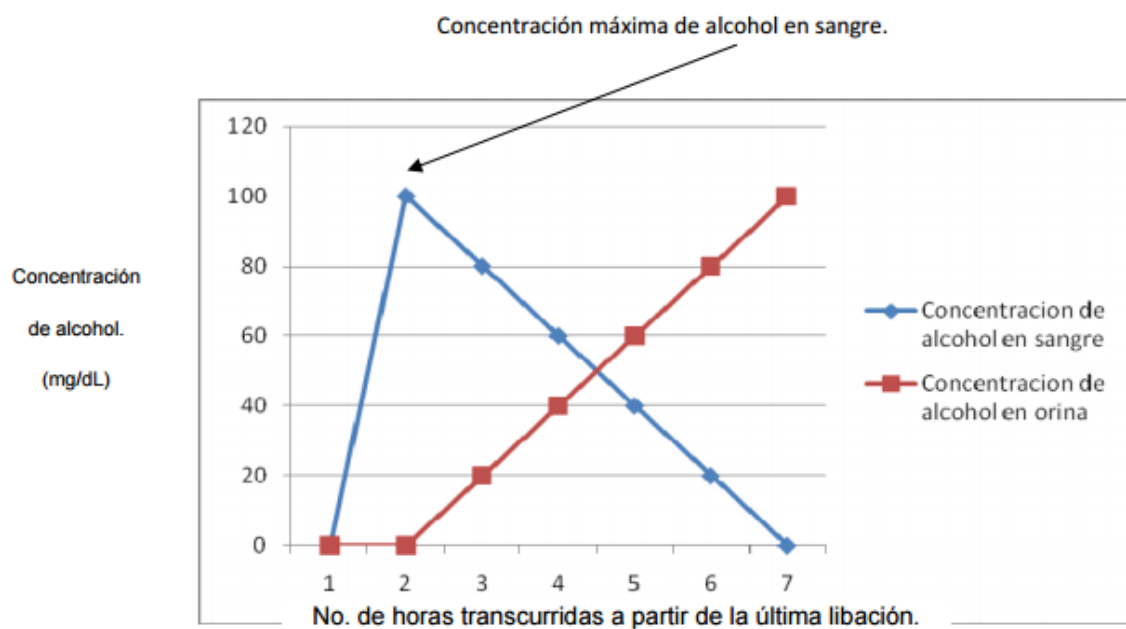


Figura 5. Representación evolutiva de la concentración de alcohol en la sangre con relación a la concentración de alcohol en la orina.

Fuente: Adaptado de (Cedilo,2010).

2.3.3. Análisis de alcohol en la saliva.

El etanol se difunde rápidamente en la saliva durante la ingesta de alcohol, e inmediatamente después de la concentración salival el alcohol se encuentra temporalmente en mayor cantidad en el plasma. Al transcurrir alrededor de 30 minutos, la concentración de etanol

en la saliva se nivela con el que se encuentra en el plasma, por lo que se puede decir que existe la posibilidad de que el etanol penetre fácilmente en la cavidad bucal (Waszkiewicz et al, 2011).

Sin embargo, no existe una relación científicamente comprobada entre la concentración de alcohol en sangre y la concentración de alcohol en saliva por lo que los resultados obtenidos no son precisos y tienen menos fiabilidad que una prueba de BAC (Prada y Martínez, 2003).

2.3.4. Análisis de alcohol en el aliento.

La mayoría de las Agencias de Tránsito hacen uso del análisis de alcohol en el aliento para determinar que el conductor ha ingerido bebidas alcohólicas, ya que la facilidad que presenta realizar esta prueba para medir la concentración de etanol en el aire exhalado hace que sea empleada con frecuencia. Además de que existe una relación comprobada entre el etanol que se encuentra en el aire espirado y el etanol en la sangre. La ejecución de esta prueba tiene iguales beneficios y fiabilidad que los métodos que se realizan en un centro de salud cuando se mide el BAC y presenta ventajas sobre la CAS Y CAO dentro de las que se encuentran el hecho de no ser una prueba invasiva, la facilidad para obtener una muestra de aliento, además de que se puede obtener el resultado de manera inmediata. El factor de conversión entre la cantidad de alcohol en el aliento y la sangre se basa en el supuesto de que 2100 mL. de aire exhalado por los pulmones contienen 1 mL. de etanol en la sangre. El resultado de las medidas se expresa en porcentaje de alcohol en sangre (Domínguez, 2008).

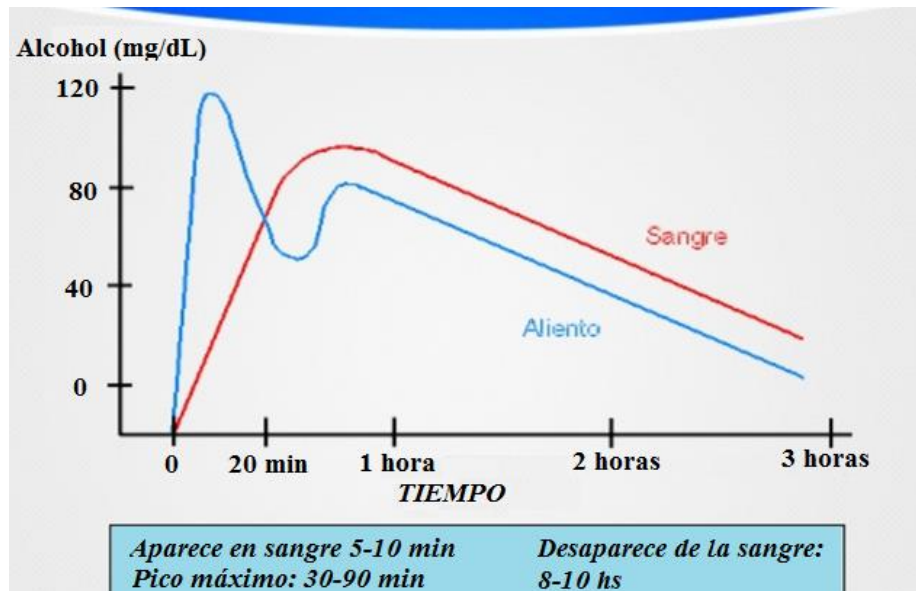


Figura 6: Relación entre el aire espirado y el contenido de alcohol en la sangre con el transcurso del tiempo.

Fuente: Adaptado de (Aranda, 2014)

Existe diversos dispositivos que analizan la concentración de alcohol en el aliento entre los que se encuentran: Dispositivos de Prueba de Aliento Evidenciales, Desechables y de Bloqueo de vehículos.

2.3.4.1. Dispositivos de prueba de aliento evidenciales.

Estos dispositivos permiten obtener resultados con carácter penal en caso de que el conductor se encuentre bajo la influencia de alcohol. Se emplean en situaciones que necesitan gran seguridad y precisión de medida. Estos dispositivos son equipos sofisticados y se encuentran en diferentes marcas como: Dräger, Intoxilyzer, Lifeloc, AlcoHAWK PT500P y PT750; el inconveniente que presentan es el elevado costo, mantenimiento, reparación y calibrado por personal certificado (Akers Biosciences, 2003).

2.3.4.2. Dispositivos de desechables.

En la Figura 7 se muestra un ejemplo de los dispositivos desechables desarrollado por la empresa Aker Biosciences el cual es denominado dispositivo BreathScan; este dispositivo es empleado para realizar una prueba desechable de alcohol cualitativa a través del aliento, los resultados de esta prueba no tienen valor legal, dentro del mercado son los más económicos esto se debe a que están diseñados específicamente para usarse una sola vez; el resultado se indica a través del cambio de color de los cristales de dicromato de potasio, el cual será positivo si la mayoría de los cristales son de color aguamarina es decir un color entre verde y azul y se obtendrá un resultado negativo si el color de los cristales es amarillo suave. Para que los resultados sean confiables es necesario que la prueba sea realizada como se especifica en las instrucciones descritas por el fabricante (Akers Biosciences, 2008).



Figura 7. Dispositivo desechable Breath Scan para realizar una prueba de alcohol.

Fuente: Adaptado de (Akers Biosciences, 2008)

2.3.4.3. Dispositivos de bloqueo en vehículos.

El avance tecnológico en la industria automotriz ha permitido que se puedan incorporar dispositivos de alcoholemia al encendido del automóvil para verificar el estado en el que se encuentra el conductor, para realizar la prueba se necesita una muestra de aliento y según se

manifiesten los resultados se procede a desbloquear o no el encendido del automóvil. Los equipos interlock son incorporados en mayor cantidad en países de Europa y Estados Unidos; el fabricante más importante de estos dispositivos es la empresa Interlock XT Dräger en la Figura 8 se muestra un ejemplo del dispositivo, el cual presenta alta fiabilidad y una gran estructura mecánica; además de cumplir con todas las homologaciones requeridas (Dräger, 2010).



Figura 8. Dispositivo de bloqueo de vehículos Dräger Interlock XT.

Fuente: Adaptado de (Dräger, 2017)

2.3.4.4. Sensores de detección de gases.

La estructura básica de un alcoholímetro está constituida por sensores electroquímicos cuya finalidad es revelar la presencia de etanol, para ello se emplean sensores de detección de gas compuestos por una capa de óxido de estaño. Dentro de los sensores de alcohol se encuentran los siguientes: TGS 2620, MQ-3, MQ303 y MR513.

- **TGS 2620.** _ “La serie Fígaro está constituido por una capa de semiconductor de óxido de metal formada en un substrato de la alúmina de una viruta de detección junto con un

calentador integrado el cual se encarga de determinar la presencia de etanol” (Figaro, s.f, p.01). Este sensor puede emplearse en diferentes aplicaciones dentro de las cuales se encuentran: probadores de alcohol haciendo uso de su alta sensibilidad a una variedad de gases combustibles como monóxido de carbono, detectores y alarmas de vapores solventes orgánicos, además de detectores o disolventes para fábricas, tintorerías e industrias de semiconductores. Dentro de las principales características se encuentra el bajo consumo de energía, alta sensibilidad, larga vida útil y bajo costo. (FIGARO, 2017)

- **MQ3.** _ La serie MQ de sensores presenta alta sensibilidad, bajo costo y rápida respuesta, el sensor MQ3 está diseñado para determinar la presencia de etanol en el aire emitido (HANWEI ELETRONICS CO.,LTD, 2010).
- **MQ303.** _ Al ser un sensor de la serie MQ también presenta alta sensibilidad y rápida respuesta por lo que este sensor semiconductor está diseñado para medir la presencia de etanol, se lo usa en detectores portables de alcohol, el inconveniente que presenta en su elevado costo (Seed Studio, s.f).
- **MR513.** _ Este sensor detecta la concentración de gas a través de la resistencia de bucle de platino. Es usado en aplicaciones como: detector portátil de alcohol y sistemas para activar o desactivar el encendido del automóvil para terreno doméstico e industrial, debido a su alta sensibilidad, bajo consumo de energía, diseño en miniatura y respuesta rápida. (Futurlec, s.f.)

2.3.5. Análisis de los cambios térmicos en personas en estado etílico.

El desarrollo de la biometría ha surgido como una alternativa a las formas tradicionales de identificación, anteriormente se realizaba las identificaciones mediante el uso de tarjetas que no están embebidas en las características físicas. La investigación en varias modalidades biométricas incluyendo el rostro, la huella digital, el iris, y el reconocimiento de la retina ha producido diversos grados de éxito. El reconocimiento facial es la modalidad más atractiva, ya que es el modo natural de identificación entre los humanos y totalmente discreto, sin embargo, es una de las modalidades más desafiantes en el área de visión artificial. (Zhao, Chellappa, Phillips, & Rosenfeld, 2003)

Recientemente, se ha dado énfasis a la adquisición de información de los rostros haciendo uso del espectro infrarrojo térmico. La razón principal de esto es que la temperatura del rostro depende principalmente de la condición fisiológica de la persona y la embriaguez es una condición fisiológica; de la cual se ha comprobado que la temperatura incrementa en el rostro de una persona en estado etílico debido a que las arterias y los vasos sanguíneos aumentan su actividad. (Wu, Xia, & Xie, 2009)

2.3.5.1. Detección de cambios térmicos en el rostro.

Estudios preliminares presentan que la temperatura fácil incrementa cuando una persona se encuentra en estado etílico y con el objetivo de vigilar los posibles cambios de la temperatura con el consumo de alcohol, se seleccionaron 20 puntos diferentes en el rostro de las personas. Estos puntos se obtuvieron en la misma posición para todos los rostros y las

adquisiciones que se realizaron. De acuerdo con ello, los resultados que se obtuvo fueron en: la nariz, las cejas, la barbilla y otros puntos del rostro como se muestra en la Figura 9.

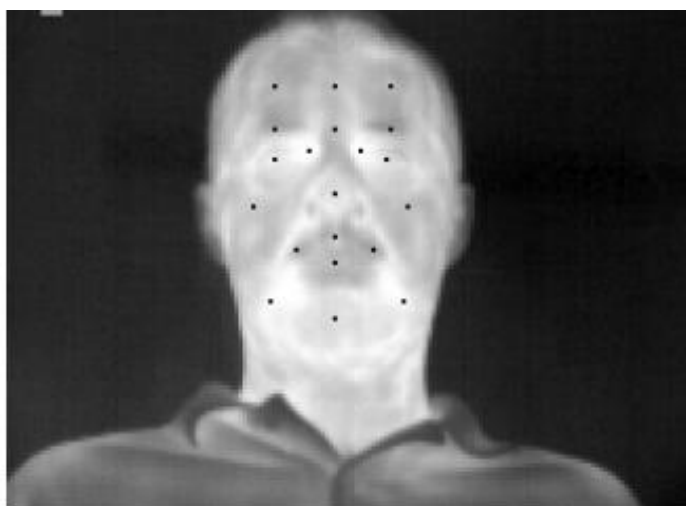


Figura 9. Puntos para determinar los cambios de temperatura en el rostro.
Fuente: Adaptado de (Koukiou, Panagopoulos, & Anastassopoulos, 2009)

En el enfoque que se desarrollado para identificar la variación de temperatura en lugares específicos del rostro cuando una persona consume alcohol, se concluye como resultado que se presenta elevación de la temperatura con respecto a otras regiones. Los resultados experimentales mostraron un aumento de temperatura en las regiones cerca de la boca y la nariz con respecto a las regiones en la frente. A continuación, en la Tabla 3 se muestran los cambios de temperatura haciendo énfasis en la nariz y la frente.

Tabla 3. Incremento de temperatura en las regiones de nariz y frente del rostro en una persona ebria

PERSONA	Sobrio		Ebrio		Incremento in la diferencia de temperatura entre la nariz y frente
	Frente	Nariz	Frente	Nariz	
A	230.5	221.1	214.7	233.9	$19.2 - (-9.5) = 28.7$
B	230.4	226.5	197.2	226.7	$29.4 - (-3.9) = 33.3$
C	229.7	183.8	193.4	232.3	$38.9 - (-45.9) = 84.8$
D	220.0	218.7	190.8	227.4	$36.5 - (-1.3) = 37.8$
E	233.7	186.4	220.1	218.3	$-1.8 - (-47.3) = 45.5$
F	212.7	219.2	200.0	232.6	$32.6 - (-6.4) = 26.2$

Fuente: Adaptado de (Koukiou, Panagopoulos, & Anastassopoulos, 2009)

Además de determinar que existen puntos del rostro que incrementan su temperatura, estudios preliminares manifiestan que existe una distribución de la temperatura en todo el ojo antes y después del consumo de alcohol, para visualizar los resultados se hace uso de algoritmos simples de procesamiento de imágenes y técnicas de modificación del histograma, con los cuales se determina que existe una diferencia de temperatura entre la esclerótica y el iris cuando alguien consume alcohol. El iris se vuelve más oscuro, lo que significa que su temperatura en comparación con la esclerótica disminuye, esto se debe a que la esclerótica está llena de vasos sanguíneos que aumentan la temperatura con el consumo de alcohol. En la Figura 10 se muestra que el iris presenta menor temperatura después del consumo de alcohol.

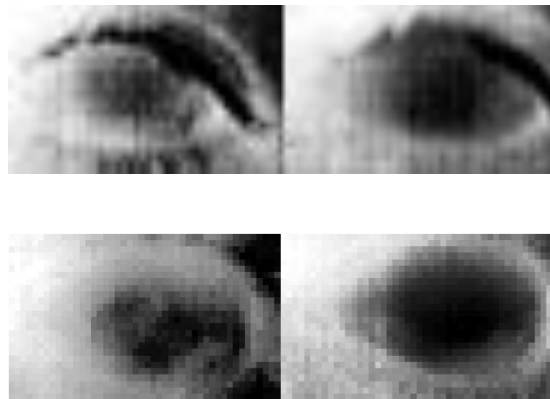


Figura 10. Las imágenes térmicas del ojo izquierdo representan a personas sobrias, mientras que las del lado derecho corresponden a personas en estado etílico.

Fuente: Adaptado de (Koukiou & Anastassopoulos, 2013)

2.3.6. Relación entre el consumo de alcohol y la dilatación de la Pupila.

La pupilometría se ha convertido en una herramienta de gran interés en las Ciencias de la Visión Artificial. La investigación realiza en el artículo denominado: Utilidad de la pupilometría dinámica en el control de alcoholemia de los conductores afirma que:

El diámetro basal aumenta significativamente a partir de concentraciones de alcohol en aire espirado iguales o mayores de 0.25 mg/l. Para el estudio se utilizó un diseño de

carácter cuasi-experimental, con 19 voluntarios participando en dos condiciones (pre y post ingesta de alcohol). En la sesión con alcohol, los participantes consumieron 27.5 ml de alcohol en intervalos de 60 minutos durante 4 horas. Se llevaron a cabo registros de pupilometría mediante el Power Refractor II, utilizándose cuatro tipos de estimulación lumínica: blanca (5600 K), azul (450 nm), verde (510 nm) y roja (600 nm). (Lobato, et al., 2013, (p. 137-145).

En la Tabla 4 se muestra los resultados obtenido de valores pupilares antes y después de la ingesta de alcohol, se debe considerar que el diámetro basal es la amplitud total de apertura de la pupila sin intervención de ningún tipo de estímulo. Esta amplitud de apertura normalmente es medida en mm, y se registra ante condiciones estables de determinado nivel de luminosidad.

Tabla 4. Valores pupilares en la sesión sin alcohol (pre-test) y en la sesión final (post-test).

	PRE-TEST		POST-TEST		P-valor
	Nivel de alcohol=0.00		Nivel de alcohol=0.49		
	Media \pm DT	(Min/Max)	Media \pm DT	(Min/Max)	
Diámetro basal (mm)	5.88 \pm 1.00	4.15 / 7.67	6.24 \pm 0.93	3.98 / 7.24	0.03
Luz Blanca					
Latencia (ms)	189.45 \pm 48.01	81 / 260	195.24 \pm 76.21	40 / 280	0.83
Amplitud (mm)	1.88 \pm 0.40	0.79 / 2.36	1.96 \pm 0.44	0.67 / 2.92	0.39
Recuperación (ratio)	91 \pm 4.77	84.9 / 99.01	89.83 \pm 5.36	80.58 / 100	0.44
Luz azul (450 nm)					
Latencia (ms)	228.44 \pm 52.59	159 / 295.8	235.24 \pm 64.18	120 / 360	0.88
Amplitud (mm)	1.44 \pm 0.35	0.79 / 2.31	1.39 \pm 0.31	0.67 / 1.91	0.63
Recuperación (ratio)	93.3 \pm 4.79	80.27 / 102.82	92.88 \pm 6.49	82.98 / 109.92	0.19
Luz verde (510 nm)					
Latencia (ms)	204.07 \pm 52.35	100 / 273.33	230.71 \pm 55.8	80 / 320	0.11
Amplitud (mm)	1.73 \pm 0.3	1.27 / 2.72	1.68 \pm 0.29	1.12 / 2.25	0.67

Recuperación (ratio)	92.6±6.19	75.41 / 109.32	92.14±9.62	83.84 / 124.06	0.93
Luz roja (600 nm)					
Latencia (ms)	249.9±68.71	79 / 360	247.5±79.18	80 / 360	0.67
Amplitud (mm)	1.10±0.24	0.65 / 1.80	0.97±0.26	0.67 / 1.35	0.00
Recuperación (ratio)	98.81±9.68	90.21 / 129.9	94.77±7.36	85.61 / 118.41	0.43

Fuente: Adaptado de (Lobato, et al., 2013, p. 145)

El artículo publicado en el Congreso Internacional de Seguridad Vial denominado “visión nocturna y consumo de alcohol: Riesgo en la conducción” afirma que:

Para altas tasas de alcoholemia ($BrAC > 0,25\text{mg/l}$), el índice de alteración visual fue significativamente mayor, el diámetro pupilar promedio pasó de $(5,5 \pm 0,9)$ mm a $(5,9 \pm 1,0)$ mm tras la ingesta de alcohol, lo que indica que en promedio se produjo un aumento del diámetro pupilar indicando un deterioro en la capacidad de discriminación visual. (Pereira, 2015, p.17-24)

Al finalizar la recopilación de la información referente a métodos para detección de alcoholemia se procede a seleccionar las variables que el sistema censara para la recolección de los datos que se emplearan para el entrenamiento del algoritmo de aprendizaje automático. Para medir el nivel de alcohol en el aliento se emplea el análisis de la concentración de etanol en el aire exhalado, para medir los cambios de temperatura se hará uso de la relación que existe entre el aumento de temperatura facial y la ingesta de bebidas alcohólicas y para censar el comportamiento humano se empleará la correlación existente entre consumir alcohol y la dilatación de la pupila.

2.4. Visión por computador

El área de visión por computador se encarga de transformar de datos que se obtienen de una cámara fija o de video en una representación digital para procesar y entender imágenes y videos. “El objetivo de la visión computacional es extraer características de una imagen para su descripción e interpretación por la computadora” (Sucar y Gómez, 2011, p.01). Los datos de entrada pueden incluir alguna información contextual como cámaras instaladas en automóviles para comprobar que hay una persona en el vehículo o un telémetro laser para detectar 14 células cancerígenas en una diapositiva. (Kaehler & Bradski, 2016)

El número de aplicaciones relacionadas con la Visión Computacional aumenta cada día. En la Tabla 5 se citan algunas de las múltiples aplicaciones en las que la visión artificial se aplica a diferentes disciplinas.

Tabla 5. Aplicaciones de la visión artificial

Área de producción	Aplicación
Control de calidad	Inspección de productos (papel, aluminio, acero, etc.). Identificación de piezas Inspección de circuitos impresos
Robótica	Control de soldaduras Guiado de robots (vehículos no tripulados)
Astronomía	Exploración del Espacio.
Reconocimiento de Caracteres	Control de cheques, inspección de textos
Control de tráfico	Matrículas de coches. Tráfico viario.
Meteorología	Predicción del tiempo.
Agricultura	Interpretación de fotografías aéreas. Control de plantaciones.
Militares	Seguimiento de objetivos. Vigilancia por satélites.
Reconocimiento y Clasificación	Clasificación de objetos por su tamaño, forma y color. Reconocimiento de caras de personas. Reconocimiento de huellas dactilares.

Fuente: Adaptado de (Platero, 2014)

2.4.1. Etapas básicas de una aplicación en visión artificial.

Cada aplicación de visión artificial o por computadora tiene sus especificaciones, sin embargo, existen etapas comunes entre ellas y no necesariamente se debe cubrir todas las fases en una implementación concreta. En la Figura 11 se muestra las etapas de visión artificial generalizadas para el desarrollo del proyecto, en la cual existe un encadenamiento temporal de una etapa sobre otra.

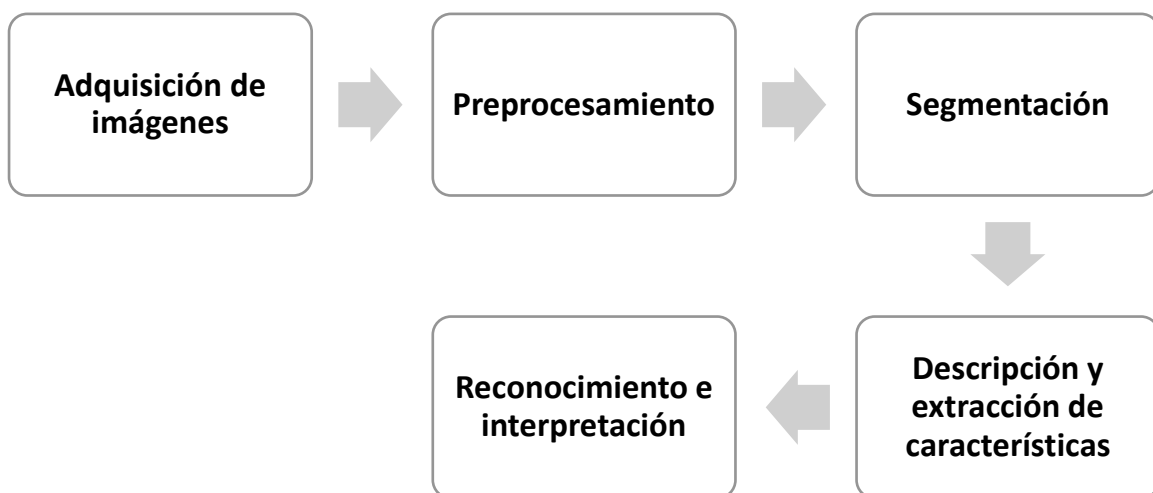


Figura 11. Diagrama de bloques con las etapas de Visión Artificial.

Fuente: Adaptado de (Platero, 2014) & (Sobrado Malpartida & Tafur Sotelo, 2003)

La primera etapa es la adquisición de la imagen que permite la captura y/o adquisición de imágenes, el principal objetivo de esta fase es realzar las características visuales de los objetos que se encuentren en la imagen mediante técnicas fotográficas (Sobrado y Tafur, 2003).

Una vez adquirida la imagen se pasará a la etapa de preprocesado, el principal objetivo es mejorar la calidad informativa de la imagen adquirida, eliminando los problemas de ruido granular de cuantización o de iluminación (Platero, 2014).

La fase de segmentación se encarga de evaluar si cada pixel de la imagen pertenece o no al objeto de interés en función de los valores RGB o HSV, esta etapa genera una imagen binaria, representados por 0 a los que no pertenecen y por uno a los pixeles que si pertenecen al objeto (Sobrado y Tafur, 2003).

La etapa de descripción y extracción de características permite obtener un modelo que representa las características relevantes que posee un objeto en específico para el sistema de visión computacional (Sobrado y Tafur, 2003).

Por último, se tiene la fase de Reconocimiento en donde se “clasifican los objetos con características comunes del espacio 3D y se restringe el espacio mediante técnicas de triangulación para que se seleccione una única imagen y esta pueda ser interpretada” (Álvarez, 2014, p.10).

2.4.2. Librería de Visión Computarizada de Código Abierto.

OpenCV³ es una librería de código abierto para el desarrollo de funciones de programación dirigidas principalmente a visión por computador y con un fuerte enfoque en aplicaciones que necesitan ser interpretadas en tiempo real. OpenCv puede ser utilizado tanto en aplicaciones académicas como comerciales bajo una licencia BSD que permite el libre uso, distribución y adaptación. (Laganière, 2016)

La biblioteca OpenCv cuenta con más de 2500 algoritmos optimizados, que incluyen un amplio conjunto de algoritmos de visión artificial por computador y algoritmos aprendizaje

³ Librería de Visión Computarizada de Código Abierto

automático. La aplicación de estos algoritmos tiene varios fines, sin embargo, las principales funciones son: detectar y reconocer rostros, identificar objetos según su tamaño, forma o color, clasificar las acciones humanas en videos, movimientos de cámara, objetos en movimiento, extraer modelos 3D, cámaras estéreo, fusionar imágenes, encontrar imágenes similares de una base de datos, eliminar los ojos rojos de las imágenes tomadas con flash, seguir los movimientos de los ojos, reconocer paisajes, realidad aumentada, etc. Cuenta con interfaces de C++, C, Python, Java y MATLAB, además es compatible con Windows, Linux, Android y Mac OS (Espinosa Peralta, 2016).

2.4.2.1. Estructura de OpenCv.

OpenCV tiene una estructura modular, es decir divide los componentes que la forman en módulos. Cada uno de estos módulos contiene estructuras de datos, funciones y algoritmos relativos a un estrecho margen de especialización indicado por el nombre del propio modulo. (Brydko, 2015)

- *core*: define una estructura de información básica que incluye las estructuras de datos básicas y las funciones básicas de procesamiento de imágenes en la matriz multidimensional y en las funciones básicas de distintos módulos.
- *highgui*: este módulo provee una “interfaz fácil de manipular para la captura de video, imagen y codecs de video, tales como las capacidades simples de la interfaz de usuario; además de capacidades como la de capturar eventos del ratón” (Rodríguez Bazaga, 2015).

- *Imgproc*: es un módulo para el procesamiento de imágenes, el cual incluye filtros, transformaciones de imágenes geométricas, histogramas, conversión de espacios de color, etc.
- *Video*: módulo de análisis de video que incluye estimación de movimiento, sustracción de fondo y algoritmos de seguimiento de objetos.
- *Objdetect* incluye “algoritmos de detección y reconocimientos de objetos e instancias de clases predefinidas como reconocimiento de rostros, ojos, boca, personas, etc.” (Calderón Jácome & Maldonado Andrade, 2014).
- *Calib3d*: permite la calibración de cámara, estimación de posición de objetos, algoritmos básicos de geometría de visión múltiple.
- *Features2d*: Detectores de características sobresalientes, descriptores y descriptor de coincidencias.
- *Ml*: contiene una amplia gama de algoritmos de aprendizaje de la máquina.
- *Gpu*: Algoritmos acelerados por GPU de diferentes módulos OpenCV.
- *Photo*: herramientas útiles para la fotografía computacional.
- *Contribut*: Este módulo contiene nuevas cosas que aún no se han integrado en toda la librería.

Es necesario considerar que en OpenCv 3.0 la librería se ha dividido en dos partes: OpenCV desarrollado completamente y el estado actual de la técnica en funcionalidad de visión más amplia en `opencv_contrib`. El primero es sustentado por el núcleo del equipo OpenCV y contiene (en su mayoría) código estable, mientras que el segundo es menos estable, es sustentado y desarrollado principalmente por la comunidad, puede tener partes bajo licencia y puede incluir algoritmos patentados. A continuación, se describen algunos de los módulos disponibles en el repositorio `opencv_contrib`. (Laganière, 2016)

- *Dnn*: Redes neuronales profundas.
- *face*: Reconocimiento facial.
- *text*: Detección y reconocimiento de texto; opcionalmente puede usar código abierto OCR.
- *rgbd*: Procesamiento de mapas de profundidad RGB + los obtenidos con kinect y otros sensores de profundidad (o simplemente computados con algoritmos de correspondencia estéreo).
- *bioinspired*: Visión inspirada.
- *ximgproc*, *xphoto*: Algoritmos avanzados de procesamiento de imágenes y fotografía computacional.
- *tracking*: Modernos algoritmos que permiten conocer la ruta de un objeto.

2.4.3. Detección de rostros.

La detección facial es una tecnología de visión artificial que determina el lugar y el tamaño de rostros humanos en imágenes o videos. Es uno de los subtipos del modelo de detección de objetos, cuya tarea es encontrar la localización y el tamaño de los objetos en una imagen perteneciente a la clase dada (Viola & Jones, 2001). Existen varios métodos para la detección facial con la finalidad de mejorar y reducir el tiempo de un sistema electrónico para detectar el rostro, sin embargo, el algoritmo de Viola-Jones, se lo considera como el más eficaz, obteniendo un mayor porcentaje de aciertos respecto a sus pares, además de más rapidez. Este algoritmo integra un nuevo concepto, la imagen integral, que junto con el algoritmo de boost como método de entrenamiento, forman un clasificador complejo y preciso. (Yang, Kriegman, & Ahuja, 2002)

2.4.3.1. Detección de rostros: Viola & Jones.

El método de detección de rostros de Viola-Jones, puede ser entrenado para identificar muchos tipos de objetos rígidos, sin embargo, el principal enfoque es para la detección de rostros. El rendimiento del algoritmo es semejante a los demás algoritmos de detección, la diferencia que presenta éste es que, al ser utilizado en tiempo real su detector es capaz de correr 15 veces más rápido que los algoritmos anteriores, sin necesidad de recurrir a técnicas como la detección de diferencias en la imagen o color de la piel. Por otra parte, la adición de estas fuentes alternativas de información resultará en la consecución de velocidades aún más altas. (Viorica Andreea, 2016)

Este detector se fundamenta en tres conceptos. El primero de ellos es conocido como el "Imagen Integral", que permite que las características tipo "haar" utilizadas por este detector se puedan computar muy rápido. El segundo es un algoritmo de aprendizaje automático, "Adaboost", que selecciona sólo las características importantes de todo el conjunto. El tercer concepto es la creación de una estructura en "cascada", la combinación de clasificadores complejos, que rechaza el fondo de la imagen de entrada pasando más tiempo de cálculo en las áreas que puedan contener el objeto de interés.

- *Las características tipo Haar:* El algoritmo de detección facial buscará características específicas que sean comunes en un rostro humano. Estas características por extraer son invariantes respecto al tamaño de la zona de búsqueda de la imagen. En la Figura 12 se muestran las más comunes como: las que se asemejan a los bordes (2 rectángulos con distintos valores de nivel de gris en las distintas posiciones posibles), de línea (3 rectángulos con una variación central suficiente en las distintas posiciones

posibles y con distintos grosores de línea) y de contornos concéntricos (representados por una zona cuadrada dentro de la otra con las posibles rotaciones).

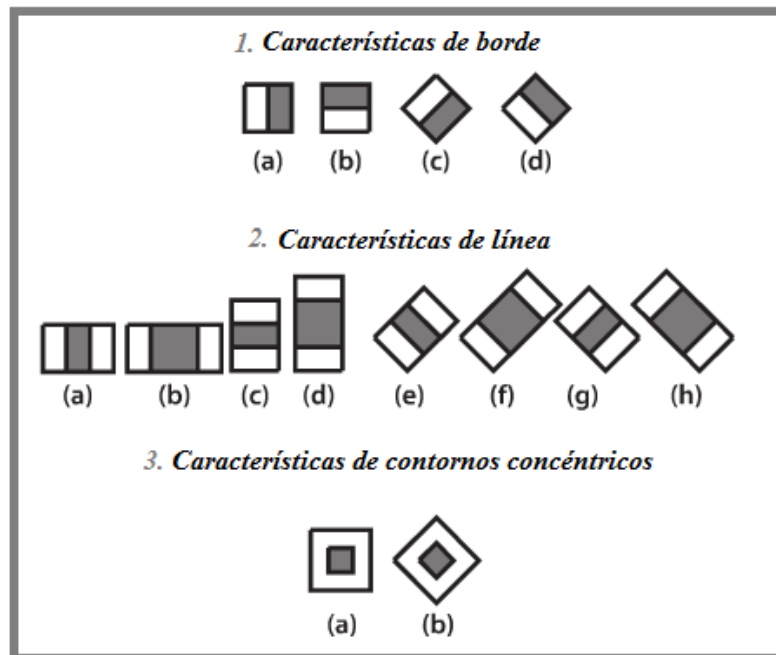


Figura 12. Características principales que comprueba el detector Viola & Jones.

Fuente: Adaptado de (Bronte Palacios, 2008)

- **Imagen Integral:** Se emplea el término “Imagen Integral” dentro de la estructura de detección de objetos para referirse a un método rápido y eficiente para calcular la suma de los valores de los píxeles en cualquier zona rectangular desde la esquina superior izquierda (referencia de coordenadas). En la Ecuación 1 se puede observar en términos de ecuación:

$$h_i = \begin{cases} 1 & \text{si } p_i f_i(x) < p_i \theta_i \\ 0 & \text{en otro caso} \end{cases}$$

Ecuación 1. Ecuación de la Imagen Integral

Fuente: (Bronte Palacios, 2008)

- *Adaboost*⁴: Este algoritmo de clasificación puede ser usado junto con otros tipos de algoritmos de aprendizaje a fin de mejorar sus resultados. La idea principal es combinar la salida de algunos clasificadores débiles (learners) en una suma ponderada, creando así un clasificador final fuerte, cuyo error tienda exponencialmente a cero. Las dos primeras características seleccionadas por el clasificador AdaBoost son: la diferencia de niveles entre los ojos y la parte superior de los pómulos y la diferencia de niveles entre los ojos y el puente formado por la nariz.
- *Filtro en cascada*: El detector de Viola-Jones usa la técnica de Adaboost, pero organiza los clasificadores como una cascada de nodos de rechazo. Este proceso en cascada se realiza para cada nodo e intenta optimizar el rechazo de los negativos y el rechazo de falsos positivos, para que de esta manera se termine instantáneamente el cálculo computacional. Tan sólo el candidato que logre atravesar la cascada entera será clasificado como un rostro. De esta manera el coste computacional se reduce significativamente, ya que la mayoría de las áreas que no contienen el objeto de interés son rechazadas en alguna de las etapas de nuestra cascada.

El detector desarrollado por Paul Viola y Michael Jones. Viene implementado directamente en OpenCV, lo que permite el uso del modelo bajo una licencia libre además de caracterizarse por la rapidez en la detección de objetos en la escena, un alto índice de aciertos y un bajo porcentaje de falsos positivos. (Bronte Palacios, 2008)

⁴ Adapting Boosting (Boost adaptativo) es un meta algoritmo de machine learning creado por Robert Schapire y Yoav Freund en 2003.

2.4.4. Reconocimiento facial.

Una aplicación de reconocimiento facial puede identificar un ser humano en un cuadro de imagen o de vídeo digital dado. Estos sistemas se utilizan cuando la identidad del individuo es desconocida, y es necesario identificarlo sobre todo en áreas de seguridad junto con otras tecnologías de autenticación biométrica.

Existen dos tipos de técnicas de reconocimiento facial: técnicas basadas en apariencia y técnicas basadas en modelos. En la Figura 13 se muestra un esquema esquematizado con las técnicas de reconocimiento facial.

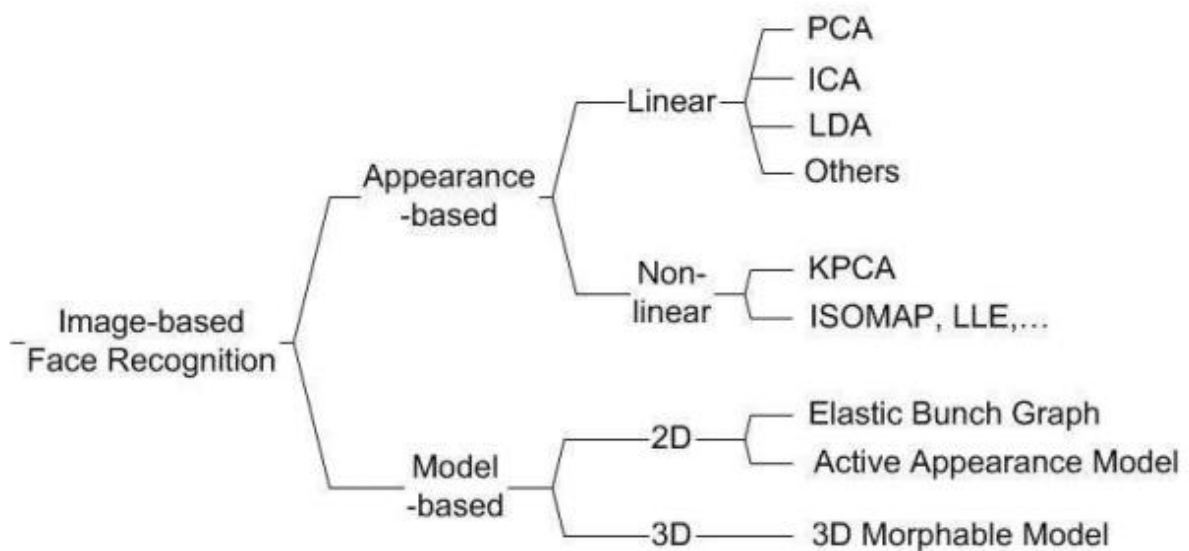


Figura 13. Métodos de Reconocimiento Facial.

Fuente: Adaptado de (Xiaoguang, 2005)

Debido a que OpenCV dispone de una clase para el reconocimiento de rostros FaceRecognizer, se analizará los métodos basados en apariencia lineales ya que dentro de esta clase se dispone de los algoritmos: Eigenfaces (basado en PCA), Fisherfaces (basado en FLD) y Local Binary Patterns Histograms.

2.4.4.1. Eigenfaces.

En una imagen de entrada existen componentes principales o características comunes como: ojos, labio, nariz y distancias entre estos componentes, esos componentes principales son llamados eigenfaces (Tayal, Kumar Pandey, & Singh, 2013). El algoritmo de reconocimiento de rostro Eigenfaces sigue los siguientes pasos:

- El primer paso es poseer un conjunto de imágenes de entrenamiento de diferentes personas, este proceso es conocido como etapa de entrenamiento, donde las imágenes poseen el mismo tamaño.
- A continuación, eigenfaces convierte las imágenes en una matriz para obtener los componentes principales (eigenfaces o autovectores) de la matriz de covarianzas formada por las imágenes de la base de datos y la imagen de entrada.
- Luego se procede a comparar las distancias entre el vector de la imagen original con el resto.
- Finalmente, la imagen que posea la menor distancia es considerada como el rostro de la imagen de entrada y de esta manera se obtendrá el rostro positivamente.

2.4.4.2. Fisherfaces.

Esta técnica considera las imágenes de entrenamiento de un mismo individuo como clases, por lo tanto, existen el mismo número de clases que personas. Una vez definida las clases se procede a calcular dos matrices: la matriz de dispersión entre clases y la matriz de dispersión dentro de clases. Una vez calculada estas matrices se obtiene una matriz de proyección donde cada columna será la base del nuevo sub-espacio, denominada Fisherfaces.

Las FisherFaces también permiten una reconstrucción de la imagen, es decir utilizan la comparación de imágenes mediante la distancia euclidiana para determinar qué imagen del conjunto de entrenamiento se parece más a la imagen de entrada (Espinoza Olgún & Jorquera Guillen, 2015).

2.4.4.3. Local binary pattern.

Es un operador de textura que etiqueta los pixeles de una imagen por thresholding o umbral, método de segmentación de imagen que a partir de una en escala de grises se crea una imagen binaria, cada vecino del pixel y el resultado de la operación se considera como un numero binario. Una de las propiedades más importantes es su indiscriminación frente a los cambios en la escala de grises como por ejemplo la luminosidad.

En este algoritmo se dice que para cada pixel de la imagen a procesar se examina sus 8 vecindades alrededor del pixel central en una figura con dimensiones de 3x3. Para cada pixel en la vecindad se determinará la intensidad de ella por medio de dos valores: 1 si el valor es mayor al del pixel central o 0 si el valor es menor al del pixel, como se puede ver en la Figura 14.

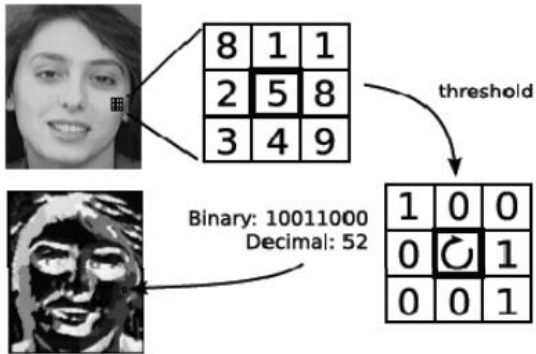


Figura 14. Creación de vector de algoritmo LBP.
 Fuente: Adaptado de (Espinoza Olgún & Jorquera Guillen, 2015)

Al ya tener calculada la matriz de número con valores 1 y 0 se tienen 2 elevado a 8 posibles patrones, se procede a multiplicar cada valor del pixel con la matriz para así calcular la distribución de textura de forma similar a los histogramas de escala de grises. En la Figura 15 se muestra un ejemplo del reconocimiento facial con los patrones binarios locales.

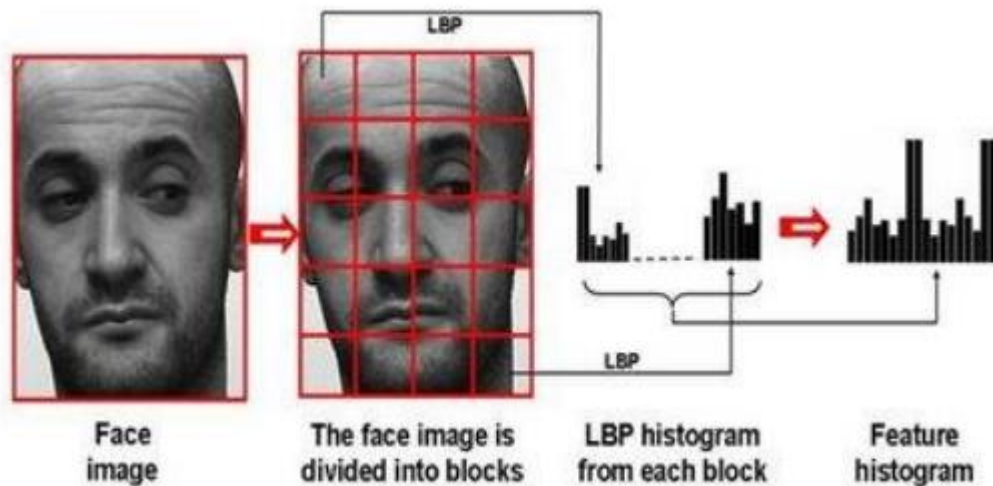


Figura 15. Reconocimiento facial con los patrones binarios locales.

Fuente: Adaptado de (Espinoza Olgún & Jorquera Guillen, 2015)

2.5. Aprendizaje automático

El aprendizaje automático (Machine Learning) es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. Un sistema puede aprender de un conjunto de datos mediante la extracción de ciertos patrones y luego ser capaz de responder a preguntas relacionadas con un nuevo conjunto de datos (Viorica Andreea, 2016). En la última década, el aprendizaje automático ha permitido el desarrollo de diversas disciplinas que incluyen motores de búsqueda, diagnósticos médicos, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, juegos, robótica, vehículos autónomos, entre otros (Sánchez García de Blas, 2015). Las aproximaciones metodológicas al diseño de las etapas de aprendizaje pueden ser: aprendizaje

supervisado, aprendizaje no supervisado y aprendizaje mediante refuerzo (García Chang, 2009). Para el desarrollo del proyecto se empleará un aprendizaje supervisado ya que los valores con los que se establecerá la función son proporcionados por un supervisor o profesor.

2.5.1. Aprendizaje automático supervisado.

En el aprendizaje supervisado, se proporcionan ejemplos de la forma (x_i, y_i) y se asume una función de aprendizaje f tal que, $f(x_i) = y_i$. El objetivo es encontrar la función f , de tal forma que dicha función capture los “patrones generales” presentes en los datos de entrenamiento y se pueda aplicar para predecir valores y , a partir de diversos valores de x . Generalmente cada x_i es una descripción de algún objeto, situación o evento y cada y_i es un descriptor simple. No es necesario expresar a la función f como un conjunto de reglas, ésta, puede ser escrita en muy diversas notaciones: como una definición lógica, un procedimiento, un árbol de decisión, un polinomio, una red neuronal, etc. (Villegas Quezada, 2005).

Se denomina “supervisado” porque se puede pensar que los valores de y_i son proporcionados por un supervisor o profesor.

Cuando se tienen pocos valores de y_i , se les denomina clases, y la función f asignará cada x a su correspondiente clase. Si se tienen únicamente dos posibles valores para y_i , se les considera como ejemplos positivos o negativos de algún determinado concepto. Por lo tanto, la meta general del aprendizaje supervisado es producir una regla de clasificación que asigne de manera correcta nuevos ejemplos a las clases propuestas (Villegas Quezada, 2005).

Scikit-learn es la biblioteca de código abierto para el aprendizaje automático diseñada para el lenguaje de programación Python. La finalidad de esta es proporcionar herramientas de aprendizaje automático eficientes y bien establecidas dentro un entorno de programación

accesible para los expertos en aprendizaje no automático y reutilizable en varias áreas científicas. Esta biblioteca, incluye algoritmos de aprendizaje clásico, herramientas de evaluación y selección de modelos, así como procedimientos de preprocesamiento de datos. (Buitinck1 et al., 2013) A continuación, se detallan los beneficios y los inconvenientes que presenta esta biblioteca.

- *Beneficios:*
 - Interfaz consistente para los modelos de aprendizaje automático.
 - Proporciona muchos parámetros de ajuste, pero con valores predeterminados razonables.
 - Documentación excepcional.
 - Amplio conjunto de funcionalidades para tareas complementarias.
 - Comunidad activa para desarrollo y apoyo.

- *Posibles inconvenientes:*
 - Más difícil (que R) para comenzar con el aprendizaje automático.
 - Menos énfasis (que R) en la interpretabilidad del modelo.

2.6. Sistemas de bloqueo del automóvil

Debido a que no es suficiente con que el conductor realice una prueba de alcoholemia, y sea consciente de las consecuencias que conlleva manejar un vehículo en estado de ebriedad y no conduzca si la prueba marca un valor alto de contenido de alcohol, es indispensable implementar una etapa de bloqueo que lo obligue a salvaguardar su vida y la de los ocupantes

del vehículo, se procede a analizar las posibles opciones que se pueden tener para bloquear el encendido del vehículo.

Los sistemas de bloqueo de un vehículo consisten básicamente en métodos empleados para suprimir la liberación de energía peligrosa almacenada en equipos de forma no deseada con la finalidad de evitar accidentes.

Para la etapa de bloqueo se mencionan 3 opciones como las principales y accesibles para evitar que el usuario del vehículo pueda no conducir si sobrepasa el límite establecido por la Ley de Tránsito y Transporte Terrestre; a continuación, se detallan los siguientes sistemas de bloqueo:

- Sistema de bloqueo mecánico a través de la llave de contacto.
- Sistema de bloqueo a través de la bomba del combustible.
- Sistema de bloqueo de encendido mediante la función anti-arranque o anti-encendido.

2.6.1. Sistema de bloqueo mecánico a través de la llave de contacto.

La llave de contacto realiza la conexión eléctrica de los diferentes circuitos eléctricos. Para ello dispone de varias posiciones que se van consolidando mediante el giro, en sentido horario de la llave. La llave de contacto de un automóvil generalmente realiza dos funciones: Interruptor eléctrico y bloqueo de la dirección.

El proceso de funcionamiento al emplear la llave de contacto como interruptor eléctrico es el siguiente: Inicialmente la llave de contacto es introducida o retirada del orificio en una sola posición, debido a que en las demás posiciones la llave se mantiene bloqueada por seguridad. Por otro lado, cuando la llave se encuentra fuera del orificio es posible activar el bloqueo mecánico de la dirección, basta solo con girar el volante de dirección al menos $\frac{1}{4}$ de vuelta en cualquier sentido.

2.6.2. Sistema de bloqueo a través de la bomba del combustible.

Este sistema consiste en evitar el flujo de combustible (Diésel o Gasolina) hacia el riel de inyectores del motor, evitando de esta manera el encendido del automóvil. En la mayoría de los casos el encendido se inhibe a través del sistema inmovilizador o cuando el vehículo tiene instalada una alarma contra robo.

2.6.3. Sistema de bloqueo de encendido mediante la función anti-arranque o anti-encendido.

Esta opción impide el arranque del motor, cuando no se ha presionado el pedal del embrague para los vehículos equipados con transmisión mecánica y para los vehículos equipados con transmisión automática se debe colocar la palanca de control de cambios en posición 'N' (Neutro) o 'P' (Parking) para permitir el encendido; sin embargo, esta opción no impide que el conductor conduzca si el resultado de la prueba de alcoholemia marca un valor alto de alcohol en el aliento.

Después de analizar las tres opciones anteriores se muestra en la Tabla 6 las ventajas y desventajas de cada sistema de bloqueo para posteriormente realizar la elección del sistema más idóneo.

Tabla 6. Ventajas y desventajas de los sistemas de bloqueo del vehículo

Tipo	Ventajas	Desventajas
Sistema de bloqueo mecánico, mediante la llave de contacto.	<ul style="list-style-type: none"> • No existe consumo elevado de la batería, debido a que la llave con el chip no usa batería y recibe la energía cuando se acerca a la chapa de encendido a través de la antena receptora. • El sistema de bloqueo garantiza en alto grado la inviolabilidad del vehículo. ya que el automóvil solo se será puesto en marcha con esa llave. 	<ul style="list-style-type: none"> • El vehículo queda inmobilizado siendo virtualmente imposible hacer funcionar el motor para quien no tenga la llave, mando o teclado programado. • A un conductor en estado de ebriedad no le impedirá encender el vehículo.
Sistema de bloqueo a través de la bomba de combustible	<ul style="list-style-type: none"> • El motor no se encenderá debido a la falta de combustible. • La implementación del bloqueo del vehículo para el control de la bomba eléctrica del combustible se emplea un módulo relé de 12V. a 10 A. • Se puede utilizar los demás accesorios del vehículo como radio, luces, calefacción, entre otros. 	<ul style="list-style-type: none"> • Pérdida de potencia del vehículo, debido a que el combustible se envía a los cilindros por medio de una bomba, si no llega la suficiente tensión para darle energía a la bomba, esta no trabajará al 100% y la mezcla no será perfecta.
Sistema de bloqueo de encendido mediante la función anti-arranque o anti-encendido	<ul style="list-style-type: none"> • No existe consumo elevado de la batería. 	<ul style="list-style-type: none"> • El vehículo es desbloqueado presionando el embrague o poniendo la marcha en neutro dependiendo del vehículo, por lo que permite el encendido del automóvil.

Fuente: Autoría

2.7. Metodologías de desarrollo

El desarrollo de un nuevo proyecto implica esfuerzo físico e intelectual, con diversas dificultades. Por lo que es necesario la implementación de una metodología que permita guiar a los desarrolladores para el cumplimiento del proyecto; se define a una metodología como un modo sistemático de realizar, gestionar y administrar un proyecto con altas posibilidades de éxito (Carhuaricra, 2015). Existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo, así como diversas posibilidades de dividir a un proyecto extenso en sub-módulos llamados etapas que se centran a una actividad específica, que ayuda a definir entradas y salidas para cada una de las etapas; además de normalizar el modo de administración del proyecto. A continuación, se indican los principales modelos de desarrollo.

2.7.1. Modelo lineal.

Es el modelo más simple de las distintas metodologías, posee las etapas del proyecto divididas y secuenciadas para ser desarrolladas independientemente ya que no existe retroalimentación. En la Figura 16 se muestra el ciclo de vida de este modelo.

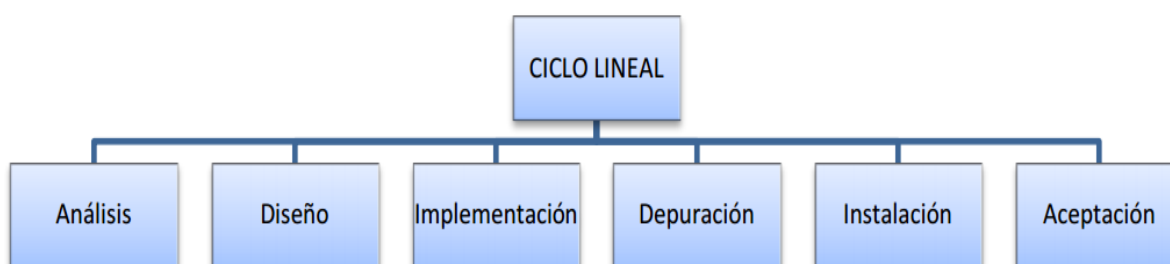


Figura 16. Etapas del Modelo Lineal.

Fuente: Adaptado de (SENA Servicio Nacional de Aprendizaje, 2013)

Con relación a la gestión, es necesario conocer con exactitud lo que se desarrollara en cada una de las etapas antes de iniciarlas para evitar posibles errores en el proyecto; la implementación de esta metodología es recomendable en proyectos pequeños debido a su administración económica y temporal; no se recomienda en proyectos que requieran retroalimentación de alguna etapa.

2.7.2. Modelo en cascada.

El modelo en cascada es un modelo de proceso clásico desde los años 70, desarrollado por Winston Royce. Es una metodología basada en la mentalidad de línea de ensamblaje, es decir admite interacciones; el desarrollo del proyecto pasa a través de una serie de etapas y el resultado final solo se observa al terminar la última etapa, así como también algún error que haya existido en una de las etapas. En la Figura 17 se muestra el proceso en cascada.

Modelo en cascada (*waterfall*)

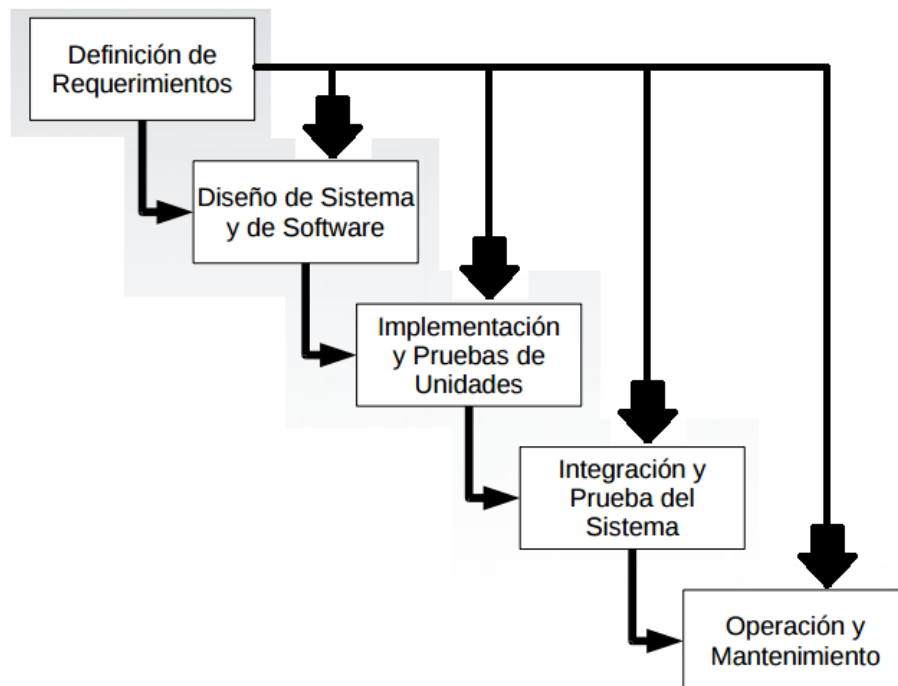


Figura 17. Etapas del Modelo en Cascada.

Fuente: Adaptado de (Gutierrez, 2011)

2.7.3. Modelo en v.

En la Figura 18 se indican las etapas del modelo en V que fue desarrollado por Alan Davis, es una variación del modelo en cascada, en esta metodología se incluye el proceso de Verificación (V) en las fases de análisis y diseño; además de incorporar retroalimentación entre las etapas. Este modelo es útil para proyectos, que necesitan alta confiabilidad.

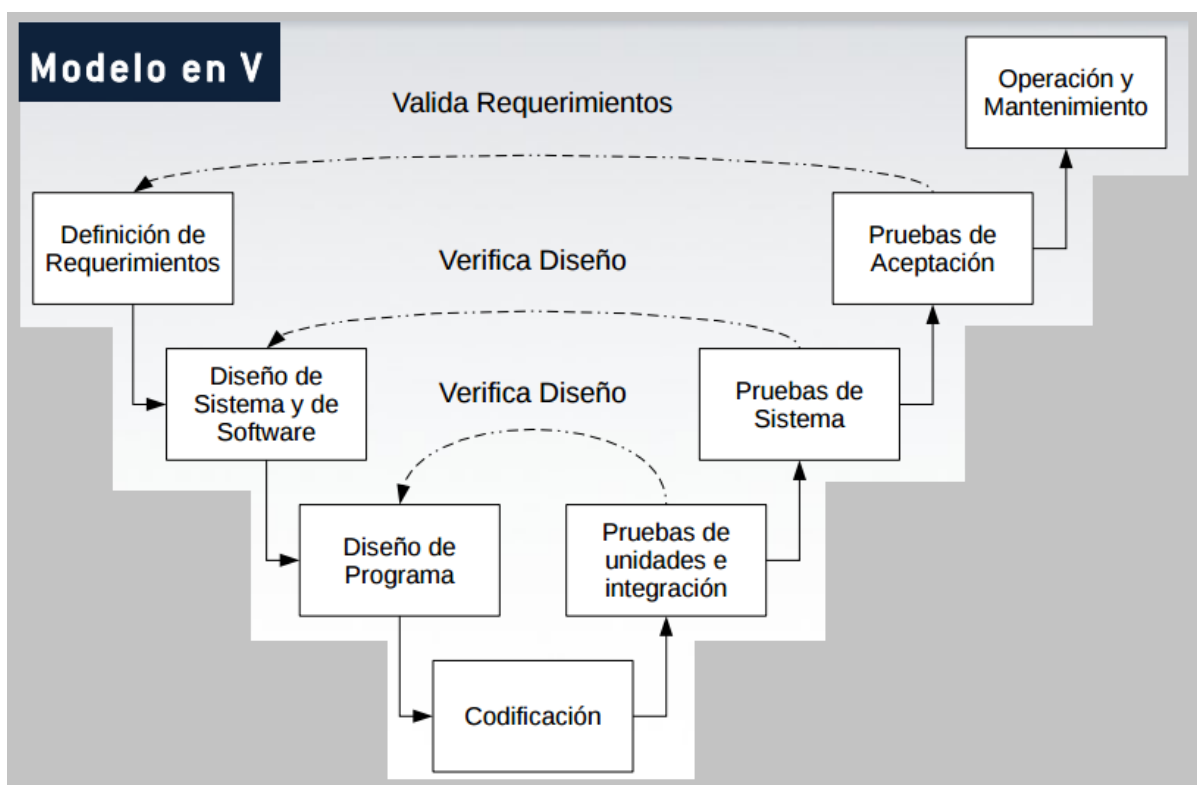


Figura 18. Etapas del Modelo en V.

Fuente: Adaptado de (Gutierrez, 2011)

2.7.4. Modelo de riesgos o de espiral.

Este modelo fue diseñado por Boehm. Esta metodología se basa en una serie de ciclos repetitivos, se puede asociar cada giro con cada fase del proceso de desarrollo. A medida que

el ciclo se cumple se construye nuevos modelos del proyecto sucesivos que van ganando la satisfacción del cliente o usuario como se muestra en la Figura 19.

Modelo de Proceso de Espiral

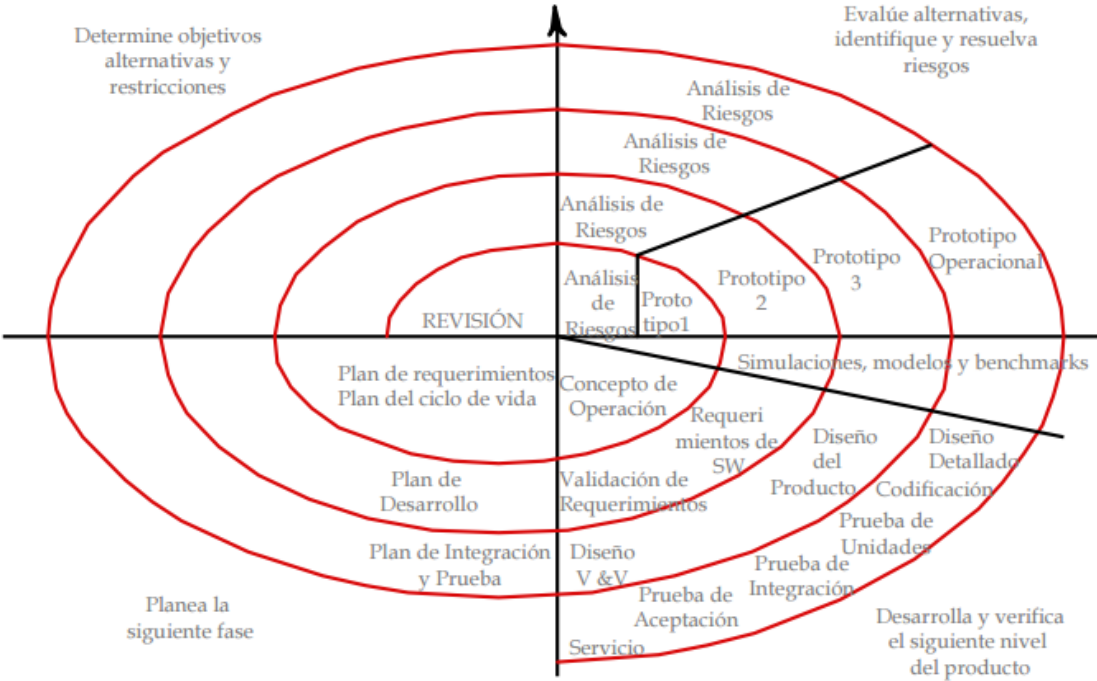


Figura 19. Etapas del Modelo Espiral.
Fuente: Adaptado de (Méndez, 2009)

Capítulo III. Desarrollo Experimental

En el presente capítulo se indica el desarrollo experimental del proyecto en base a la metodología de desarrollo del Modelo en V, con la finalidad de cumplir con las distintas herramientas de desarrollo de cada etapa que especifica el modelo.

3.1. Metodología

El modelo de desarrollo a emplearse para la realización de la investigación es el “Modelo en V”, el cual cumple con el proceso adecuado a través del cual se crea el proyecto, ya que este modelo relaciona las pruebas de validación con las fases de análisis, diseño y desarrollo del proyecto; de esta manera se puede identificar en qué fase se debe iniciar si alguna prueba falla, lo que garantiza altas posibilidades de éxito y minimiza los riesgos en la fase final.

En la Figura 20 se muestra las dos áreas y los niveles en la que se encuentra dividido el Modelo en V; el área izquierda contiene el análisis, los requerimientos y el diseño del sistema hasta llegar a la implementación; mientras que el área derecha comprende las validaciones del proyecto diseñadas específicamente para cada fase. A continuación, se especifica cada uno de los niveles con sus respectivas fases.

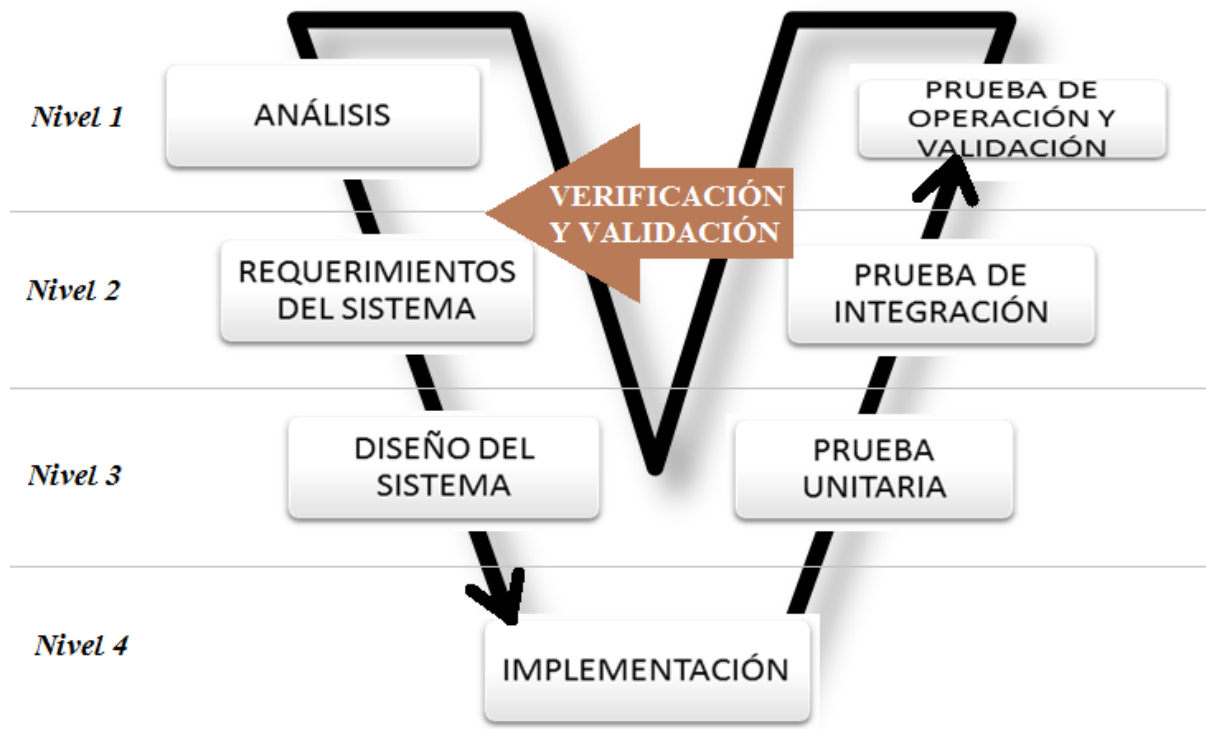


Figura 20. Modelo en V para el desarrollo de Proyectos.

Fuente: Adaptado de (Perez, Berreteaga, Ruiz de Olano, Urkidi, & Perez, 2006)

En el nivel primer nivel se deben definir y documentar los diferentes requisitos y especificaciones del proyecto a desarrollar, recolectando datos e información que permitan conocer los factores que influyan en la implementación del proyecto.

El nivel segundo se determina cuáles son los requerimientos del sistema; este nivel comprende una de las fases más importantes para que el proyecto tenga éxito, debido a que definen la funcionalidad del sistema propuesto. En la fase de integración del sistema se realizan pruebas para verificar la funcionalidad de cada módulo que forma el proyecto.

En el nivel tercero se definen todos los estados que se desarrollara en cada parte del sistema con la finalidad de obtener un diseño y visión general del proyecto. En la fase de prueba

unitaria se realiza la verificación independiente de cada parte del proyecto, comprobando la correcta implementación, coherencia y cumplimiento de las especificaciones respectivas.

En el último nivel se encuentra la fase de implementación con la cual se materializa el diseño del proyecto y se procede a la programación para cada parte del sistema.

3.2. Análisis

Para determinar los factores que intervienen en el desarrollo del Detector de Alcoholemia, se necesita realizar un análisis del entorno en el cual va a ser implementado el proyecto, en este caso un automóvil marca Chevrolet, modelo Corsa Evolution del año 2007; a partir del cual se obtendrá la información necesaria para determinar la ubicación y modelo del hardware a emplearse.

3.2.1. Análisis de las características del vehículo.

Mediante la investigación documental se realiza la recolección de los datos relevantes del vehículo para la implementación el Detector de alcoholemia; se analizará las dimensiones generales del vehículo, descripciones generales del tablero del automóvil (Panel frontal interior del vehículo) y especificaciones del calentamiento, ventilación y acondicionador de aire.

3.2.1.1. Dimensiones generales del vehículo.

En la Figura 21 se muestran la vista frontal del vehículo, la Tabla 7 contiene las dimensiones expresadas en milímetros.

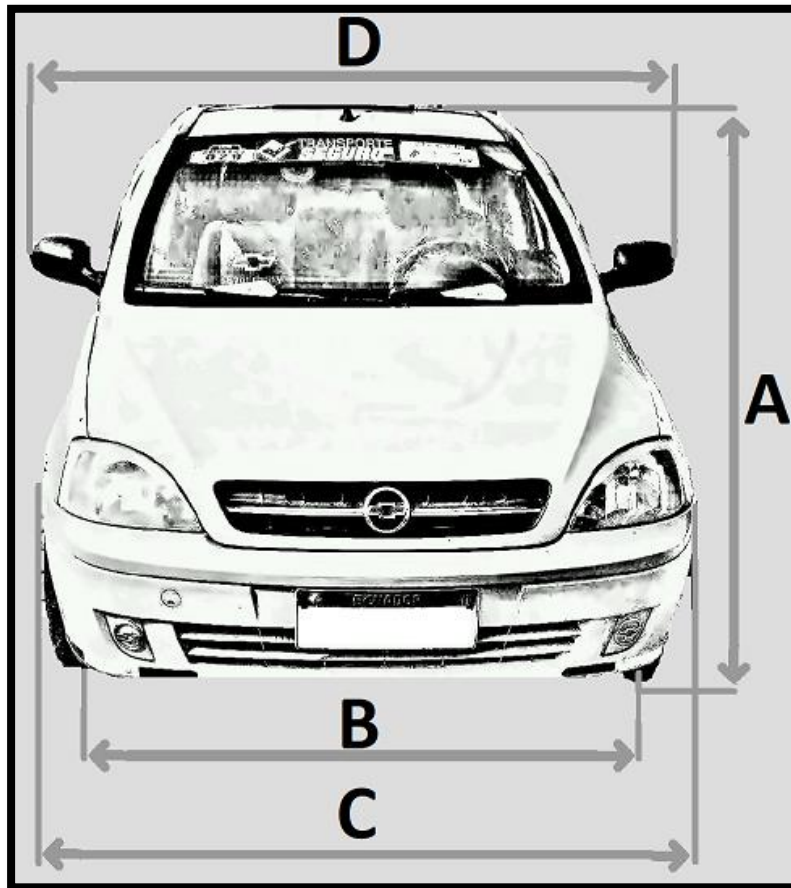


Figura 21. Vista Frontal del vehículo.
Fuente: Adaptado de (Chevrolet, 2007)

Tabla 7. Dimensiones de la vista frontal del vehículo.

		Dimensiones (mm)
A	Altura total	1.432
B	Trocha delantera	1.429
	Trocha trasera	1.420
C	Anchura total	1.646
D	Anchura total (espejo a espejo)	1.954

Fuente: (Chevrolet, 2007)

En la Figura 22 se muestran la vista lateral del vehículo, la Tabla 8 contiene las dimensiones expresadas en milímetros.

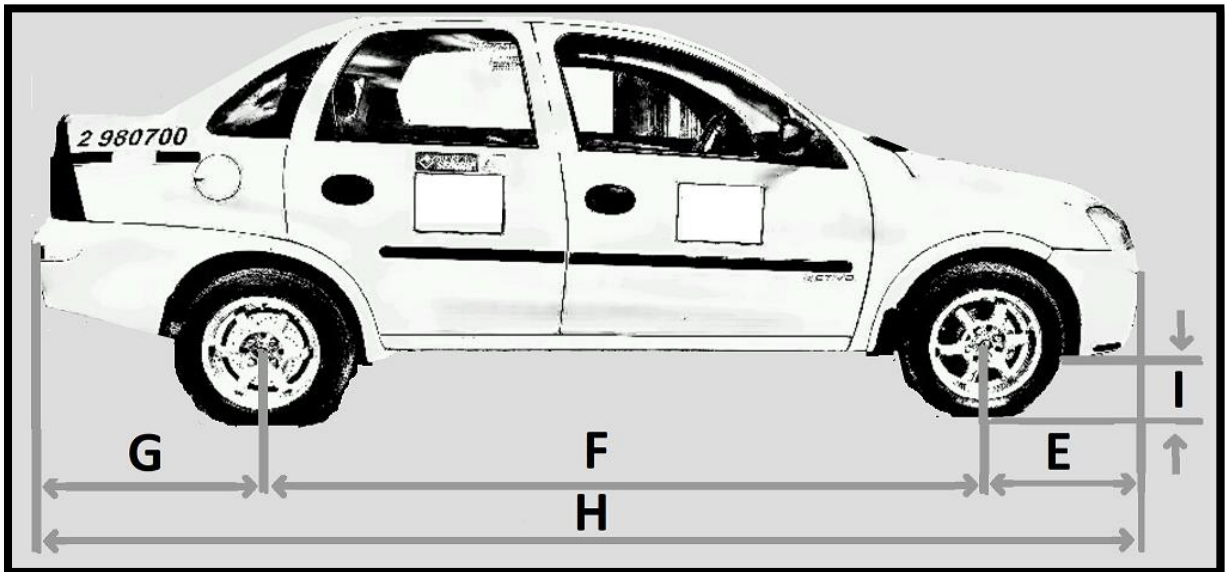


Figura 22. Vista lateral del vehículo.
Fuente: Adaptado de (Chevrolet, 2007)

Tabla 8. Dimensiones de la vista lateral del vehículo.

		Dimensiones (mm)
E	Distancia entre el centro de la rueda delantera y el paragolpes delantero	755
F	Distancia entre ejes	2.491
G	Distancia entre el centro de la rueda trasera y el paragolpes trasero	924
H	Largo total	4.170
I	Vano libre (vehículo cargado)	128

Fuente: (Chevrolet, 2007)

A continuación, se indican las dimensiones del interior del vehículo, la finalidad de conocer dichas dimensiones es determinar en la parte de requerimientos del sistema el

número de sensores y la ubicación del hardware. La Figura 23 se muestra las líneas de cota, que serán especificadas en la Tabla 9 las dimensiones en milímetros.

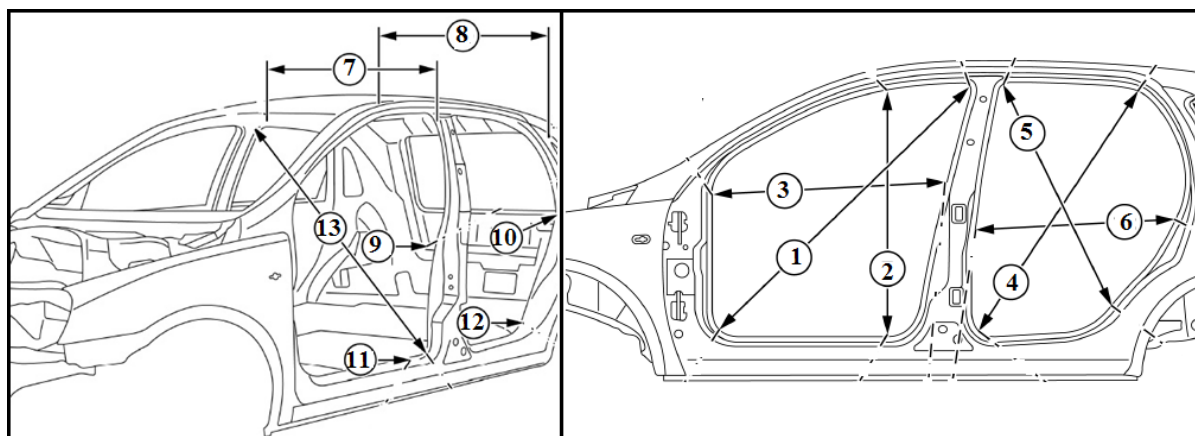


Figura 23. Vista Interior del vehículo.

Fuente: Adaptado de (Chevrolet, 2007)

Tabla 9. Dimensiones de la vista Interior del vehículo.

		Dimensiones
		(mm)
1	Diagonal del bastidor de la puerta delantera.	1308
2	Altura del bastidor de la puerta delantera.	934
3	Recorte del bastidor de la puerta delantera (columna bisagra de la puerta delantera hasta tope de la puerta delantera).	840
4	Diagonal del bastidor de la puerta trasera.	1098
5	Altura del bastidor de la puerta trasera.	975
6	Recorte del bastidor de la puerta trasera (columna tope de la puerta delantera hasta tope de la puerta trasera)	720
7	Separación entre las columnas del tope de las puertas delanteras (superior, interior).	1069
8	Separación entre las columnas del tope de las puertas traseras (superior).	1155
9	Separación entre las columnas del tope de las puertas delanteras (centro).	1327

10	Separación entre las columnas del tope de las puertas traseras (centro).	1342
11	Separación entre las columnas del tope de las puertas delanteras (inferior).	1350
12	Separación entre las columnas del tope de las puertas traseras (inferior).	1355
13	Diagonal de la columna del tope de la puerta delantera, superior, hasta la columna del tope de la puerta delantera, inferior, (inferior izquierda, superior derecha)	1565

Fuente: (Chevrolet, 2007)

En la Figura 24 se muestra un índice ilustrado para facilitar la localización de las partes del panel frontal interior del vehículo. La Tabla 10 denomina cada punto señalado en la imagen 25.

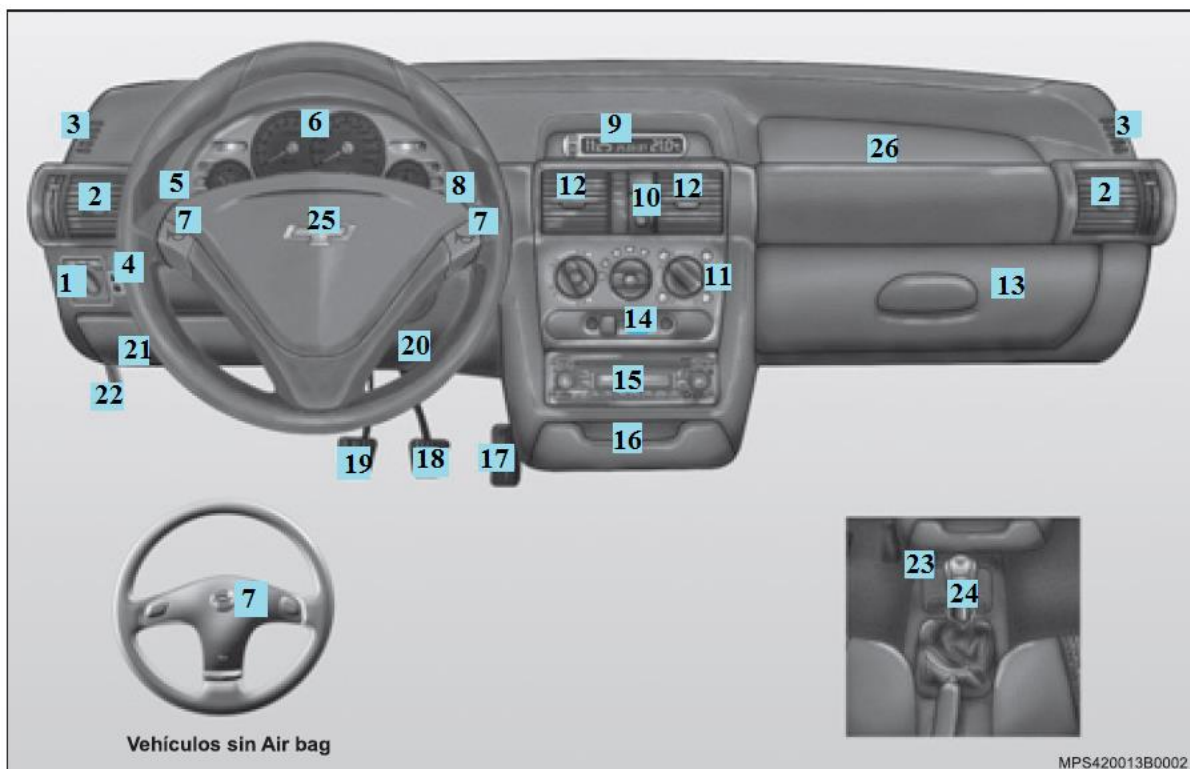


Figura 24. Equipamiento del tablero del vehículo.

Fuente: Adaptado de (Chevrolet, 2007)

Tabla 10. Equipamiento del vehículo

Número	Partes del tablero	Número	Partes del Tablero
1	Mando de las luces	14	Interruptor de recirculación de aire
2	Difusores de la salida de aire laterales	15	Sistema de audio (de estar equipado)
3	Desempañadores laterales hacia los vidrios delanteros	16	Cenicero y encendedor de cigarrillos o toma de accesorios y portaobjetos (de estar equipado)
4	Interruptor del faro antiniebla (de estar equipado)	17	Pedal del acelerador
5	Palanca de accionamiento: luz alta, destellador y señalizadores de giro	18	Pedal de freno
6	Tablero de instrumentos	19	Pedal del embrague
7	Bocina	20	Interruptor de encendido y arranque (no visible)
8	Palanca de accionamiento: limpia/ lavaparabrisas	21	Caja de fusibles
9	Módulo digital de función triple (de estar equipado)	22	Palanca de apertura del capó
10	Interruptor del señalizador de emergencia	23	Portaobjetos
11	Mandos de calentamiento, ventilación y refrigeración	24	Palanca de cambio de marchas – transmisión manual
12	Difusores de la salida de aire centrales	25	Air bag (conductor – de estar equipado)
13	Guantera	26	Air bag (lado del pasajero – de estar equipado)

Fuente: (Chevrolet, 2007)

3.2.1.2. Sistema de ventilación y acondicionador de aire.

Además de la circulación de aire natural que entra hacia el habitáculo por los difusores de aire (flechas) ubicados en el panel delantero, mientras el vehículo esté en movimiento, se puede también accionar un ventilador para que aumente la circulación de aire. Para mejor comodidad, este aire se puede calentar o enfriar (con acondicionador de aire). El sistema de mezcla de aire permite graduar la cantidad de aire caliente con aire enfriado, para que se pueda regular rápidamente la temperatura según el nivel requerido bajo cualquier velocidad. La circulación de aire se determina a través de la revolución del ventilador y puede ser influida por la velocidad del vehículo.

- *Difusores de salidas del aire:* En la Figura 25 se indican los cuatro difusores de aire ajustables (A) en la parte frontal del tablero de instrumentos, dos salidas laterales (D), salidas hacia el parabrisas (B) y salidas en la parte inferior del tablero de instrumentos (C) suministran ventilación agradable con aire en temperatura de ambiente, calentado o enfriado (con acondicionador de aire).




Figura 25. Difusores de salidas de aire en el vehículo.

Fuente: Adaptado de (Chevrolet, 2007)

- *Dirección del aire:* En la Tabla 11 se indica la dirección de los difusores de aire ajustables en la que se deben mover los interruptores para para dirigir el aire, según se requiera.



Tabla 11. Dirección del aire para los difusores ajustables.




Interruptor rotatorio izquierdo	Temperatura
Sentido horario	Flujo de aire más caliente
Sentido antihorario	Flujo de aire más frío
Interruptor rotatorio central	Ventilador
	Ventilador desconectado
1	Revolución mínima
2 y 3	Revoluciones intermedias
4	Revolución máxima

Fuente: (Chevrolet, 2007)

- *Distribución del flujo de aire:* La Tabla 12 hace referencia hacia donde se debe girar el interruptor rotatorio derecho para obtener un flujo de aire en una posición específica.

Tabla 12. Distribución del flujo de aire para diferentes posiciones.

Interruptor rotatorio derecho	Distribución de aire
	<i>Desempañado del parabrisas:</i> El flujo de aire es dirigido hacia el parabrisas.
	<i>Área de los pies y desempañado del parabrisas:</i> Una porción del flujo de aire es dirigida hacia los difusores de aire del parabrisas

	y la otra porción es dirigida hacia los difusores de aire del área de los pies.
	<i>Pies:</i> El flujo de aire es dirigido hacia los difusores de aire, ubicadas en el área de los pies. Ajuste el mando de temperatura según la condición más cómoda.
	<i>Área de los pies y cabeza:</i> Una porción del flujo de aire es dirigida hacia los difusores de aire del área de los pies y la otra porción es dirigida hacia los difusores de aire del área de la cabeza.
	<i>Cabeza:</i> El flujo de aire es dirigido hacia los difusores frontales centrales y laterales

Fuente: (Chevrolet, 2007)

El sistema de acondicionador de aire opera solamente con las gamas de temperatura en que son requeridas. El sistema se desconecta automáticamente cuando la temperatura externa alcanza cerca de +4°C.

La etapa de análisis del entorno del vehículo es parte fundamental para el desarrollo del proyecto debido a que este permite recolectar información necesaria para determinar cuál debe ser la ubicación idónea para el hardware y los captadores de información; además permite detallar que las limitaciones del sistema con respecto al lugar de los sensores y la cámara. A continuación, se muestran las limitaciones que se deben tener presentes en el diseño del proyecto.

- Una de las limitaciones que se presentan es la ubicación de la cámara y el sensor térmico, debido a que se debe considerar que la cámara y el sensor enfoquen el rostro frontalmente, además de que no obstaculice la visibilidad del conductor.

- Las ubicaciones de los sensores de gas deben ser lo más cerca al rostro del conductor para disminuir la sensibilidad de estos y de esta manera solo detectar el gas etanol del conductor y no de los tripulantes del vehículo.

3.3. Requerimientos del sistema

Para el análisis de los requerimientos del sistema se tomó como referencia el estándar ISO/ IEC / IEEE 29148: 2011 el mismo que contiene directrices para el proceso relacionado a la ingeniería de requisitos, específicamente ha sido desarrollado para ser implementado en los sistemas y productos de software y servicios a lo largo del ciclo de vida, ya que permite definir cada función que se requiere en el proyecto, las restricciones necesarias, y especificar los requisitos y funciones del sistema (ISO/IEC/IEEE, 2011).

A continuación, se indican las consideraciones del estándar para el diseño de tablas que contengan la información relevante del proyecto, éstas contienen los requerimientos iniciales del sistema, requerimientos de arquitectura y los requerimientos de stakeholders, la finalidad de la elaboración de tablas es presentar de una manera resumida el contenido y que la información este clara para posteriormente realizar la selección de hardware y software a emplear en el desarrollo del proyecto. La Tabla 13 muestra los acrónimos empleados para referirse de forma abreviada a cada requerimiento analizado.

Tabla 13. Definición de Acrónimos.

Acrónimo	Descripción
SySR	Requerimientos iniciales del Sistema.
SRSR	Requerimientos de Arquitectura.
StSR	Requerimientos del Stakeholder.

Fuente: (Barreto, 2014)

Los criterios que se tuvieron en cuenta para determinar la prioridad de los requerimientos del proyecto fueron la importancia para el usuario, el rendimiento del sistema y el riesgo para el diseño y la implementación del proyecto. La Tabla 14 indica la prioridad que se le asigna a cada requerimiento del sistema.

Tabla 14. Prioridad de los Requerimientos del sistema.

Prioridad	Descripción
Alta	Es un requerimiento crítico que debe incluirse durante el desarrollo del sistema. Si no se implementa puede afectar la funcionalidad.
Media	El no incluir este tipo de requerimiento puede afectar la decisión final del sistema, sin embargo, se puede omitir este requerimiento en condiciones de fuerza mayor.
Baja	Si no se incluye este requerimiento no se espera un impacto significativo en la decisión final del sistema.

Fuente: (Pérez García, 2013)

3.3.1. Requerimientos iniciales del sistema.

En los requerimientos iniciales del sistema (SySR) se definen los límites funcionales del sistema en términos de comportamiento y propiedades del proyecto, para la captura de los requerimientos funcionales se analizaron las necesidades del proyecto, con la finalidad de dividirlos en categorías. A continuación, se describen los requerimientos iniciales según su categoría:

- *Requerimientos de Interfaces* Dentro de los requerimientos comunes de las interfaces del sistema se contemplan ciertos parámetros que definen cómo se requiere que el

sistema interactúe con las interfaces de usuario, hardware y software para contribuir al buen funcionamiento del sistema. Las interfaces de usuario a considerar en el desarrollo del proyecto es la pantalla que el usuario debe manipular para realizar una función específica, en las interfaces de hardware se contemplan parámetros como: la comunicación a través del puerto serial del sistema embebido y los pines de entrada y salida de la información, dentro de las interfaces de software se contempla la compatibilidad del software de programación con las placas reducidas de desarrollo disponibles en el mercado.

- *Requerimientos de performance* Dentro de los requerimientos de performance se consideran parámetros que definen el rendimiento del sistema y en qué condiciones una función debe realizarse. Estos son requisitos cuantitativos verificables individualmente. Los requerimientos de performance a analizar en el proyecto son: la temperatura, reconocimiento facial y concentración de alcohol en el aire.
- *Requerimientos de modo y estados* Dentro de los requerimientos de modo y estados se considera el conjunto de funcionalidades del sistema que se activan o desactivan con la ocurrencia de un evento, envío de una señal y posterior entrada a un modo de operación. Los modos de operación típicos del proyecto son: encendido, apagado que garantiza la integridad temporal del sistema.
- *Requerimientos físicos* Dentro de los requerimientos físicos se contemplan restricciones con respecto a la ubicación de los sensores y la dimensión del hardware a emplear en el proyecto.

En la Tabla 15 se muestran los requerimientos iniciales del sistema (SySR) la prioridad a cada requerimiento se la realizo con relación a la Tabla 14.

Tabla 15. Requerimientos Iniciales del Sistema.

SySR						
REQUERIMIENTOS INICIALES DEL SISTEMA						
#		REQUERIMIENTOS	PRIORIDAD			RELACIÓN
			Alta	Media	Baja	
REQUERIMIENTO DE INTERFAZ						
SySR 1		Pines de entrada y salida de información del sistema embebido.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SySR 2		Comunicación a través del puerto serial del sistema embebido con una placa para disminuir el procesamiento.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SySR 3		El sistema electrónico (Sistema embebido, los sensores, la cámara y la alimentación eléctrica) deberán interactuar entre sí.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
REQUERIMIENTO DE PERFORMANCE						
SySR 4		Reconocimiento facial para impedir la suplantación de identidad	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SySR 5		Medir la temperatura facial del conductor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SySR 6		Censar constantemente los sensores de gas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SySR 7		Establecer el estado del conductor con el algoritmo de aprendizaje automático.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	StRS 4:6
SySR 8		Desbloquear el encendido del vehículo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	StRS 7
REQUERIMIENTO DE MODOS/ESTADOS						
SySR 9		El sistema debe permanecer encendido el tiempo en el que el vehículo funcione	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
REQUERIMIENTO FÍSICOS						
SySR 10		La cámara y el sensor térmico deben ubicarse en un lugar que enfoque directamente el rostro del conductor y que no interfiera con la visibilidad de este.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SySR 11		Los sensores de gas deben ubicarse cerca del conductor y a la vez no hacer contacto con el mismo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SySR 12		Tamaño reducido del sistema embebido.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Fuente: Autoría

3.3.2. Requerimiento de arquitectura.

En los requerimientos de arquitectura (SRSH) se definen los requerimientos de hardware, software y el sistema eléctrico. A continuación, se describen los requerimientos de arquitectura considerados para el diseño del sistema:

- *Requerimientos de Diseño* Dentro de los requerimientos de diseño se considera parámetros de la ubicación de los sensores para la recolección de la información y las modificaciones que se deben realizar en el hardware.
- *Requerimientos de Hardware* Dentro de los requerimientos de hardware se contemplan las especificaciones técnicas de los elementos que se van a emplear como son: la cámara a emplear para la visión artificial, el sensor de gas para medir la concentración de alcohol en el aire, el sensor de temperatura, el sistema embebido que realizará el procesamiento de la información y la ejecución del script que contiene el algoritmo de aprendizaje automático y la placa reducida para disminuir el procesamiento del sistema embebido.
- *Requerimientos de Software* Dentro de los requerimientos de software se considera la compatibilidad con el sistema embebido, las bibliotecas de visión artificial de OpenCV y las librerías de los sensores a utilizar en el desarrollo del proyecto.
- *Requerimientos de Eléctricos* Dentro de los requerimientos eléctricos se contemplan parámetros como: la fuente de alimentación para el sistema embebido y los reguladores de voltaje a emplear.

Las directrices descritas en los requerimientos de Arquitectura se muestran en la Tabla 16 y son indispensables para la selección del hardware y software a emplear en el proyecto.

Tabla 16. Requerimientos de Arquitectura.

SRSH					
REQUERIMIENTO DE ARQUITECTURA					
#	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		Alta	Media	Baja	
REQUERIMIENTOS DE DISEÑO					
SRSH 1	Para sostener la cámara y el sensor térmico debe diseñarse un soporte que no obstaculice la visión del conductor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SRSH 2
SRSH 2	La cámara y el sensor térmico deber situarse en el mismo soporte	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SRSH 1
SRSH 3	El cable del sensor térmico y la cámara deben ser extendidos para conectarse a la placa reducida	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	SRSH 1-2
SRSH 4	Los sensores de gas deben conectarse a otra placa para reducir el procesamiento de la placa principal	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
REQUERIMIENTOS DE HARDWARE					
SRSH 5	Sistema embebido compatible con la librería de OpenCV	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 6	El sistema embebido requiere un procesador que soporte el tratamiento de imágenes en tiempo real	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 7	El sistema embebido requiere más de 2 puestos USB para comunicación serial.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SRSH 8	Sistema embebido con entradas y salidas análogas y/o digitales para manejo de sensores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 9	Se debe considerar el tamaño de la cámara	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 10	Se requiere que una cámara con alta resolución para el reconocimiento facial.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SRSH 11	Se debe considerar que el conector de la cámara sea compatible con el sistema embebido.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 12	Se requiere una cámara para uso en el día y la noche	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 13	Las imágenes de la cámara deben ser procesadas con rapidez en el sistema embebido	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 14	Considera la sensibilidad de los sensores de gas.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SRSH 15	Se requiere bajo consumo de energía del sensor de gas.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 16	En el sensor térmico considerar el rango de temperatura	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SRSH 17	En el sensor térmico considerar la distancia a cuál detecta la temperatura.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

REQUERIMIENTOS DE SOFTWARE

SRSH 18	Sistema Operativo y Lenguaje de programación de código abierto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 19	Compatibilidad con Librería OpenCV y cámara	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SRSH 20	Se requiere software que permita ejecutar el código de visión artificial desde un sistema embebido en tiempo real	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SRSH 21	Compatibilidad de software con la biblioteca para aprendizaje automático	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
REQUERIMIENTOS ELÉCTRICOS				
SRSH 22	Regulador de voltaje de 12V a 5V	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fuente: Autoría

3.3.3. Requerimientos del stakeholder.

Un stakeholder es todo grupo o individuo que tiene un interés directo en el resultado obtenido por el desarrollo del proyecto. La definición de los requerimientos del stakeholder (StSR) tiene como finalidad identificar los requisitos de los interesados para el sistema. En la Tabla 17 se listan a los implicados o stakeholder que se toman en cuenta para el desarrollo de este proyecto:

Tabla 17. Lista de Stakeholders del sistema.

Lista de stakeholders
1. Conductor del vehículo
2. Tripulantes del vehículo
3. Ing. Jaime Michelena Director de Presente Trabajo de Titulación
4. Liceth Portilla Desarrolladora del Proyecto
5. Ing. Sandra Narváez Codirectora de Presente Trabajo de Titulación

Fuente: Autoría

Con el propósito de definir los requisitos de los interesados en el desarrollo del sistema, se analiza un conjunto común de los requerimientos que expresan la interacción que el sistema tendrá con su entorno con los usuarios involucrados. La Tabla 18 contiene los requisitos operacionales y de usuario considerados en los requerimientos del stakeholder.

Tabla 18. Requerimientos del Stakeholder.

StSR					
REQUERIMIENTOS DE STAKEHOLDERS					
#	REQUERIMIENTO DE USO	PRIORIDAD			RELACIÓN
		Alta	Media	Baja	
REQUERIMIENTOS OPERACIONALES					
StSR 1	El sistema debe implementarse en un vehículo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
StSR 2	El sistema debe presentar bajo consumo de energía	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
StSR 3	Adquisición de datos de entrenamiento para construcción del algoritmo de aprendizaje automático	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
REQUERIMIENTOS DE USUARIOS					
StSR 4	El conductor debe sentarse de manera correcta y mantener su visión en el carril del vehículo.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
StSR 5	Para salir del vehículo el conductor debe apagar el sistema.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
StSR 6	El usuario no debe manipular ningún elemento del sistema.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Fuente: Autoría

3.3.4. Selección de hardware y software.

La selección del hardware y software se realiza a través de una tabla comparativa de especificaciones de hardware y software, para ello se obtendrá una valoración a dichas especificaciones y se elegirá las que cumplan con los requisitos establecidos en la Tabla 16, la cual contiene los requerimientos de Arquitectura.

3.3.4.1. Hardware.

La selección del Hardware se la realiza en base a los requerimientos de hardware establecidos en la Tabla 16 sobre Requerimientos de Arquitectura. Se seleccionará el sistema embebido, la placa secundaria para disminuir el procesamiento de la principal, la cámara a emplearse para la visión artificial, el sensor térmico y el sensor de gas.

3.3.4.1.1. Sistema Embebido.

Para la elección del sistema embebido se seleccionó 4 opciones, las cuales se adaptaban a las necesidades del proyecto. El modo de valoración se realizó de la siguiente manera: si cumple con el requerimiento con un valor de 1 y si no cumple con un valor de 0. La Tabla 19 muestra la valoración de cada requerimiento para la elección del sistema embebido.

Tabla 19. Elección del sistema embebido.

HARDWARE	REQUERIMIENTOS				VALORACIÓN
	SRSH 5	SRSH 6	SRSH 7	SRSH 8	TOTAL
Raspberry Pi 3	1	1	1	1	4
Intel Galileo Gen 2	1	1	0	1	3
Banana Pi.	1	1	0	1	3
PC-duino	1	1	0	1	3

1 Cumple
0 No cumple

Elección: El hardware idóneo según los requerimientos del hardware especificado en la tabla de requerimientos de arquitectura es Raspberry Pi 3, se optó por esta placa ya que cumple con todos los requerimientos, además ser una unidad compacta lo que facilita la instalación en el interior del vehículo y del uso de las cámaras desarrolladas para esta placa a través del conector CSI.

Fuente: Autoría y Adaptado de: (Raspberry Pi Foundation)

A continuación, en la Tabla 20 se muestran las Características técnicas principales de la placa Raspberry Pi 3, la información completa de la placa se puede visualizar en la página oficial de raspberry https://www.raspberrypi.org/documentation/hardware/computemodule/RPI-CM-DATASHEET-V1_0.pdf

Tabla 20. Especificaciones técnicas de la placa Raspberry Pi 3.



Raspberry Pi 3 Model B



Chip	Broadcom BCM 2835
CPU	1,2 GHz de 64 bits con cuatro núcleos ARMv8
GPU	VideoCore IV
Memoria del sistema	1 Gb LPDDR2
Almacenamiento	Tarjeta microSD, ranura para un máximo de 32 GB
Salida de video	HDMI, jack 3.5 mm, Conector LVDS de a bordo
Audio I/O	HDMI, 3.5mm jack de salida estéreo
Red	Ethernet 10/100Mbps RJ45 Wifi, 802.11n Wireless LAN
USB	4 puertos USB 2.0
Expansión	UART, PWM, GPIO, I2C, SPI, conector CSI cámara, conector monitor LVDS
Consumo	2,5 A /12,5W /5V
Dimensiones	85 x 56 x 17 mm.
Peso	45 g.
Sistemas operativos	CPU Rasbian, Pidora, Arch Linux. Media center Openelec, Raspbmc

Fuente: (Raspberry Pi Foundation)

3.3.4.1.2. Cámara.

Para la elección de la cámara se seleccionó 3 opciones, las cuales se adaptaban a las necesidades del proyecto. La Tabla 21 muestra la elección de la cámara y el modo de valoración se realizó de igual forma que para el sistema embebido.

Tabla 21. Elección de cámara.

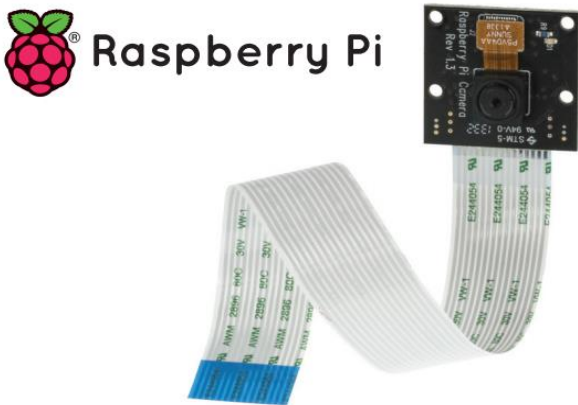
HARDWARE	REQUERIMIENTOS					VALORACIÓN
	SRSH 9	SRSH 10	SRSH 11	SRSH 12	SRSH 13	TOTAL
ELP- USB130W01M T-L21-S	1	1	1	1	0	4
Raspberry Pi camera module v1 (Raspicam)	1	1	1	0	1	4
Raspberry Pi camera module v2 (Pi Noir)	1	1	1	1	1	5
1 Cumple						
0 No cumple						

Elección: La cámara idónea según los requerimientos del hardware especificado en la tabla de requerimientos de arquitectura, se optó por Raspberry Pi camera module v2 (Pi Noir), debido a la compatibilidad que tiene con la placa Raspberry Pi 3 modelo B que fue el sistema embebido seleccionado y el cumplimiento de todos los requerimientos de la Tabla 16.

Fuente: Autoría

A continuación, en la Tabla 22 se muestran las Características técnicas principales de la Raspberry Pi camera module v2 (Pi Noir), la información completa del módulo PiNoir se encuentra en el siguiente enlace: <http://docs-europe.electrocomponents.com/webdocs/127d/0900766b8127db33.pdf>

Tabla 22. Especificaciones técnicas Raspberry Pi camera module v2 (Pi Noir).

	
Sensor de imagen	Sensor de imagen Omnivision 5647 CMOS
Resolución	5 megapíxeles
Resolución de imagen fija	2592 x 1944
Tasa máxima de transferencia de imágenes	1080p: 30fps (encode and decode) 720p: 60fps
Conexión con el Raspberry PI	15 pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2)
Rango de temperatura	Operating: -30° to 70° Stable image: 0° to 50°
Tamaño de la lente	1/4"
Dimensiones	20 x 25 x 10mm
Peso	3g

Fuente: (Europe Electrocomponents)

3.3.4.1.3. Sensor de Gas.

Para la elección del sensor de gas se seleccionó 3 opciones, las cuales se adaptaban a las necesidades del proyecto. La Tabla 23 indica los requerimientos y la valoración para la selección del sensor de gas.

Tabla 23. Elección del sensor de gas.

HARDWARE	REQUERIMIENTOS		VALORACIÓN
	SRSH 14	SRSH 15	TOTAL
MQ-3	1	1	2
MQ303	1	1	2
MR513	1	0	1
1 Cumple			
0 No cumple			
Elección: Se decidió emplear el sensor MQ-3 debido a la disponibilidad de este en el mercado y al cumplimiento de todos los requerimientos de hardware especificados en la Tabla 16.			

Fuente: Autoría

A continuación, en la Tabla 24 se muestran las Características técnicas principales del sensor MQ-3, las especificaciones completas se pueden visualizar en el enlace siguiente <https://www.pololu.com/file/0J310/MQ3.pdf> .

Tabla 24. Especificaciones técnicas del Sensor MQ-3).



Alimentación	5 V. DC
Temperatura de operación	-10 a 70 grados Celsius
Resistencia de Carga	Ajustable
Resistencia de Calentamiento	$31\Omega \pm 3\Omega$ (temperatura Ambiente)
Potencia De Calentamiento	≤ 900 Mw
Sensibilidad	R_s (en el aire) / R_s (0.4 mg/L de alcohol) ≥ 5

Fuente: (HANWEI ELETRONICS CO.,LTD)

3.3.4.1.4. Sensor de Temperatura.

Para la elección del sensor de temperatura se seleccionó 2 opciones, las cuales se adaptaban a los requerimientos del proyecto para adquirir la temperatura facial del conductor sin hacer contacto. A continuación, en la Tabla 25 se muestra la valoración de los requerimientos de hardware se consideran todos los parámetros establecidos en la tabla de Requerimientos de Stakeholders para la elección del sensor de temperatura.

Tabla 25. Elección del sensor de temperatura.

HARDWARE	REQUERIMIENTOS		VALORACIÓN TOTAL
	SRSH 14	SRSH 15	
MLX90614	1	1	2
MLX90621	1	1	2

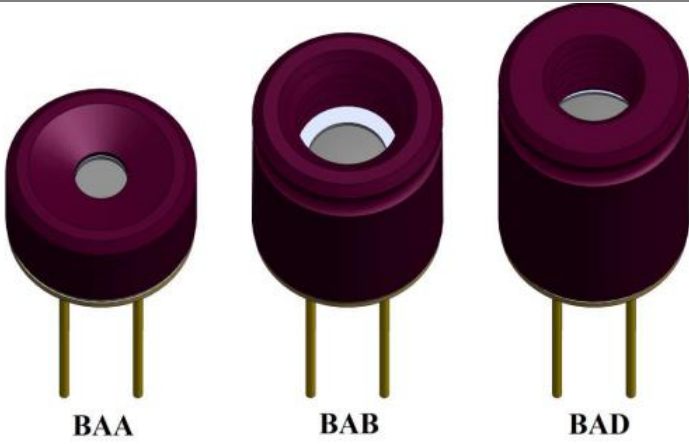
1 Cumple
0 No cumple

Elección: Se decidió optar por el sensor MLX90621 considerando las mejoras en términos de velocidad y resolución de temperatura y el campo de visión en comparación a la versión anterior, además de presenta un arreglo de píxeles de 16 x 4 sensible a la radiación infrarroja térmica y que cumple todos los requerimientos de hardware especificados en la Tabla 15.

Fuente: Autoría

A continuación, en la Tabla 26 se muestran las Características técnicas principales del sensor MLX90614, la información completa de las especificaciones del sensor se pueden obtener en el siguiente enlace: <https://www.melexis.com/en/product/mlx90621/far-infrared-sensor-array-high-speed-low-noise> .

Tabla 26. Especificaciones técnicas del Sensor MLX90621.



Censado de la temperatura	sin contacto físico
Modelo	MLX 90621 BAB
Matriz térmica	16 x 4 píxeles
rango de temperatura ambiente	-40 a 85°C
rango de temperatura del objeto	-20 a 300°C
Campo de visión	30° x 120°, 60° x 15° o 40° x 10° FVO
métodos de salida	PWM y SMBus (TWI, I ² C)
Índice de marco programable	0.5 Hz y 512 Hz

Fuente: (MELEXIS [Melexis Microelectronic Systems])

3.3.4.2. Software.

Una vez seleccionado el hardware se procede a la selección del software, la cual se la realiza en base a los requerimientos de software establecidos en la Tabla 16 sobre Requerimientos de Arquitectura. El software seleccionado debe tener compatibilidad con las bibliotecas de visión artificial OpenCV, las librerías de raspberry pi, las librerías para el sensor térmico y que sea software libre; estas especificaciones se permiten reducir a 2 opciones de software. La Tabla 27 muestra la selección de software con la valoración igual a la del hardware.

Tabla 27. Elección del Software.

HARDWARE	REQUERIMIENTOS			VALORACIÓN
	SRSH 18	SRSH 19	SRSH 20	TOTAL
Python	1	1	1	3
Java	1	1	0	2

1 Cumple
0 No cumple

Elección: El software idóneo según los requerimientos del software especificado en la tabla de requerimientos de arquitectura, se optó por Python ya que este permite la ejecución del código en tiempo real, lo que permite que no existan retrasos considerables al momento de tomar una decisión, además de que cuenta con la biblioteca para aprendizaje automático scikit-learn.

Fuente: Autoría

3.4. Diseño del sistema

Una vez realizado el análisis descrito en el presente capítulo y la recolección de información para la selección de los componentes que integran el sistema, se procede a realizar el diseño con la finalidad de presentar una visión general del funcionamiento del sistema de una forma estructurada; para definir las directrices del diseño del sistema se deben considerar los criterios y limitaciones tomados en cuenta durante el análisis y los requerimientos del sistema que permitirán el desarrollo e implementación del detector de alcoholemia en vehículos.

Como parte del diseño del sistema se muestra a continuación el diagrama de bloques, diagrama circuital y diagrama de flujo que guiará el funcionamiento y los procesos para poder

desarrollar adecuadamente los scripts y el algoritmo de aprendizaje automático, para posteriormente realizar la fase de prueba unitaria para realizar la verificación independiente de cada parte del proyecto, con la finalidad de realizar una adecuada implementación.

3.4.1. Diagrama de bloques general del sistema.

EL proceso de funcionamiento del sistema se presenta mediante un diagrama de bloque, que permite observar la estructura física del sistema, los elementos que lo componen destacando el tipo de Hardware y cómo este se conecta entre sí, además de la ubicación dentro del espacio físico. A continuación, en la Figura 26 se puede apreciar el diagrama de bloques general el cual se divide en 2 etapas: La primera etapa formada por la de adquisición de datos, procesamiento, almacenamiento de la información y la construcción del algoritmo de aprendizaje automático y la segunda etapa constituida de la adquisición de datos, uso del algoritmo de aprendizaje automático, captura del conductor para realizar el reconocimiento facial o almacenar al conductor que se encuentra en estado étílico para proceder a bloquear o desbloquear el encendido del vehículo y accionamiento del botón de pánico según sea el caso.

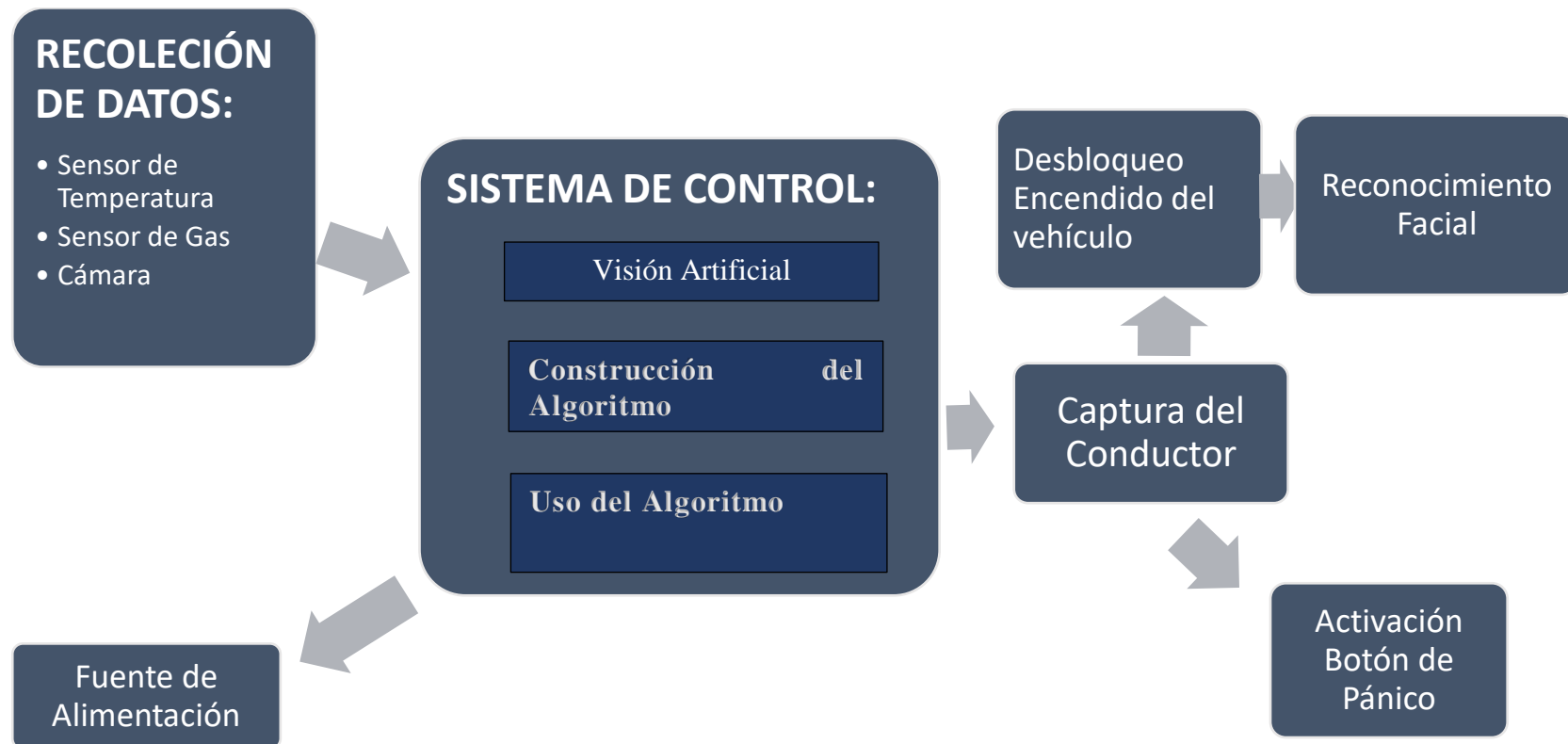


Figura 26. Diagrama de Bloques General.

Fuente: Autoría

3.4.1.1. Diagrama de bloques de la primera etapa.

El diagrama de bloques que se presenta a continuación es el que comprende la primera fase, desarrollada para la construcción del algoritmo de aprendizaje automático para la detección de alcoholemia, la primera etapa está formado por 4 bloques los cuales a su vez contienen varios subprocesos; se han planteado los bloques dependiendo de las funciones que cada uno debe desarrollar. En la Figura 27 se puede observar el diagrama de bloques de la primera fase con cada uno de los subprocesos.

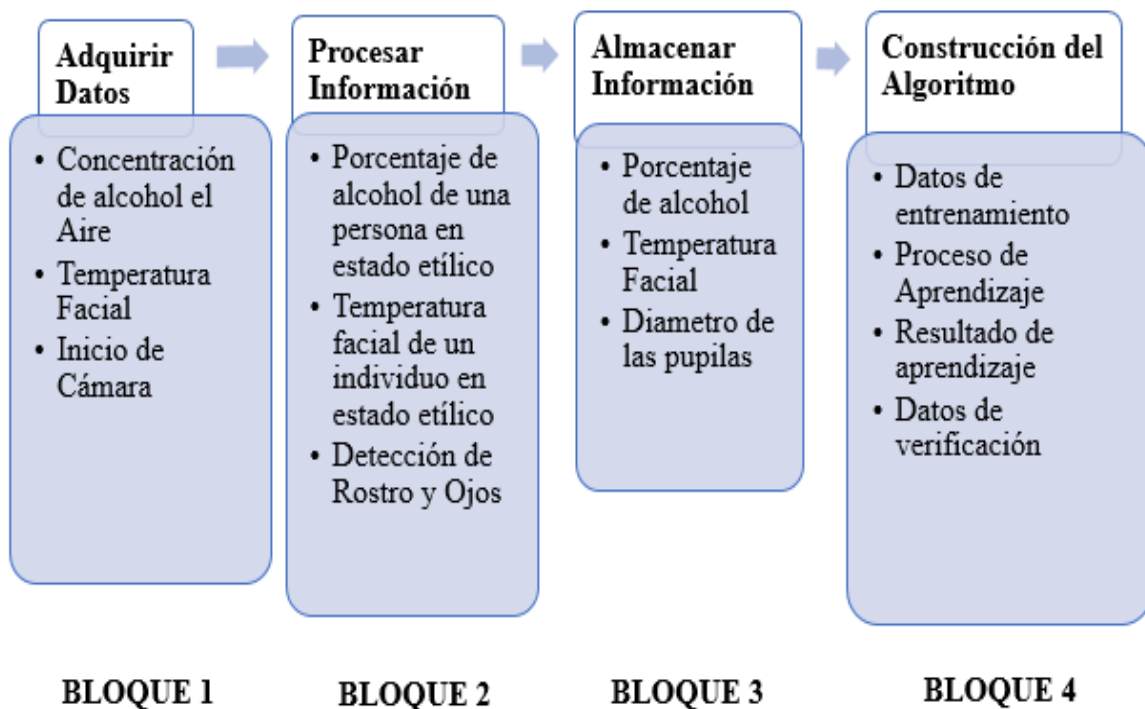


Figura 27. Diagrama de Bloques Primera Etapa.

Fuente: Autoría

La primera fase del proyecto comienza con la adquisición de datos en el bloque 1, este es el encargado de censar la concentración de alcohol en el aire a través del sensor MQ3, censar la temperatura del objeto de estudio y ambiental mediante el sensor MLX90621 y del inicio de

la cámara para adquirir las imágenes, para ello debe desarrollarse un script en la plataforma de Arduino la cual se encargara de adquirir los datos de los sensores de gas y temperatura y un script en Python para arrancar la cámara en el sistema embebido Raspberry Pi 3.

En el bloque 2 se inicia el procesamiento de la información, en el cual se procede a clasificar el porcentaje de etanol en el aire y la temperatura facial de un individuo en estado etílico en la plataforma Arduino, además de iniciar el trabajo de OpenCV, en el cual se procede a detectar el rostro y los ojos del conductor en donde se realizará el análisis de los cambios faciales que presenta un individuo en estado de ebriedad.

El siguiente bloque se encargará de recibir los datos clasificados en el bloque anterior para iniciar el almacenar la información en un archivo .csv el cual se lo desarrolla en el sistema embebido y está formado por el porcentaje de alcohol, la temperatura facial y el diámetro de la pupila del conductor.

Finalmente, se procede a la construcción del algoritmo de aprendizaje automático en el bloque 4, el cual hace uso del archivo creado en el bloque 3 que representan los datos de entrenamiento, posteriormente se realiza el proceso de aprendizaje mediante el lenguaje de Python y la biblioteca para aprendizaje automático scikit-learn, a continuación se procede a entrenar un modelo y mostrar los resultados del aprendizaje, para finalizar se realiza la verificación de los datos a través de una matriz de confusión en corpus de entrenamiento.

3.4.1.2. Diagrama de bloques de la segunda etapa.

El diagrama de bloques de la segunda etapa del sistema se presenta con la finalidad de conocer el funcionamiento del detector de alcoholemia. La Figura 28 muestra el diagrama de bloques formado por 4 bloques con funciones que realiza cada uno.

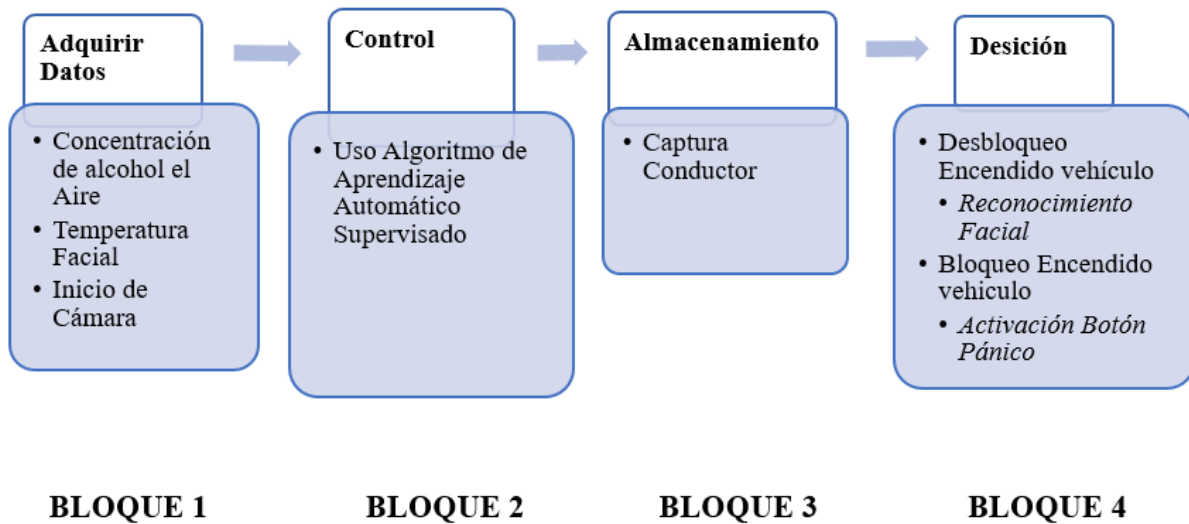


Figura 28. Diagrama de Bloques Segunda Etapa.

Fuente: Autoría

El bloque 1 de la segunda etapa comprende la adquisición de datos y se realiza de igual manera que la primera etapa, las funciones a realizar en este bloque medir la concentración de alcohol en el aire a través del sensor MQ3, medir la temperatura del conductor y ambiental mediante el sensor MLX90621; además de aplicar técnicas de visión artificial para analizar los cambios faciales del conductor.

En el bloque 2 comprende la etapa de control en la cual se aplica el algoritmo de aprendizaje automático, construido en la primera etapa del proyecto, la función que realiza este bloque es determinar el estado del conductor.

El siguiente bloque se encargará de almacenar la información del conductor designado, se procede a guardar el rostro del conductor.

Finalmente, en el bloque de decisión se realiza la función de bloquear o desbloquear el encendido del vehículo, para posteriormente ejecutar el reconocimiento facial si se ha determinado que el conductor es apto para conducir o se procede a activar el botón de pánico si el individuo se encuentra en estado etílico.

3.4.2. Diagrama de conexión del sistema.

Para el desarrollo del proyecto se emplean diversos dispositivos, por lo que es necesario realizar un diagrama de conexión que muestre los pines que se van a emplear en el sistema; los dispositivos de los cuales se realizarán los diagramas son: El sistema embebido Raspberry Pi 3, la placa reducida Arduino Uno R3, el sensor MQ-3 y el sensor MLX90621.

3.4.2.1. Diagrama de Pines y Puertos - Raspberry Pi 3.

Del sistema embebido Raspberry Pi 3 se emplean los pines GPIO para conectar la pantalla Lcd, mediante la cual se visualizan mensajes del funcionamiento del alcoholímetro, el puerto CSI para conectar la cámara, también se hace uso de un puerto USB para la comunicación serial con la placa Arduino, los tres puertos USB restantes se los emplea para la navegación en la interfaz del sistema embebido como son mouse y teclado; además de utilizar los pines GND para conectar las tierras en común con la placa Arduino UNO. En la Figura 29 se muestra la distribución de pines y puertos del sistema embebido Raspberry Pi 3.

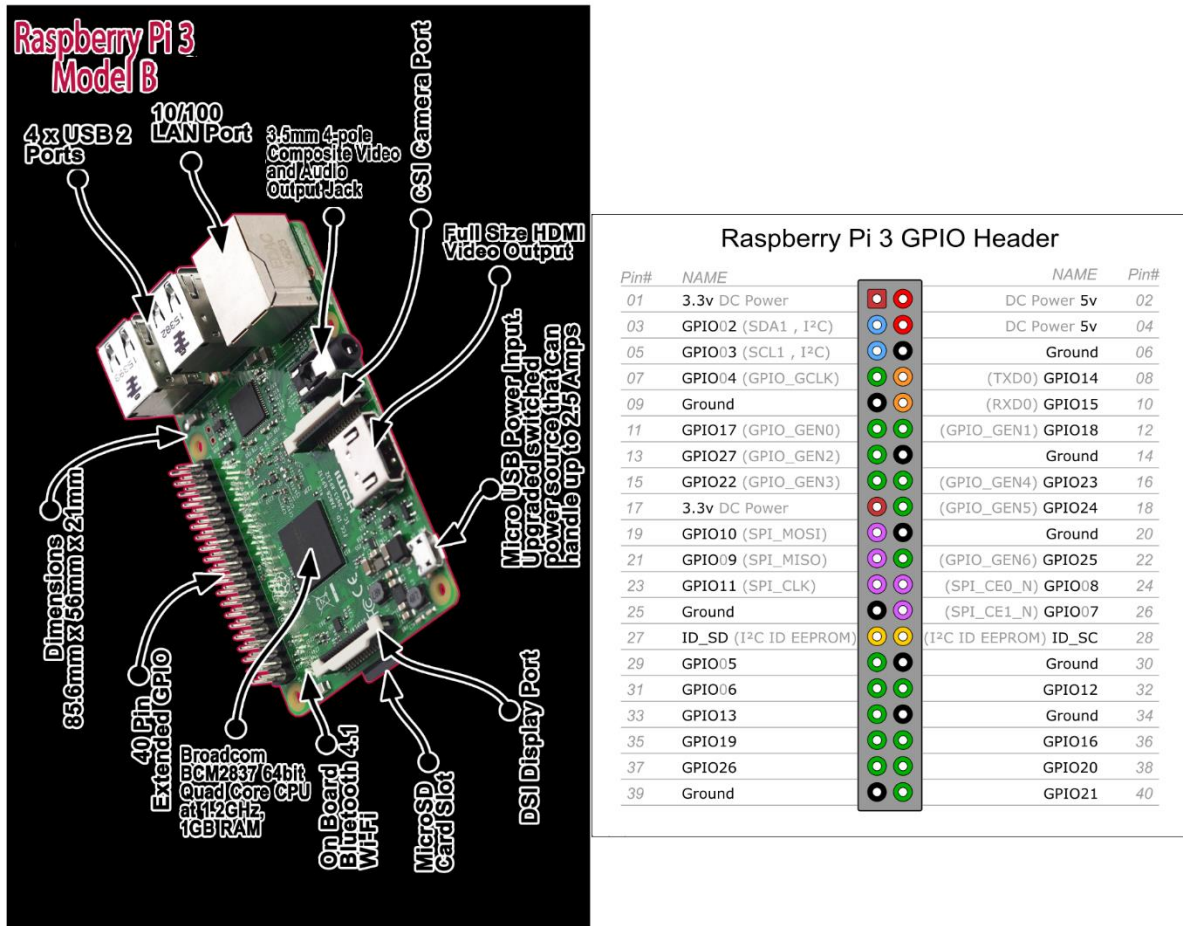


Figura 29. Pines y Puertos del Sistema Embebido Raspberry Pi 3.

Fuente: Adaptado de (Element I4 Community, 2016)

3.4.2.2. Diagrama de Pines y Puertos – Arduino UNO R3.

De la placa reducida Arduino UNO R3 se emplean las entradas analógicas conectar el sensor MQ-3 con el cual se mide la concentración de alcohol en el aire, los pines gnd para tierra común y los pines scl y sda para la comunicación I2C con el sensor de temperatura MLX9021. A continuación, en la Figura 30 se muestra la distribución de pines de la placa Arduino UNO R3.

Arduino Uno R3 Pinout

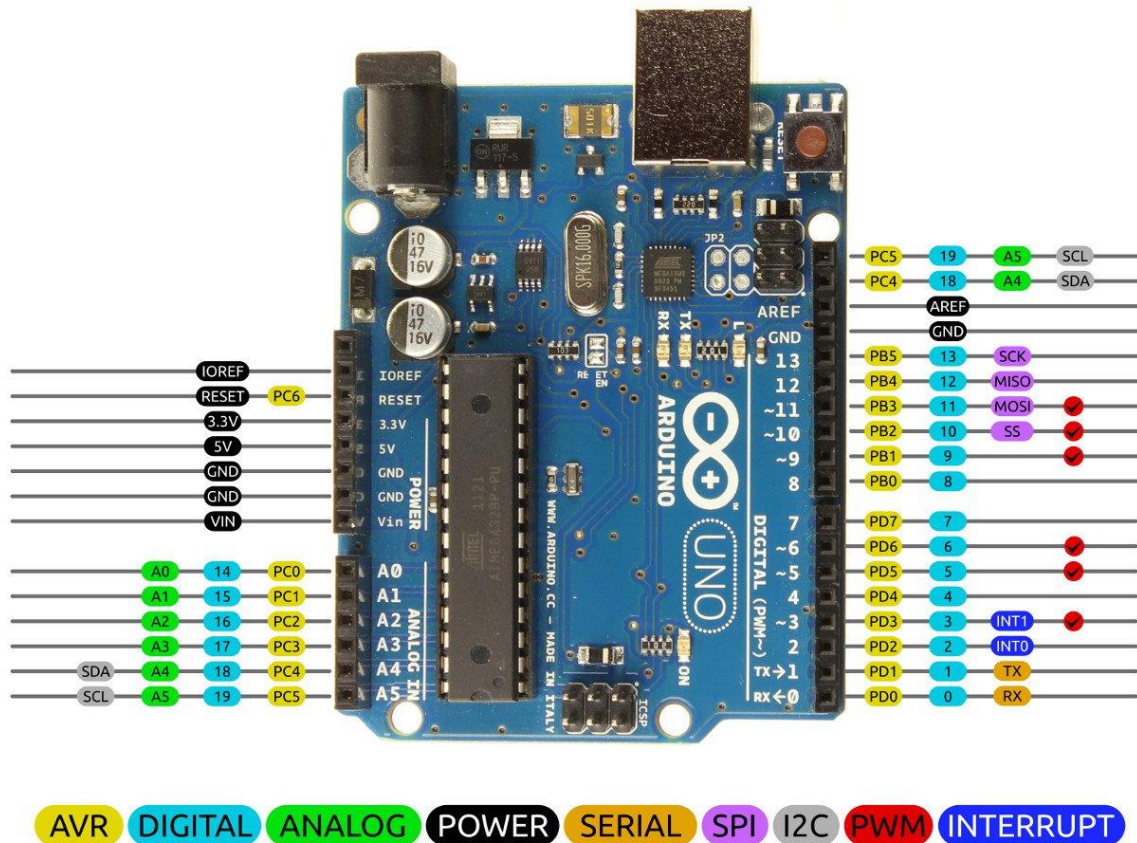


Figura 30. Distribución de Pines Arduino UNO R3.

Fuente: Adaptado de (Tawil, 2016)

3.4.2.3. Diagrama de Pines – Módulo Sensor MQ-3.

Para el desarrollo del proyecto se emplear el módulo del sensor MQ-3, ya que facilita la conexión a la placa reducida, este módulo realiza la función de determinar la concentración de alcohol en el aire, la alimentación se realiza a través de la placa Arduino UNO. En la Figura 31 se muestra la distribución de pines del sensor.

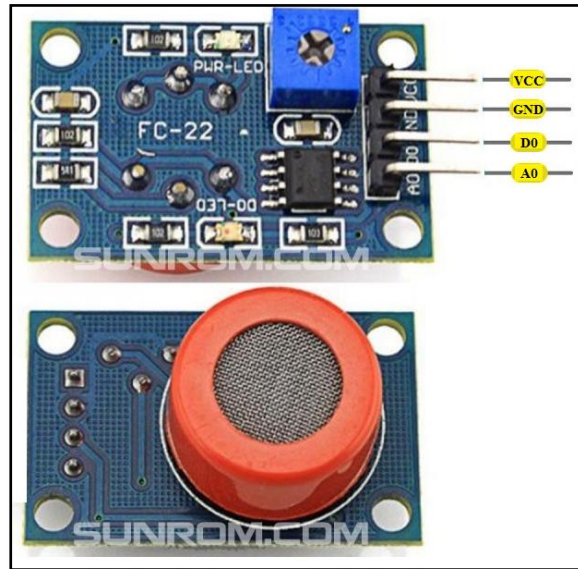


Figura 31. Distribución de Pines Modulo MQ-3.

Fuente: Adaptado de (SOUNROM Electronics/Techonologies, s.f.)

3.4.2.4. Diagrama de Pines – Módulo Sensor MLX90621.

El diagrama de pines del sensor MLX90621 que se muestra en la Figura 32 se emplea para determinar a qué pines del Arduino UNO R3 se conecta; la función que realiza este módulo es medir la temperatura ambiente y del conductor dentro del vehículo.



Figura 32. Distribución de Pines Modulo MLX90621.

Fuente: Adaptado de (HIFI, 2016)

3.4.3. Diagrama de circuitual del sistema.

En esta sección se realiza el diseño de la fuente de alimentación, seguido del diagrama de circuito de esta. Para realizar el diseño de la fuente se considera el voltaje y la corriente que necesita el sistema embebido el cual se encargara de alimentar los demás elementos del sistema. El raspberry Pi 3 requiere de una corriente de 2.5 A y un voltaje de 5 V, para un desempeño eficiente. En la Figura 32 se muestra el diagrama circuitual de la fuente de alimentación del sistema que emplea el Regulador DC-DC Step Down LM2596, ya que permite conducir una corriente de hasta 3A.

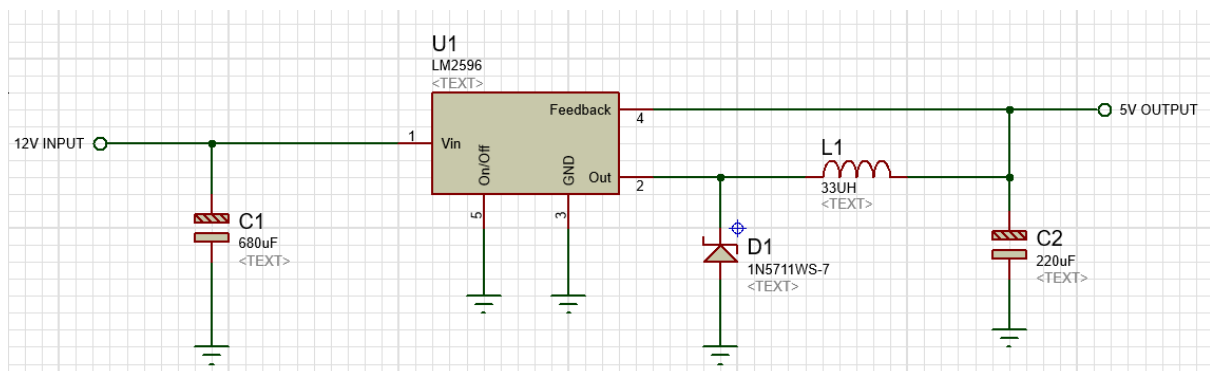


Figura 33. Diagrama Circuitual Fuente de Alimentación.

Fuente: Autoría.

Debido a que en el mercado existe un reductor de voltaje de 12 VDC a 5VDC con una corriente de 2.5A comercial, no se realizará la implementación.

A continuación, en la Figura 34 se presenta el diagrama circuitual del detector de alcoholemia, para el funcionamiento del sistema se requiere el sistema embebido Raspberry Pi 3, el Raspberry camera module v2, la placa Arduino UNO R3, el sensor MQ-3, el sensor MLX90621 y la fuente de alimentación micro USB.

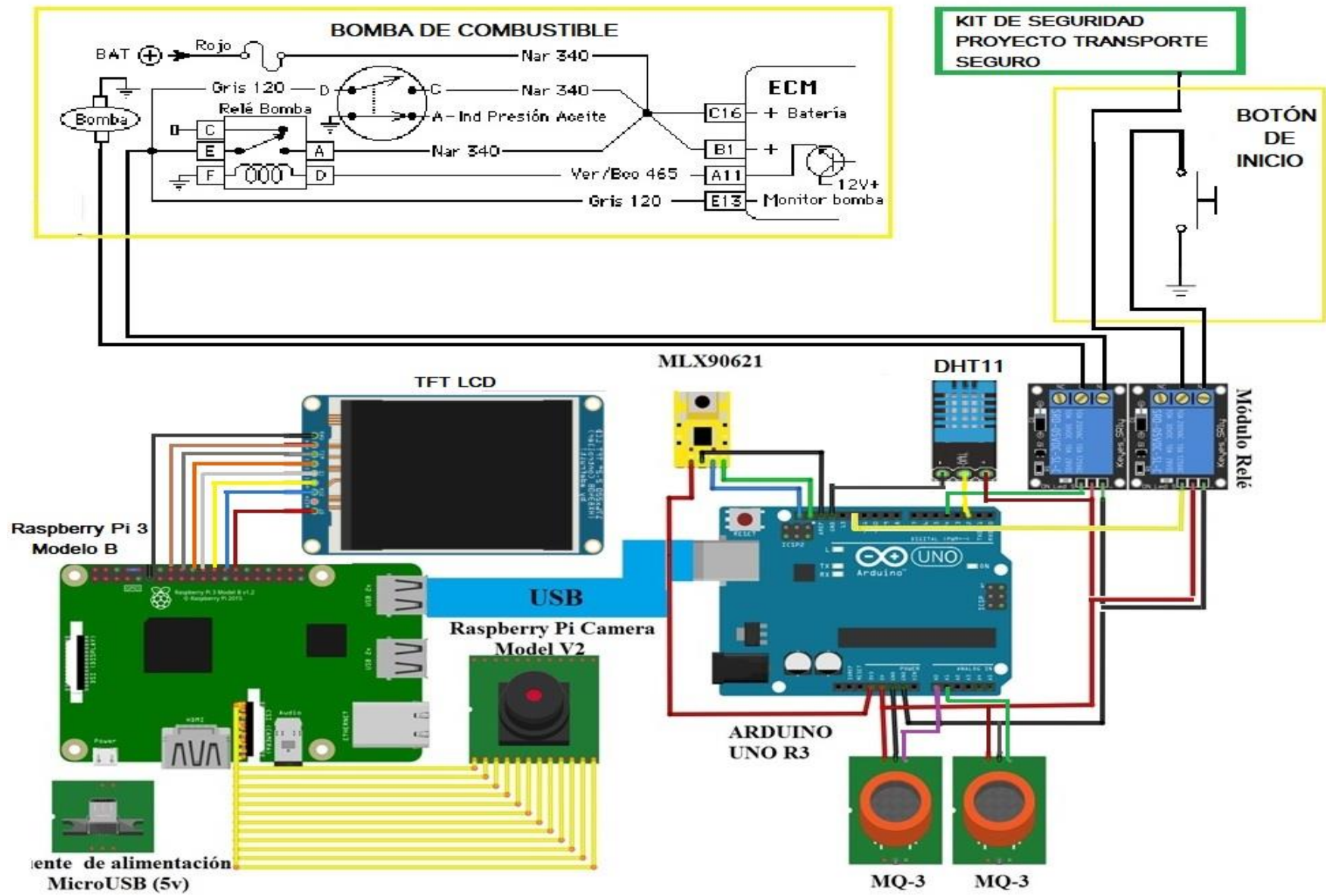


Figura 34. Diagrama Circuital del Detector de Alcholemla.
Fuente: Autoría.

3.4.4. Diagrama de flujo del sistema.

Inicialmente se realiza el análisis de la lectura que debe hacer cada sensor, con la finalidad de obtener las ecuaciones que se emplearan para establecer la concentración de alcohol en el aire, la temperatura ambiental y la temperatura vista por el sensor para posteriormente elaborar el diagrama de flujo del sistema.

- **Lectura Sensor MQ-3**

El sensor MQ-3 cumple la función de leer la concentración de alcohol en el aire y el script que se programa que mide el valor de la salida analógica del sensor para posteriormente realizar una relación entre el valor del sensor y el nivel de alcohol.

Para el desarrollo del proyecto se necesita conocer el valor del sensor en unidades correspondientes a la medición del gas y escalar el valor leído, el inconveniente que se presenta es que la relación entre la lectura analógica del sensor y el valor real no es lineal. Por lo que se procede a estimar la curva que se muestra en la Figura 35 de las características de sensibilidad que proporciona la hoja de especificaciones. Debido a que como datos se obtiene la curva y no una ecuación, es necesario estimar y por regresión encontrar la ecuación.

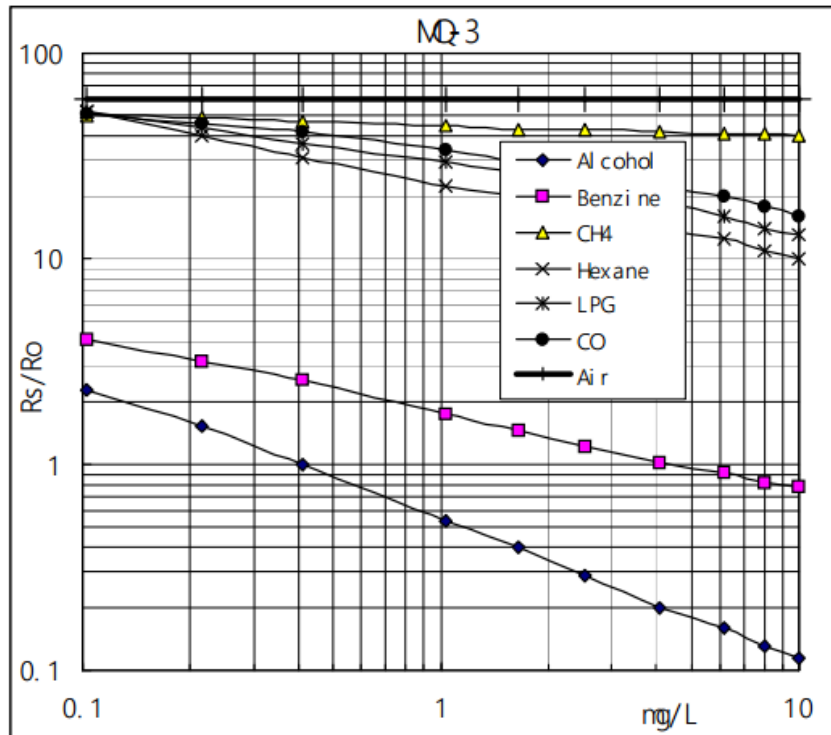


Figura 35. Características de sensibilidad del sensor MQ-3.

Fuente: (HANWEI ELETRONICS CO.,LTD)

A continuación, se indica el proceso que se realizó para obtener la ecuación de regresión potencial a través de Microsoft Excel que es una aplicación de hojas de cálculo desarrollada por Microsoft Corporation. En la Tabla 28 se indican las coordenadas de los puntos que se obtuvieron a partir de la curva de las características de sensibilidad de la Figura 35, en el eje x se encuentran los valores de la resistencia eléctrica (R_s/R_o) y en el eje y la cantidad de alcohol en el aire espirado en unidades de (mg/L).

Tabla 28. Tabla de valores de la curva de las características de sensibilidad del sensor MQ3.

(R_s/R_o)	Alcohol (mg/L)
2,23	0,1
1,7	0,2
1,65	0,25
1,28	0,3
1	0,4
0,9	0,5

0,8	0,6
0,7	0,7
0,65	0,8
0,6	0,9
0,55	1
0,4	1,75
0,35	2
0,3	2,6
0,25	3
0,2	4
0,18	5
0,17	6
0,15	7
0,132	8
0,12	9
0,1	10

Fuente: Autoría.

En la Figura 36 se muestra el gráfico de dispersión para mostrar la relación del conjunto de valores de la Tabla 28.

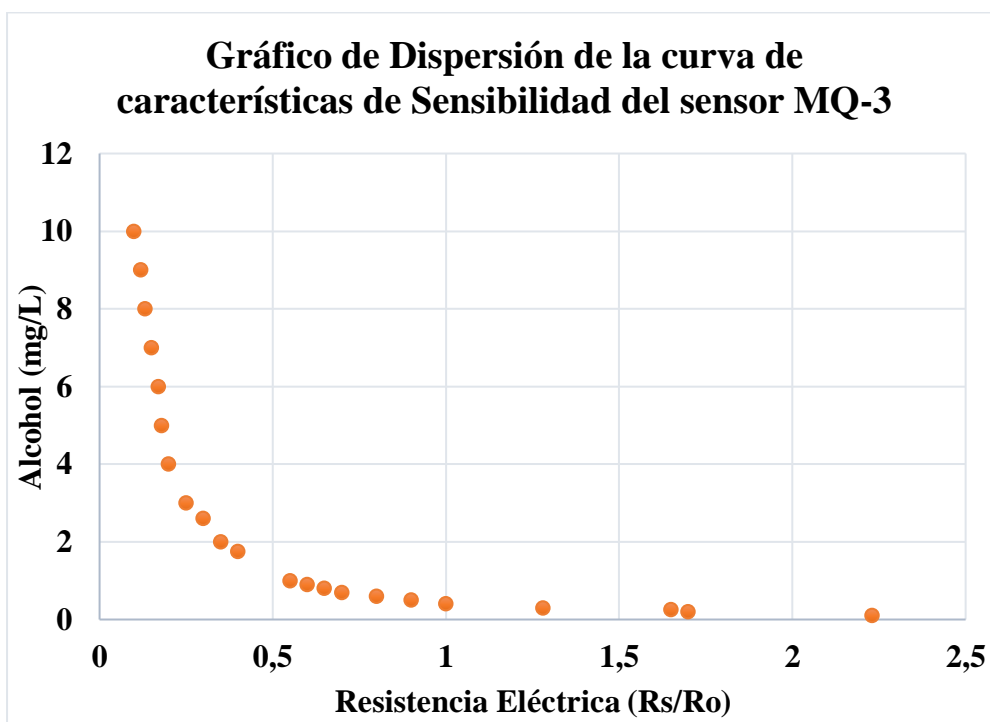


Figura 36. Gráfico de Dispersión de la curva de características de Sensibilidad del sensor MQ-3.

Fuente: Autoría.

Se procede a agregar la línea de tendencia de tipo potencial al gráfico de dispersión de la Figura 36 para obtener la ecuación. La ecuación de una línea de tendencia de potencia calcula el ajuste de mínimos cuadrados a través de los puntos. La Figura 37 indica el gráfico de dispersión con la línea de tendencia potencial y la ecuación.

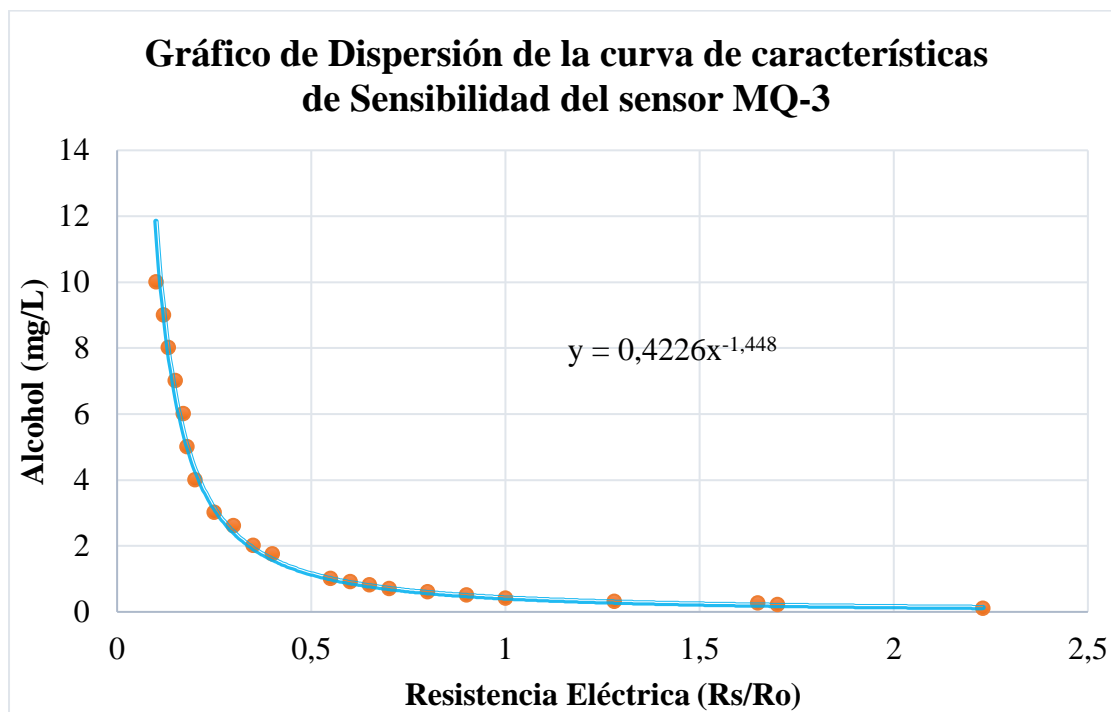


Figura 37. Línea de tendencia potencial y ecuación del gráfico resultante de la curva de características de Sensibilidad del sensor MQ-3.

Fuente: Autoría.

La variable y de la Ecuación que se indica en la Figura 37 representa el valor de concentración de alcohol en el aire, mientras que la variable x representa la relación de la resistencia eléctrica del sensor (R_s/R_o). A continuación, en la Ecuación 2 se indica la concentración de alcohol en el aire.

$$Alcohol = 0,4226 \left(\frac{R_s}{R_o} \right)^{-1,448}$$

Ecuación 2. Ecuación de la concentración de alcohol en función de la resistencia eléctrica.

Fuente: Autoría.

Donde:

Alcohol = Concentración de alcohol en el aire.

R_o = Constante que equivale al valor de la resistencia del sensor cuando se lo expone a una concentración de 0.4mg/L.

R_s = Resistencia del sensor (salida de analógica).

En el gráfico de la tendencia lineal potencial se puede apreciar que el valor de la resistencia eléctrica (R_s/R_o) se acerca a 0.1 (mg/L); de la cual se puede deducir que para valores superiores a la máxima concentración de alcohol que el sensor puede censar y en los puntos cercanos a este, la variación de la resistencia eléctrica es mínima lo que disminuye el porcentaje de error. A continuación, asumiendo que el sensor se satura con 10mg/L el valor de la resistencia eléctrica correspondiente es de 0.12 este valor se puede observar en la gráfica de las curvas características de la sensibilidad. Para simular esta situación se requiere generar un ambiente con bastante concentración de alcohol, para posteriormente realizar la medición de la salida analógica del sensor, como resultado se obtiene un voltaje de 3.02, que equivale a un $R_s=655$. A través de la Ecuación 3 se obtiene el valor aproximado de R_o .

$$\left(\frac{R_s}{R_o}\right) = 0,12$$

Ecuación 3. Ecuación para obtener el valor aproximado de R_o .

Fuente: Autoría.

$$R_o = \left(\frac{R_s}{0,12}\right)$$

$$R_o = \left(\frac{655}{0,12}\right)$$

$$R_o = 5458,33 \approx 5458$$

Según el cálculo aproximado de la constante que equivale al valor de la resistencia del sensor cuando se lo expone a una concentración de 0.4mg/L se obtiene un Ro de 5458. Estos datos se emplearán cuando se procesa a desarrollar el programa en la plataforma Arduino.

- **Lectura Sensor de Temperatura MLX90621**

Para realizar la lectura del sensor de temperatura, en el cual se desarrolla el programa que se comunica a través del bus de comunicación I2C del sensor. Las ecuaciones que se emplean para la lectura del sensor MLX90621 se obtienen de la hoja de especificaciones del sensor.

Para realizar el cálculo de la temperatura absoluta del chip T_a (temperatura del sensor). Se considera la señal de salida del sensor IR la cual es relativa a la temperatura de unión fría. Por lo que es necesario conocer la temperatura de la matriz para poder calcular la temperatura del objeto 'vista' por cada píxel.

- **Temperatura Ambiente del Sensor (T_a)**

La T_a se puede calcular usando la fórmula de la Ecuación 4:

$$T_a = \frac{-K_{T1} + \sqrt{K_{T1}^2 - 4K_{T2}[V_{TH}(25) - PTAT_{data}]}}{2K_{T2}} + 25, \quad [^{\circ}C]$$

Ecuación 4. Ecuación de la Temperatura Ambiente del Sensor.

Fuente: (Melexis INSPIRED ENGINEERING, 2016).

Donde:

T_a

Temperatura Absoluta del chip

$V_{TH}(25)$, K_{T1} y K_{T2}	Constantes almacenadas en la EEPROM en las direcciones que se muestran en la Tabla 29 como dos valores de complemento.
PTAT_data	Dato de entrada

Tabla 29. Tabla de las constantes almacenadas en la EEPROM.

Dirección EEPROM	Nombre de celda	Almacenado como	Parámetro
0xDA	VTH_L	Complemento de 2	V_{TH0} de la Temperatura absoluta del sensor
0xDB	VTH_H	Complemento de 2	
0xDC	K_{T1_L}	Complemento de 2	K_{T1} de la Temperatura absoluta del sensor
0xDD	K_{T1_H}	Complemento de 2	
0xDE	K_{T2_L}	Complemento de 2	K_{T2} de la Temperatura absoluta del sensor
0xDF	K_{T2_H}	Complemento de 2	
0xD2	K_T_scale	No asignado	[7:4] – K_{T1_scale} [3:0] – K_{T2_scale}

Fuente: (Melexis INSPIRED ENGINEERING, 2016)

- **Temperatura Vista por píxeles específicos en la matriz (To)**

La T_o se puede calcular usando la fórmula de la Ecuación 5:

$$T_{o(i,j)} = \sqrt[4]{\frac{V_{IR(i,j)compensado}}{\alpha_{comp(i,j)} * (1 - K_{S4} * 273,15) + S_{x(i,j)}}} + T_{aK^4} - 263,15, \quad [^{\circ}C]$$

Ecuación 5. Ecuación de la Temperatura vista del Sensor.

Fuente: (Melexis INSPIRED ENGINEERING, 2016)

Donde:

$V_{IR(i,j)compensado}$

Señal IR parasítica libre compensada

$\alpha_{comp(i,j)}$	Es el coeficiente de sensibilidad compensado para cada píxel.
K_{s4}	Es el factor de compensación para la sensibilidad, para BAB y BAD $K_{s4} = 0$
$T_{aK^4} = (T_a + 273.15)^2$	Donde T_a es la temperatura ambiente calculada en la ecuación 4.

$$S_{x(i,j)} = K_{s4} \sqrt[4]{\alpha_{comp(i,j)}^3 * V_{IR(i,j)compensado} + \alpha_{comp(i,j)}^4 * T_{aK^4}}$$

El diagrama de flujo desarrollado para el detector de alcoholemia representa los procesos que se realiza en la programación de scripts, se especifican las condicionantes de las variables y las funciones a emplear; de manera que permite ejecutar el programa y elaborar correcciones. En la Figura 38 se presenta el diagrama de flujo en el cual se representa el código de compilación para el sistema:

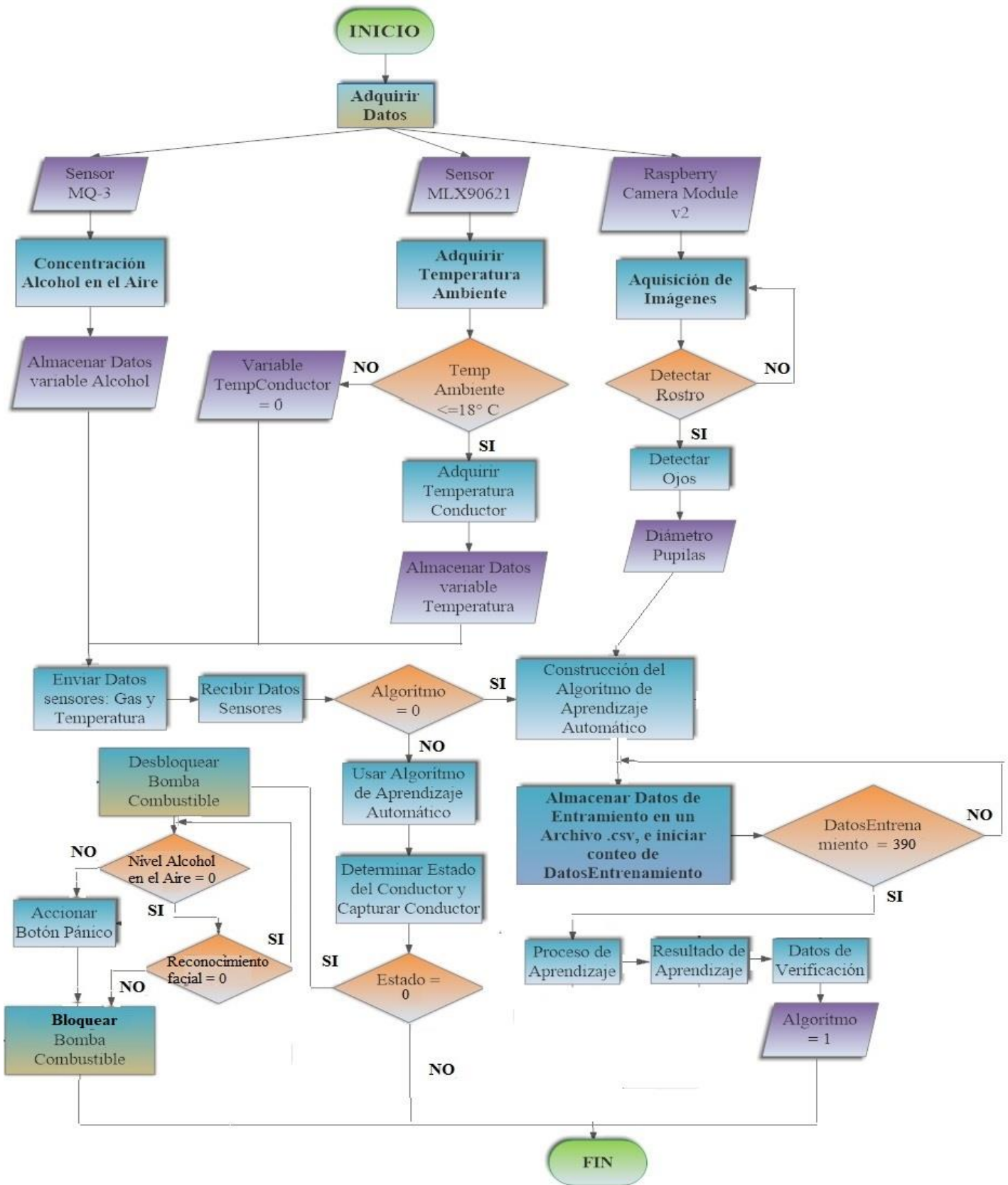


Figura 38. Diagrama de Flujo del Sistema.

Fuente: Autoría.

3.4.5. Desarrollo de software (programación).

El desarrollo del software de programación permite verificar cada parte del proyecto en la fase de prueba unitaria. Durante el desarrollo de software se realiza la programación en la placa reducida Arduino UNO R3 para la lectura de los sensores de gas y temperatura y en el sistema embebido Raspberry Pi 3 el cual se encarga de procesar la información de visión artificial y la construcción del algoritmo de Aprendizaje Automático. Para la placa Arduino se emplea el lenguaje de programación C++ con una adaptación que proviene de avr-libc que provee de una librería de C de alta calidad para usar con GCC en los microcontroladores AVR de Atmel y muchas funciones específicas para los MCU AVR de Atmel (Arduino.cc, s.f.) y para el sistema embebido se usa el lenguaje de programación de Python, donde se crea un archivo ejecutable para cada función del sistema; se debe considerar que el desarrollo el software del sistema se realiza según los requerimientos especificados anteriormente.

3.4.5.1. Lectura del Sensor MQ-3.

Para la lectura del Sensor MQ-3 se procede a leer la salida analógica del sensor, posteriormente se convierte la lectura en un valor de voltaje, a continuación, se calcula la resistencia del sensor (R_s) y finalmente se emplea la ecuación de la curva de características de sensibilidad establecida en la Ecuación 2. La Figura 39 muestra el comando para el cálculo de la concentración de alcohol en el aire en el lenguaje de programación de la plataforma Arduino.

```
double alcohol=0.5491*pow(Rs/5463, -0.682); // calculamos la concentración de alcohol.
```

Figura 39. Calculo de la Concentración de alcohol en el IDE de Arduino.

Fuente: Autoría.

A continuación, en la Figura 40 se observa la respuesta del sensor de MQ-3 ante la presencia y ausencia de alcohol. El código de programación para la lectura del sensor de gas se encuentra en el Anexo 1.

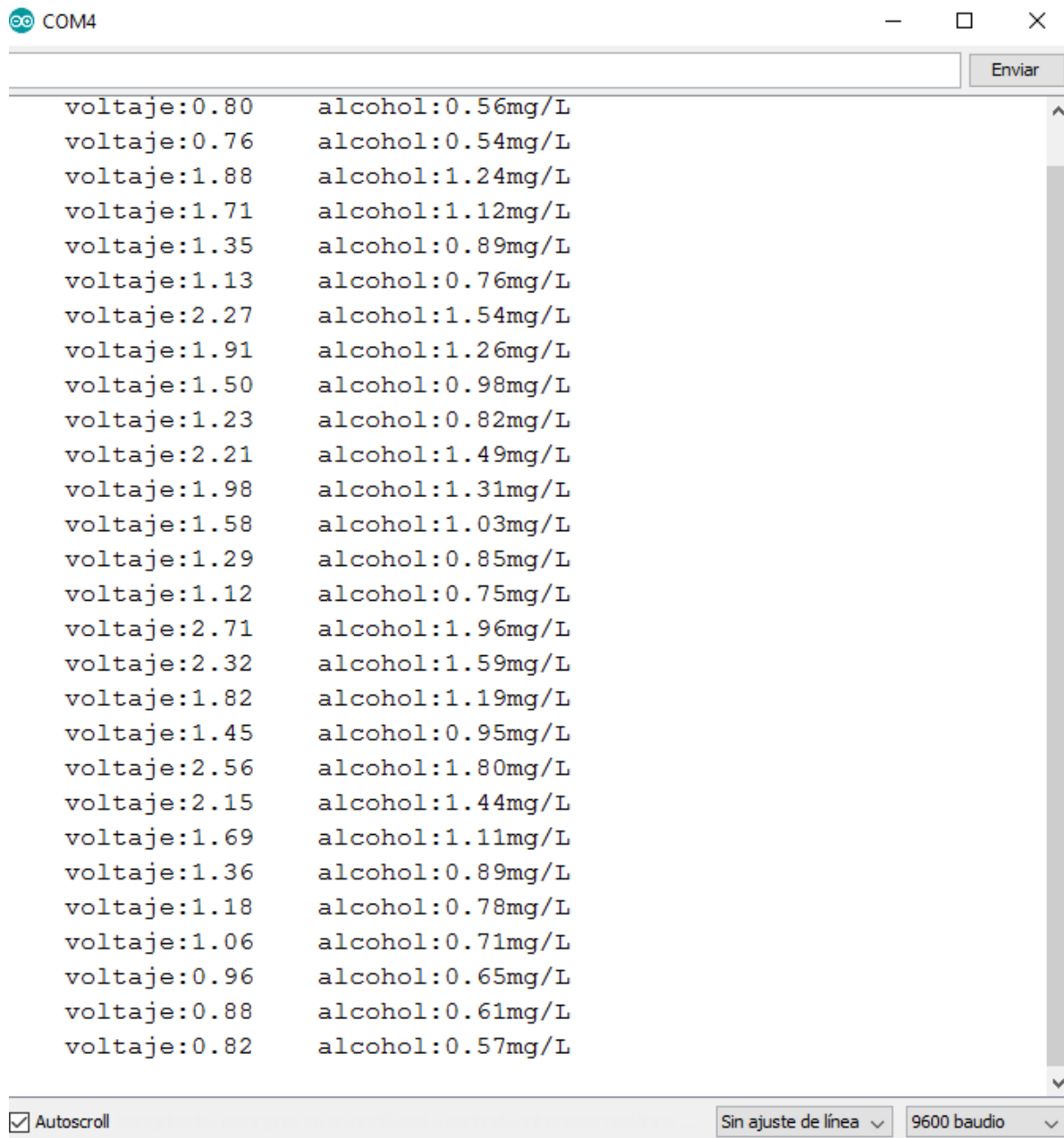


Figura 40. Respuesta del Sensor MQ-3 ante la presencia y ausencia de Alcohol.

Fuente: Autoría.

Al finalizar el desarrollo del script de programación que se encargara de adquirir el nivel de alcohol a través del sensor MQ-3 se verifica que sin la presencia de alcohol el sensor marca valores entre 0,2 y 0,30, esto se debe a que el sensor MQ-3 no solo censa la concentración de etanol sino la presencia de todos los gases; por lo que se establece como condición para determinar que el conductor se encuentra en estado de ebriedad a quienes superen este nivel.

3.4.5.2. Lectura del Sensor MLX90621.

El Sensor MLX90621 para realizar la lectura en el Arduino UNO se requiere incluir las librerías que se muestran en la Figura 41.



```
readMLX90621TemperaturesWire Arduino 1.8.2
Archivo Editar Programa Herramientas Ayuda

readMLX90621TemperaturesWire $ MLX90621.cpp wireMLX90621.h

#include <SPI.h>
#include <Arduino.h>
#include <Wire.h>
#include "wireMLX90621.h"
```

Figura 41. Librerías para el sensor MLX90621.

Fuente: Autoría.

Los pines de comunicación entre el sensor MLX90621 y la placa Arduino son los pines SDA y SCL mediante el bus de comunicación I2C.

A continuación, en la Figura 42 se observa la respuesta del sensor de MLX90621 de la Temperatura Ambiente, mientras que las líneas de código empleadas para la lectura de la temperatura ambiente se encuentran en el Anexo 1.

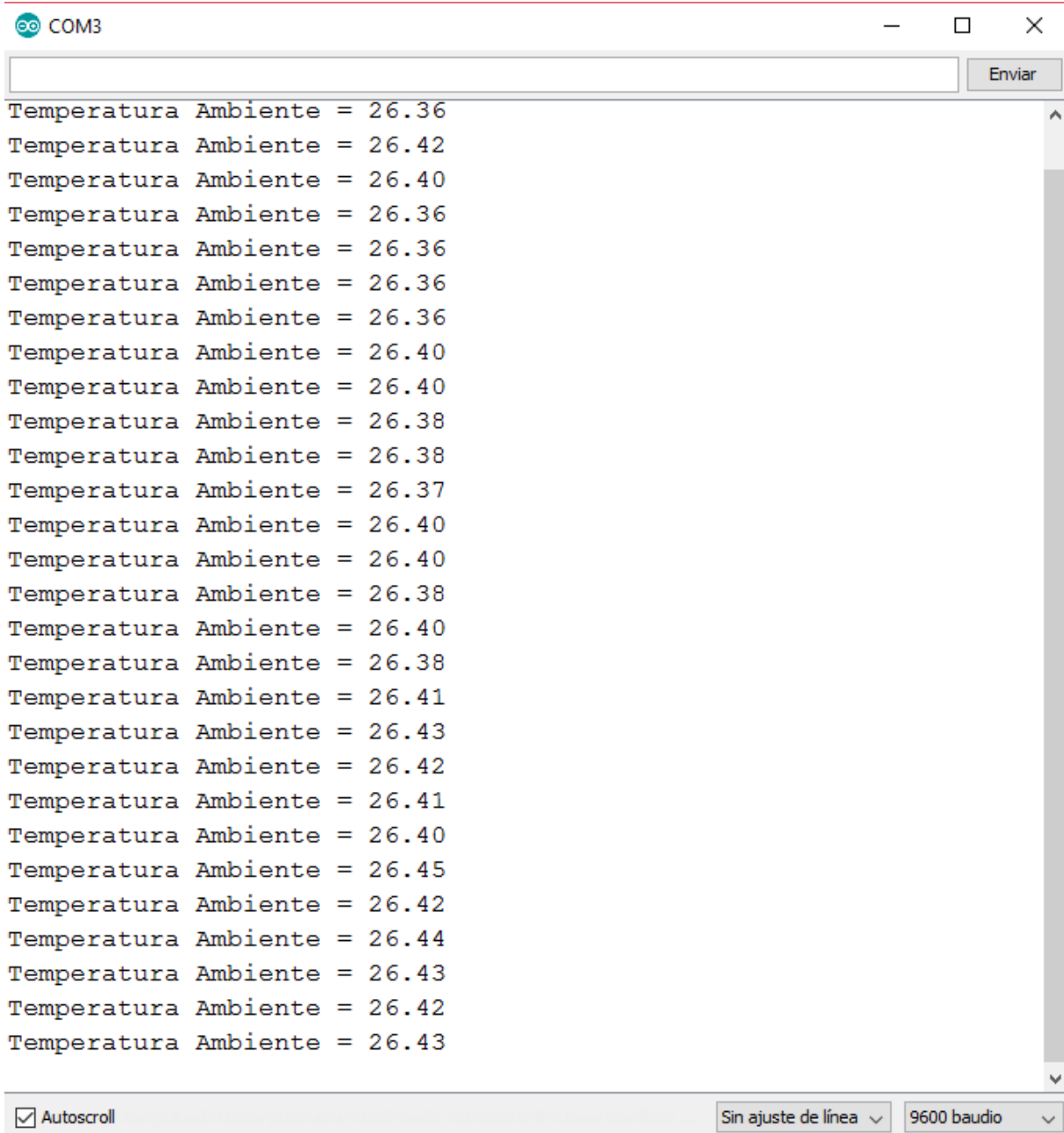


Figura 42. Temperatura Ambiente del sensor MLX90621.

Fuente: Autoría.

El resultado al medir la temperatura del objeto se muestra en la Figura 43, la cual contiene una matriz de 4x16 con los valores de temperatura en grado centígrados, la distancia a la que se realiza la medición es de 0,60 cm. El código de programación se puede apreciar en el Anexo 1.

```

COM3
Temperatura Ambiente: 25.21
6.26 8.61 9.79 11.34 14.14 20.62 23.80 25.88 25.54 25.65 25.34 24.95 24.63 24.52 20.84 17.71 0
7.60 10.89 11.62 14.04 19.63 24.56 25.56 26.51 27.08 27.45 26.25 25.89 25.99 24.26 21.38 19.52 1
6.72 10.31 12.76 18.18 24.57 25.18 25.04 25.05 27.31 26.93 26.16 21.58 20.08 22.26 21.81 21.63 2
7.05 9.29 12.23 18.74 23.11 24.93 23.82 24.48 25.01 24.47 23.04 15.37 15.15 14.33 20.67 19.57 3
Temperatura Ambiente: 25.25
6.64 8.95 10.10 12.05 17.14 22.58 24.04 25.79 25.66 26.20 26.00 25.53 25.01 24.54 20.40 17.55 0
6.97 10.62 12.41 17.15 23.29 25.46 26.07 25.06 26.52 27.66 26.56 25.80 25.90 23.33 21.66 19.54 1
7.39 10.61 13.56 21.11 24.90 25.30 25.06 25.17 26.66 27.04 26.47 19.17 18.85 22.39 22.86 21.94 2
6.36 9.45 13.67 21.73 23.24 24.72 24.38 24.92 24.48 24.48 21.70 14.89 14.91 14.77 20.97 19.10 3
Temperatura Ambiente: 25.26
6.08 8.61 9.79 11.06 13.48 18.86 22.53 25.24 24.90 25.87 26.31 25.98 24.51 24.39 21.29 18.06 0
6.95 10.46 11.36 12.94 16.65 23.15 25.26 26.32 26.51 27.17 26.84 27.31 25.78 24.14 21.25 20.28 1
6.88 10.17 11.84 15.79 22.67 24.77 26.21 25.45 25.31 26.93 27.25 26.06 18.38 20.22 21.81 22.07 2
6.70 8.98 11.07 15.13 22.77 23.81 24.48 23.61 24.47 25.00 25.03 20.84 15.15 13.32 16.79 19.25 3
Temperatura Ambiente: 25.25
5.55 8.31 9.97 10.95 13.91 18.41 20.69 24.94 24.71 25.79 26.44 26.23 24.78 24.29 19.95 20.31 0
7.47 10.49 10.61 12.60 13.37 20.82 24.89 25.85 26.63 27.29 26.67 27.33 23.73 22.15 21.54 19.85 1
6.91 9.63 11.34 13.73 17.77 23.87 25.85 26.07 25.43 26.39 26.98 26.29 19.20 14.89 17.30 21.36 2
6.73 8.70 10.22 11.96 18.31 23.84 24.07 24.39 23.42 24.60 24.83 23.61 15.43 12.63 13.09 14.50 3
Temperatura Ambiente: 25.27
5.91 8.78 9.81 11.07 12.96 18.99 21.85 25.14 24.91 25.56 25.90 25.76 23.28 18.87 17.00 17.20 0
6.47 10.48 10.85 12.09 14.19 21.53 24.37 25.55 25.74 26.12 25.67 25.70 21.27 16.10 16.78 16.45 1
7.06 9.33 10.79 13.72 17.30 22.18 24.07 24.37 24.65 24.75 25.28 23.66 18.51 14.88 13.66 15.86 2
6.54 7.74 9.46 11.27 15.35 21.44 22.07 22.64 22.22 22.22 22.26 20.39 16.05 12.76 12.60 11.57 3

```

Figura 43. Temperatura Objeto del sensor MLX90621.

Fuente: Autoría.

3.4.5.3. Recibir Datos de la placa Arduino.

Para realizar la recepción de los datos se desarrolla un script en el que se importa la librería serial para la comunicación de los dispositivos y la librería time para establecer un tiempo entre los dos dispositivos, posteriormente se define una variable para identificar la placa y asignar el puerto USB al cual está conectado, además de asignar la velocidad en este caso 9600 Baudios. Para recibir los datos de la placa Arduino UNO-R3 se realizan scripts por cada dato recibido; para el nivel de alcohol se desarrolla en el script denominado sensorGas.py cuyo código de programación se encuentra especificado en el Anexo 2, se recibe el dato de la temperatura ambiente en el script temperaturaAmbiente.py que se encuentra en el Anexo 3 y la temperatura facial se especifica en el script temperatura.py el cual se muestra en el Anexo 4.

La Figura 44 muestra la definición de las librerías que se incluyen y las especificaciones del Arduino para todos los scripts que reciben los datos de la placa Arduino UNO-R3.

```
nui sensorGas.py - /home/pi/RecoleccionDatos/SensorGas/sensorGas.py (3.4.2)
File Edit Format Run Options Windows Help
1 import subprocess      #Modulo empleado para los mensajes de audio
2 import serial          #Modulo empleado para establecer la comunicacion con la plataforma ARDUINO
3 import cv2             #Modulo empleado para utilizar las funciones de vision por computador
4 import time            #Modulo empleado para trabajar con fechas y/o horas
5
6 ser = serial.Serial('/dev/ttyUSB0', 9600)      #Define la variable que identifica el puerto al cual
7
```

Figura 44. Definir las librerías necesarias para Recibir los datos de la Placa Arduino.

Fuente: Autoría

Una vez declarado la variable Arduino con los parámetros correspondientes se procede a abrirla, mediante el comando `arduino.open ()`. Posteriormente se inicializa una variable en blanco en la que se va a almacenar los datos recibidos. Seguidamente se crea un bucle infinito en el cual al inicio se establece un tiempo de respuesta, para luego crear un bucle en el que cada vez que existan datos para leer en Arduino UNO R3 se asignen a la variable inicializada anteriormente. En la Figura 45 se muestran la recepción de los datos del sensor de temperatura.

```
04.6606.9707.3608.9409.6410.2710.2911.6811.2211.4312.0611.7111.5009.7209
112.2112.8912.4312.9211.7709.9204.8808.1709.0609.9610.9211.2212.3011.701
7408.8810.1410.7110.4610.5610.6110.9811.3311.5710.8811.0909.91
```

Figura 45. Librerías para Recibir los datos de la Placa Arduino.

Fuente: Autoría

3.4.5.4. *Habilitación de la cámara.*

Una vez conectado el módulo Pi Camera a la placa Raspberry Pi se procede a habilitar el módulo Pi Camera v2, el cual es la cámara seleccionada para el desarrollo del proyecto con

una resolución de imagen de 5 Megapíxeles (2592 x 1944 píxeles) y una velocidad de transferencia de imágenes de 1080p: 30fps y 780p: 60fps. El módulo de la cámara se conecta al Raspberry Pi a través del conector CSI diseñado específicamente para este periférico., con la finalidad de finalizar el estado de reposo. A continuación, se indica el comando que se debe ejecutar en el terminal para habilitar el módulo de la cámara:

sudo raspi-config, Este comando se emplea para visualizar el menú de configuración del sistema embebido. La Figura 46 muestra la opción de habilitar y deshabilitar la cámara:

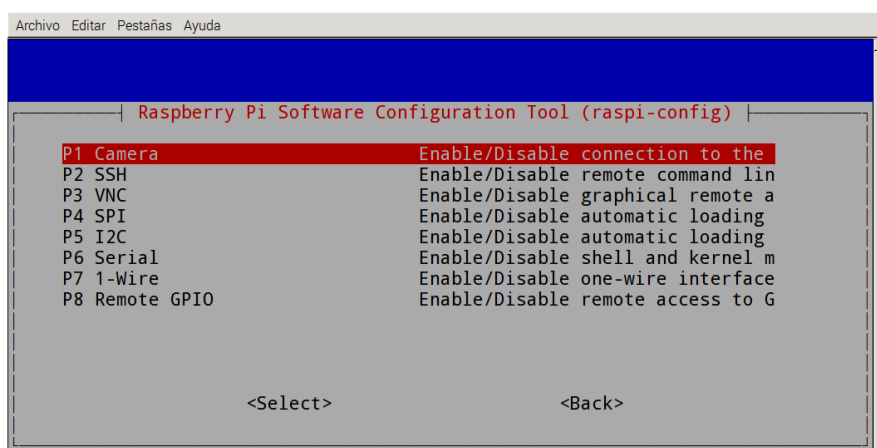


Figura 46. Menú para la habilitación del Módulo de la cámara en el sistema embebido.

Fuente: Autoría

3.4.5.5. Adquisición de imágenes.

El primer paso para adquirir los cambios faciales de conductor es la adquisición de la imagen y para realizar este proceso se agregan los módulos necesarios en la parte superior del script denominado `adquirirImagen.py`, el cual se puede visualizar en el Anexo 5, posteriormente se crea un objeto, mediante el cual se llama al módulo cámara, a continuación, se definen la resolución de la imagen, el formato y se convierte la imagen en una matriz.

Finalmente se crea, escribe, lee y visualiza la imagen con el módulo OpenCV. La Figura 47 muestra los comandos empleados para adquirir la imagen.

```
Archivo Editar Pestañas Ayuda
GNU nano 2.2.6 Fichero: adquirirImagen.py

#AGREGAR LOS MODULOS NECESARIOS
import io #Modulo utilizado parqa leer y escribir imagenes en varios formatos
import picamera #Modulo empleado para interacturar conel modulo de la camara del Raspberry Pi
import cv2 #Modulo empleado para utilizar las funciones de vision por computador
import numpy #Modulo empleado para operar con vectores o matrices

stream=io.BytesIO() #Crear una secuencia de memoria para guardar las imagenes

with picamera.PiCamera() as camera: #Crea un objeto para llamar a la camara
    camera.resolution = (1920,1080) #Define la resolucio
    camera.capture(stream,format='jpeg') #Define el formato
    buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8) #Convierte la imagen en una matriz numpy
image=cv2.imdecode(buff,1) #Crea una imagen empleando OpenCV

cv2.imwrite('AdquisicionImg2.jpg',image) #Guarda el resultado
img = cv2.imread('AdquisicionImg2.jpg') #Leer la imagen
cv2.imshow('AdquisicionImagen2',img) #Visualizar la imagen
```

Figura 47. Comandos empleados para adquirir una imagen con el módulo PiCamera.

Fuente: Autoría

La imagen que se obtiene del script `adquirirImagen.py` es una imagen RGB, esto se debe a que “El espacio RGB es el espacio de color más extendido y el que utilizan la gran mayoría de cámaras de video y fotográficas para construir una imagen de color” (Gil, Torres, & Ortiz Zamora, 2004), la cual se muestra en la Figura 48.

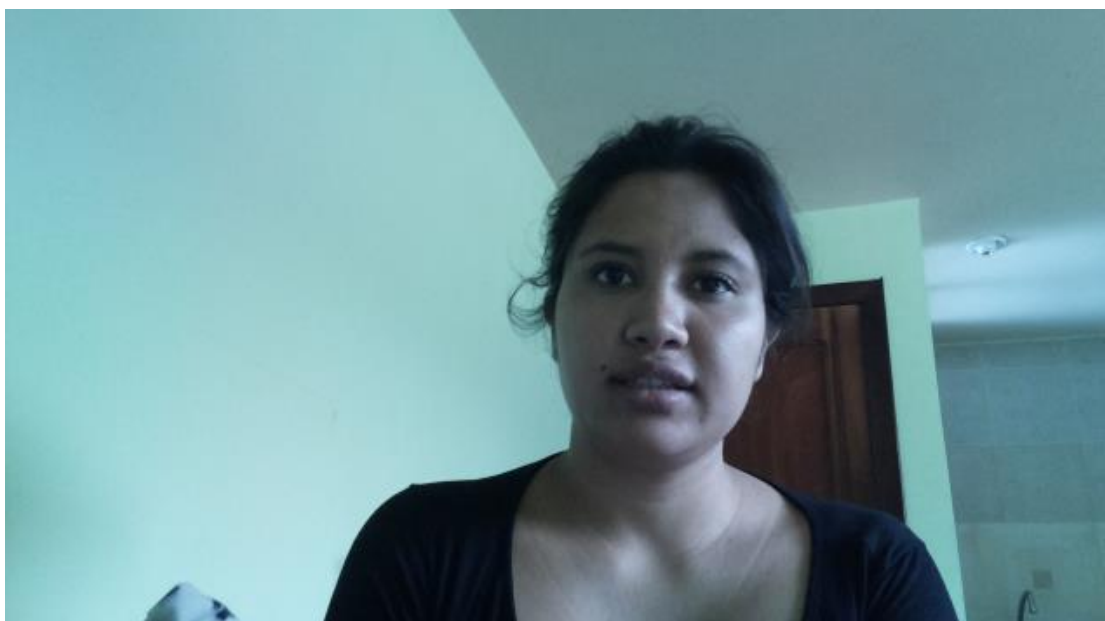


Figura 48. Imagen en RGB.

Fuente: Autoría

3.4.5.6. Preprocesamiento de imágenes.

Una vez realizado el proceso de adquisición de la imagen RGB se procede a una etapa de Preprocesamiento, con la cual se busca eliminar el ruido de la imagen original. A continuación, se detallan las técnicas que pueden ser aplicadas durante la etapa de Preprocesamiento de las imágenes:

Conversión de RGB a escala de grises: La conversión a escala de grises permite que la imagen tenga un solo canal a diferencia de las imágenes RGB que poseen 4 canales, para realizar este proceso se emplea la función `cvtColor` del módulo OpenCv, el tipo de transformación que se emplea es: `CV_BGR2GRAY`. La Figura 49a muestra la imagen capturada por el módulo Pi Camera v2, la cual es una imagen de tipo RGB, la conversión a escala de grises se observa en la Figura 49b al emplear la función `cvtColor` con el comando: `cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)`.

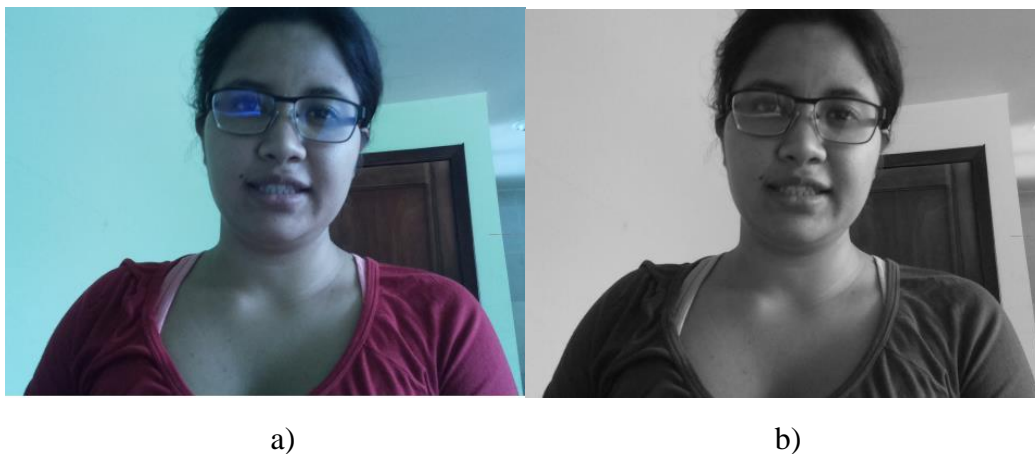


Figura 49. Conversión de una imagen RGB a una imagen a escala de grises.

Fuente: Autoría

Ecualización del histograma: Otra técnica que se puede emplear en el preprocesamiento de imágenes es emplear la ecualización del histograma para mejora el contraste en una imagen y busca agrupar los píxeles de manera que se normalizan los niveles de grises, aumentando el contraste entre las zonas oscuras y las zonas claras. En la Figura 50a se muestra la imagen RGB capturada con el módulo Pi Camera v2, a la que se le aplica la conversión a escala de grises que se indica en la Figura 50b, para posteriormente aplicar la ecualización del histograma para mejorar el contraste y normalizar los niveles de grises como se muestra en la Figura 50c, el comando que se emplea para la ecualización es: `cv2.equalizeHist(src, dst)`.



a)

b)

c)

Figura 50. Imagen RGB aplicado escala de grises y ecualización del histograma.

Fuente: Autoría

El script empleado para realizar el preprocesamiento de la imagen se denomina `ecualizar.py`, el código de programación se puede apreciar en el Anexo 6.

3.4.5.7. Detección de Rostro.

Para la detección de rostro se emplean el algoritmo de reconocimiento facial, el cual se encarga de buscar las características específicas que sean comunes en un rostro humano. Los

clasificadores para emplear en la Detección de rostro son los Haar-Like Features. El Script que se realiza en python hace uso de la herramienta OpenCV para detectar un rostro en la imagen, posteriormente enmarcarlo dentro de un rectángulo. La función que se emplea de OpenCV es `cv2.CascadeClassifier` el cual se basa en el clasificador `faces.xml`. En el Anexo 7 se muestra la estructura del script denominado `detectarRostro.py`. A continuación, en la Figura 51 se muestra el resultado del Script elaborado en Python, el cual contiene el rostro enmarcado en color Azul.



Figura 51. Prueba de Detección de Rostro.

Fuente: Autoría

3.4.5.8. Detección de Ojos.

Para detectar los ojos se emplea el script de la detección de Rostro, en el cual ya se tiene enmarcada la cara y se le agrega el clasificador en cascada entrenado para la detección de ojos dentro del bucle que detecta el rostro, con la finalidad de disminuir la región de búsqueda. La Figura 52 contiene los resultados obtenidos con la ejecución del script denominado `detectarOjos.py` en python con el rostro y ojos detectados y enmarcados utilizando rectángulos. Las líneas de código del script `detectarOjos.py` se encuentra en el Anexo 10.

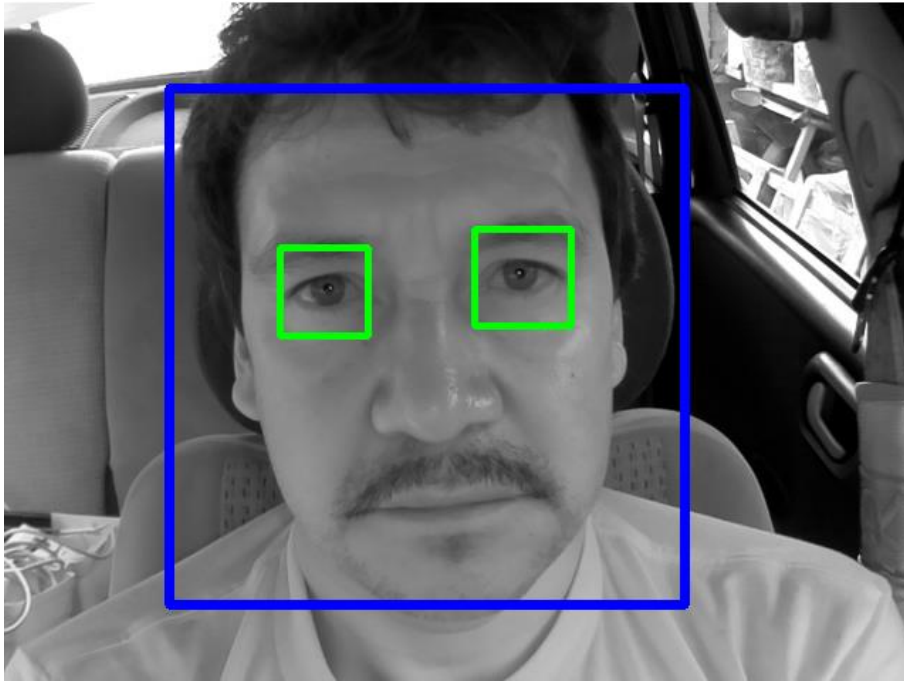


Figura 52. Prueba de Detección de Rostro y Ojos.

Fuente: Autoría

3.4.5.9. Extraer Ojos.

Es necesario extraer los ojos y almacenarlos para posteriormente realizar la medición de la pupila, a la imagen en la que se detectan los ojos se le aplican técnicas de escalado con el objetivo de redimensionar la imagen pero manteniendo la relación del aspecto de la misma, es decir la relación entre el ancho y alto de la imagen, estos parámetros se asigna manualmente; posteriormente se procede a almacenar la imagen en formato .pgm con cada ejecución del script se almacenan 20 imágenes en una carpeta denominada Ojos. La Figura 53 contiene las imágenes que se guardaron.

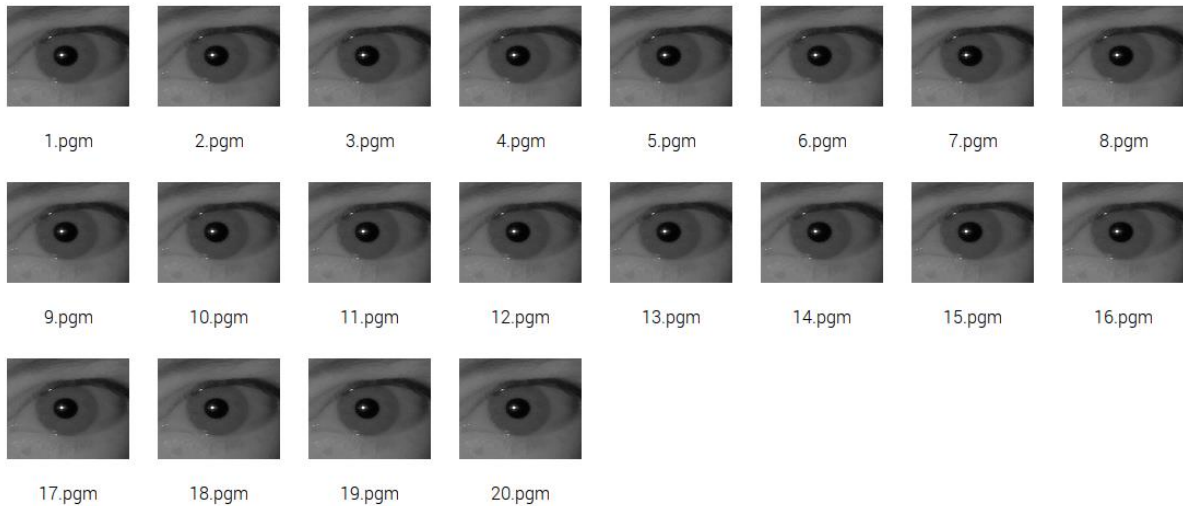


Figura 53. Prueba de Almacenamiento de Ojos.

Fuente: Autoría

3.4.5.10. Detectar la pupila del Ojo.

Debido a que la pupila del ojo es el círculo negro del centro del iris que deja que la luz entre al ojo, se utiliza esta relación geométrica para identificarla; para ello se hace uso de la función `HoughCircles` de OpenCV para detectar círculos, la cual devuelve como resultado el punto central del círculo en coordenadas (x, y) y el radio (r) . Antes de aplicar la función para detectar círculos se procede a aplicar el filtro Gaussiano para reducir la detección de círculos no existentes. Posteriormente se procede a aplicar la función `Houghcircles` en la que se establecen los parámetros que se indican a continuación:

`cv2.HoughCircles(image, method, minDist, param1, param2, minRadius, maxRadius)`

- **Image:** que corresponde a la imagen a analizar es decir la imagen con el ojo.
- **method:** el cual es configurado como `cv2.HOUGH_GRADIENT`
- **minDist:** es la mínima distancia que debe existir entre dos círculos para que sean considerados círculos diferentes.

- param1: El valor de gradiente utilizado para manejar la detección de bordes.
- param2: Valor umbral del acumulador para el método cv2.HOUGH_GRADIENT. Cuanto más pequeño sea el umbral, más círculos serán detectados (incluyendo círculos falsos). Cuanto mayor sea el umbral, más círculos serán potencialmente devueltos.
- minRadius: Tamaño mínimo del radio (en píxeles).
- maxRadius: Tamaño máximo del radio (en píxeles).

En la Figura 54 se observa los valores que se emplearon en el script de detección pupilas denominado pupilas.py y se encuentra en el Anexo 9.

```
#Detectar círculos
circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1, 200, param1=5, param2=10, minRadius=0, maxRadius=0)
```

Figura 54. Función Houghcircles.

Fuente: Autoría

Finalmente se procede a dibujar los círculos encontrados con la función cv2.circle colocando como parámetros el centro del círculo detectado en coordenadas (x, y) y el radio (r). En la Figura 55 se indica los resultados obtenidos para la detección de la pupila, en la Figura 55a muestra la imagen en escala de grises del Ojo del conductor, en la Figura 55b se puede observar el Ojo después de aplicar el filtro Gaussiano y en la Figura 55c se muestra la imagen del ojo dibujado el círculo alrededor de la pupila del conductor.

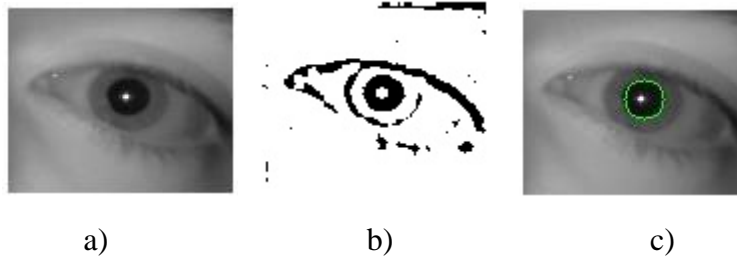


Figura 55. Detección de la Pupila

Fuente: Autoría

3.4.5.11. Alineación de la salida del sensor de Temperatura MLX90621 con la imagen óptica.

Para el desarrollo del Script en el cual se realiza la alineación del sensor de Temperatura con la imagen, se importan los módulos que se muestran en la Figura 56. De los cuales los módulos colores y posicionesMatriz se realizaron con la finalidad de disminuir las líneas de código del script principal denominado Temperatura.py, el cual se puede observar en el Anexo 4.

```

Archivo  Editar  Pestañas  Ayuda
GNU nano 2.2.6                               Fichero: Temperatura.py

#MODULOS
import subprocess          #Modulo empleado para los mensajes de audio
import colores             #Modulo creado para asignar un color a cada valor de temperatura
import posicionesMatriz   #Modulo creado para alinear los datos recibidos del sensor de Temperatura con la imagen
import serial             #Modulo empleado para establecer la comunicacion con la plataforma ARDUINO
import cv2                #Modulo empleado para utilizar las funciones de vision por computador
import time               #Modulo empleado para trabajar con fechas y/o horas
from picamera.array import PiRGBArray #Biblioteca empleada para construir matrices n-dimensional numpy
#a partir de la salida de la cámara
from picamera import PiCamera #Biblioteca empleada para hacer uso de las clases para interactuar con
#el modulo de la camara del Raspberry Pi
import skimage            #Modulo usado para el procesamiento de imagenes
from skimage import exposure #Biblioteca empleada para mejorar el contraste de una imagen
import numpy as np       #Modulo empleado para operar con vectores o matrices
import os                #Modulo empleado para utilizar la funcionalidad dependiente del sistema operativo
import matplotlib        #Modulo empleado para realizar graficas en 2 y 3 dimensiones

```

Figura 56. Módulos empleados en el script para medir la temperatura facial del conductor.

Fuente: Autoría

Después de realizar la adquisición de la imagen, convertirla a escala de grises y disminuir el ruido mediante la ecualización del histograma; se procede a enviar el carácter 'A' a la placa Arduino Uno R3 a través del puerto serial, se verifica que la longitud de la cadena recibida sea de tamaño 320, este tamaño se lo establece considerando que se deben recibir los valores de temperatura de una matriz de 16x4. A continuación, en la Ecuación 6 se indica cómo se obtiene el tamaño de la cadena que se recibe de la placa Arduino.

$$b = \text{filas} \times \text{columnas} \times (\text{numEnteros} + \text{numDecimales} + \text{coma})$$

Ecuación 6. Ecuación para obtener el tamaño de la cadena que contiene las temperaturas de objeto de la matriz 16x4.

Fuente: Autoría.

Donde:

b = Tamaño de la cadena que contiene las temperaturas del objeto.

filas = 4 filas de la matriz.

columnas = 16 columnas de la matriz.

numEnteros = 2 dígitos enteros debido a que no hay temperaturas que lleguen a 100 grados centígrados.

numDecimales = 2 dígitos decimales de la temperatura del Objeto.

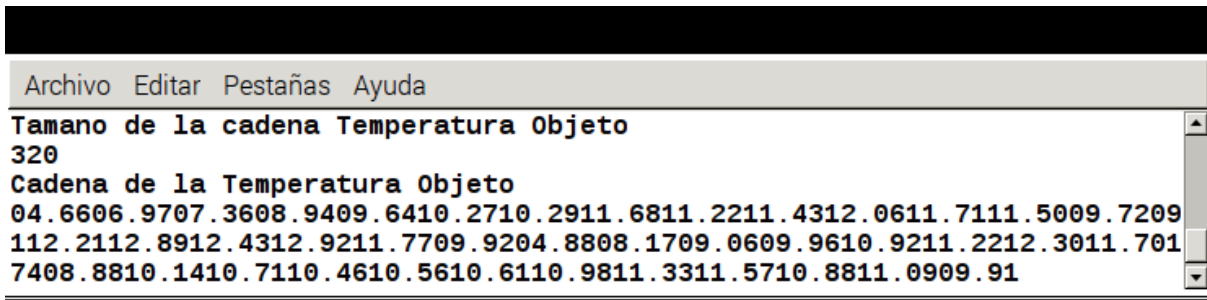
coma = 1 dígito que representa la separación de la parte entera con la parte decimal.

Remplazando en la Ecuación 6 los valores se obtiene el valor de 320 como tamaño de la cadena que se debe recibir.

$$TamCadena = 4 \times 16 \times (2 + 2 + 1)$$

$$TamCadena = 320 \text{ caracteres.}$$

En la Figura 57 se muestra la cadena que se recibe de la placa Arduino uno R3.

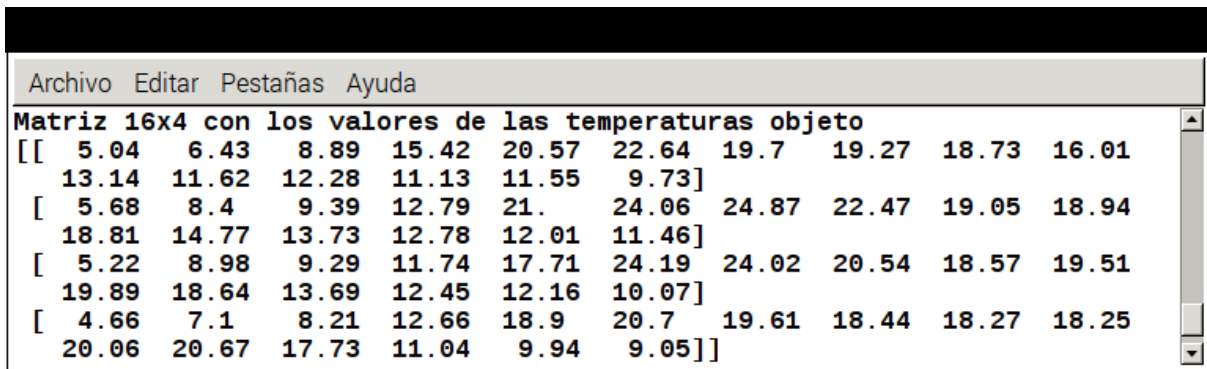


```
Archivo Editar Pestañas Ayuda
Tamaño de la cadena Temperatura Objeto
320
Cadena de la Temperatura Objeto
04.6606.9707.3608.9409.6410.2710.2911.6811.2211.4312.0611.7111.5009.7209
112.2112.8912.4312.9211.7709.9204.8808.1709.0609.9610.9211.2212.3011.701
7408.8810.1410.7110.4610.5610.6110.9811.3311.5710.8811.0909.91
```

Figura 57. Tamaño y cadena que se reciben de la placa Arduino uno R3 de la Temperatura Objeto.

Fuente: Autoría

Se procede a separar la cadena recibida en subcadenas de 5 caracteres cada una, obteniendo como resultado 64 datos de temperatura Objeto. En la Figura 58 se muestra el resultado de separar cada valor de temperatura objeto y almacenarlo en una matriz de 16x4.



```
Archivo Editar Pestañas Ayuda
Matriz 16x4 con los valores de las temperaturas objeto
[[ 5.04 6.43 8.89 15.42 20.57 22.64 19.7 19.27 18.73 16.01
 13.14 11.62 12.28 11.13 11.55 9.73]
 [ 5.68 8.4 9.39 12.79 21. 24.06 24.87 22.47 19.05 18.94
 18.81 14.77 13.73 12.78 12.01 11.46]
 [ 5.22 8.98 9.29 11.74 17.71 24.19 24.02 20.54 18.57 19.51
 19.89 18.64 13.69 12.45 12.16 10.07]
 [ 4.66 7.1 8.21 12.66 18.9 20.7 19.61 18.44 18.27 18.25
 20.06 20.67 17.73 11.04 9.94 9.05]]
```

Figura 58. Matriz 16x4 con los valores de la Temperatura Objeto.

Fuente: Autoría

A continuación, se establece una relación entre el valor de temperatura medido con el sensor MLX90621 y los colores que se asignaran a cada temperatura, posteriormente se crea un script denominado colores.py, que se puede apreciar en el Anexo 10; el cual es el encargado de asignar el color a cada valor de temperatura, la Figura 59 contiene los rangos de temperatura facial establecidos.



-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,>40

Figura 59. Rangos de Temperatura Facial.

Fuente: Autoría

Debido a que se establece una escala con números enteros para los rangos de Temperatura facial se debe convertir cada valor de temperatura a una variable tipo entero a la que se le asignara el color mediante el script color.py. A continuación, en la Figura 60 se muestra el código que se emplea para dividir la cadena y asignar el color que le corresponde.

```

ser.write('A')
ir_raw = ser.readline()
ir_raw = ir_raw.strip()

b = len(ir_raw)

if (b==320):
    for j in range (64):
        for i in range(d,e):
            texto = texto + ir_raw[i]
            entero = int(float(texto))
            color = colores.ColoresTemperatura(entero)

```

Figura 60. Asignar el color a cada valor de Temperatura.

Fuente: Autoría

Posteriormente se procede a buscar los rostros en la imagen haciendo uso del clasificador en cascada entrenado para la detección de rostros, al encontrar una cara el algoritmo devuelve las posiciones de los rostros detectados como una función de Rectangulo (x, y, w, h); estos valores de las posiciones del rostro se emplearán posteriormente para extraer solo los valores de temperatura que coincidan con estas posiciones.

Se hace uso de la función `cv2.rectangle` con el objetivo de dibujar un rectángulo sobre la imagen capturada por la cámara y de esta manera alinear la imagen con la matriz de los valores de temperatura leídos por el sensor. La Figura 61 contiene el código empleado para dibujar el rectángulo sobre la imagen y alinear los valores a la misma.

```
fila=fila+35
cv2.rectangle(image, (fila, columna), (fila+35, columna+80), color, -10)

if(j==15):
    fila= -20
    columna=220
if(j==31):
    fila= -20
    columna=140
if(j==47):
    fila= -20
    columna= 60

if (e==d+5):
    d=e
    e=d+5
    texto = ""
```

Figura 61. Dibujar rectángulos sobre la imagen capturada por la cámara con los valores de Temperatura del sensor MLX90621.

Fuente: Autoría

Finalmente se procede a crear un vector que almacene los datos de la temperatura facial que agrupe los valores de temperatura que se encuentran sobre el rostro del conductor para ello se hace uso de las posiciones (x, y, w, h) del rostro detectado con el clasificador en cascada para establecer las posiciones de inicio y fin de las filas y columnas que contienen el rostro del conductor. Una vez creado el vector se procede a extraer la temperatura facial máxima y mínima del conductor. En la Figura 62 se muestra el vector creado con las temperaturas faciales del conductor, así como el valor de la temperatura máxima y mínima.

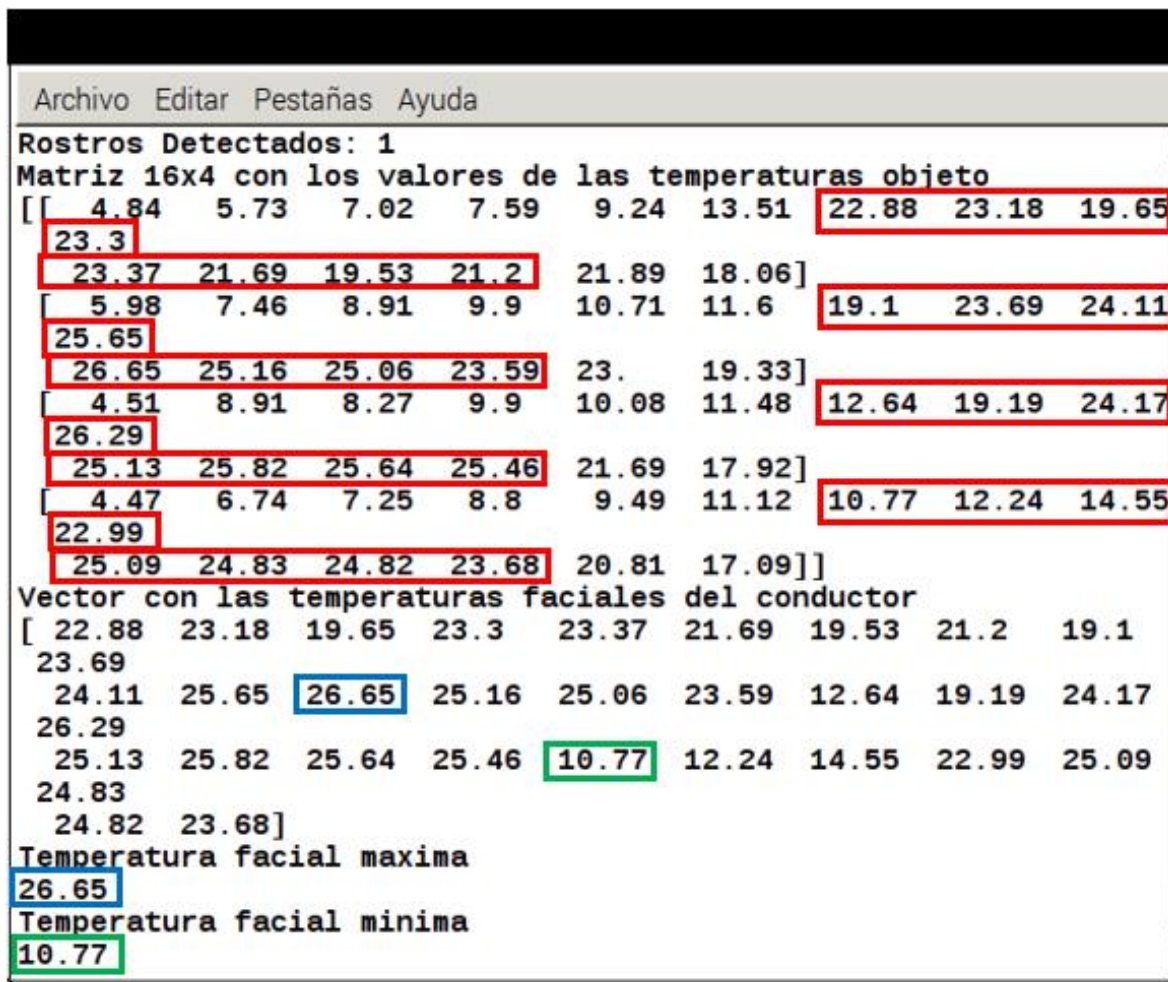
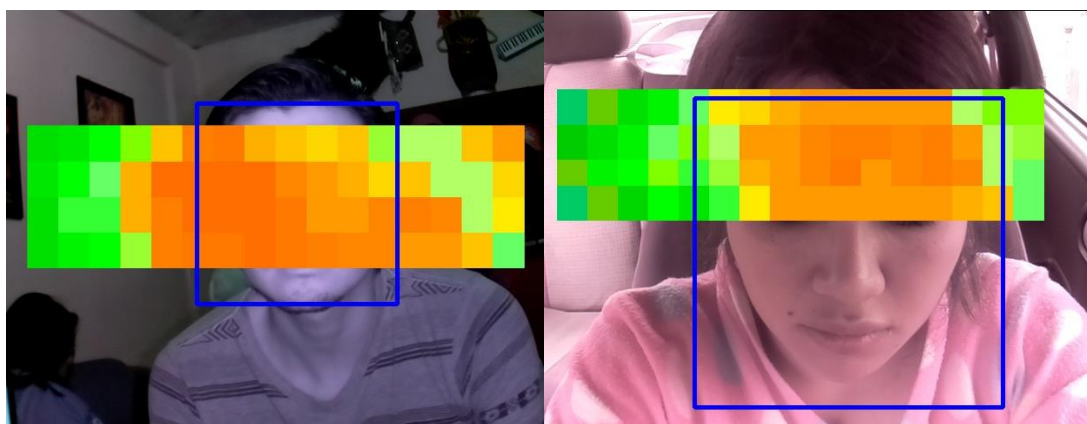


Figura 62. Vector con las Temperaturas faciales del conductor y valor de la temperatura máxima y mínima.

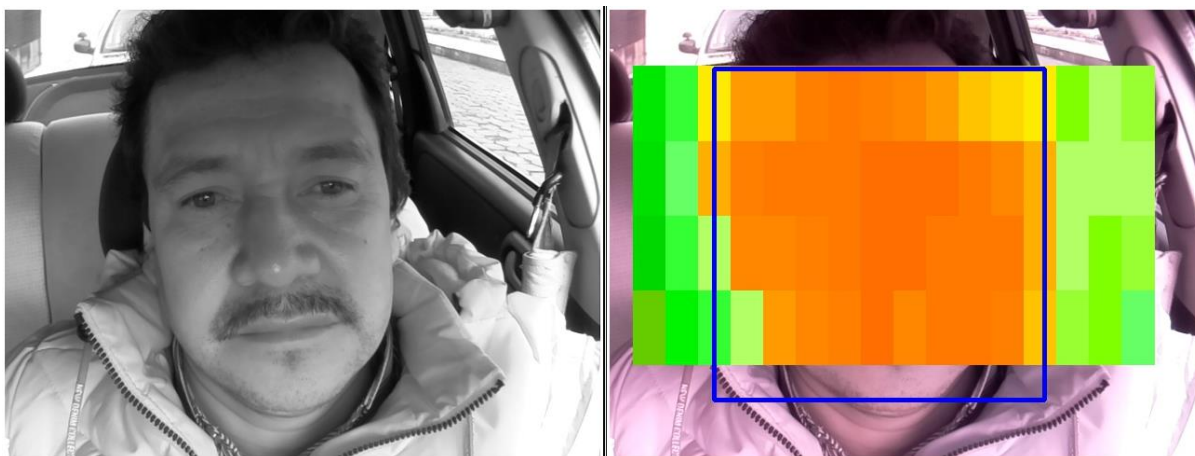
Fuente: Autoría

La Figura 63a contiene los resultados obtenidos al medir la temperatura Facial alineados para mediar la temperatura cuando el usuario se encuentra de 40 a 80 cm de la cámara y el sensor, sin embargo, en la Figura 63b al colocar la cámara y el sensor en el interior del vehículo a 40 cm del conductor se observa existe una desalineación del sensor con la imagen capturada, la Figura 63c muestra los resultados obtenidos al alinear correctamente la imagen del conductor al adquirir los datos del sensor de temperatura variando la distancia.



a)

b)



c)

Figura 63. Alineación del sensor de temperatura MLX90621 con la imagen capturada por el módulo Pi Camera.

Fuente: Autoría

Es necesario que la cámara y el sensor de temperatura se encuentren apuntando hacia el mismo lugar para evitar almacenar falsos datos de la temperatura facial.

3.4.6. Cronograma de Recolección de Información.

La recolección de la información se realiza de acuerdo con el cronograma de actividades durante el transcurso del desarrollo del trabajo de titulación. La Tabla 30 que se muestra a continuación presenta las actividades realizadas para la recopilación de los datos, se realizaron 6 reuniones, en la primera toma de datos se recopilieron 80 datos, en la segunda 125, en la

tercera 66, la cuarta 54, la quinta 65 y la sexta 45 datos; el objetivo es reunir el mayor número de datos para realizar el entrenamiento del algoritmo de aprendizaje automático. El Anexo 13 contiene la información almacenada con diferentes escenarios, en el Anexo 14 se encuentran las imágenes de las pupilas almacenadas y el Anexo 15 muestra las imágenes de la temperatura facial almacenada.

Tabla 30. Cronograma de la recolección de Información.

ACTIVIDAD		DURACIÓN				DATOS					
TIEMPO EN DIAS	Duración	Comienzo	Fin	1	2	3	4	5	6		
1	Primera toma de datos	2 días Vie 13/05/2017	Sab 14/05/2017	X						80	
2	Segunda toma de datos	2 días Sab 09/09/2017	Dom 10/09/2017		X					125	
3	Tercera toma de datos	2 días Dom 15/10/2017	Lun 16/10/2017			X				66	
4	Cuarta toma de datos	1 día Lun 13/11/2017	Lun 13/11/2017				X			54	
5	Quinta toma de datos	1 día Vie 17/11/2017	Vie 17/11/2017					X		65	
6	Sexta toma de datos	1 día Mie 25/04/2018	Mie 25/04/2018						X	45	
TOTAL										435	

Fuente: Autoría

La primera sesión para almacenar el conjunto de datos se la ejecuto a 6 Personas, la prueba de alcoholemia se realizó antes y durante la ingesta de alcohol para conocer el comportamiento del cuerpo humano al tomar Ron.

La segunda recolección de datos se la realizo a 3 Personas Sobrias y a 8 personas antes y durante el consumo de distintos tipos de alcohol; la prueba en los 8 individuos inicio con la ingesta de Ron, posteriormente se les dio a tomar Whisky para finalizar con Ron nuevamente, durante la realización de la prueba se optó porque los individuos mastiquen chicle para intentar vulnerar al sistema; los datos recolectados al ingerir una bebida alcohólica y masticar chicle se encuentran numerados del 189 al 196 del fichero que contiene el conjunto de datos.

La tercera y cuarta recolección de los datos se la hizo con el sistema implementado en el vehículo a un grupo de 19 personas de la cooperativa de Taxis Atahualpa, en esta recolección se pudo almacenar los datos de 15 personas Sobrias y 4 personas en estado etílico.

Para la quinta recolección de datos se realizó la prueba de alcoholemia a 8 Personas que habían ingerido cerveza hace una hora y seguirían ingiriendo cerveza durante el transcurso del almacenamiento de los datos.

La recopilación de los últimos 45 datos se la elaboró con la finalidad de determinar si el sistema es capaz de detectar el estado del conductor realizando la prueba en diferentes circunstancias y con diferentes bebidas:

- La primera prueba se la llevo a cabo con tres personas una mujer y dos hombres, los cuales tenían que ingerir solo una copa con cerveza para conocer el comportamiento de esta bebida en pocas dosis y cómo reacciona el metabolismo entre los dos géneros. Después se realizó la prueba con un vaso cervecero y se finalizó el almacenamiento de los datos 30 minutos después de haber consumido esta bebida.

- La siguiente prueba que se elaboro fue con dos personas que inicialmente debían enjuagar su boca con un enjuague bucal, posteriormente tomar una copa de cerveza, luego tomar un vaso cervecero y para terminar el almacenamiento se les pidió a las dos personas que se enjuaguen la boca y realicen la prueba nuevamente para verificar si el sistema puede ser vulnerado.
- Finalmente, a dos personas más se les realizo la misma prueba anterior, pero cambiando de bebida alcohólica a Ron.

3.4.7. Construcción del modelo de Aprendizaje Automático Supervisado.

Construir un modelo de Aprendizaje Automático o Machine Learning, no se reduce solo a utilizar un algoritmo de aprendizaje o utilizar una librería de Machine Learning; sino que implica todo un proceso, a continuación, se describen los pasos que involucra el desarrollo de un modelo de aprendizaje automático.

3.4.7.1. Recolectar los datos.

El proceso de recolectar los datos se lo realiza a través del prototipo electrónico, para ello se desarrolla un script denominado RecolectarInformación.py el cual se encuentra en el Anexo 16; en el cual se agregan los módulos creados para la adquirir el nivel de alcohol, la temperatura ambiental, el diámetro de la pupila, la temperatura facial máxima y mínima. Después de llamar a cada función para adquirir la información se procede a escribir un fichero, el cual recopila la fecha, la hora, el porcentaje de alcohol en el aliento, la temperatura ambiental, las temperaturas faciales máxima y mínima, el radio de la pupila y el estado del conductor del vehículo. La información se almacena un archivo .csv para posteriormente realizar el

preprocesamiento de los datos. En la Figura 64 se visualiza el archivo que contiene una muestra de los datos recopilados.

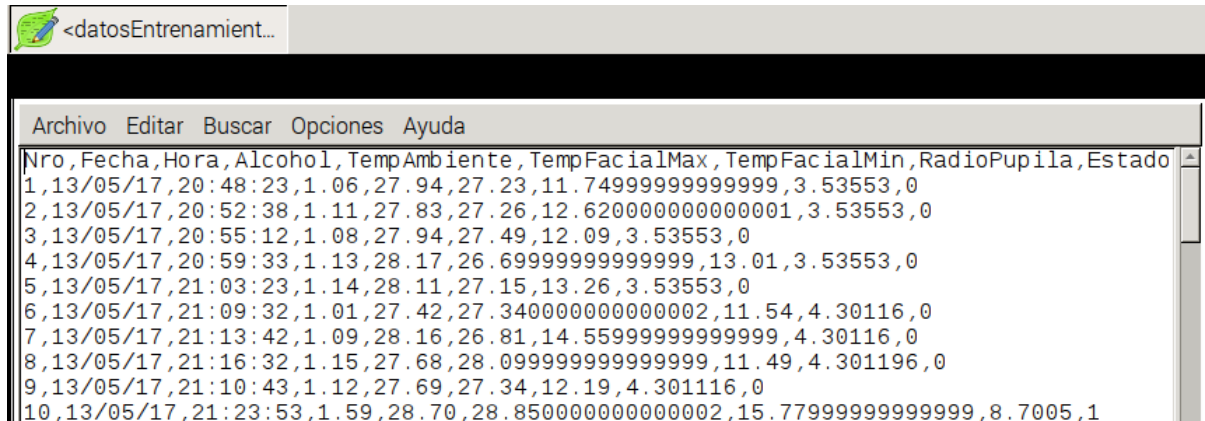


Figura 64. Visualización del archivo con los datos almacenados en la recolección de datos.

Fuente: Autoría

Con los datos recolectados se procede a resolver el problema el cual es determinar si el conductor del vehículo se encuentra en estado de ebriedad. El conjunto de datos con el cual se trabajará es el archivo DatosAprendizaje.csv. Para iniciar el proceso de aprendizaje de maquina se comienza importando las siguientes librerías: IPython, sklearn, numpy, matplotlib, csv y pyparsing.

3.4.7.2. Pre procesar los datos.

Para el preprocesamiento de los datos, inicialmente se procede a cargar el conjunto de datos. Dentro de la lista de atributos que incluye se encuentran: Nro, Fecha, Hora, Alcohol, TempAmbiente, TempFacialMax, TempFacialMin, RadioPupila y Estado (0=Sobrio, 1=Ebrio). En el Anexo 17 se muestra el script denominado preprocesamientoDatos.py, el cual cargar el conjunto de datos en una matriz numpy; el archivo se encuentra en la dirección:

/home/pi/RecoleccionDatos/AprendizajeAutomatico/datosEntrenamientoPruebas.csv En la Figura 65 se muestra las matrices creadas a partir del conjunto de datos para nombres de los atributos y los atributos del conjunto de datos.

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python preprocesamientoDatos.py
//////////Nombres de los atributos del conjunto de datos//////////
['Nro' 'Fecha' 'Hora' 'Alcohol' 'TempAmbiente' 'TempFacialMax'
 'TempFacialMin' 'RadioPupila' 'Estado']
//////////Atributos//////////
[['1' '13/05/17' '20:48:23' ..., '11.749999999999999' '3.53553' '0']
 ['2' '13/05/17' '20:52:38' ..., '12.620000000000001' '3.53553' '0']
 ['3' '13/05/17' '20:55:12' ..., '12.09' '3.53553' '0']
 ...
 ['433' '25/04/18' '18:26:16' ..., '16.140000000000001' '8.63134' '1']
 ['434' '25/04/18' '18:29:49' ..., '16.09' '6.5192' '1']
 ['435' '25/04/18' '18:34:28' ..., '17.23' '6.5192' '1']]
//////////NOMBRES DE LAS CLASES//////////
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico#
```

Figura 65. Matrices con los nombres de los atributos y los atributos del conjunto de datos.

Fuente: Autoría

La Figura 66 contiene la matriz con el nombre de las clases y las clases; las clases del conjunto de datos son dos [0,1] 0 para especificar que el conductor esta sobrio y 1 para conductores ebrios.

```

Archivo  Editar  Pestañas  Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python preprocesamientoDatos.py
['Estado']
//////////////////////////////////Clases//////////////////////////////////
['0' '0' '0' '0' '0' '0' '0' '0' '0' '1' '1' '1' '1' '1' '1' '0' '0' '0'
'0' '0' '0' '0' '0' '0' '0' '0' '1' '1' '1' '1' '1' '0' '0' '0' '0' '0'
'0' '0' '1' '1' '1' '1' '1' '1' '1' '0' '0' '0' '0' '0' '1' '1' '1' '1'
'1' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '1' '1'
'1' '1' '1' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
'0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
'0' '0' '1' '1' '1' '1' '1' '1' '1' '1' '0' '0' '0' '0' '0' '1' '1' '1'
'1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1'
'1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1'
'1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '1'
'1' '1' '1' '1' '1' '1' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
'0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
'0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0'
'0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '1' '1' '0' '0' '0' '0'
'0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '1' '1' '0' '0' '0' '0'
'0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '0' '1' '0' '0' '0' '0'
'0' '1' '1' '1' '1' '1' '1' '1' '0' '0' '1' '1' '1' '1' '1' '1' '1' '1'
'1' '1' '1' '1' '1' '1' '1' '1' '1' '0' '0' '1' '1' '1' '1' '1' '1' '1'
'1' '1' '1' '1' '1' '1' '1' '1' '1' '1' '0' '1' '1' '1' '1' '1' '1' '1'
'1' '1' '1' '1' '0' '0' '0' '0' '0' '0' '0' '0' '1' '0' '1' '1' '0'
'1' '0' '1' '1' '0' '1' '0' '1' '1' '0' '0' '0' '1' '0' '0' '1' '1' '0'
'0' '1' '0' '0' '1' '1' '1' '1' '1' '0' '0' '0' '1' '1' '1' '1' '1' '1'
'1' '1' '1']
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# █

```

Figura 66. Vector con los nombres de las clases y las clases del conjunto de datos.

Fuente: Autoría

Debido a que se tienen 7 atributos para cada clase del conjunto de datos se procede a disminuir el número de atributos considerando los más representativos, de los cuales se eligen los atributos: Alcohol, TempFacialMax y RadioPupila. La Figura 67 indica las nuevas matrices para los nombres de los atributos y los atributos tipo float.

```
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python preprocesamientoDatos.py
//////////Nombres de los atributos Seleccionados//////////
['Alcohol' 'TempFacialMax' 'RadioPupila']
//////////ATRIBUTOS SELECCIONADOS//////////
[[ 0.      27.23    3.53553]
 [ 0.      27.26    3.53553]
 [ 0.      27.49    3.53553]
 ...
 [ 1.67    26.88    8.63134]
 [ 1.85    26.74    6.5192 ]
 [ 1.83    27.13    6.5192 ]]
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# █
```

Figura 67. Matrices con los nombres de los atributos y atributos seleccionados para construir el algoritmo.

Fuente: Autoría

Para finalizar con el preprocesamiento de los datos se procede a dividir el conjunto de datos en: conjunto de entrenamiento y conjunto de prueba.

Para obtener los dos tipos de datos, se procede a dividir los datos muestrales en dos partes; la primera parte se emplea como conjunto de entrenamiento, con la finalidad de determinar los parámetros del clasificador y la segunda parte, denominada conjunto de prueba se utiliza para estimar el error de generalización que debe ser pequeño, con el fin de evitar el sobreajuste, el cual consiste en una sobrevaloración de la capacidad predictiva de los modelos obtenidos; por lo que, no tiene sentido evaluar la calidad de un modelo sobre los datos que han servido para construirlo ya que esta práctica nos lleva a ser demasiado optimistas acerca de su calidad. (García Cambronero & Gómez Moreno, 2006)

Para la división del conjunto de datos se hace uso de la función **train_test_split**, los parámetros que requiere esta función son los atributos y las clases. Debido a que el conjunto de datos es pequeño el 80% se utiliza para entrenar al modelo de aprendizaje automático y el 20% se asigna como un conjunto de datos de validación. El resultado de esta función devuelve

cuatro parámetros: X_train (x para entrenar), y_train (y para entrenar), X_test (x para pruebas) y y_test (y para pruebas). En la Figura 68 se indica que X_train es una matriz de 348x3 la cual especifica que se tienen 348 casos del Estado del conductor con 3 atributos, y_train es un vector de 348 que especifica las clases; mientras que X_test es una matriz de 87x3 la cual especifica que se tienen 87 casos del Estado del conductor con 3 atributos, y_test es un vector de 87 que especifica las clases.

```
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python preprocesamientoDatos.py
////////////////X_ENTRENAMIENTO////////////////
(348, 3)
////////////////y_ENTRENAMIENTO////////////////
(348,)
////////////////X_PRUEBA////////////////
(87, 3)
////////////////y_PRUEBA////////////////
(87,)
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico#
```

Figura 68. Atributos y clases del conjunto de entrenamiento y el conjunto de prueba.

Fuente: Autoría

3.4.7.3. Entrenar el algoritmo.

El aprendizaje es el proceso de estimación de una dependencia desconocida, utilizando un número limitado de observaciones para nuestro caso de estudio el conjunto de datos que se especificó en el punto 3.4.7.2; el esquema de aprendizaje incluye tres componentes: un generador de vectores de entrada aleatorios, un sistema que proporciona la salida para un vector de entrada dado y una máquina de aprendizaje que estima una relación desconocida del sistema a partir de las muestras observadas. Dentro de esta sección se analizan los dos algoritmos de aprendizaje automático supervisado empleados para la clasificación: K-VECINOS MÁS PRÓXIMOS (“K Nearest Neighbor”) y ÁRBOL DE DECISIÓN (Decision Tree).

3.4.7.3.1. K-Vecinos más Próximos (“K Nearest Neighbor” KNN)

El método de los k-vecinos o k-nn es un método retardado y supervisado (pues su fase de entrenamiento se hace en un tiempo diferente al de la fase de prueba) cuyo argumento principal es la distancia entre instancias.

La principal dificultad de este método consiste en determinar el valor de k, ya que si toma un valor grande se corre el riesgo de hacer la clasificación de acuerdo con la mayoría (y no al parecido), y si el valor es pequeño puede haber imprecisión en la clasificación a causa de los pocos datos seleccionados como instancias de comparación.

Para enfrentar este problema se plantearon diferentes variaciones del método: en cuanto a la forma de determinar el valor de k, por ejemplo 1-nn, que no es otra cosa más que usar como instancia de comparación al primer vecino más cercano encontrado.

También el valor de k puede hallarse tomando un radio de comparación o mediante el uso de diagramas de Voronoi.

Una característica importante e interesante de k-nn es que el método puede cambiar radicalmente sus resultados de clasificación sin modificar su estructura, solamente cambiando la métrica utilizada para hallar la distancia. Por lo tanto, los resultados pueden variar tantas veces como métodos de hallar distancia entre puntos haya.

A continuación, se presentan algunas de las diferentes métricas para medir la distancia sólo se mostrarán los modelos matemáticos generales de cada métrica, sin detalles, con la finalidad de recordar que además de la distancia clásica euclidiana existen métricas alternativas.

En la Ecuación 7 se muestra el modelo matemático para la distancia Euclidiana que será la que se empleara para el desarrollo del algoritmo.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Ecuación 7. Distancia Euclidiana.

Fuente: (Rodríguez Rodríguez, Rojas Blanco, & Franco Camacho, 2007)

En la Ecuación 8 se muestra el modelo matemático para la distancia de Manhattan.

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Ecuación 8. Distancia de Manhattan.

Fuente: (Rodríguez Rodríguez, Rojas Blanco, & Franco Camacho, 2007)

En la Ecuación 9 se muestra el modelo matemático para la distancia de Chebychev.

$$d(x, y) = \max_{i=1..n} |x_i - y_i|$$

Ecuación 9. Distancia de Chebychev.

Fuente: (Rodríguez Rodríguez, Rojas Blanco, & Franco Camacho, 2007)

En la Ecuación 10 se muestra el modelo matemático para la distancia del coseno.

$$d(x, y) = \arccos\left(\frac{x^T y}{\|x\| \cdot \|y\|}\right)$$

Ecuación 10. Distancia del coseno.

Fuente: (Rodríguez Rodríguez, Rojas Blanco, & Franco Camacho, 2007)

En la Ecuación 11 se muestra el modelo matemático para la distancia de Mahalanobis.

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}$$

Ecuación 11. Distancia de Mahalanobis.

Fuente: (Rodríguez Rodríguez, Rojas Blanco, & Franco Camacho, 2007)

En la Ecuación 12 se muestra el modelo matemático para la distancia usando la función delta.

$$d(x, y) = \omega \sum_{i=1}^n \delta(x_i, y_i)$$

Ecuación 12. Distancia usando la función delta.

Fuente: (Rodríguez Rodríguez, Rojas Blanco, & Franco Camacho, 2007)

Para la implementación del algoritmo de aprendizaje automático supervisado K-Vecinos más próximos (KNN), el primer paso es seleccionar el tipo de distancia con la que se va a trabajar. Debido a que se empleara la librería scikit-learn de aprendizaje automático para Python se seleccionó la distancia euclidiana, el modelo matemático se puede apreciar en la Ecuación 7.

En la Figura 69 se presenta un gráfico de los atributos en un plano de 3 dimensiones. Los ejes o dimensiones (a1, a2, a3), representan atributos de los datos. La distancia euclidiana entre los puntos x_i y x_j (que son dos casos particulares de un conjunto de datos) es igual a la longitud de la recta que los separa.

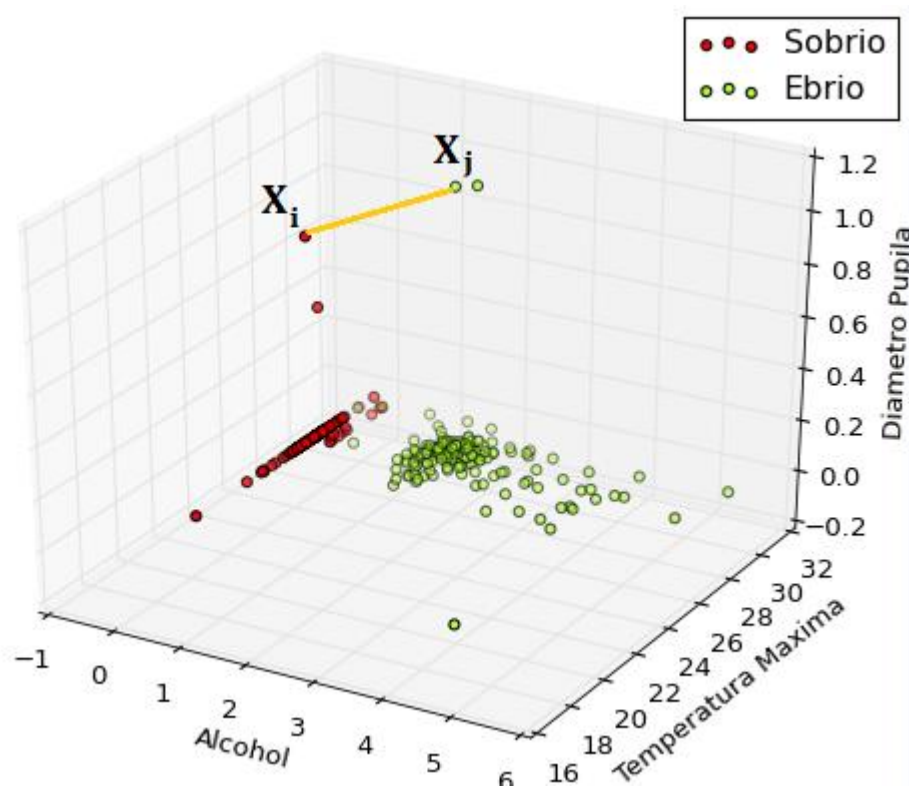


Figura 69. Representación de la distancia euclidiana en un plano en 3 dimensiones.

Fuente: Autoría

Después de interpretar la Ecuación 7 correspondiente a distancia euclidiana y de adaptar la ecuación a los términos de K-NN, se procede a emplear el módulo neighbors de la biblioteca sklearn, el cual posee función principal del algoritmo. Es necesario que los datos de entrenamiento sean normalizados, es decir que los valores de los atributos se dejen entre 0 y 1 con el fin de evitar que los resultados de las distancias se queden en valores medios y deterioren los resultados de la clasificación. Para normalizar se usa la función standarScaler, la cual tiene como fórmula matemática la Ecuación 13 que se muestra a continuación.

$$V = \frac{(V' - \text{min}A)}{\text{max}A - \text{min}A} (\text{nuevoMax}A - \text{nuevoMin}A) + \text{nuevoMin}A$$

Ecuación 13. Normalización de los datos de Entrenamiento.

Fuente: (Rodríguez Rodríguez, Rojas Blanco, & Franco Camacho, 2007)

Donde V es el nuevo valor por hallar para cierto dato perteneciente al atributo A , v' es el valor actual del dato antes mencionado, $\min A$ y $\max A$ son los valores mínimo y máximo de datos que pertenecen a A respectivamente y $\text{nuevoMax}A$ y $\text{nuevoMin}A$ son los nuevos valores máximo y mínimo que podrá tomar V .

El módulo `scikit-learn` para aprendizaje automático en python implementa el clasificador de vecinos más cercanos: `KNeighborsClassifier`, en el cual el aprendizaje se basa en los k vecinos más cercanos de cada punto de consulta, donde k se especifica en un valor entero que es asignado por el usuario. La elección óptima del valor k depende en gran medida de los datos: en general, una mayor k suprime los efectos del ruido, pero hace que los límites de clasificación sean menos distintos. En este punto es donde realmente se emplea las técnicas de aprendizaje automático, con el objetivo de extraer información útil de los datos almacenados para posteriormente poder hacer predicciones. Para entrenar el algoritmo se emplea la función `fit`, que recibe como parámetros `X_train` y `y_train`, es decir x/y de entrenamiento. En la Figura 70a se muestra el resultado de la función `score` que representa que tan bien aprendió el algoritmo cuando el conjunto de Datos es 360, los parámetros que recibe esta función son `X_test` y `y_test`, que hace referencia a x/y de validación; mientras que al incrementar la base de datos a 435 el resultado de exactitud se muestra en la Figura 70b.

```
Archivo Editar Pestañas Ayuda
Exactitud con el clasificador Vecinos Cercanos
0.935897435897
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico#
```

a)

```
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python clasifi
cacionKNN.py
Exactitud con el clasificador Vecinos Cercanos
0.988505747126
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico#
```

b)

Figura 70. Exactitud del algoritmo de aprendizaje automático supervisado mediante el clasificador KNN cuando la base de datos es 360 y 435 datos.

Fuente: Autoría

En este caso se obtiene que el algoritmo pudo predecir el 93,58% de datos del conjunto de pruebas, considerando $K=7$, sin embargo, al incrementar el conjunto de datos el algoritmo de aprendizaje automático basado en KNN puede predecir el 98,85%. Debido a la variación de exactitud al incrementar el conjunto de datos se procederá a verificar si no existe sobreajuste, para ellos se calcula la exactitud, pero con X/Y de entrenamiento. Si el aprendizaje es 100% quiere decir que hay una sobre estimación de la calidad de predecir en el algoritmo. La Figura 71 muestra el resultado de la función Score realizada sobre x/y de entrenamiento, a través de la cual se verifica que no existe sobreajuste y que al incrementar el conjunto de datos se mejora la calidad de predicción del algoritmo.

```
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python clasifi
cacionKNN.py
Exactitud con el clasificador Vecinos Cercanos
0.988505747126
Sobreajuste
0.979885057471
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# █
```

Figura 71. Calculo de Sobreajuste en el algoritmo de aprendizaje basado en KNN.

Fuente: Autoría

Para predecir un nuevo dato se hace uso de la función predict, sobre la variable que se empleó para seleccionar el clasificador, en este caso la variable es denominada knn; el parámetro que requiere esta función es el vector con los nuevos datos a predecir. En la Figura 72a se indica el vector que se emplea para predecir y el resultado de la clase a la que pertenece los datos cuando el conjunto de datos es 360, mientras que en la Figura 72b se muestra la clase cuando el conjunto es 435.

```
Archivo Editar Pestañas Ayuda
//////////DATOS A PREDECIR//////////
[[ 1.61 27.33 9.5 ]]
//////////CLASE A LA QUE PERTENECE//////////
[ 1.]
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# █
```

a)

```
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python clasifi
cacionKNN.py
//////////DATOS A PREDECIR//////////
[[ 1.61 27.33 9.5 ]]
//////////CLASE A LA QUE PERTENECE//////////
[ 1.]
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# █
```

b)

Figura 72. Predicción de la clase a la que pertenecen los atributos dados en el vector.

Fuente: Autoría

En la Figura 72 se puede apreciar que cuando el conductor presenta un nivel de alcohol de 1.61, la temperatura Facial Máxima de 27.33 y un diámetro de la pupila de 9.5; el algoritmo es capaz de predecir que esos atributos pertenecen a la clase 1 que corresponde a conductores en estado de Ebriedad. El código de programación para el desarrollo del algoritmo de aprendizaje automático supervisado a través de knn vecinos más cercanos se encuentra en el Anexo 18.

3.4.7.3.2. *Árbol de Decisión (Decision Tree)*

El árbol de decisión es un método de aprendizaje supervisado no paramétrico utilizado para la clasificación. El objetivo es crear un modelo que prediga el valor de una variable objetivo mediante el aprendizaje de reglas simples de decisión inferidas a partir de las características de los datos.

Para la modelación matemática de este método esta descrito se consideran el vector con los atributos del conjunto de datos mediante la Ecuación 14 y el vector con las clases mediante la Ecuación 15. La finalidad que tiene el método de árbol de decisión es particionar recursivamente el espacio de modo que las muestras con las mismas etiquetas se agrupen juntas.

$$x_i \in R^m, i = 1$$

Ecuación 14. Función del vector que contiene los atributos del conjunto de datos.

Fuente: (Scikit-learn Organization, 2017)

$$y \in R^l$$

Ecuación 15. Función del vector que contiene las clases del conjunto de datos.

Fuente: (Scikit-learn Organization, 2017)

Cuando el resultado de clasificación a través, del árbol de decisión toma los valores de $K=0,1,2, 3,\dots,n$; para el nodo m que Representa una región R_m con N_m observaciones, el modelo matemático se describe en la Ecuación 16.

$$P_{mk} = \frac{1}{N_m} \sum_{x_i \in R^m} I(y_i = k)$$

Ecuación 16. Modelo matemático del algoritmo de aprendizaje automático basado en el árbol de decisión.

Fuente: (Scikit-learn Organization, 2017)

Las medidas comunes de impureza son Gini que se muestran en la Ecuación 17.

$$H(X_m) = \sum_K P_{mk}(1 - P_{mk})$$

Ecuación 17. Medidas comunes de impureza son Gini.

Fuente: (Scikit-learn Organization, 2017)

Mientras que la clasificación errónea está dada por la Ecuación 18, en donde X_m son los datos de entrenamiento en el nodo m .

$$H(X_m) = 1 - \max(P_{mk})$$

Ecuación 18. Clasificación errónea del modelo árbol de decisión.

Fuente: (Scikit-learn Organization, 2017)

Antes de iniciar con el entrenamiento del algoritmo de aprendizaje automático supervisado basado en el árbol de decisión, se procederá a seleccionar las variables más dominantes del conjunto de datos haciendo uso de la característica `feature_importances`, la cual permite conocer la importancia de cada atributo en específico para seleccionar a que clase pertenece. En la Figura 73 se puede observar que el atributo que tiene más relevancia es el nivel de alcohol del conductor, por lo que en la Figura 74 se muestra la importancia entre los atributos

de Temperatura Facial Máxima, mínima y el radio de la pupila para proceder a seleccionar los otros dos atributos para el entrenamiento del algoritmo.

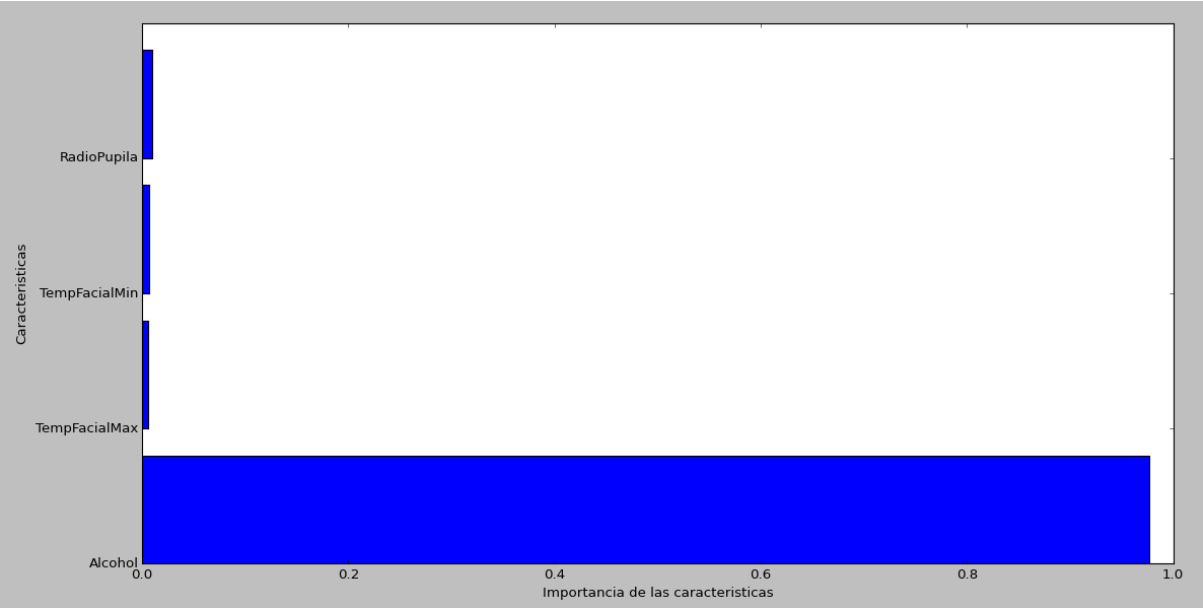


Figura 73. Nivel de Alcohol como atributo de mayor Importancia para seleccionar a que clase pertenecen.

Fuente: Autoría

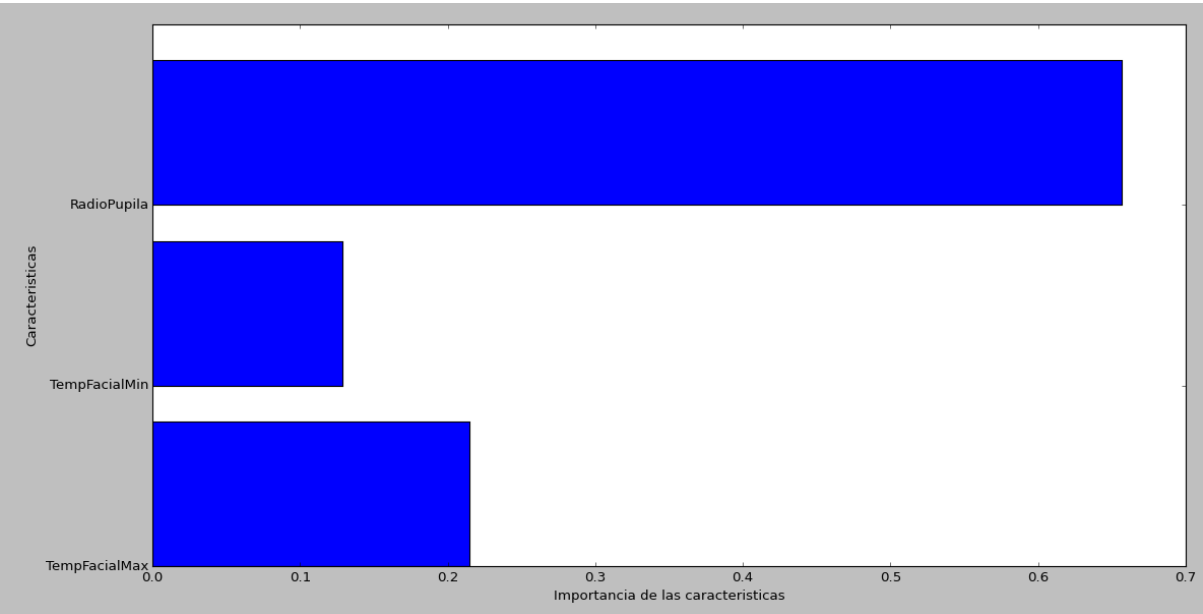
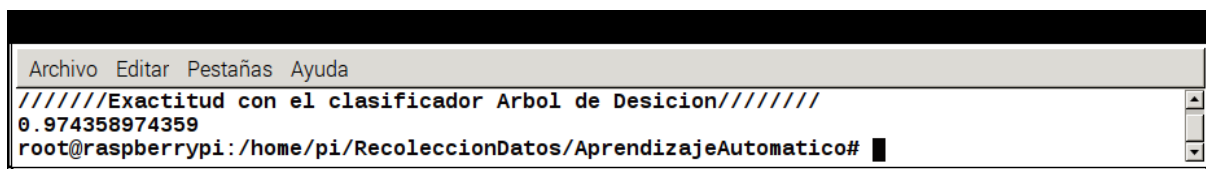


Figura 74. Radio de la pupila y temperatura máxima como siguientes atributos con mayor Importancia para seleccionar a que clase pertenecen.

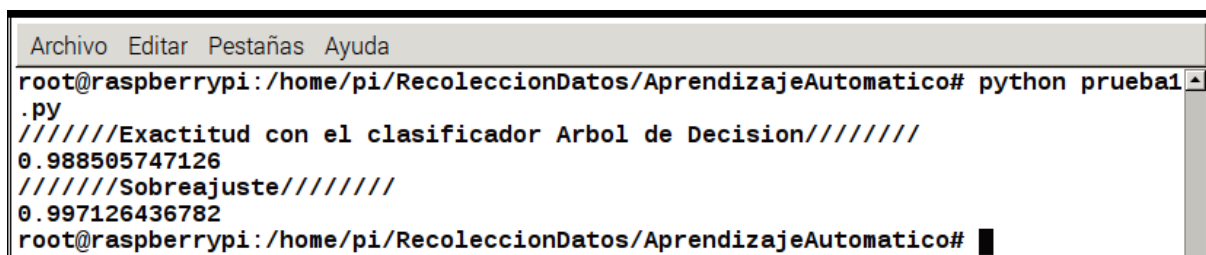
Fuente: Autoría

Para iniciar con el entrenamiento del modelo basado en árbol de decisión, se hará uso de la librería scikit-learn para aprendizaje automático en python, la cual contiene el clasificador de árbol de decisión: DecisionTreeClassifier, que se basa en la agrupación de las clases con los atributos que sean similares creando, condiciones para cada rama. Para entrenar el algoritmo se emplea la función fit, que recibe como parámetros X_train y y_train, es decir x/y de entrenamiento. En la Figura 75a se muestra el resultado de la función score que representa que tan bien aprendió el algoritmo cuando el conjunto de datos es 360, los parámetros que recibe esta función son X_test y y_test, que hace referencia a x/y de validación; mientras que en la Figura 75b se indica la exactitud cuándo se tienen 435 datos para entrenar el algoritmo.



```
Archivo Editar Pestañas Ayuda
////////Exactitud con el clasificador Arbol de Decision////////
0.974358974359
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico#
```

a)



```
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python prueba1
.py
////////Exactitud con el clasificador Arbol de Decision////////
0.988505747126
////////Sobreajuste////////
0.997126436782
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico#
```

b)

Figura 75. Exactitud del algoritmo de aprendizaje automático supervisado mediante el clasificador de árbol de decisión con 360 y 435 datos.

Fuente: Autoría

Para almacenar en un archivo .dot la información del árbol de decisión que se crea con los datos de entrenamiento se hace uso de la función: export_graphviz. En la Figura 76 se muestra el árbol de decisión que se guarda en formato png con los 390 datos después de graficarlo, esta clasificación se realiza en tres niveles, en el primer nivel analiza el nivel de alcohol en el cual si es menor igual a 0.695 el conductor es asignado a la clase 0 sin importar

que radio de pupila o temperatura facial máxima tenga; en el segundo nivel se considera que la temperatura facial es menor o igual a 29.405 el conductor es asignado a la clase 0, para finalizar se establece la condición del tercer nivel en la cual se considera que para una temperatura facial máxima menor o igual a 29,84 el conductor será asignado a la clase 0; mientras que en la Figura 77 se muestra el árbol de decisión cuando el conjunto de datos es 425, en el cual la clasificación se hace en 5 niveles, en el primero y segundo solo se considera el nivel de alcohol, en el tercero se considera la temperatura facial máxima, en el cuarto también se considera el radio de la pupila y en el último nivel se pone como condición solo el nivel de alcohol pero para ello se debe primero pasar los anteriores cuatro niveles.

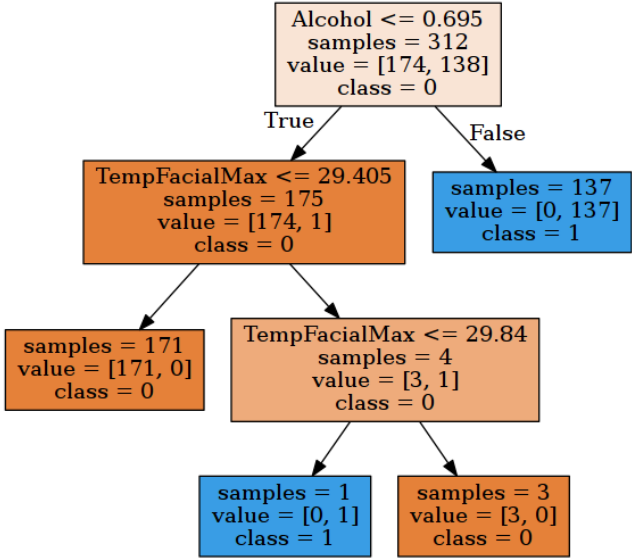


Figura 76. Árbol de decisión con 390 datos en el conjunto de datos mediante el cual el algoritmo clasifica si el conductor esta sobrio o ebrio.

Fuente: Autoría

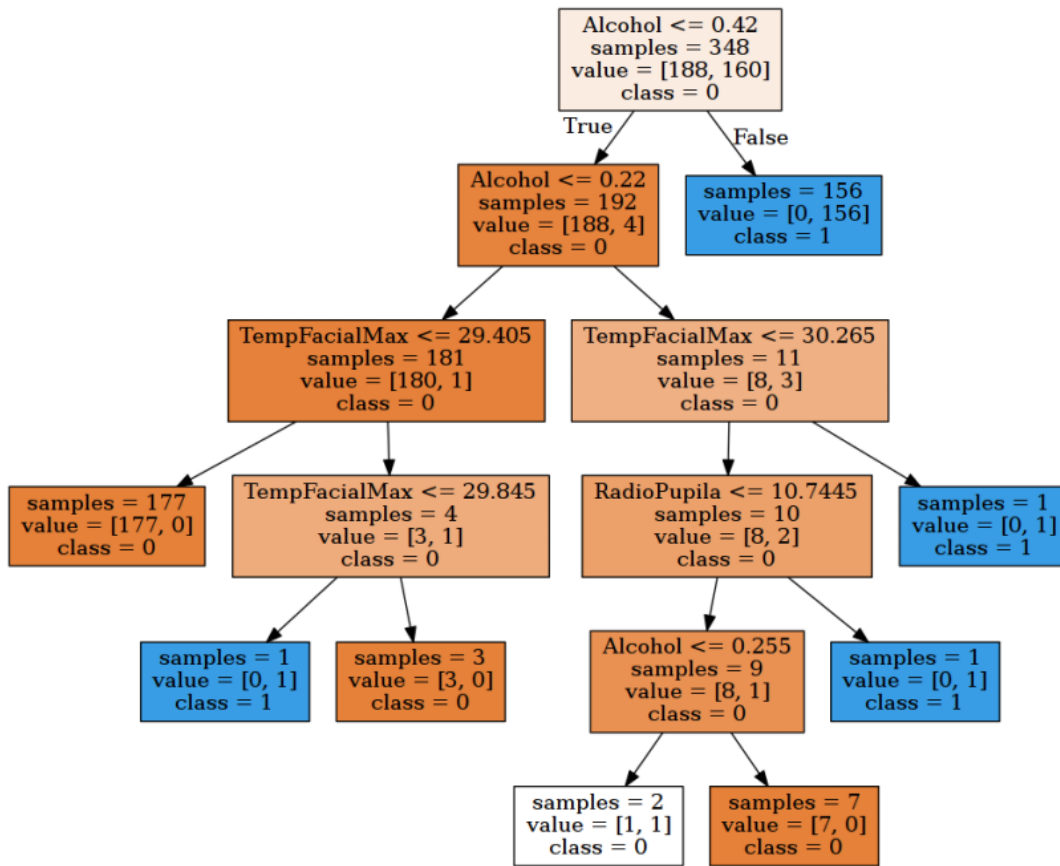


Figura 77. Árbol de decisión con 425 datos en el conjunto de datos mediante el cual el algoritmo clasifica si el conductor esta sobrio o ebrio.

Fuente: Autoría

Para predecir un nuevo dato se hace uso de la función predict, sobre la variable que se empleó para seleccionar el clasificador, en este caso la variable es denominada árbol; el parámetro que requiere esta función es el vector con los nuevos datos a predecir. En la Figura 78 se indica el vector que se emplea para predecir y el resultado de la clase a la que pertenece los datos cuando el conjunto de datos es 390; mientras que cuando la base de datos aumenta a 435 se puede observar en la Figura 79.

```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
/////////////////DATOS A PREDECIR/////////////////  
[[ 1.41 23.33 7.5 ]]  
/////////////////CLASE A LA QUE PERTENECE/////////////////  
[ 1.]  
root@raspberrypi: /home/pi/RecoleccionDatos/AprendizajeAutomatico#
```

Figura 78. Predicción de la clase a la que pertenecen los atributos con 390 datos.

Fuente: Autoría

```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
/////////////////DATOS A PREDECIR/////////////////  
[[ 1.41 23.33 7.5 ]]  
/////////////////CLASE A LA QUE PERTENECE/////////////////  
[ 1.]  
root@raspberrypi: /home/pi/RecoleccionDatos/AprendizajeAutomatico#
```

Figura 79. Predicción de la clase a la que pertenecen los atributos con 435 datos.

Fuente: Autoría

En la Figura 78 se puede apreciar que haciendo uso de los mismos datos del conductor para predecir que en la Figura 72; el algoritmo con el clasificador de árbol de decisión también es capaz de predecir que esos atributos pertenecen a la clase 1 que corresponde a conductores en estado de Ebriedad. El código de programación para el desarrollo del algoritmo de aprendizaje automático supervisado a través de árbol de decisión se encuentra en el Anexo 19

Considerando la exactitud de los dos clasificadores y las características que presenta cada uno se selecciona como algoritmo de aprendizaje automático supervisado el modelo de árbol de decisión ya que además de tener un 98.85% de exactitud al predecir un nuevo dato, tiene la característica de seleccionar el atributo que posee mayor importancia dentro del conjunto de datos.

Para finalizar con la etapa que realiza el entrenamiento del algoritmo se realizará un análisis considerando solo como atributos del conjunto de datos a la temperatura facial máxima

y el radio de la pupila, sin embargo, debido a que solo se presenta variación de estos datos al consumir altos niveles de alcohol no será factible la implementación del algoritmo solo con estos dos atributos en un vehículo de transporte público.

- Procesamiento de los Datos, en esta fase se procede a seleccionar las dos variables como atributos del conjunto de datos. En la Figura 80 se muestra el resultado que indica los nombres de los atributos y los atributos seleccionados, posteriormente se realiza la división del conjunto de datos en: conjunto de prueba y de entrenamiento.

```
root@raspberrypi:/home/pi/RecoleccionDatos/Ap
.py
['TempFacialMax' 'RadioPupila']
[['27.23' '3.53553']
 ['27.26' '3.53553']
 ['27.49' '3.53553']
 ['26.699999999999999' '3.53553']
 ['27.15' '3.53553']
 ['27.340000000000002' '4.30116']
 ['26.81' '4.30116']
 ['28.099999999999999' '4.301196']
 ['27.34' '4.301116']
 ['28.850000000000002' '8.7005']
 ['28.94' '8.7005']
 ['28.689999999999999' '8.005']
 ['28.32' '9.92472']
 ['28.07' '9.92472']
 ['28.85' '9.92472']
 ['27.10' '3.80789']
 ['27.26' '3.80789']
 ['27.739999999999999' '3.80789']
```

Figura 80. Matrices con los nombres de los atributos y atributos seleccionados para construir el algoritmo.

Fuente: Autoría

- En la Figura 81 se muestra que el radio de la pupila tiene mayor importancia con relación a la temperatura facial.

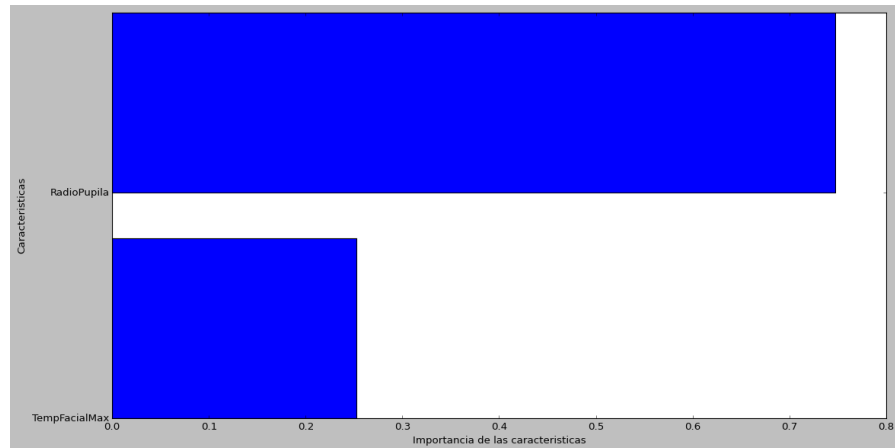


Figura 81. Característica más importante del conjunto de datos al elegir como atributos la temperatura fácil máxima y el radio de la pupila.

Fuente: Autoría

- Seleccionar el clasificador y proceder a entrenar el algoritmo para conocer la exactitud del algoritmo. En la Figura 82 se muestra el resultado de que valor fue capaz de predecir el algoritmo usando el conjunto de prueba.

```

Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico# python prueba1.py
////////Exactitud con el clasificador Arbol de Decision/////
///
0.758620689655
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutomatico#

```

Figura 82. Exactitud del algoritmo de aprendizaje automático supervisado basado en el árbol de decisión usando como atributos solo la temperatura facial máxima y el radio de la pupila.

Fuente: Autoría

- A continuación, se indica las condiciones de cada rama para determinar el estado del conductor, la clasificación se realiza en tres niveles, en el primer nivel se considera el radio de la Pupila el cual debe ser menor o igual a 8.5178, si esta condición se cumple se procederá analizar la temperatura facial para asignar a una clase esos datos, para finalizar con el análisis de la temperatura o radio de la pupila según sea el caso particular.

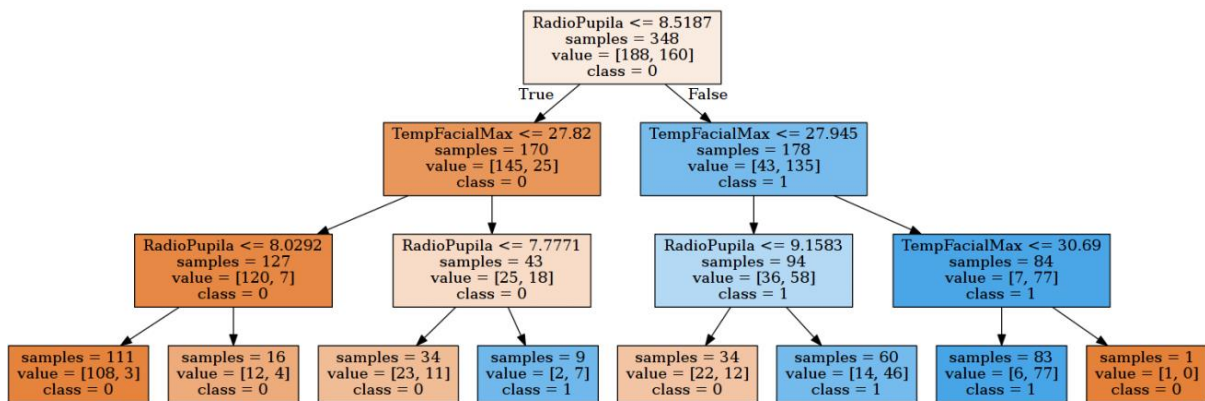


Figura 83. Exactitud del algoritmo de aprendizaje automático supervisado basado en el árbol de decisión usando como atributos solo la temperatura facial máxima y el radio de la pupila.

Fuente: Autoría

- Para finalizar el entrenamiento del algoritmo se procede a realizar una predicción con un nuevo conjunto de datos. El resultado de la predicción se puede apreciar en la Figura 84.

```

Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutom
atico# python prueba1.py
////////Exactitud con el clasificador Arbol de Decision/////
///
0.724137931034
//////////DATOS A PREDECIR//////////
[[ 27.33  9.5 ]]
////////CLASE A LA QUE PERTENECE////////
[ 1.]
root@raspberrypi:/home/pi/RecoleccionDatos/AprendizajeAutom
atico#
  
```

Figura 84. Predicción de un nuevo dato usando como atributos solo la temperatura facial máxima y el radio de la pupila.

Fuente: Autoría

Sin embargo, para que el algoritmo realice una predicción correcta el conductor debe tener altos niveles de alcohol para que se presenten cambios en la pupila y la temperatura fácil.

3.4.8. Mensajes de Alerta Visuales y Auditivos.

Para que el conductor pueda realizar la prueba de Alcoholemia se procede a incorporar en el sistema mensajes de alerta visuales y auditivos, que permitan asimilar las instrucciones que el sistema brinda para los usuarios.

3.4.8.1. Mensajes Visuales.

Los mensajes de alerta que se muestran en la pantalla lcd de la placa Raspberry Pi3 se muestran en la Figura 85, para que se pueda visualizar se hace uso de la función `.show()` del módulo OpenCv; el llamado de la función para visualizar las imágenes se realiza en los diferentes scripts según corresponda el mensaje de Alerta.



CONDUCTOR EBRIO.jpg



CONDUCTOR SOBRIO.jpg



ENCENDIDO VEHICULO BLOQUEADO.jpg



PRESIONE EL BOTON.jpg

Figura 85. Mensajes de alerta visuales para el Conductor del vehículo.

Fuente: Autoría

3.4.8.2. Mensajes Auditivos.

Para los mensajes de alerta auditivos, se hace uso del Festival que es un sintetizador de texto a voz, con el cual se puede hacer que lea documentos o páginas web, además de poder dar instrucciones de las aplicaciones, se emplea módulo subprocesses para llamar al puerto que está conectado el parlante y hacer uso de Festival. En la Figura 86 se muestra la función que se emplea para el audio y los parámetros que se asignan. La Tabla 31 contiene el texto de todos los mensajes de audio implementados en el sistema.

```
#Mensaje que se emitira  
VozInformacion = "El encendido del vehiculo se encuentra bloqueado realice la prueba de alcoholemia para desbloquear"  
#Se llama el programa festival a traves del modulo subprocesses  
subprocess.call('echo '+VozInformacion+'|festival --tts --language spanish', shell=True)
```

Figura 86. Mensajes de alerta auditivos para el Conductor del vehículo.

Fuente: Autoría

Tabla 31. Mensajes de Audio para el sistema.

FUNCIÓN	MENSAJE
Mensaje al encender el sistema	VozInformación = '"El encendido del automóvil se encuentra bloqueado realice la prueba de alcoholemia para desbloquear"'
Instrucción para realizar la prueba de alcoholemia	VozIndicaciones = '"Presione el botón para iniciar la prueba de alcoholemia"'
Mensaje de inicio de prueba	VozInicio = '"Iniciando prueba de alcoholemia"'
Instrucción para medir la temperatura facial	VozTemperaturaFacial = '"Mire al frente para proceder a medir la temperatura facial"'
Instrucción cuando no se ha detectado el rostro para almacenarlo	VozRostro = '"Mire al frente no se ha detectado el rostro"'
Mensaje cuando no se puede medir la temperatura fácil	VozUbicarse = '"No se completó el almacenamiento de la temperatura mirar fijamente a la cámara"'
Mensaje de Rostro detectado al medir la temperatura facial	VozRostroDetectado = '"Rostro del Conductor Detectado"'
Mensaje cuando la temperatura facial se almacena	VozTempFacial = '"Temperatura Facial Almacenada"'
Instrucción para proceder a medir el diámetro de la pupila	VozPupila = '"Mirar fijamente a la cámara para medir el diámetro de la pupila"'
Instrucción para medir la pupila cuando no se ha detectado los ojos	VozPupilaNoDetectada = '"Por favor acérquese a la cámara para medir el diámetro de la pupila"'

Mensaje cuando los ojos han sido detectados	VozOjosDetectados = '"Ojos del conductor detectados"'
Mensaje cuando la pupila ha sido detectada	VozPupilaDetectada = '"Pupila del Conductor Detectada"'
Mensaje cuando la pupila del Conductor ha sido almacenada	VozPupila = '"Diámetro de la pupila almacenado"'
Instrucción para medir el nivel de alcohol en el aliento.	VozAlcohol = '"Sople para medir su nivel de alcohol"'
Instrucción cuando no se ha detectado que el conductor ha soplado	VozHumedad = '"Por favor Sople Nuevamente"'
Mensaje de que no se ha detectado que el conductor ha soplado.	VozNoSoplo = '"Imposible medir el nivel de alcohol"'
Mensaje del nivel de alcohol almacenado.	VozNivelAlcohol = '"Nivel de alcohol almacenado"'
Mensaje de que no se ha podido detectar que el Conductor ha soplado	VozNoSoplo = '"Automóvil bloqueado realice la prueba de alcoholemia correctamente"'
Mensaje cuando el conductor se encuentra sobrio.	VozEstadoSobrio = '"Conductor Sobrio ya puede encender el automóvil"'
Mensaje de advertencia	Advertencia = '"Advertencia en caso de que se detecte alcohol en el aire se procederá a comunicar al ECU novecientos once y el automóvil será apagado"'
Instrucción para realizar el reconocimiento facial.	VozSuplantacion = '"Mire al frente para realizar el reconocimiento facial"'
Mensaje cuando el conductor es suplantado.	ConductorSuplantado = '"El conductor ha sido suplantado el automóvil se apagará en pocos segundos"'

Mensaje cuando se ha detectado alcohol en el aire.	VozAlcoholAmbiente = '"Se ha detectado alcohol en el ambiente se procederá a comunicar al ECU novecientos once y se apagará el automóvil"'
Instrucción cuando no se ha detectado el rostro del Conductor para realizar el reconocimiento Facial	intentos = '"No se ha detectado el rostro mire al frente"'
Mensaje cuando no se ha detectado el Conductor.	Vozsinrostro = '"Imposible detectar el rostro se procederá a apagar el automóvil"'
Mensaje cuando el Conductor se encuentra Ebrio.	VozEstadoEbrio = '"Conductor Ebrio por favor cambie de conductor"'

Fuente: Autoría.

3.4.9. Reconocimiento facial del Conductor.

3.4.9.1. Aprender rostro del conductor.

Para realizar el reconocimiento facial del conductor del vehículo es necesario que crear una pequeña base de datos con los rostros del conductor para ello se emplea el script denominado CapturaConductor.py el código de programación se encuentra en el Anexo 20. Para que el algoritmo de reconocimiento trabaje con mayor exactitud es necesario que se almacenen 30 imágenes. Se emplea el código en el que se detecta el rostro, después de encontrar el rostro se procede a recortarlo de la imagen y almacenarlo en una carpeta. La Figura 87 muestra los rostros del conductor almacenados, es necesario que las imágenes sean guardadas en formato .pgm para realizar el reconocimiento esto se debe a que este formato se emplea específicamente para imágenes en escala de grises.

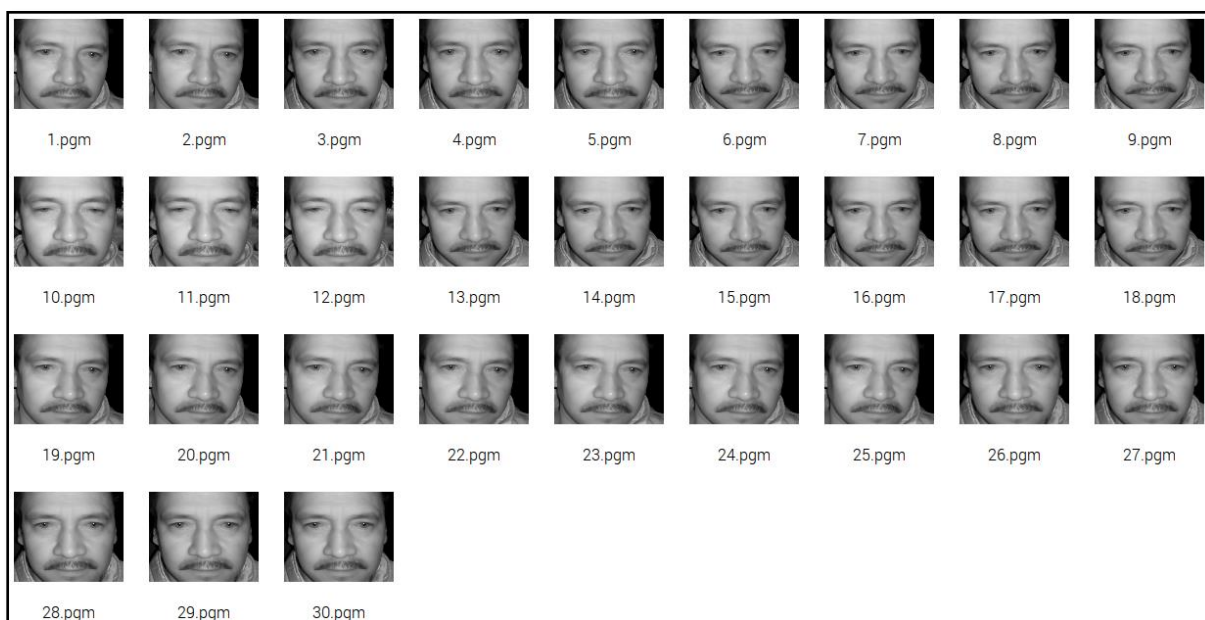


Figura 87. Rostros del conductor para el reconocimiento facial.

Fuente: Autoría

Las capturas del conductor designado se realizan al iniciar la prueba de alcoholemia para evitar que posteriormente el conductor sea suplantado.

3.4.9.2. Reconocimiento facial.

El algoritmo de reconocimiento que se emplea es el Fisherfaces, este algoritmo de Fisherfaces aprende una matriz de transformación específica de clase, por lo que no capturan la iluminación tan obviamente como el método de Eigenfaces, por lo que emplea el Análisis discriminante para encontrar las características faciales y discriminar entre las personas. El rendimiento de este algoritmo depende en gran medida de los datos de entrada es por esto por lo que es necesario capturar como mínimo 30 rostros del conductor que está realizando la prueba.

Fisherfaces permite una reconstrucción de la imagen proyectada, pero dado que solo identificamos las características para distinguir entre sujetos, no puede esperar una buena reconstrucción de la imagen original. El script de programación para el reconocimiento facial se denomina PruebaReconocimiento.py y éste se encuentra en el Anexo 21. La Figura 88 muestra el resultado del reconocimiento del conductor designado. Mientras que en la Figura 89 al cambiar de conductor es detectado como Desconocido.



Figura 88. Verificación del conductor designado a través del Reconocimiento facial.

Fuente: Autoría

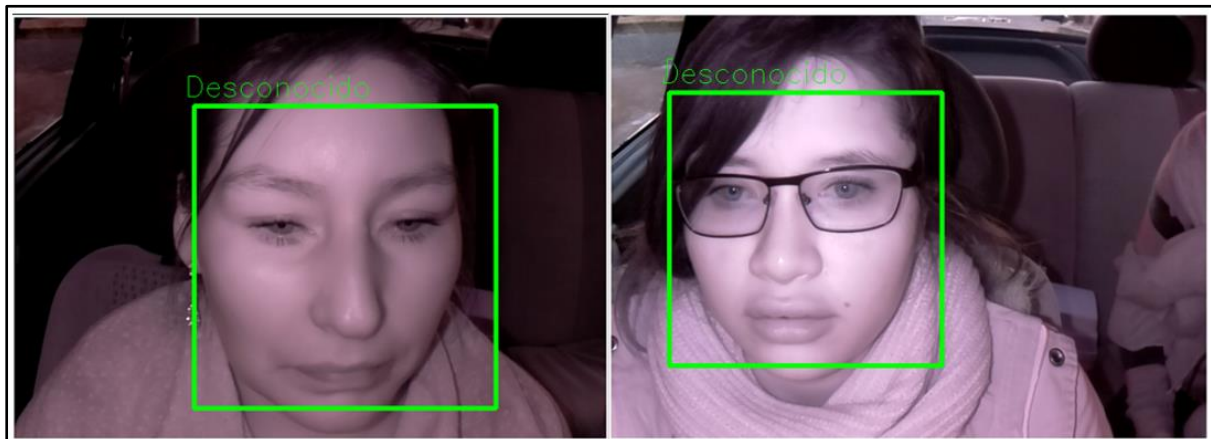


Figura 89. Conductor detectado como desconocido a través del Reconocimiento Facial.

Fuente: Autoría

El reconocimiento facial se lo realiza en tiempo real e inicia al finalizar la prueba de alcoholemia cuando el conductor haya sido detectado como sobrio.

Una vez terminado el desarrollo de software se finaliza la Fase de Prueba Unitaria de la metodología de desarrollo del modelo en V; en la cual se verificó el funcionamiento adecuado de manera individual de las partes que conforman el Detector de Alcoholemia.

3.4.10. Desarrollo de Hardware.

En la fase de implementación de la metodología de desarrollo en base al modelo en V se materializa el diseño del proyecto y la unificación del sistema, para ello se realiza el desarrollo del hardware que permite la integración de las partes que requiere el sistema en las distintas secciones, para lo cual se realiza el montaje de las placas, la conexión eléctrica para la alimentación de estas en el vehículo y la conexión eléctrica para la activación de la bomba de combustible.

3.4.10.1. Conexión eléctrica para alimentación de la placa Raspberry Pi 3.

Para la alimentación del sistema se procede a conectar el convertidor de 12 V CD a 5 V CD a través del encendedor del vehículo, para que el sistema funcione solo cuando el vehículo se encuentre encendido. Antes de realizar el cableado es necesario desconectar la batería por seguridad, debido a que el vehículo en el que se implementará el sistema ya tiene instalado un encendedor de cigarrillos desde la fuente activa de alimentación al terminal central en la toma y colocado el respectivo fusible, se procede a realizar un empalme del terminal positivo y la tierra. La Figura 90 muestra la realización del empalme de los dos terminales.



Figura 90. Empalme de los terminales positivo y tierra del Encendedor de cigarrillos.

Fuente: Autoría

Al finalizar el empalme se vuelve a conectar la batería y se procede a realizar el montaje del encendedor en el lado izquierdo inferior del tablero del vehículo, a continuación, en la Figura 91 se muestra el cableado de la toma eléctrica para el sistema.

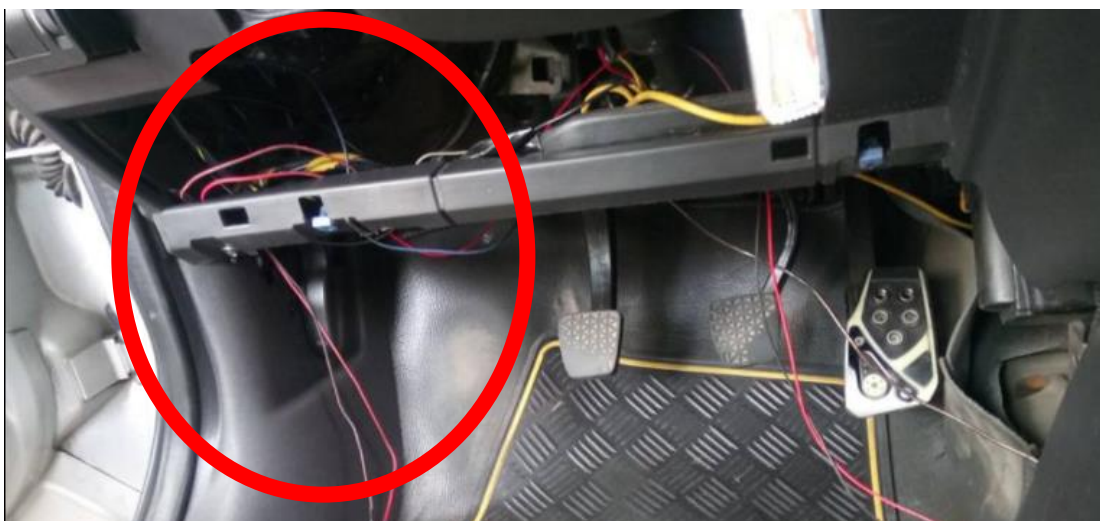


Figura 91. Cableado eléctrico.

Fuente: Autoría

Antes de realizar la conexión del encendedor de cigarrillos, se procede a medir el voltaje de salida de los dos terminales; la Figura 92 muestra el voltaje que se obtiene con el multímetro.

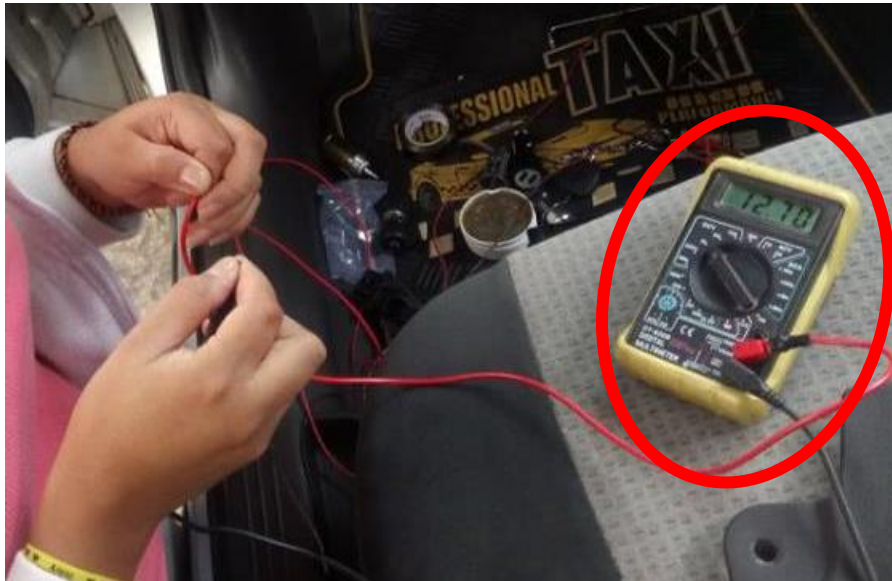


Figura 92. Comprobación del voltaje.

Fuente: Autoría

En la Figura 93 se observa la instalación del encendedor de cigarrillos en la parte inferior izquierda del tablero del puesto del conductor del vehículo.



Figura 93. Conexión eléctrica para el sistema.

Fuente: Autoría

3.4.10.2. Alineación del sensor de temperatura MLX90621 con la cámara para que apunten al mismo objetivo.

Se realiza el montaje del sensor de temperatura y la cámara para que los dos elementos apunten hacia la misma dirección, con la finalidad de que no exista una falsa toma de datos, el sensor se ubica bajo el módulo cámara Pi Noir en posición vertical. La Figura 94 muestra el montaje de la cámara, el sensor de temperatura y la luz infrarroja en una sola pieza.

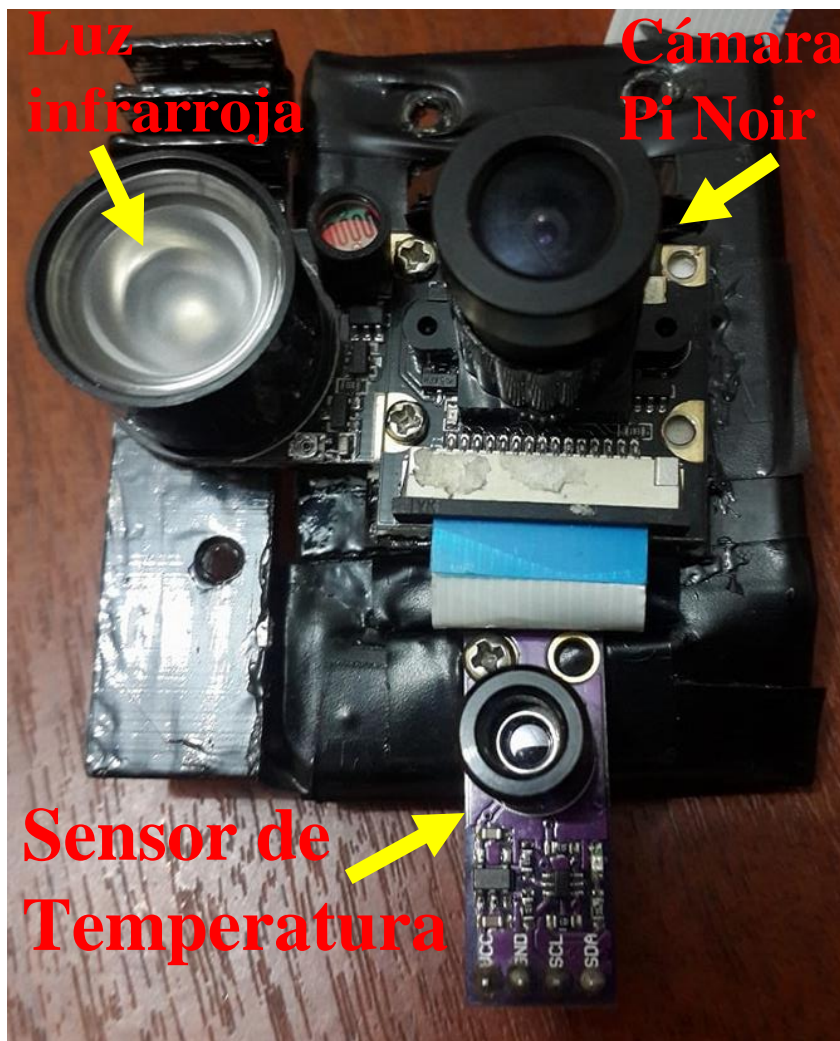


Figura 94. Montaje del sensor de temperatura, la cámara y la luz infrarroja.

Fuente: Autoría

3.4.10.3. Montaje del sensor de gas y humedad.

Cuando se inició las pruebas de integración del sistema se determinó que es necesario implementar un sensor que determine que el conductor si ha soplado, al realizar una prueba de aliento se puede medir la humedad del aliento. Considerando esta variable se procede a implementar el sensor DHT11 junto al sensor de gas MQ-3 para asegurar que el conductor sople cuando se lo solicita, la función que realiza este sensor es medir la humedad, debido a su alta fiabilidad, precisión del 95% y bajo precio se seleccionó este sensor, en la Tabla 32 se muestran los parámetros que presenta este sensor.

Tabla 32. Parámetros del sensor DHT11.

SENSOR DHT11	
PARAMETROS	
Fiabilidad.	Alta
Estabilidad.	Humedad: $\pm 1\%$ /año Temperatura: $\pm 1^\circ\text{C}$ /año
Tiempo de respuesta.	Humedad: 6-15 segundos Temperatura: 6-30 segundos
Precisión.	Humedad: 95% Temperatura: 90%
Precio.	\$5.90
Disponibilidad.	Quito

Fuente: Adaptado de (Ortiz Arciniega, 2017)

El sensor DHT11 y el MQ3 se colocan en la misma caja para realizar las mediciones respectivas. La Figura 95 indica el montaje de los dos sensores. La programación del script para medir la humedad en el aliento se encuentra en el Anexo 22.

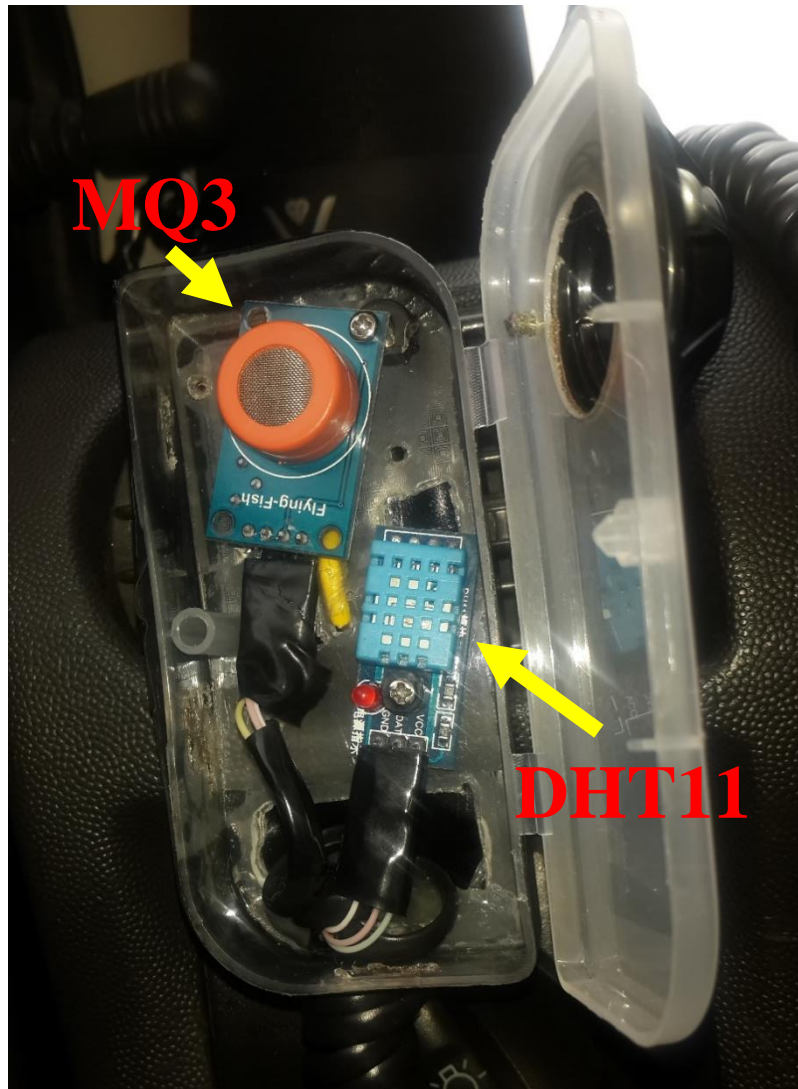


Figura 95. Montaje del sensor de Alcohol MQ3 y el sensor de Humedad DHT11.

Fuente: Autoría

3.4.10.4. Montaje del botón para iniciar el sistema.

Se realiza la implementación de un botón para que inicie la prueba de alcoholemia, debido a que se considera que no siempre el conductor va a encender el vehículo al momento de subirse al automóvil. En la Figura 96 se muestra el montaje del botón de activación en el vehículo, el cual se encuentra ubicado en el armazón de la puerta del conductor cerca al parabrisas del vehículo.



Figura 96. Montaje del botón de activación del sistema.

Fuente: Autoría

3.4.10.5. Conexión eléctrica para activar y desactivar la bomba de combustible para controlar el encendido del vehículo.

Después de analizar las ventajas y desventajas de los sistemas de encendido del automóvil en la Tabla 6 se procede a seleccionar el Sistema de bloqueo a través de la bomba de combustible, el inconveniente que presenta de que puede ocasionar pérdida de potencia del vehículo causada por la falta de tensión, se lo soluciona verificando el voltaje a la salida del módulo Relé que se emplea para la activación y desactivación de esta.

En la Figura 97 se muestra el circuito para la activación y desactivación de la bomba de combustible.

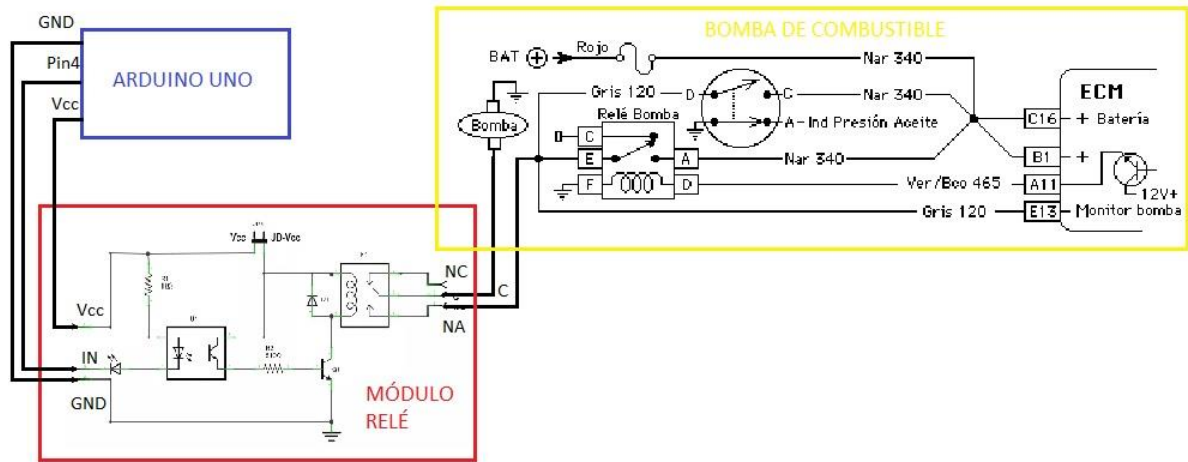


Figura 97. Circuito para Activar la Bomba de Combustible.

Fuente: Autoría

Para controlar la activación de la bomba de combustible se procede a empalmar los cables de alimentación de esta. La Figura 98 muestra el empalme de los cables de alimentación y tierra de la cámara de combustible que se encuentra ubicada debajo del asiento de la parte posterior del vehículo.



Figura 98. Empalme de los terminales de la cámara de combustible.

Fuente: Autoría

Al finalizar el empalme se procede a realizar la conexión de los cables al módulo relé de dos canales con el objetivo de activar de manera automática a la bomba de combustible cuando el resultado de la prueba de alcoholemia determine que el conductor se encuentra sobrio, a continuación, en la Figura 99 se muestra la conexión de los cables que vienen de la bomba de combustible hacia el módulo relé.

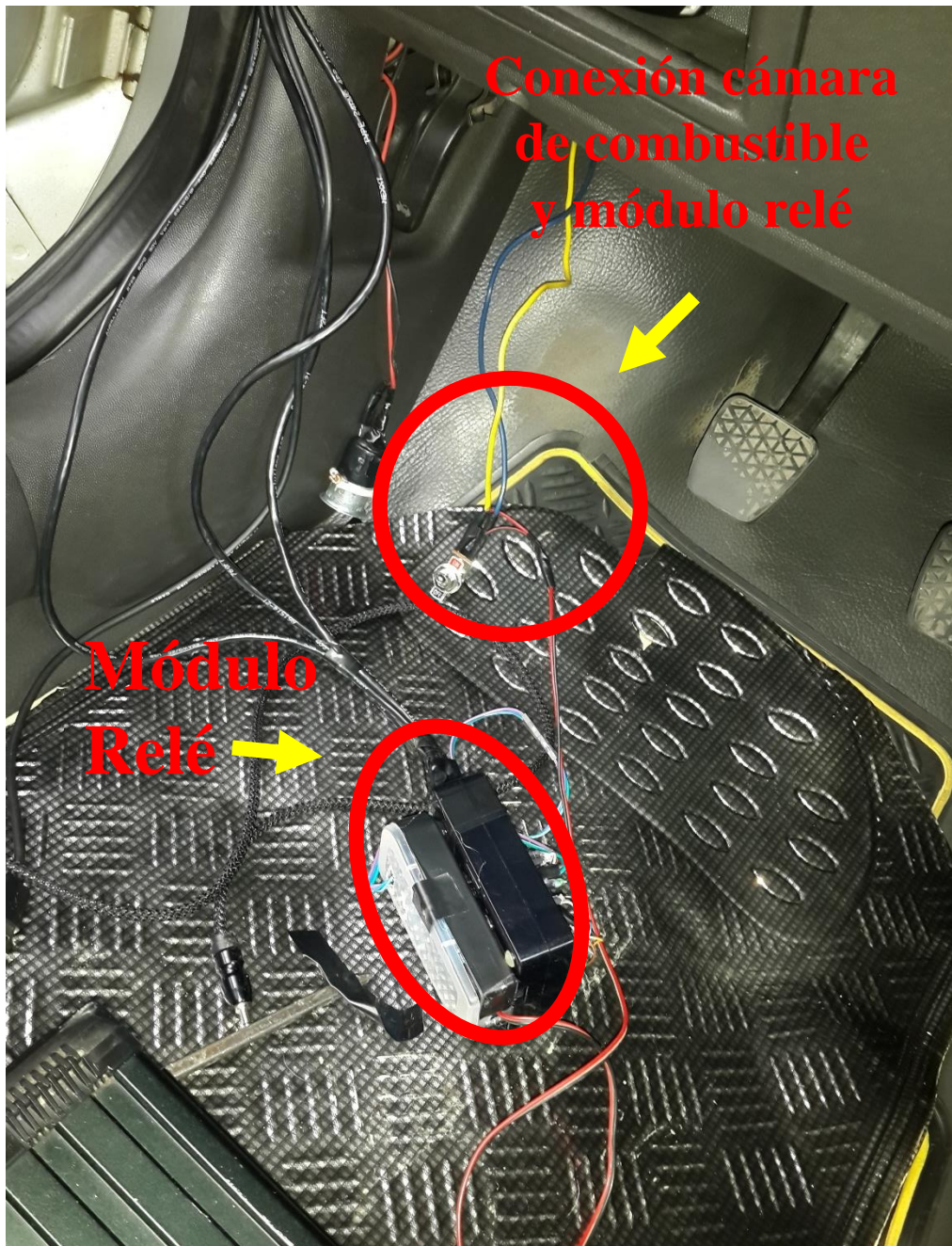
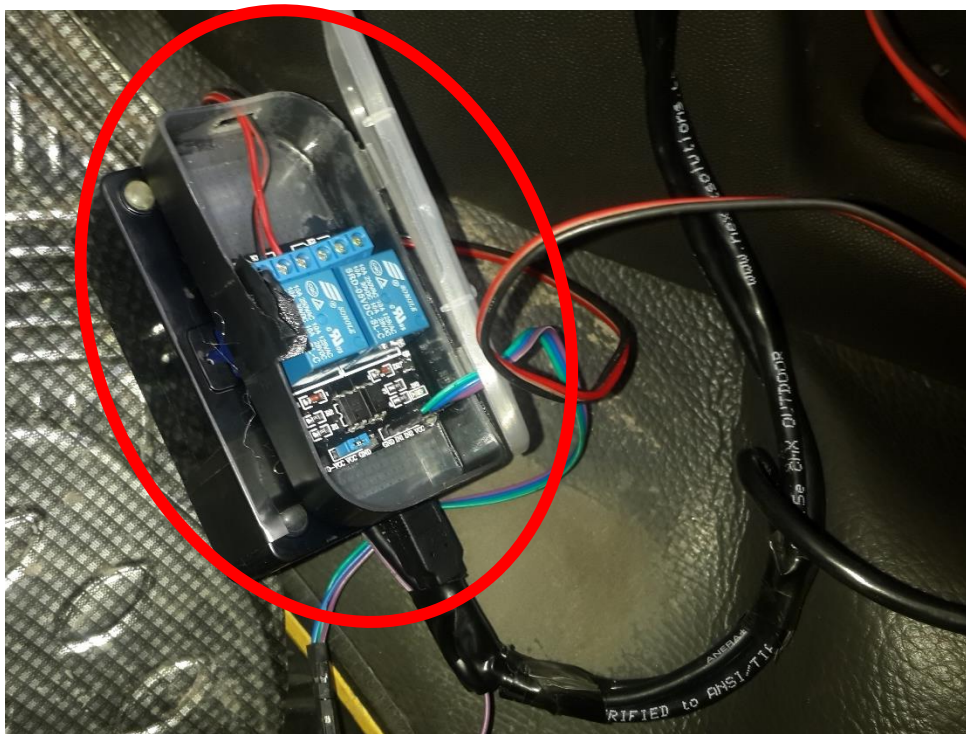


Figura 99. Conexión del módulo relé y la bomba de combustible.

Fuente: Autoría

En la Figura 100 se muestra el módulo Relé de dos canales empleado para la activación de la bomba de combustible.



*Figura 100. Módulo Relé que se implementó en el vehículo.
Fuente: Autoría*

3.4.10.6. Activación del botón de pánico del ECU 911.

Para proceder a realizar la activación del botón de pánico del Servicio Integrado de Seguridad ECU 911, se envió un oficio dirigido al Supervisor de Monitoreo del Transporte Seguro Ing. Danny Ontaneda a través del Gerente de la Cooperativa de Taxis “Atahualpa” el Sr. Álvaro Cano. En el Anexo 23 se adjunta la Respuesta del Oficio con la autorización para poder activar el botón de pánico. La activación se realizará cuando el conductor proceda a ingerir bebidas alcohólicas después de haber encendido el vehículo, se consideró esta opción solo en el caso en el cual al momento de realizar la prueba de alcoholemia el conductor se encuentre sobrio, pero luego de haber encendido el vehículo proceda a ingerir alcohol. Para realizar la activación se procede a activar el segundo canal del módulo relé por 3 segundos,

además se emitirá una alerta auditiva que indique que se ha detectado alcohol en el aire y se procede a desactivar la bomba de combustible conectada al canal uno del módulo. El diagrama de conexión del sistema al pulsador del kit de Seguridad de Proyecto Transporte Seguro se muestra en la Figura 101.

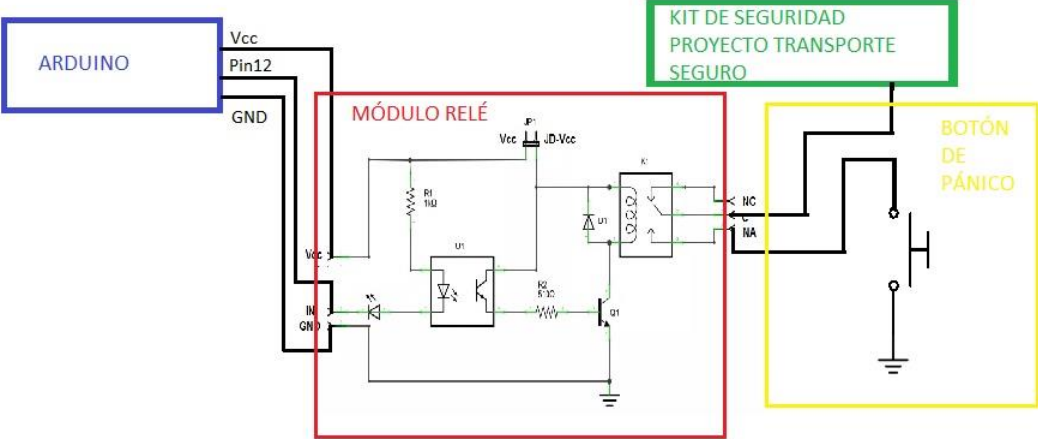


Figura 101. Diagrama de Conexión para el Botón de Pánico.

En la Figura 102 se muestra la conexión del módulo relé al botón de pánico implementado por el Servicio Integrado de Seguridad ECU 911.

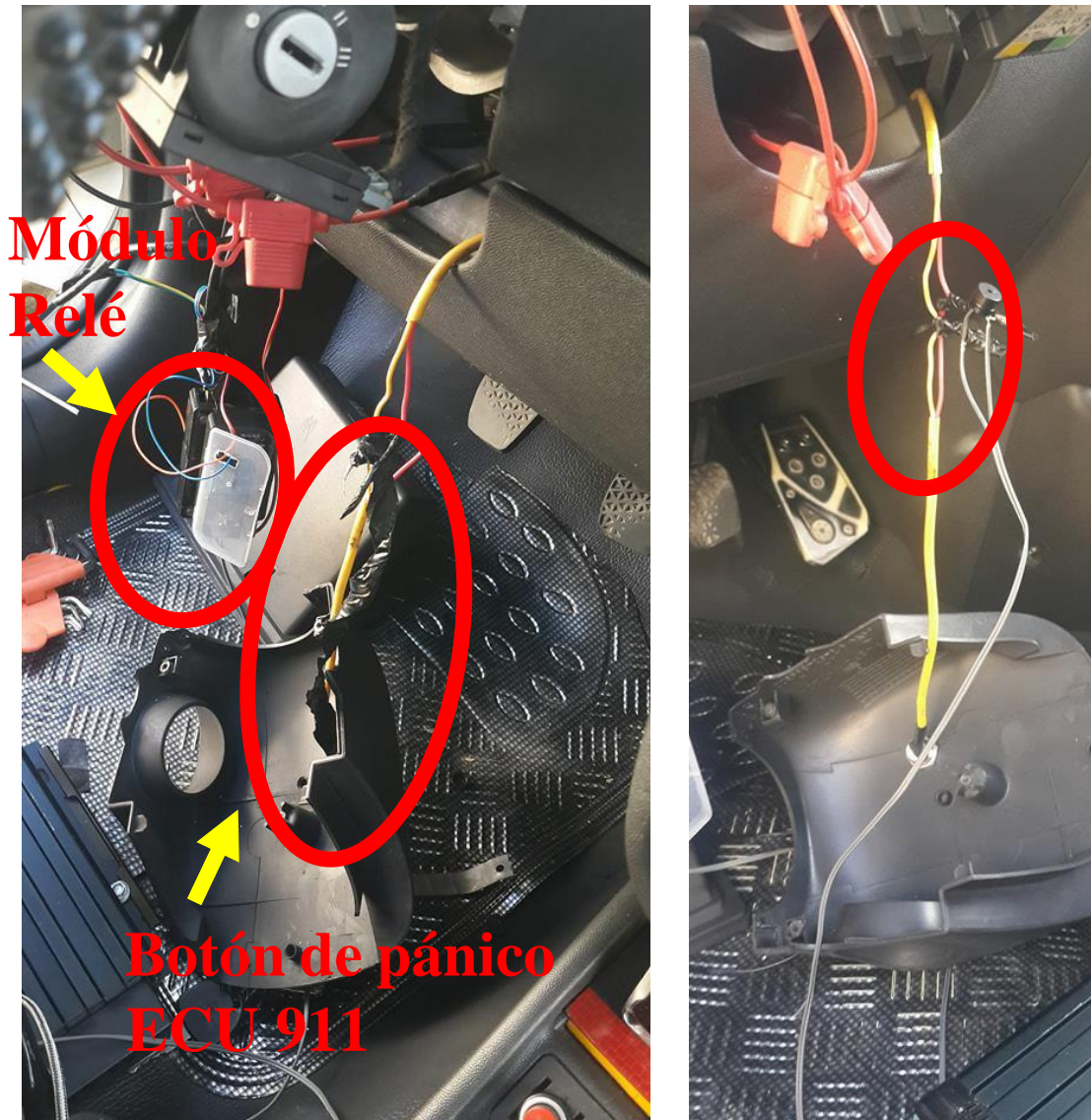


Figura 102. Conexión del módulo relé al botón de pánico implementado por el Servicio Integrado de Seguridad ECU 911.

Fuente: Autoría

3.4.10.7. Implementación del Detector de Alcoholemia en el vehículo.

Para la implementación del proyecto en el vehículo se toma como referencia la Tabla 9 en la cual se encuentran las dimensiones de la vista interior del vehículo. Debido a que para que se comuniquen los sensores y la cámara a las placas de procesamiento es necesario ocultar los cables en la puerta del conductor, a continuación, en la Figura 103 se muestra el cableado

del proyecto con la finalidad de evitar que los colores llamen demasiada atención hacia el conductor y que no se obstaculice la visión de la carretera.

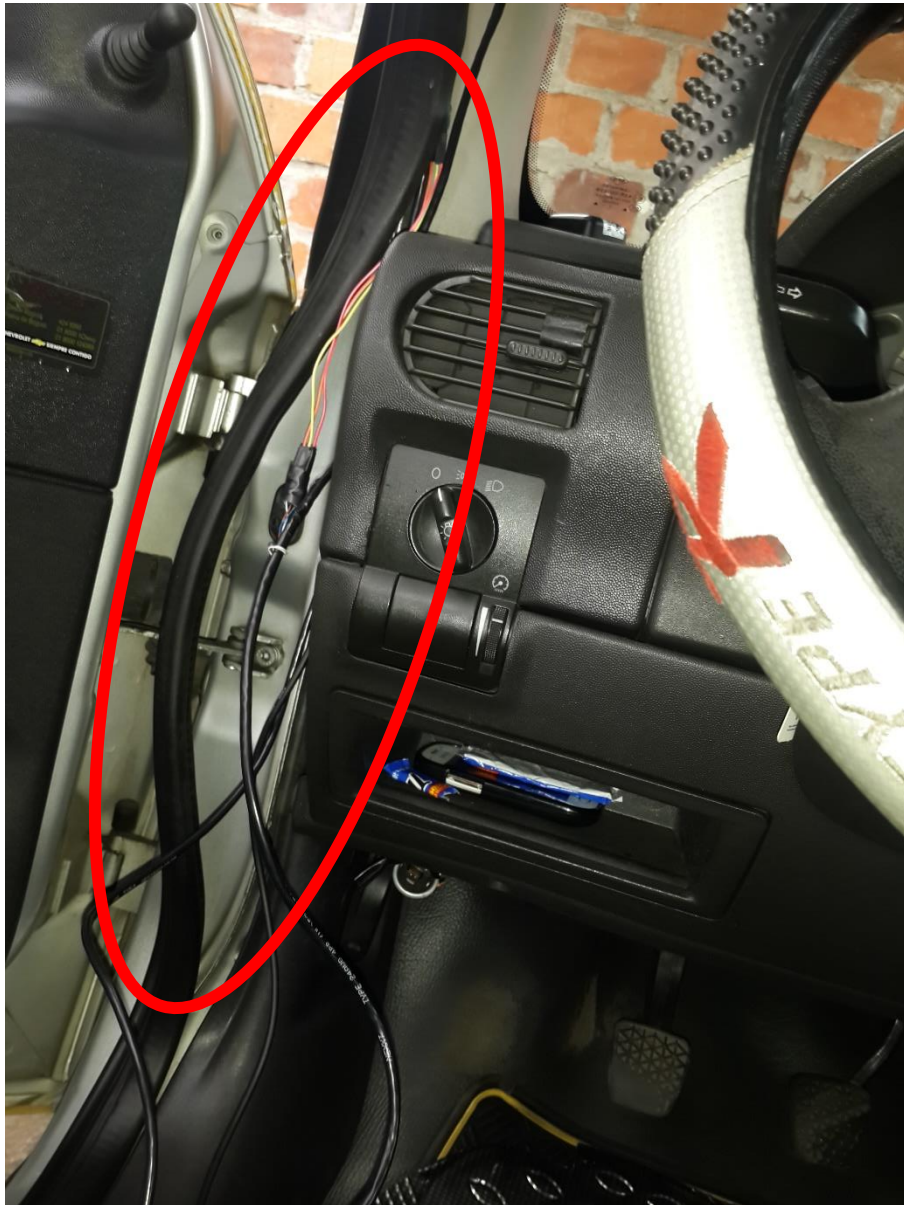


Figura 103. Cableado del Sistema.

Fuente: Autoría

De la Tabla 9 se toman como referencia las siguientes distancias: la distancia del recorte del bastidor de la puerta delantera la cual es de 0,84 metros, la altura del bastidor de la puerta que es de 0.934 metros y la separación entre las columnas del tope de las puertas delanteras que tiene una distancia de 1.327 metros.

Para la implementación del sensor de temperatura y la cámara se considera el alto, ancho y la profundidad. En la Figura 104 se muestra las dimensiones que se tomara en cuenta para la implementación.



Figura 104. Alto ancho y profundidad para instalar la cámara y el sensor de temperatura.

Fuente: Autoría

La distancia de la profundidad es de 0,50 metros debido a que es necesario considerar que el proyecto analiza variables faciales del conductor, por lo que a una distancia más corta el sistema no detectara el rostro en la imagen que captura la cámara.

Para el alto se considera la distancia medida desde el techo hacia el piso, la cual es de 0,12 metros, se obtuvo este valor realizando pruebas para la detección del rostro considerando que no todos los conductores tienen la misma estatura.

Para terminar el ancho se considera que la cámara y el sensor apunten al rostro del conductor, se determinó mediante pruebas que la distancia óptima para que sean capturados las variables a faciales del conductor es de 0,40 metros desde la puerta del conductor.

En la Figura 105 se muestra la ubicación de la cámara y el sensor de temperatura en el vehículo según las dimensiones establecidas anteriormente.

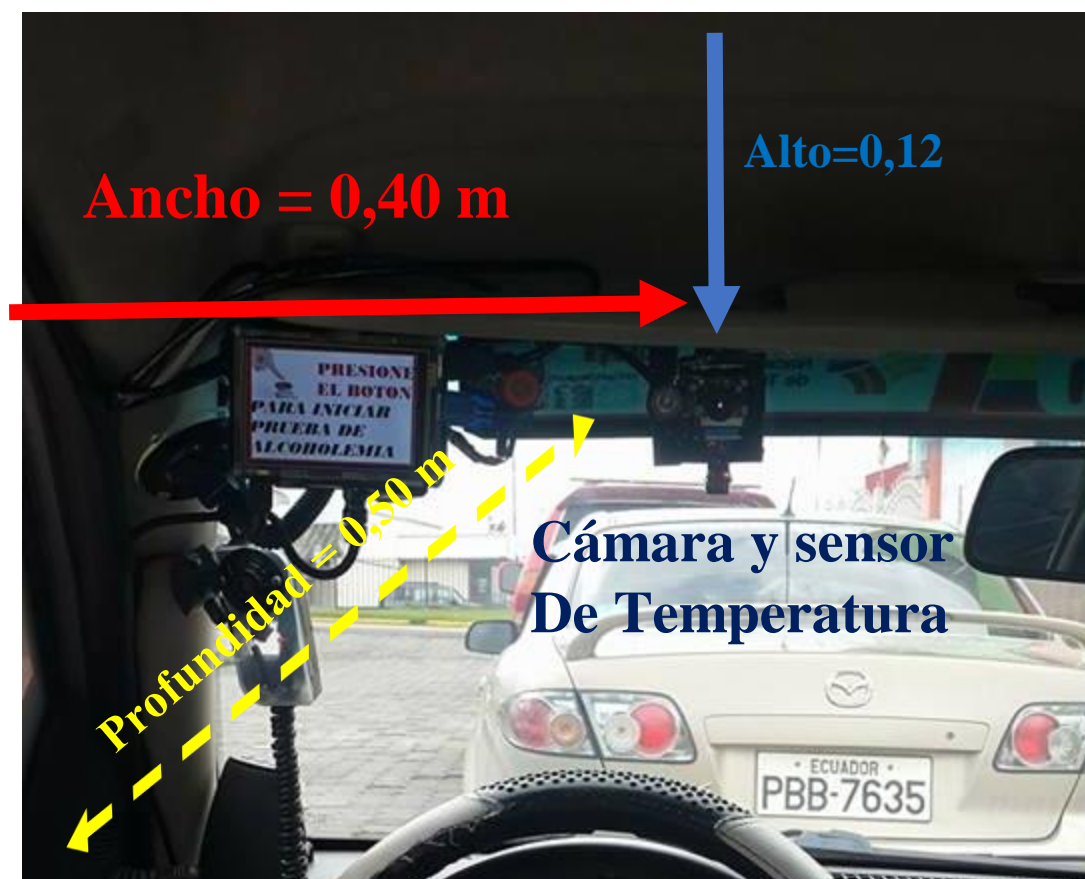


Figura 105. Ubicación de la cámara y el sensor de temperatura en el interior del vehículo.

Fuente: Autoría

Para la implementación de la pantalla Lcd en la que se muestran los mensajes visuales se tienen las siguientes dimensiones: alto = 0,12 metros, ancho = 0,06 metros y profundidad de 0,50 metros. En la Figura 106 se muestra la ubicación de la pantalla.



Figura 106. Ubicación de la pantalla LCD en el interior del vehículo.

Fuente: Autoría

Finalmente, en la Figura 108 se muestra todos los elementos que intervienen en la fase de pruebas de integración de la metodología de desarrollo del modelo en V, en la cual se realiza la unificación de las partes que se verificaron en la fase de prueba unitaria. En la Figura 107 se muestra un diagrama de bloques de cómo está conectado el sistema en el interior del vehículo, mientras que la Tabla 33 contiene los nombres de las partes que se indican en la Figura 108.

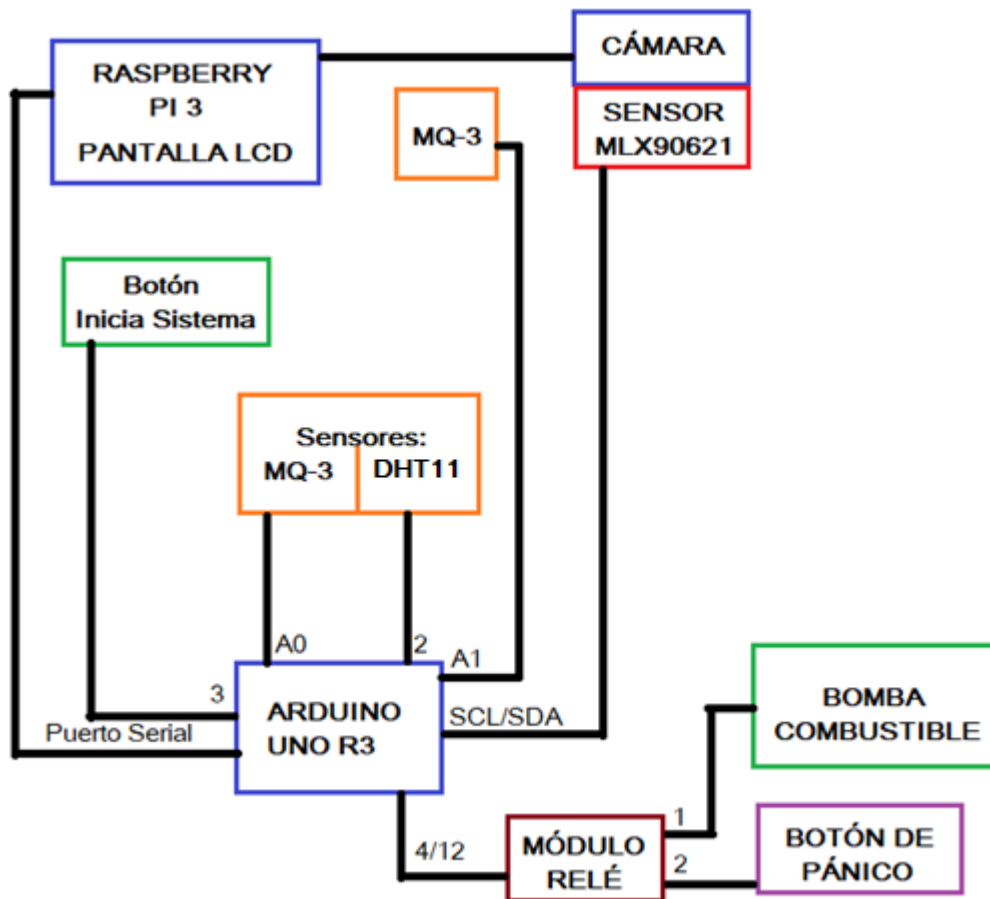


Figura 107. Diagrama de bloques de cómo está conectado el sistema en el interior del vehículo.

Fuente: Autoría



Figura 108. Elementos que intervienen en el Detector de alcoholemia.

Fuente: Autoría

Tabla 33. Partes del Detector de alcoholemia

Nomenclatura		Nombre
1	A	Pantalla LCD para mostrar los mensajes visuales
2	B	Sensor de Temperatura y módulo cámara Pi Noir
3	C	Botón de activación del Sistema
4	D	Parlante para los mensajes auditivos
5	E	Sensor de MQ-3 y sensor de Humedad
6	F	Sensor MQ-3 para medir nivel de alcohol en el aire

Fuente: Autoría

Capítulo IV. Pruebas de funcionamiento y resultados

En el presente capítulo se indica las pruebas de funcionamiento del software y hardware del detector de alcoholemia, se desarrollan los resultados obtenidos en el estudio y cuáles fueron las novedades y problemas en su realización, dentro de las pruebas que se realizan se encuentran: la activación y desactivación del encendido del vehículo, la ubicación del sensor de temperatura y la cámara para captar la temperatura facial y el diámetro de las pupilas del conductor, los mensajes de alerta visuales y auditivos que garanticen la aceptabilidad del sistema por los usuarios; para la recolección de resultados se procederá a emplear técnicas para recopilación de datos con la finalidad de determinar los resultados obtenidos y la aceptación de los conductores del proyecto. Con las pruebas de funcionamiento y los resultados obtenidos se finaliza la metodología de desarrollo basada en el modelo en V en la fase de Pruebas de operación y validación.

4.1. Análisis de resultados obtenidos.

Para conocer los resultados obtenidos y la aceptación de los conductores del trabajo de titulación denominado “DETECTOR DE ALCOHOLEMIA PARA CONDUCTORES QUE ANALIZA VARIABLES FACIALES Y AMBIENTALES DEL AUTOMÓVIL MEDIANTE EL APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA REDUCCIÓN DE ACCIDENTES DE TRÁNSITO” se emplea un método de la investigación descriptiva para recolectar los datos necesarios, este método corresponde a la encuesta. Se selecciona la encuesta debido a que se requiere conocer las ideas, opiniones y datos reales acerca de la aceptación que tiene el proyecto.

4.1.1. Encuesta.

Se estable como población la Cooperativa de taxis Atahualpa de la ciudad de Tulcán la cual esta dividida en tres grupos A, B y C, por lo que para la encuesta se toma como muestra a los señores conductores del grupo C, debido a que el vehículo en el cual se implementará el proyecto es parte de este grupo, lo que se busca conocer principalmente es la aceptación de las instrucciones, la ubicación del detector de alcoholemia, el sistema de bloqueo y activación del botón de pánico y las variables que se censa en el proyecto por los conductores. Es necesario que el conductor del vehículo realice una prueba del funcionamiento del sistema para proceder posteriormente a llenar la encuesta. En la Figura 109 se indica el momento en el que un conductor del grupo C realiza la prueba de funcionamiento del sistema. En el Anexo 24 se adjunta la Autorización por parte del Sr. Álvaro Cano Gerente de la Cooperativa de Taxis “Atahualpa” para realizar las pruebas de funcionamiento del sistema.



Figura 109. Prueba de funcionamiento del Detector de Alcoholemia.

Fuente: Autoría

Para el desarrollo de la encuesta se formulan 9 preguntas de opción múltiple de las cuales 6 son preguntas cerradas de elección según los criterios del conductor, mientras que las tres últimas son preguntas semiabiertas en las que se debe justificar la opción de la respuesta elegida. Las preguntas de respuestas cerradas se diseñaron con la finalidad de facilitar la cuantificación y tabulación de resultados, mientras que las preguntas semiabiertas se desarrollaron para proporcionar a los señores encuestados la libertad de responder según su criterio permitiendo obtener respuestas más profundas que en un inicio no estaban consideradas dentro de los formularios de tabulación. Para poder apreciar el formato de la encuesta ver el Anexo 25.

4.1.1.1. Identificación de la población.

Para determinar la población a la cual se va a aplicar la encuesta se utiliza la Ecuación 19 la cual corresponde a la fórmula propuesta por Fisher y Navarro (1994).

$$n = \frac{N \times K^2 \times p \times q}{(e^2 \times (N - 1) + K^2 \times p \times q)}$$

*Ecuación 19. Cálculo de muestra, propuesto por Fisher y Navarro.
Fuente: (Torres & Paz, 2014)*

Donde:

n = el tamaño de la muestra.

N = tamaño de la población.

K^2 = nivel de confianza

e = error porcentual

e = Límite aceptable de error muestra que, generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% (0,01) y 9% (0,09), valor que queda a criterio del encuestador. (Torres & Paz, 2014)

p = probabilidad a favor

q = probabilidad en contra

Parámetros utilizados:

Para este caso de estudio se tiene un tamaño de la población de 135 señores conductores del grupo C de la Cooperativa de taxis Atahualpa de la ciudad de Tulcán. Los parámetros considerados para la fórmula de Fisher y Navarro son los siguientes: el valor de N corresponde a los 135 señores conductores, para el nivel de confianza K se ha tomado un valor del 95,5% ($K=2$) esperando que únicamente el 4,5% de los resultados obtenidos no correspondan a información verídica; tanto para el valor de la probabilidad a favor p como el valor de la probabilidad en contra q se tomó el valor de 0,5 considerando obtener datos reales de los encuestados, el error porcentual e con un valor de $e=0,10$ teniendo un porcentaje de respuestas correctas del $\pm 10\%$ de los resultados totales obtenidos. El cálculo de la muestra aplicando la ecuación 3 se presenta a continuación:

$$n = \frac{135 \times 2^2 \times 0,5 \times 0,5}{(0,1^2 \times (135 - 1) + 2^2 \times 0,5 \times 0,5)}$$

$$n = 57,6923 \cong 58 \text{ conductores.}$$

Según el cálculo de muestra se determina que el número de conductores encuestados debe de ser aproximadamente 58 conductores de taxis en la Figura 110 se muestra un ejemplo de cuando un señor conductor llenaba la encuesta. En el Anexo 26 se encuentran parte de los encuestados, no se adjuntan todas las imágenes debido a que algunos conductores manifestaron que no desean que su rostro aparezca en el documento. Las respuestas de las encuestas se encuentran disponibles en el siguiente enlace: https://utneduec-my.sharepoint.com/:b:/g/personal/lgportilla_utn_edu_ec/EZaV9iQOfrxFhUX994MoEh0BxbroSJseY2sbgLAQs_UX0w?e=7y38dF

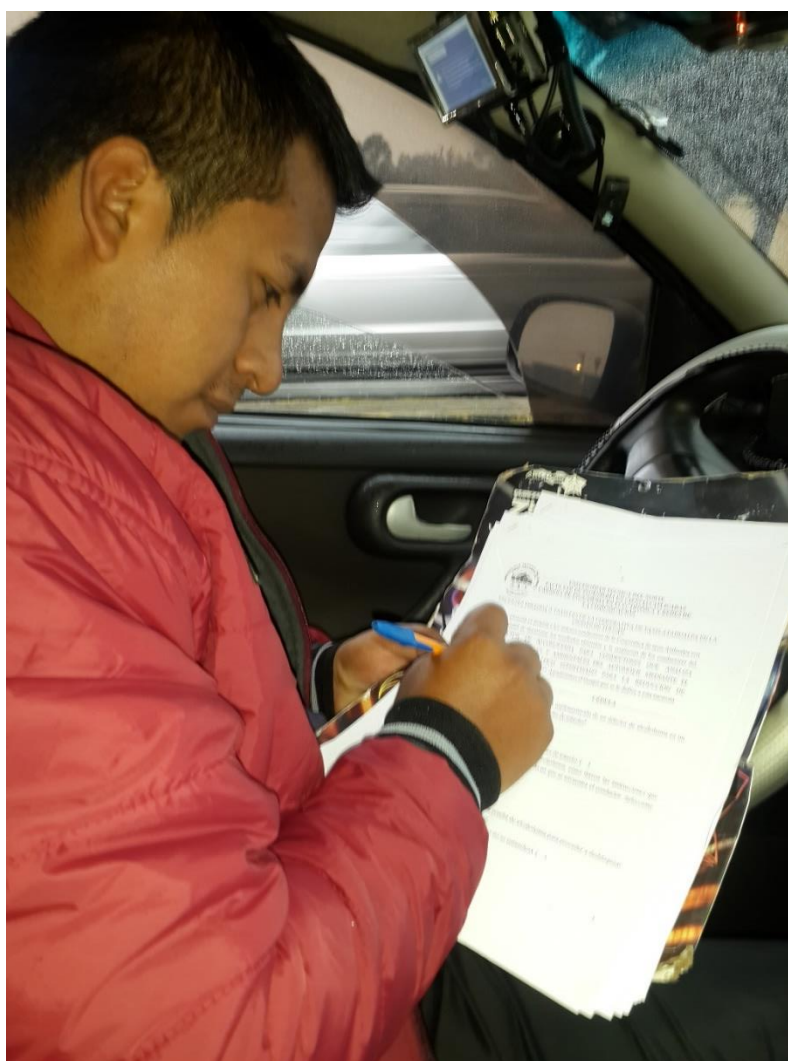


Figura 110. Conductor del grupo C de la Cooperativa de Taxis Atahualpa de la ciudad de Tulcán respondiendo la encuesta.

Fuente: Autoría

4.1.1.2. *Análisis y conclusión de la encuesta.*

Una vez aplicada la encuesta a los 58 señores conductores del grupo C de la Cooperativa de taxis Atahualpa de la ciudad de Tulcán; se obtuvo resultados que fundamentan la aceptabilidad de este proyecto, a continuación, se indican las conclusiones derivadas de la encuesta. Los resultados de la encuesta se pueden apreciar en el Anexo 27.

El resultado en la encuesta para la pregunta uno se puede ver en Figura 111, donde el 83% de los encuestados consideran que la implementación de un detector de alcoholemia en un vehículo ayuda a disminuir los accidentes de tránsito entre un 50 y 80 %, esto indica que es factible la implementación del proyecto en un vehículo, el cual permita disminuir los accidentes de tránsito.

El resultado de la encuesta para la pregunta dos se puede ver en Figura 112, donde el 76% de los usuarios encuestados considera que las instrucciones que proporciona el sistema para verificar el estado en que se encuentra el conductor son muy buenas y el 24% restante buenas, obteniendo como conclusión que los usuarios captaron las instrucciones del proyecto de manera clara y no es necesario realizar modificaciones en las mismas.

Continuando con el resultado para la pregunta tres se puede observar en la Figura 113, en la cual el 84% de la muestra afirma que no tuvieron dificultades al realizar la prueba de alcoholemia y el 16% restante considera que la realización de la prueba fue de mediana dificultad, por lo que se concluye que los señores conductores pueden finalizar la prueba de alcoholemia sin dificultades.

Los resultados de la pregunta 4 se pueden apreciar en la Figura 114, en la cual se tiene un total de 9 opciones ya que la pregunta considera múltiples selecciones de las cuales el 28% indica que solo se debería medir el nivel de alcohol en el aliento, seguido de un 21% que establecen que se debería tomar en cuenta todas las variables, por tal razón se determina que el uso los parámetros que se analizan en el proyecto son los correctos, sin embargo debido a que la mayoría considera que solo se debería medir el nivel de alcohol en el aliento se procede a realizar una corrección del proyecto para que solo en el primer análisis del estado del conductor se analicen todas las variables, en el caso de que el conductor sea la misma persona y necesite volver a encender el vehículo solo se solicite la prueba del aliento con la finalidad de disminuir el tiempo que toma realizar la prueba.

El resultado de la encuesta para la pregunta cinco se puede estimar en la Figura 115, donde el 86% de los señores encuestados afirmaron que la ubicación del detector de alcoholemia no obstaculiza la visión, de manera que se concluye que no es necesario realizar ningún cambio en la ubicación del proyecto en el vehículo.

Los resultados de los conductores encuestado en la pregunta seis se pueden ver en la Figura 116, en la cual el 59% de la muestra afirma que de los mensajes de alerta que emite el del detector de alcoholemia los dos mensajes son necesarios, de manera que se puede concluir que para emitir alertas o dar instrucciones al conductor de cómo realizar la prueba los dos mensajes ayudan a que sea captadas las órdenes.

Los resultados de la pregunta siete de la encuesta se pueden observar en la Figura 117 en donde el 95% de la muestra está de acuerdo con bloquear o desbloquear el encendido del vehículo según el estado en que se encuentre el conductor, en la Figura 118 se puede apreciar

las razones del porque los usuarios están de acuerdo, donde el 31% considera que se debe bloquear el encendido para prevenir o evitar accidentes de tránsito. Con estos resultados se concluye que el proyecto puede ser rentable ya que ningún usuario no se opuso a la medida de prevención que se asignó en el proyecto para evitar accidentes de tránsito.

Los resultados de la pregunta ocho de la encuesta se pueden observar en la Figura 119 en donde el conductor del vehículo proceda a ingerir bebidas alcohólicas después de haber desbloqueado el encendido del automotor, la mayoría de la muestra están totalmente de acuerdo y la razón principal por la que están de acuerdo es que es necesario informar a las autoridades competentes para evitar accidentes de tránsito. Con estos resultados se concluye que el proyecto permite que la sociedad tenga seguridad en las vías al comunicar del evento a las autoridades competentes.

Los resultados de la última pregunta se pueden visualizar en la Figura 120 en la cual se pone en discusión el estar de acuerdo o desacuerdo con la implementación del detector de alcoholemia en un taxi, considerando los riesgos que conllevan manejar en estado de ebriedad, como son las pérdidas materiales, personas heridas, la muerte del conductor o los tripulantes, además de arriesgar la vida de terceras personas, en este caso el 98% de la muestra manifiesta estar de acuerdo, por lo que en la Figura 121 se detallan las razón del porque están de acuerdo y la principal razón es que la implementación de un detector de alcoholemia en vehículos permite evitar accidentes de tránsito y salvar vidas. Con estos resultados se concluye que la realización del proyecto permite cumplir con la finalidad del sistema que es reducir los accidentes de tránsito.

Al finalizar las pruebas de operación y funcionamiento del sistema, en el Anexo 28 se adjunta el certificado emitido por el Sr. Álvaro Cano Gerente de la Cooperativa de Taxis “Atahualpa”, indicando que los resultados son los esperados del trabajo de Titulación.

Capítulo V. Análisis Económico

En este capítulo se realiza el análisis económico del Detector de Alcoholemia que analiza variables faciales y ambientales del automóvil mediante el aprendizaje automático supervisado para la reducción de accidentes de tránsito, en el cual se considera el monto referencial del presupuesto que se invirtió en el diseño y la implementación del proyecto, y las etapas que se realizan para evaluar el Costo-Beneficio de este.

5.1. Presupuesto del Hardware

En este punto se consideran los elementos electrónicos empleados durante la implementación del proyecto, la Tabla 34 muestra los costos de los elementos electrónicos que se emplearon para el desarrollo del proyecto.

Tabla 34. Presupuesto de componentes electrónicos.

Cantidad	Material	Costo Unitario	Costo Total
1	Arduino Uno R3	\$ 15,00	\$ 15,00
1	Raspberry Pi 3	\$ 65,00	\$ 65,00
1	Micro SD 32Gb	\$ 18,00	\$ 18,00
3	Sensor MQ3	\$ 5,00	\$ 15,00
1	Sensor MLX90621	\$ 129,66	\$ 129,66
1	Sensor DHT11	\$ 4,50	\$ 4,50
1	Raspberry Pi camera module v2 (Pi Noir)	\$ 35,69	\$ 35,69
1	LED de Infrarrojos para Pi Noir	\$ 10,06	\$ 10,06
1	Cinta FFC para Pi Noir	\$ 1,70	\$ 1,70

1	Modulo LCD TFT 3,5 pulgada + case	\$ 22,99	\$ 22,99
1	Modulo Relé 2 canales	\$ 13,70	\$ 13,70
1	Pulsador	\$ 0,30	\$ 0,30
1	Resistencia 10k Ω	\$ 0,10	\$ 0,10
1	Parlante	14,00	14,00
TOTAL			\$ 345,70

Fuente: Autoría.

En la Tabla 35 se indica el presupuesto necesario para la implementación del Detector de Alcholemla en el vehículo.

Tabla 35. Presupuesto de materiales para la implementación del Detector de Alcholemla en el vehículo.

Cantidad	Material	Costo Unitario	Costo Total
1	Soportes para móvil	8,00	8,00
1	Adaptador de Corriente para auto 2A.	16,00	16,00
1	Cable Micro USB (3m)	5,00	5,00
TOTAL			\$ 29,00

Fuente: Autoría.

5.2. Presupuesto del Software

Para el desarrollo del Detector de Alcholemla se utilizó software bajo la arquitectura Open Source y de descarga libre, en la Tabla 36 se detalla el software empleado.

Tabla 36. Presupuesto de Software.

Cantidad	Material	Costo Unitario	Costo Total
1	IDE de Arduino	\$ 0,00	\$ 0,00
1	Python	\$ 0,00	\$ 0,00
1	OpenCV	\$ 0,00	\$ 0,00
	TOTAL		\$ 0,00

Fuente: Autoría.

El diseño y la implementación del Detector de Alcoholemia en un vehículo tiene un costo total de \$ 376,70.

5.3. Análisis Costo Beneficio

La técnica de Análisis de Costo/Beneficio, tiene como finalidad el proporcionar una medida de la rentabilidad de un proyecto, mediante la comparación de los costos previstos con los beneficios esperados en la realización del proyecto y de esta manera tomar la mejor decisión ante la posible evaluación (Arturo K, 2012).

Para la evaluación del Costo Beneficio que tiene la implementación del Detector de Alcoholemia comprende el análisis de los siguientes puntos: Análisis de Costos, Análisis de beneficios y Análisis de la relación Costo Beneficio.

5.3.1. Análisis de costos.

En el análisis de costos se considera el método de costeo TCO (Total Costo Ownership) en el cual se discriminan los costos directos e indirectos para el desarrollo del

proyecto, en este caso serían los materiales empleados en el diseño y la implementación del Detector de Alcoholemia, tomando en cuenta como unidad de medida económica al dólar americano.

TCO es una práctica internacional utilizada para el análisis y mejora de los procesos de adquisición de tecnología y costos asociados a estas. Esta práctica es incorporada para el análisis de beneficios económicos.

Debido a que se está realizando el análisis de costos de un proyecto se tienen solo costos directos en los cuales se incluyen: Costo de Hardware y comunicaciones, costo de software y los costos de la implementación en el automóvil. En la Tabla 37 se pueden apreciar los Costos del proyecto.

Se debe considerar al ingresar la información la moneda, es decir se deberá ingresar toda la información monetaria para este caso de estudio se empleará el dólar americano de los Estados Unidos.

Tabla 37. Costos del proyecto

Descripción	Valor
Costos de los elementos electrónicos	\$ 345,70
Costo de los materiales para la implementación en el vehículo	\$ 29,00
Costo de Software	\$ 0,00
COSTO TOTAL	376,70

Fuente: Autoría.

5.3.2. Análisis de los Beneficios.

Para realizar el cálculo de los beneficios del proyecto se consideran los Stakeholder o involucrados en el proyecto que se pueden apreciar en Tabla 17 que especifica los Requerimientos de Stakeholders; considerando los stakeholders del proyecto se tienen los siguientes beneficios: los percibidos por las personas y los beneficios monetarios.

5.3.2.1. Beneficios Sociales.

Las personas involucradas para determinar estos beneficios son: el conductor del vehículo, los tripulantes, el director, la codirectora y la desarrolladora del presente trabajo de titulación. En este punto se analiza la calidad y la eficiencia del proyecto.

Este proyecto tiene como finalidad principal reducir Accidentes de tránsito, por lo que es considerado como un proyecto social, en el cual existe una inversión económica pero el beneficio que se obtiene no es monetario. A continuación, se mencionan los beneficios enfocados los stakeholders después de la implementación en el vehículo.

El principal beneficio que el Detector de Alcoholemia es prevenir accidentes de tránsito, especialmente causados por personas irresponsables que no miden el riesgo de conducir en estado de ebriedad, este beneficio no solo favorece a los involucrados en el proyecto, sino que permite que los peatones puedan caminar sin riesgos de ser atropellados en las vías.

La implementación de este proyecto es una gran opción para que los dueños de los vehículos tengan a la mano una nueva herramienta para combatir las muertes causadas por conductores ebrios.

Este proyecto también contribuye a la seguridad vial, esto se logra gracias al bloqueo del encendido del vehículo cuando una persona se encuentra en estado etílico, lo que garantiza que los tripulantes y el conductor lleguen a salvo a su destino.

El desarrollo de este proyecto permite a los involucrados hacer uso de las herramientas de visión artificial y aprendizaje automático para determinar el estado en el que se encuentra el conductor antes de salir de viaje.

La implementación de mensajes visuales y auditivos permiten informar al señor conductor el estado en el que se encuentra, mientras que mediante la activación del botón de pánico del ECU911 garantiza que conductores necios insistan en movilizarse en el automotor.

El reconocimiento facial garantiza que no sea se cambie de conductor sin realizar la prueba respectiva para verificar el estado en el que se encuentra antes de encender el vehículo. Además de incrementar la confiabilidad de los pasajeros en los transportes de servicio público.

5.3.2.2. Beneficios monetarios.

El único stakeholder que hace uso de este beneficio es el conductor del vehículo, ya que gracias a la implementación del Detector de alcoholemia se elimina el costo de infracciones de tránsito por conducir en estado de ebriedad. En la Tabla 38 se detallan los montos, con relación

al “Acuerdo Ministerial MDT-2017-0195, el cual establece: “Fijar a partir del 1 de enero del 2018 el salario básico unificado del trabajador en general, incluidos los trabajadores de la pequeña industria, trabajadores agrícolas y trabajadores de maquila; trabajadora o trabajador remunerado del hogar, operarios de artesanía y colaboradores de la microempresa en \$ 386,00 dólares de los Estados Unidos de América, mensuales” (Ministerio del Trabajo, 2017, p.3).

Tabla 38. Beneficios económicos

NIVEL	Nivel de alcohol por litro de sangre (gramos)	Multa en salario básico unificado del trabajador en general	Valor
1	0,3 a 0,8	1	\$ 386,00
2	0,8 a 1,2	2	\$ 772,00
3	Mayor 1,2	3	\$ 1158,00

Fuente: (Zuñiga, et al,2014)

Para determinar el beneficio monetario se considera que un conductor vaya a tener por lo menos una infracción de los tres niveles de concentración de alcohol. En la Tabla 39 se muestra el beneficio monetario en la cual el monto que debería cancelar el conductor en estado de ebriedad se lo considerará como ingresos debido a que con la implementación de este proyecto no tendrá que cancelar este valor.

Tabla 39. Beneficio monetario

Concentración de Alcohol	Valor
0,3 a 0,8	\$ 386,00
0,8 a 1,2	\$ 772,00
Mayor 1,2	\$ 1158,00
Total	\$ 2316,00

Fuente: Autoría.

5.3.3. Análisis de la relación Beneficio-Costo.

Debido a que para la implementación de Detector de Alcoholemia no se espera obtener una tasa por la inversión hecha es decir no se tiene establecida una Tasa Mínima Aceptable de Rendimiento (TMAR), se procede a calcular la relación beneficio costo (B/C), haciendo uso de la suma de los beneficios y se divide para la suma de los costos totales.

Para el primer caso de estudio se considera que al conductor del vehículo fue sancionado con una concentración de alcohol entre 0,3 y 0,8, por la suma de los beneficios monetarios es de \$ 386,00 dólares.

$$B/C = \frac{\sum \text{Beneficios monetarios}}{\sum \text{Costos directos}}$$

$$B/C = \frac{386,00}{376,70}$$

$$B/C = 1,025$$

Para el segundo caso de estudio se considera que al conductor del vehículo fue sancionado con una concentración de alcohol entre 0,8 y 1,2, por la suma de los beneficios monetarios es de \$ 772,00 dólares.

$$B/C = \frac{\sum \text{Beneficios monetarios}}{\sum \text{Costos directos}}$$

$$B/C = \frac{772,00}{376,70}$$

$$B/C = 2,049$$

Para el tercer caso de estudio se considera que al conductor del vehículo fue sancionado con una concentración de alcohol es mayor a 1.2, por la suma de los beneficios monetarios es de \$ 1158,00 dólares.

$$B/C = \frac{\sum \text{Beneficios monetarios}}{\sum \text{Costos directos}}$$

$$B/C = \frac{1158,00}{376,70}$$

$$B/C = 3,074$$

En el último caso que se considera es que el conductor del vehículo fue sancionado tres ocasiones con las distintas concentraciones de alcohol, por lo que la suma de los benéficos es de \$ 2316,00 dólares.

$$B/C = \frac{\sum \text{Beneficios monetarios}}{\sum \text{Costos directos}}$$

$$B/C = \frac{2316,00}{376,70}$$

$$B/C = 6,148$$

La relación beneficio costo determina la rentabilidad del proyecto en términos generales y el resultado expresa el dinero ganado en cada dólar que se ha invertido en la implementación del Detector de Alcholemia.

5.3.4. Conclusión del análisis Costo - Beneficio.

De acuerdo con la relación beneficio costo que se calculó en el literal anterior se demuestra que el proyecto es monetariamente viable, ya que para todos los casos que se consideró que podrían aplicarle un sanción al conductor esta relación es mayor a 1

incrementando su valor según la concentración de alcohol en la que se encuentre; con cada dólar invertido en la implementación se puede ganar, sin embargo, este valor puede variar dependiendo del número de infracciones de tránsito que cometa el conductor y la concentración de alcohol.

Capítulo VI. Conclusiones y Recomendaciones

Con la finalización del presente proyecto de titulación denominado Detector de alcoholemia que analiza variables faciales y ambientales del automóvil mediante el aprendizaje automático supervisado; se presentan las siguientes conclusiones y recomendaciones.

6.1. Conclusiones

Con la implementación del Detector de Alcoholemia se demostró que se puede determinar el estado del conductor haciendo uso del aprendizaje automático supervisado basado en el árbol de decisión; los resultados y la aceptación del proyecto por parte de los señores conductores contribuyó a que el desarrollo de este proyecto tenga implicaciones significativas al momento de disminuir los accidentes de tránsito a causa de conductores en estado de ebriedad con el bloqueo del encendido del automóvil y la activación del botón de pánico implementado por el Servicio Integrado de Seguridad ECU 911.

La recopilación de los diferentes métodos para determinar si una persona ha consumido alcohol garantizó la selección adecuada de las variables que se deben tomar en cuenta al momento de entrenar al algoritmo de aprendizaje automático supervisado basado en el árbol de decisión.

La aplicación de la metodología de desarrollo basada en el modelo en V permitió que se realice el diseño adecuado de un prototipo de detección de alcoholemia para la recolección de las variables que el sistema analiza para determinar el estado del conductor.

La selección de la temperatura facial máxima, el diámetro de la pupila y el nivel de alcohol en el aliento como atributos del conjunto de datos facilitó el desarrollo del algoritmo de aprendizaje automático supervisado basado en arboles de decisión, debido a que ayudó a que la agrupación de la clase 1 que corresponde a los conductores ebrios o la clase 0 que identifica a los conductores sobrios con los atributos que contienen características similares.

La fase de la prueba unitaria en donde se verificó de manera independiente el funcionamiento de cada parte del proyecto y la prueba de integración que avaló que el sistema se encuentre unificado, garantizaron que en la prueba de operación y validación no se presenten inconvenientes por lo que al finalizar la metodología de desarrollo basada en el modelo en V se concluyó que en el Detector de Alcoholemia no se debe realizar ninguna corrección.

6.2. Recomendaciones

Para la implementación del detector de alcoholemia en el vehículo se debe realizar un adecuado montaje de cada uno de los elementos que se emplean en el proyecto con la finalidad de evitar contratiempos al realizar las pruebas de funcionamiento.

Para evitar tener inconvenientes en la prueba unitaria es fundamental que se consideren todos los posibles fallos que se podrían presentar en el desarrollo del detector de alcoholemia en la fase de análisis.

En la fase de requerimientos del sistema solo se deben considerar los que tengan prioridad media y alta, siendo los datos más relevantes los que presentan prioridad alta en la fase de diseño.

Es necesario que el conjunto de datos que se emplea para el algoritmo de aprendizaje automático contenga un número de datos significativo y que hayan sido recolectados en diferentes escenarios.

Antes de realizar las pruebas de operación y funcionamiento del sistema, las pruebas unitarias y de integración deben ser exitosas para evitar realizar modificaciones al final del proyecto.

Para realizar estudios similares de sistemas de detección de alcoholemia se recomienda analizar otros métodos para la detección de alcohol con la finalidad de cambiar los sensores y disminuir el tiempo en la fase de adquisición de datos para mejorar el rendimiento del sistema.

Bibliografía

Bronte Palacios, S. (2008). *Sistema de Detección y Reconocimiento Facial de Conductores Mediante Sistemas de Visión Computacional*. Madrid.

MELEXIS [Melexis Microelectronic Systems]. (s.f.). Obtenido de MLX90614 Datasheet:
<http://pdf1.alldatasheet.es/datasheet-pdf/view/224153/MELEXIS/MLX90614.html>

Rodríguez Rodríguez, J. E., Rojas Blanco, E. A., & Franco Camacho, R. O. (2007). Clasificación de datos usando el método k-nn. *VINCULOS*, 10-14. Obtenido de <http://revistavinculos.udistrital.edu.co/files/2012/12/Clasificaci%C3%B3n-de-datos-ED7.pdf>

Sánchez García de Blas, R. J. (Junio de 2015). *Reconocimiento de caracteres mediante imágenes en contadores de gas en entornos reales*. Madrid,. Obtenido de Reconocimiento de caracteres mediante imágenes en contadores de gas en entornos reales: <http://eprints.ucm.es/32992/1/Memoria.pdf>

Agencia Nacional de Tránsito. (Febrero de 2017). *Estadísticas de transporte terrestre y seguridad vial*. Obtenido de Estadísticas sobre Accidentes de Tránsito: <http://www.ant.gob.ec/index.php/noticias/estadisticas#.WOUN22jhC03>

Akers Biosciences. (23 de Junio de 2003). Obtenido de FORMAS DE DETECCION DE ALCOHOL:
<http://www.elad.com.co/descargas/alcohol/Formas%20de%20Detecci%C2%A2n%20de%20Alcohol.pdf>

Akers Biosciences, Inc. (Agosto de 2008). Obtenido de BreathScan® Detector de Alcohol:
http://www.diinsel.com/antialcohol_breathscan.html

- Álvarez Durán, M. (2014). Obtenido de Análisis, diseño e implementación de un sistema de control de ingreso de vehículos basado en visión artificial:
<http://dspace.ups.edu.ec/bitstream/123456789/7060/1/UPS-CT003790.pdf>
- Andrade Spatz, G. L. (2009). *REGULACIÓN PARA LOS PILOTOS DE VEHÍCULOS AUTOMOTORES, DE LA PROHIBICIÓN DE CONDUCIR BAJO INFLUENCIA DE BEBIDAS ALCOHÓLICAS Y CONSECUENCIAS JURÍDICAS Y SOCIALES DE LAS PERSONAS QUE CONDUZCAN EN ESTADO DE EBRIEDAD*. GUATEMALA.
- Aranda, A. (17 de Septiembre de 2014). Obtenido de Etanol metanol.
- Arduino.cc. (s.f.). Obtenido de Lenguaje de programación de Arduino:
<http://playground.arduino.cc/ArduinoNotebookTraduccion/Structure>
- Arias Duque, R. (2005). Reacciones fisiológicas y neuroquímicas del alcoholismo. *DIVERSITAS*, 138-147.
- Barreto, F. (13 de Mayo de 2014). Obtenido de IEEE 29148:
<https://prezi.com/8mki1yjihkdp/ieee-29148/>
- Bausà Aragonés, J., García Gómez, C., & Zaragozaí Zaragoz, B. (Abril de 2003). Obtenido de Sensores de Temperatura: http://server-die.alc.upv.es/asignaturas/LSED/2003-04/0.Sens_Temp/ARCHIVOS/SensoresTemperatura.pdf
- Beccach Izquierdo, M. G., Cano Cabezas, M. d., Falconí Arias, E. F., González Camacho, Á. C., Guzmán Silva, G. d., Ibarra González, S., . . . Zúñiga Rocha, L. (2014). *Código Orgánico Integral Penal (COIP)*. Quito: Gráficas Ayerve C. A. Obtenido de Código Orgánico Integral Penal.
- Brydko, V. (2015). Diseño e implementación de múltiples interfaces para el control remoto de un vehículo y del sistema de visión embarcado. Valencia, España.
- Calderón Jácome, M. T., & Maldonado Andrade, D. (Octubre de 2014). Control por visión de un cuadricóptero utilizando ROS. Quito, Pichincha, Ecuador.

- Carhuaricra, J. (2015). Obtenido de INGENIERÍA DEL SOFTWARE:
https://www.academia.edu/7074168/INGENIER%C3%8DA_DEL_SOFTWARE
- Cedillo Ochoa, E. (Octubre de 2010). *ESTUDIO COMPARATIVO ENTRE MUESTRAS DE SANGRE Y ORINA PARA DETERMINACIÓN DE LA CONCENTRACIÓN DE ALCOHOL*. Obtenido de <http://repositoriodigital.ipn.mx/bitstream/123456789/9021/1/tesis%20ERIKA.pdf>
- Chevrolet. (2007). *Manual del propietario Corsa*. MAXIGRAF S.A. Obtenido de Manual del Propietario Corsa.
- Coursera Org. (2016). Obtenido de Aprendizaje Automático:
<https://www.coursera.org/learn/machine-learning/lecture/zcAuT/welcome-to-machine-learning>
- Delfino, C. (2012). *Luchemos por la Vida Asociación Civil*. Obtenido de Efectos del alcohol en la conducción: <http://www.luchemos.org.ar/revistas/articulos/rev31/pag02.pdf>
- Dominguez Reboiras, M. A. (2008). *Química la Ciencia Básica*. Madrid: Thomson Ediciones Paraninfo, S. A.
- Dräger. (2017). Obtenido de Draeger Interlock® XT: https://www.draeger.com/en-us_us/Ignition-Interlock/Products/Breath-Alcohol-and-Drug-Testing/Alcohol-Ignition-Interlock-Devices/Interlock-XT
- Element I4 Community. (22 de Febrero de 2016). Obtenido de Raspberry Pi 3 Model B Technical Specifications: <https://www.element14.com/community/docs/DOC-80899/1/raspberry-pi-3-model-b-technical-specifications>
- Espinosa Peralta, H. P. (Septiembre de 2016). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD Y ALERTA PARA VEHICULOS, BASADO EN RECONOCIMIENTO FACIAL Y LOCALIZACIÓN GPS, EN UNA RASPBERRY PI B PLUS. Quito, Pichincha.

- Espinoza Olgún, D. E., & Jorquera Guillen, P. I. (Junio de 2015). Obtenido de Reconocimiento Facial: http://opac.pucv.cl/pucv_txt/txt-1000/UCD1453_01.pdf
- Europe Electrocomponents. (s.f.). Obtenido de Pi NoIR: <http://docs-europe.electrocomponents.com/webdocs/127d/0900766b8127db33.pdf>
- FIGARO. (2017). Obtenido de TGS 2620 - for the detection of Solvent Vapors: <http://www.figarosensor.com/products/2620pdf.pdf>
- Futurlec. (s.f.). Obtenido de MR513 Hot-wire type gas sensor: <https://www.futurlec.com/Datasheet/Sensor/MR513.pdf>
- García Cambroner, C., & Gómez Moreno, I. (2006). *ALGORITMOS DE APRENDIZAJE: KNN & KMEANS*. Madrid: Universidad Carlos III de Madrid.
- García Chang, M. E. (2009). *Diseño e implementación de una herramienta de detección facial*. México.
- doi:[http://tesis.bnct.ipn.mx:8080/jspui/bitstream/123456789/5902/1/1366_Centro%20de%20Innovacin%20y%20Desarrollo%20Tecnolgico%20en%20C%3%B3mputo%20\(CIDETEC\)tesis_Febrero_2010_585879229.pdf](http://tesis.bnct.ipn.mx:8080/jspui/bitstream/123456789/5902/1/1366_Centro%20de%20Innovacin%20y%20Desarrollo%20Tecnolgico%20en%20C%3%B3mputo%20(CIDETEC)tesis_Febrero_2010_585879229.pdf)
- Gennaro, A. R. (2003). *Remington* (Vol. XX). Bueno Aires: Médica Panamericana.
- Gil, P., Torres, F., & Ortiz Zamora, F. G. (2004). *Detección de objetos por segmentación multinivel combinada de espacios de color*. Real.
- Guanche Garcell, H., Martínez Quesada, C., & Gutiérrez García, F. (2007). Efecto del alcohol en la capacidad de conducción de vehículos automotores. *Revista Cubana de Salud Pública*, 33(1), 0-0.
- Gutierrez, D. (Julio de 2011). Obtenido de Métodos de Desarrollo de Software: http://www.codecompiling.net/files/slides/IS_clase_13_metodos_y_procesos.pdf
- HANWEI ELETRONICS CO.,LTD . (s.f.). Obtenido de TECHNICAL DATA MQ-3 GAS SENSOR: <https://www.sparkfun.com/datasheets/Sensors/MQ-3.pdf>

- HANWEI ELETRONICS CO.,LTD. (8 de Febrero de 2010). Obtenido de TECHNICAL DATA MQ-3 GAS SENSOR: <https://www.jameco.com/jameco/products/prodds/2124509.pdf>
- HIFI, Y. (2016). Obtenido de 1pcs Top quality new MLX90621 MLX90621ESF BAB infrared (IR) sensor array free shipping: https://es.aliexpress.com/store/product/1pcs-Top-quality-new-MLX90621-MLX90621ESF-BAB-infrared-IR-sensor-array-free-shipping/939205_32646269531.html?spm=2114.04010208.3.9.ZPWV9S&ws_ab_test=searchweb0_0,searchweb201602_2_10152_10065_10151_10068_436_10136_1
- ISO/IEC/IEEE. (1 de Diciembre de 2011). Obtenido de 29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering: <https://standards.ieee.org/findstds/standard/29148-2011.html>
- Kaehler, A., & Bradski, G. (2016). *Learning OpenCV3*. Estados Unidos: O'Really Media, Inc.
- Koukiou, G., & Anastassopoulos, V. (5-6 de Septiembre de 2013). Eye temperature distribution in drunk persons using thermal imagery. *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the.*
- Koukiou, G., Panagopoulos, G., & Anastassopoulos, V. (5-7 de Julio de 2009). Drunk person identification using thermal infrared images. *Digital Signal Processing, 2009 16th International Conference on.*
- Laganière, R. (2016). *OpenCV 3 Computer Vision Application Programming Cookbook*. Pack Publishing.
- Lobato Ricon, L. L., Cabanillas Campos, M. C., Navarro Valls, J. J., Bonnin Arias, C., Chamoro, E., & Sanchez Ramo Roda, C. (2013). Utilidad de la pupilometría dinámica en el control de alcoholemia de los conductores. *Adicciones, 25, 9.*

Loor López, G. E. (2012). *EL ALCOHOL Y SUS EFECTOS NOCIVOS*. Herramientas de Colaboración Digital.

Melexix INSPIRED ENGINEERING. (15 de Septiembre de 2016). Obtenido de MLX90621

16X4

IR

array:

https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwic8deZIPLTAhXFJCYKHZ0_BJ8QFgg4MAA&url=https%3A%2F%2Fwww.melexis.com%2F-%2Fmedia%2Ffiles%2Fdocuments%2Fdatasheets%2Fmlx90621-datasheet-melexis.pdf&usg=AFQjCNHhYacr1pQnOmwRt6is3mv

Méndez, G. (2009). Obtenido de Proceso Software y Ciclo de Vida:

<https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/02-ProcesoCicloDeVida.pdf>

Ministerio del Trabajo. (Diciembre de 2017). Obtenido de Acuerdo Ministerial No MDT-2017-

0195):

[http://www.trabajo.gob.ec/wp-](http://www.trabajo.gob.ec/wp-content/uploads/downloads/2018/01/ACUERDO-MINISTERIAL-MDT-2017-0195-SALARIO-BASICO-UNIFICADO-386.pdf)

[content/uploads/downloads/2018/01/ACUERDO-MINISTERIAL-MDT-2017-0195-SALARIO-BASICO-UNIFICADO-386.pdf](http://www.trabajo.gob.ec/wp-content/uploads/downloads/2018/01/ACUERDO-MINISTERIAL-MDT-2017-0195-SALARIO-BASICO-UNIFICADO-386.pdf)

Ortiz Arciniega, J. L. (2017). *BENCHMARK DE SELECCIÓN DE SENSORES PARA UNA*

WSN DE RECOLECCION DE DATOS PARA UN SISTEMA DE ALERTA TEMPRANA

DE INCENDIOS FORESTALES. Ibarra. Obtenido de

<http://repositorio.utn.edu.ec/bitstream/123456789/6527/1/04%20RED%20151%20TRABAJO%20DE%20GRADO.pdf>

Pereira Moreira, R. (2015). Advances in road safety. Avances en Seguridad Vial. *Congreso*

Internacional de Seguridad Vial, 50.

Pérez García, A. (2013). Obtenido de GAMIFICATION: ENCUESTAS DE SATISFACCIÓN

DE LOS SERVICIOS DE ALIMENTACIÓN JAVERIANOS:

<https://repository.javeriana.edu.co/handle/10554/12643>

- Perez, A., Berreteaga, O., Ruiz de Olano, A., Urkidi, A., & Perez, J. (2006). Obtenido de UNA METODOLOGÍA PARA EL DESARROLLO DE HARDWARE Y SOFTWARE EMBEBIDOS EN SISTEMAS CRÍTICOS DE SEGURIDAD: [http://www.iiisci.org/journal/CV\\$/risici/pdfs/C863GM.pdf](http://www.iiisci.org/journal/CV$/risici/pdfs/C863GM.pdf)
- Platero, C. (27 de Enero de 2014). Obtenido de Robótica y Visión Artificial: http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.pdf
- Prada Pérez de Azpeitia, F. I., & Martínez Pons, J. A. (Marzo de 2003). *ALCOHOL Y ALCOHOLÍMETROS*. Obtenido de Historia, fundamentos científicos y aplicación didáctica: <file:///D:/avances/Dialnet-AlcoholYAlcoholímetros-637814.pdf>
- Pruebas de Detección, S. A. de C. V. & AdlTesr diagnósticos, S. A. de C. V. (2016). Obtenido de ALCOLÍMETROS (DETECCIÓN DE CONSUMO DE ALCOHOL): <http://www.adi-test.com/alcoholímetros-mexico-toxicologia.php>
- Raspberry Pi Foundation. (s.f.). Obtenido de RASPBERRY PI 3 MODEL B: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Revista Dräger. (Noviembre de 2010). Obtenido de Bloqueo del arranque para quien abuse del alcohol: https://www.draeger.com/Corporate/Content/revistadraeger_2_enfoque.pdf
- Rodríguez Bazaga, A. (18 de Agosto de 2015). Obtenido de OpenCV: Librería de Visión por Computador: <https://osl.ull.es/software-libre/opencv-libreria-vision-computador/>
- Ruiz, Á., Macías, F., Gómez Restrepo, C., Rondón, M., & Lozano, J. M. (Febrero de 2010). Niveles de alcohol en sangre y riesgo de accidentalidad vial: revision sistematica de la literatura. *Revista Colombiana de Psiquiatría*, 249.
- Ruiz, Á., Macías, F., Restrepo, C., Rondón, M., & Lozan, J. (2010). *Niveles de alcohol en sangre y riesgo de accidentalidad vial: revisión sistemática de la literatura*. Obtenido de <http://www.scielo.org.co/pdf/rcp/v39s1/v39s1a17.pdf>

Scikit-learn Organization. (17 de Noviembre de 2017). *scikit-learn*. Obtenido de Decision Trees: <http://scikit-learn.org/stable/modules/tree.html>

Seed Studio. (s.f.). Obtenido de MQ303A Alcohol Sensor: <http://www.seeedstudio.com/document/pdf/MQ303A.pdf>

SENA Servicio Nacional de Aprendizaje. (2013). Obtenido de Procesos para Software de Calidad: https://sena.blackboard.com/bbcswebdav/pid-8208177-dt-content-rid-1319144_1/courses/21710086_2_VIRTUAL/externalFiles_20150302081115/Material_U1.pdf

Sobrado Malpartida , E., & Tafur Sotelo, J. (2003). Obtenido de SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO Y MANIPULACIÓN DE OBJETOS UTILIZANDO UN BRAZO ROBOT: <http://www.iiis.org/cds2008/cd2008csc/cisci2008/paperspdf/c336gi.pdf>

SOUNROM Electronics/Techonologies. (s.f.). Obtenido de Alcohol Sensor Module - MQ3: <http://www.sunrom.com/p/alcohol-sensor-module-mq3>

Sucar, E., & Gómez, G. (2011). *Visión Computacional*. Puebla.

Tawil, Y. (01 de Julio de 2016). Obtenido de Understanding Arduino UNO Hardware Design: <https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/>

Tayal, Y., Kumar Pandey, P., & Singh, D. B. (2013). Obtenido de Face Recognition using Eigenface: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.300.8197&rep=rep1&type=pdf>

TRADESEGUR. (s.f.). Obtenido de FST-2 Control de Alcoholemia Portátil: <http://www.tradesegur.com/pdf/WP/fst.pdf>

- Villegas Quezada, C. (2005). *Reconocimiento de rostros utilizando análisis de componentes principales: Limitaciones del algoritmo*. México. Obtenido de <http://www.bib.uia.mx/tesis/pdf/014620/014620.pdf>
- Viola, P., & Jones, M. (2001). *Rapid Object Detection using a Boosted Cascade of Simple Features*. ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION.
- Viorica Andreea, P. (2016). *Aplicación para Detección y Reconocimiento Facial en Interiores*. Sevilla.
- Waszkiewicz, N., Zalewska, U., Szulc, U., U. K., B, K., B, Z.-S., . . . K, Z. (2011). Obtenido de The influence of alcohol on the oral cavity, salivary glands and saliva: <http://europepmc.org/abstract/med/21542250>
- Wu, Y.-c., Xia, Y.-q., & Xie, P. (19-20 de Diciembre de 2009). The Design of an Automotive Anti-Drunk Driving System to Guarantee the Uniqueness of Driver. *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*.
- Xiaoguang, L. (2005). Obtenido de Image Analysis for Face Recognition: http://face-rec.org/interesting-papers/General/ImAna4FacRcg_lu.pdf
- Yang, M.-H., Kriegman, D. J., & Ahuja, N. (2002). *Detecting Faces in Images: A Survey*. Washington, DC: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Zhao, W., Chellappa, R., Phillips, J., & Rosenfeld, A. (2003). Face recognition: A literature survey. New York: ACM Computing Surveys (CSUR).

Glosario

Siniestros de tránsito: es un suceso que ocurre con la responsabilidad de alguna parte, es decir, se produce por culpa, dolo, impericia, distracción o alguna causa imputable a personas que altera la marcha normal o prevista del desplazamiento en las vialidades.

Etanol: también conocido como alcohol etílico es una sustancia psicoactiva y es el principal tipo de alcohol presente en las bebidas alcohólicas, el cual en condiciones normales de presión y temperatura se muestra como líquido incoloro e inflamable con un punto de ebullición de 78,4 °C.

BAC: Concentración de alcohol en la sangre es el medio para establecer la concentración de alcohol en el nivel sanguíneo de una persona en un momento determinado.

CAO: Concentración de alcohol en orina permite detectar el consumo de alcohol y permiten estimar frecuentemente su concentración en sangre.

CAA: Concentración de alcohol en el aliento permite detectar el consumo de alcohol en el aire espirado.

OpenCv: Es una librería de código abierto para el desarrollo de funciones de programación dirigidas principalmente a visión por computador y con un fuerte enfoque en aplicaciones que necesitan ser interpretadas en tiempo real.

Licencia BSD: Es una licencia de software libre permisiva.

Eigenfaces: es un algoritmo empleado para el reconocimiento facial que emplea un conjunto de vectores propios cuando se usan en el problema de visión artificial del reconocimiento de rostros humanos, en donde los componentes principales o características comunes de una imagen son: ojos, labio, nariz y distancias entre estos componentes.

Fisherfaces: es un algoritmo que se encarga del reconocimiento de caras, teniendo en cuenta como se refleja la luz y las expresiones faciales, emplea el Análisis discriminante para encontrar las características faciales y discriminar entre las personas.

Local binary pattern: es un algoritmo empleado para el reconocimiento facial que utiliza un operador de textura que etiqueta los píxeles de una imagen por thresholding o umbral, método de segmentación de imagen que a partir de una en escala de grises se crea una imagen binaria, cada vecino del píxel y el resultado de la operación se considera como un número binario.

Machine Learning: Aprendizaje automático es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.

Modelo en V: Metodología de desarrollo que define un procedimiento uniforme para el desarrollo de sistemas embebidos incluye el proceso de Verificación (V) en las fases de análisis y diseño; además de incorporar retroalimentación entre las etapas. Este modelo es útil para proyectos, que necesitan alta confiabilidad.

SySR: Requerimientos iniciales del sistema se definen los límites funcionales del sistema en términos de comportamiento y propiedades del proyecto.

SRSH: Requerimiento de arquitectura se definen los requerimientos de hardware, software y el sistema eléctrico.

Stakeholder: Es todo grupo o individuo que tiene un interés directo en el resultado obtenido por el desarrollo del proyecto.

StSR: Requerimientos de stakeholders identificar los requisitos de los interesados para el sistema.

Sistema Embebido: Es un sistema de computación diseñado para realizar una o algunas funciones dedicadas frecuentemente en un sistema de computación en tiempo real.

Raspberry Pi 3: es una placa computadora (SBC) de bajo coste, considerada como una maravilla en miniatura, que guarda en su interior un importante poder de cómputo en un tamaño muy reducido.

Arduino UNO R3: es una placa electrónica basada en el microcontrolador ATmega328.

Pi Noir: Raspberry Pi modulo V2 es una cámara infrarroja de alta definición para Raspberry Pi.

Sensor MQ-3: Sensor de alcohol diseñado para determinar la presencia de etanol en el aire emitido.

Sensor MLX90621: Sensor térmico de 16x4 pixeles que ofrece un campo de visión (FOV) significativamente ampliado a 30 ° x120 ° y una velocidad de medición 4 veces más rápida o, equivalentemente, un nivel de ruido 2 veces más bajo.

Sensor DHT11: es un sensor de temperatura y humedad digital de bajo costo.

Módulo Relé: Es un dispositivo electromagnético que funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Python: Es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Scikit-learn: es una biblioteca de aprendizaje de automático y minería de datos libre para el lenguaje de programación Python.

RGB: es un modelo de color utilizado normalmente para presentar color en los sistemas de video, cámaras, y monitores de ordenadores. Representa todos los colores como combinaciones de rojo, verde y azul.

Haar-Like Features: clasificadores para emplear en la Detección de rostro.

HoughCircles: Función de la librería de visión por computador OpenCV para detectar círculos.

Atributos: también denominados características, factores, propiedades o campos son los que describen cada una de las instancias del conjunto de datos.

Clases: conocidas como instancias son cada uno de los datos de los que se disponen para hacer un análisis.

Conjunto de datos: Es la materia prima del sistema de predicción que se usa para entrenar al algoritmo que detecta los patrones. El conjunto de datos se compone de instancias o clases, y las instancias de atributos factores, características o propiedades.

Conjunto de Entrenamiento: Son parte del conjunto de datos que se utilizan con la finalidad de determinar los parámetros del clasificador.

Conjunto de Prueba: Son parte del conjunto de datos se utiliza para estimar el error de generalización, considerando que el valor del error debe ser lo más pequeño posible.

KNN: Vecinos más Próximos es un método retardado y supervisado (pues su fase de entrenamiento se hace en un tiempo diferente al de la fase de prueba) cuyo argumento principal es la distancia entre instancias.

Árbol de Decisión (Decision Tree): es un método de aprendizaje supervisado no paramétrico utilizado para la clasificación. El objetivo es crear un modelo que prediga el valor de una variable objetivo mediante el aprendizaje de reglas simples de decisión inferidas a partir de las características de los datos.

Anexo 1. Programación script IDE-Arduino para la lectura de los sensores y activación del módulo Relé

```
EnviarDatosRaspberry  DHT.cpp  DHT.h  MLX90621.cpp  wireMLX90621.h
1 //////////////////////////////////////////////////LIBRERIAS////////////////////////////////////
2 #include <SPI.h>          //Libreria para usar el bus SPI
3 #include "DHT.h"         //Libreria para el sensor DHT11
4 #include <Arduino.h>     //Libreria para usar las funciones de Arduino
5 #include <Wire.h>        //Libreria para la comunicación I2C
6 #include "wireMLX90621.h" //librerias para el sensor de Temperatura MELEXIS MLX90621 adaptadas
7                          //de (https://github.com/longjos) y https://github.com/robinvanemden/MLX90621\_Arduino\_Processing
8 #include <EEPROM.h>      //Libreria para el almacenamiento en la memoria Eeprom
9 #include <Servo.h>       // Incluimos la librería para poder controlar el servo
10
11 //////////////////////////////////////////////////DEFINIR VARIABLES////////////////////////////////////
12 #define DHTPIN 2         //Define el pin 2 para la lectura del modulo DHT11
13 #define DHTTYPE DHT11   //Define la salida del valor de temperatura y humedad del módulo DHT11
14 DHT dht(DHTPIN, DHTTYPE);
15 MLX90621 sensor;        // crea una instancia para la clase sensor de la libreria WireMLX90621
16 int bombaCombustible = 4; //Define el pin para la salida del canal uno del modulo Relé
17 int BotonPanico = 12;    //Define el pin para la salida del canal dos del modulo Relé
18 float dato;             //Variable para almacenar un dato
19 int tam;                //Variable para el tamaño de la cadena de la temperatura facial
20 int BOTON = 3;          //Define el pin de entrada para el pulsador que iniciará la prueba de Alcholeemia
21 volatile int estado_pulsador = LOW; //Variable para almacenar el estado del pulsador
22 int variableAlcoholAmbiente = 0;
```

```
23
24 //////////////////////////////////////////////////MÉTODO PARA VER EL ESTADO DEL PULSADOR////////////////////////////////////
25 void pulsador()
26 {
27     estado_pulsador = estado_pulsador + 1;
28     if(estado_pulsador == 1)
29     {
30         Serial.print(estado_pulsador);
31         delay (100);
32         Serial.println(' ');
33     }
34 }
35
36 //////////////////////////////////////////////////MÉTODO PARA LEER EL NIVEL DE ALCOHOL EN EL AIRE////////////////////////////////////
37 void alcoholAmbiente()
38 {
39     int adc_MQ = analogRead(A1); //Lemos la salida analógica del MQ
40     float voltaje = adc_MQ * (5.0 / 1023.0); //Convertimos la lectura en un valor de voltaje
41     float Rs=1000*((5-voltaje)/voltaje); //Calculamos Rs con un RL de 1k
42     double alcohol=0.5491*pow(Rs/5463, -0.682); // calculamos la concentración de alcohol.
43 // Serial.print(alcohol);
44     delay (10);
```

```
45 // Serial.println(' ');
46 if((alcohol >= 2.1) && (variableAlcoholAmbiente == 1))
47 {
48     int digito = 9;
49     Serial.print(digito);
50     delay(50);
51     Serial.println(' ');
52 //     delay(200);
53 //     Serial.print("DETECTADO ALCOHOL");
54     delay (100);
55 //     Serial.println(' ');
56     int estado_bomba = HIGH;
57     EEPROM.write(1,estado_bomba);
58 //     digitalWrite(bombaCombustible, estado_bomba);
59 //     Serial.print("APAGADO");
60     delay(50);
61 //     Serial.println(' ');
62 }
63 else{
64     int digito = 7;
65     Serial.print(digito);
66     delay(50);
```

```
67     Serial.println(' ');
68 }
69 }
70
71 //////////////////////////////////////////////////MÉTODO PARA CONVERTIR DE TIPO FLOAT A STRING////////////////////////////////////
72 String Dato;
73 String floatToString(float x, byte precision = 2) {
74     char tmp[50];
75     dtostrf(x, 0, precision, tmp);
76     return String(tmp);
77 }
78 void setup(){
79     Serial.begin(9600);           //Velocidad de la comunicación serial
80     sensor.initialise (4);       //Inicializar la lectura del sensor de Temperatura MLX90621
81     dht.begin();                 //Inicializa la lectura del modulo DHT11
82     pinMode(bombaCombustible, OUTPUT); //Define el pin de salida para el canal uno del módulo Relé
83     pinMode(BotonPanico, OUTPUT);  //Define el pin de salida para el canal dos del módulo Relé
84     digitalWrite(BotonPanico, HIGH); //Permanece la salida del rele en normalmente abierto
85     attachInterrupt (1, pulsador, RISING); //Interrupción para el pulsador
86     digitalWrite(bombaCombustible,EEPROM.read(1)); //Permanece la salida del rele en normalmente abierto
87 }
```

```
88 void loop(){
89     //motor();
90     sensor.measure(true);           //Obtiene nuevos valores del sensor de Temperatura MLX90621
91     if(Serial.available()) {        //Devuelve un entero con el número de bytes (caracteres) disponibles para leer desde
92                                     //el buffer serie, ó 0 si no hay ninguno
93         char r = Serial.read();     //Lee el caracter del buffer serie.
94         ///////////////////////////////////Enviar la cadena con la Temperatura Facial////////////////////////////////////
95         if (r=='A') {
96             for(int y=0;y<4;y++){   //Recorrer las filas de la matriz del sensor de Temperatura MLX90621
97                 for(int x=0;x<16;x++){ //Recorrer las columnas de la matriz del sensor de Temperatura MLX90621
98                     int16_t valueAtXY= sensor.irData[y+x*4]; // Extrae la temperatura en la posición x/y
99                     dato=sensor.getTemperature(y+x*4); //Obtiene el dato de la temperatura objeto
100                    String str = floatToString(dato); //Convierte el dato de float a string
101                    tam = str.length(); //Obtiene el tamaño de la cadena
102                    if (tam==4){ //Condición para aumentar un cero a la izquierda de la columna
103                        Dato ="0" + str;
104                    }
105                    if (tam==5){ //Condición para mantener la cadena en su forma original
106                        Dato = str;
107                    }
108                    Serial.print(Dato); //Enviar la cadena con los valores de la temperatura objeto
109                    //Serial.print(" ");
110                }
111            }
112        }
113    }
```

```
111     }
112     Serial.println(' ');
113 }
114 ///////////////////////////////////////////////////Enviar la cadena con la Temperatura Ambiente////////////////////////////////////
115 if (r=='B') {
116     Serial.print(sensor.getAmbient());           //Obtiene y envia la cadena con la temperatura ambiente del sensor
117                                                  //de Temperatura MLX90621
118     delay (10);
119     Serial.println(' ');
120 }
121 ///////////////////////////////////////////////////Enviar la cadena con la lectura del sensor MQ3////////////////////////////////////
122 if (r=='C') {
123     int adc_MQ = analogRead(A0); //Lemos la salida analógica del MQ
124     float voltaje = adc_MQ * (5.0 / 1023.0); //Convertimos la lectura en un valor de voltaje
125     float Rs=1000*((5-voltaje)/voltaje); //Calculamos Rs con un RL de 1k
126     double alcohol=0.5491*pow(Rs/5463, -0.682); // calculamos la concentración de alcohol.
127     Serial.print(alcohol);
128     delay (10);
129     Serial.println(' ');
130 }
131 ///////////////////////////////////////////////////Enviar la cadena con la lectura de la humedad del sensor DHT11////////////////////////////////////
132 if (r=='D') {
133     float h = dht.readHumidity();
```



```
133     float h = dht.readHumidity();
134     Serial.print(h);
135     delay(50);
136     Serial.println(' ');
137 }
138 ///////////////////////////////////////////////////Enviar la cadena con la lectura de la temperatura del sensor DHT11////////////////////////////////////
139 if (r=='E') {
140     float t = dht.readTemperature();
141     Serial.print(t);
142     delay(50);           //Recordad que solo lee una vez por medio segundo
143     Serial.println(' ');
144 }
145 ///////////////////////////////////////////////////DESACTIVA LA BOMBA DE COMBUSTIBLE////////////////////////////////////
146 if (r=='F') {
147     int estado_bomba = HIGH;
148     EEPROM.write(1,estado_bomba);
149     digitalWrite(bombaCombustible, estado_bomba);
150     Serial.print("APAGADO");
151     delay(50);
152     Serial.println(' ');
153 }
154 ///////////////////////////////////////////////////ACTIVA LA BOMBA DE COMBUSTIBLE////////////////////////////////////
155 if (r=='G') {
```

```
155     if (r=='G') {
156         int estado_bomba = LOW;
157         EEPROM.write(1,estado_bomba);
158         digitalWrite(bombaCombustible, estado_bomba);
159         Serial.print("ENCENDIDO");
160         delay(50);
161         Serial.println(' ');
162         variableAlcoholAmbiente = 1;
163     }
164     ///////////////////////////////////////////////////Enviar la cadena con la lectura del sensor MQ3////////////////////////////////////
165     if (r=='H') {
166         int adc_MQ = analogRead(A1); //Lemos la salida analógica del MQ
167         float voltaje = adc_MQ * (5.0 / 1023.0); //Convertimos la lectura en un valor de voltaje
168         float Rs=1000*((5-voltaje)/voltaje); //Calculamos Rs con un RL de 1k
169         double alcohol=0.5491*pow(Rs/5463, -0.682); // calculamos la concentración de alcohol.
170         Serial.print(alcohol);
171         delay (10);
172         Serial.println(' ');
173     }
174     ///////////////////////////////////////////////////Desactiva el pulsador////////////////////////////////////
175     if (r=='J') {
176         estado_pulsador = 0;
177         Serial.print("BOTON DESACTIVADO");
```

```
178     delay(50);
179     Serial.println(' ');
180 }
181 //////////////////////////////////////////////////DESACTIVA LA BOMBA DE COMBUSTIBLE////////////////////////////////////
182 if (r=='K') {
183     int estado_bomba = HIGH;
184     EEPROM.write(1,estado_bomba);
185     digitalWrite(bombaCombustible, estado_bomba);
186     int digito = 5;
187     Serial.print(digito);
188     delay(50);
189     Serial.println(' ');
190 }
191 //////////////////////////////////////////////////Enviar la cadena con la lectura del sensor MQ3////////////////////////////////////
192 if (r=='R') {
193     alcoholAmbiente();
194 }
195 //////////////////////////////////////////////////ACTIVA EL BOTÓN DE PÁNICO A TRAVÉS DEL CANAL DOS DEL MODULO RELÉ////////////////////////////////////
196 if (r=='Z') {
197     digitalWrite(BotonPanico, LOW);
198     delay(3000);
199     digitalWrite(BotonPanico, HIGH);
200     delay(1000);
```

```
EnviarDatosRaspberry$ DHT.cpp DHT.h MLX90621.cpp wireMLX90621.h
181 //////////////////////////////////////////////////DESACTIVA LA BOMBA DE COMBUSTIBLE////////////////////////////////////
182 if (r=='K') {
183     int estado_bomba = HIGH;
184     EEPROM.write(1,estado_bomba);
185     digitalWrite(bombaCombustible, estado_bomba);
186     int digito = 5;
187     Serial.print(digito);
188     delay(50);
189     Serial.println(' ');
190 }
191 //////////////////////////////////////////////////Enviar la cadena con la lectura del sensor MQ3////////////////////////////////////
192 if (r=='R') {
193     alcoholAmbiente();
194 }
195 //////////////////////////////////////////////////ACTIVA EL BOTÓN DE PÁNICO A TRAVÉS DEL CANAL DOS DEL MODULO RELÉ////////////////////////////////////
196 if (r=='Z') {
197     digitalWrite(BotonPanico, LOW);
198     delay(3000);
199     digitalWrite(BotonPanico, HIGH);
200     delay(1000);
201 }
202 }
203 delay(500);
204 }
```

Anexo 2. Programación script para lectura del nivel de alcohol de la placa Arduino

```
nu sensorGas.py - /home/pi/RecoleccionDatos/SensorGas/sensorGas.py (3.4.2)
```

```
File Edit Format Run Options Windows Help
1 import subprocess      #Modulo empleado para los mensajes de audio
2 import serial          #Modulo empleado para establecer la comunicacion con la plataforma ARDUINO
3 import cv2             #Modulo empleado para utilizar las funciones de vision por computador
4 import time            #Modulo empleado para trabajar con fechas y/o horas
5
6 ser = serial.Serial('/dev/ttyUSB0', 9600)      #Define la variable que identifica el puerto al cual esta conectado el Arduino UNOR3 y la velocidad
7
8 ser.close()          #Cierra la variable para la comunicacion serial
9 ser.open()           #Abre la variable para iniciar la comunicacion serial con los parametros establecidos
10
11 time.sleep(1)        #Retardo de un segundo
12 ser.setDTR(level=0) #Reset manual del arduino para evitar lectura de datos erroneos cada vez que el arduino abra el puerto serial
13 time.sleep(0.5)     #Retardo de medio segundo
14
15 while True:          #Iniciar un ciclo hasta recibir el nivel de alcohol
16     time.sleep(1)    #Retardo de un segundo
17     ser.write('C')  #Enviar el caracter 'C' para recibir el dato del nivel de alcohol de la placa Arduino
18     alcohol = ser.readline() #Lee una linea enviada por el arduino
19     alcohol = alcohol.strip() #Compara los caracteres de inicio y fin
20     b = len(alcohol) #Obtener el tamaño de la cadena
21     alcohol = (float(alcohol)) #Convierte la variable string a tipo float
22     print alcohol    #Escribir el valor de alcohol
23     VozNivelAlcohol = "Nivel de alcohol almacenado" #Mensaje para indicar que el nivel de alcohol se ha almacenado
24     subprocess.call('echo '+VozNivelAlcohol+'|festival --tts --language spanish', shell=True) #Define la funcion para el mensaje de audio
25     def Gas():      #Funcion para retornar el valor de alcohol
26         return alcohol
27     cv2.destroyAllWindows() #Cerrar la ventana en la que se ejecuta el script
28     break           #salir del ciclo
```

Anexo 3. Programación script para lectura de la temperatura ambiente de la placa Arduino

```
num temperaturaAmbiente.py - /home/pi/RecoleccionDatos/TemperaturaAmbiente/temperaturaAmbiente.py (3.4.2)
File Edit Format Run Options Windows Help
1 import serial #Modulo empleado para establecer la comunicacion con la plataforma ARDUINO
2 import cv2 #Modulo empleado para utilizar las funciones de vision por computador
3 import time #Modulo empleado para trabajar con fechas y/o horas
4
5 ser = serial.Serial('/dev/ttyUSB0', 9600) #Define la variable que identifica el puerto al cual esta conectado el Arduino UNOR3 y la velocidad
6
7 ser.close() #Cierra la variable para la comunicacion serial
8 ser.open() #Abre la variable para iniciar la comunicacion serial con los parametros establecidos
9
10 time.sleep(1) #Retardo de un segundo
11 ser.setDTR(level=0) #Reset manual del arduino para evitar lectura de datos erroneos cada vez que el arduino abra el puerto serial
12 time.sleep(0.5) #Retardo de medio segundo
13 cont = 0
14
15 while True: #Iniciar un ciclo hasta recibir el valor de la temperatura ambiente
16     time.sleep(1) #Retardo de un segundo
17     ser.write('B') #Enviar el caracter 'B' para recibir el dato de la temperatura ambiente
18     time.sleep(1) #Retardo de un segundo
19     temp_amb = ser.readline() #Lee una linea enviada por el arduino
20     temp_amb = temp_amb.strip() #Compara los caracteres de inicio y fin
21     # print temp_amb
22     b = len(temp_amb) #Obtener el tamaño de la cadena
23     def TempAmb(): #Funcion para retornar el valor de alcohol
24         return temp_amb
25     if (b>1): #Condicion para verificar que se ha recibido el dato
26         cont = 2
27     if cont == 2:
28         cv2.destroyAllWindows() #Cerrar la ventana en la que se ejecuta el script
29         break #salir del ciclo
30
```

Ln: 30 Col: 0

Anexo 4. Programación script para lectura de la temperatura facial de la placa Arduino

```
num Temperatura.py - /home/pi/RecoleccionDatos/TemperaturaFacial/Temperatura.py (3.4.2)
File Edit Format Run Options Windows Help
1      #MODULOS
2  import subprocess          #Modulo empleado para los mensajes de audio
3  import colores             #Modulo creado para asignar un color a cada valor de temperatura
4  import posicionesMatriz   #Modulo creado para alinear los datos recibidos del sensor de Temperatura con la imagen
5  import serial              #Modulo empleado para establecer la comunicacion con la plataforma ARDUINO
6  import cv2                 #Modulo empleado para utilizar las funciones de vision por computador
7  import time                #Modulo empleado para trabajar con fechas y/o horas
8  from picamera.array import PiRGBArray #Biblioteca empleada para construir matrices n-dimensional numpy a partir de la salida de la camara
9  from picamera import PiCamera #Biblioteca empleada para hacer uso de las clases para interactuar con el modulo de la camara del Raspberry Pi
10 import skimage             #Modulo usado para el procesamiento de imagenes
11 from skimage import exposure #Biblioteca empleada para mejorar el contraste de una imagen
12 import numpy as np         #Modulo empleado para operar con vectores o matrices
13 import os                  #Modulo empleado para utilizar la funcionalidad dependiente del sistema operativo
14 import matplotlib         #Modulo empleado para realizar graficas en 2 y 3 dimensiones
15
16      #PARAMETROS PARA ADQUIRIR LA IMAGEN
17 camera = PiCamera()        #Crea un objeto para llamar a la camara
18 camera.resolution = (640, 480) #Define la resolucioin
19 camera.framerate = 30      #Establece la velocidad de fotogramas en la que se ejecutan las capturas de imagen
20 Camara = PiRGBArray(camera, size=(640, 480)) #Capturar video
21 ventana_visualizacion = cv2.namedWindow("Faces") #Crea una ventana para cargar la imagen mas tarde
22
23 fn_haar = '/home/pi/data/haarcascades/haarcascade_frontalface_alt.xml' #Direccion del archivo para la clasificacion
24 haar_cascade = cv2.CascadeClassifier(fn_haar) #Cargar plantilla de clasificacion
25
26      #DEFINIR VARIABLES A EMPLEAR
27 texto = ""                 #Variable para almacenar los datos recibidos del arduino
28 textomatriz = ""          #Variable para almacenar la matriz de la temperatura facial
29 d=0                         #Variable que indica el inicio de la cadena de la temperatura facial recibida
30 e=5                         #Variable que indica el fin de la cadena que contiene la temperatura facial
31 cont = 0                   #Variable para inicializar el conteo de las veces que se busca el rostro en la imagen
32 tam = 0                    #Variable para especifica cuantos datos de la matriz corresponden a la temperatura del rostro
33 llenar = 0                 #Variable para almacenar los datos de temperatura
34 xinicio = 0                #Variable para almacenar el inicio del vector que contiene la temperatura facial
35 x = 0                      #Variable para almacenar inicio del ancho de rostro
36 y = 0                      #Variable para almacenar inicio del alto de rostro
```

num Temperatura.py - /home/pi/RecoleccionDatos/TemperaturaFacial/Temperatura.py (3.4.2)

```
File Edit Format Run Options Windows Help
37 h = 0 #Variable para almacenar fin del ancho del rostro
38 w = 0 #Variable para almacenar fin de la altura del rostro
39 numeroveces = 0 #Variable para almacenar el numero de veces que se ha terminado un ciclo de busqueda
40 guardarImagen = 0 #Variable para almacenar el resultado
41 mover = 0 #Variable para mover la posicion para almacenar la temperatura facial
42
43 matriz = np.random.rand(4,16) #Crea una matriz de 16x4 para almacenar los datos de temperatura
44 vectorResultado = np.random.rand(tam) #Crea una matriz del tamaño especificado por la variable tam para almacenar solo la temperatura del rostro
45
46 ser = serial.Serial('/dev/ttyUSB0', 9600) #Define la variable que identifica el puerto al cual esta conectado el Arduino UNOR3 y la velocidad
47 time.sleep(1) #Establece un retardo de 1 segundo
48 ser.setDTR(level=0) #Reset manual del arduino para evitar lecturas de datos erroneos cada vez que el arduino abra el puerto serial
49 time.sleep(0.5) #Establece un retardo de 0.5 segundos
50
51 #Direccion para almacenar las imagenes adquiridas
52 Direccion_Carpeta = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/'
53 Direccion_Principal = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/'
54 nombre_Carpeta1 = "TemperaturaFacial"
55 nombre_TempFacialPGM = "TempFacialPGM"
56 nombre_TempFacialJPG = "TempFacialJPG"
57 nombre_RostroPGM = "RostroPGM"
58 nombre_RostroJPG = "RostroJPG"
59 TemperaturaPGM = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/TempFacialPGM'
60 TemperaturaJPG = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/TempFacialJPG'
61 RostroPGM = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/RostroPGM'
62 RostroJPG = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/RostroJPG'
63
64 #CREA CARPETAS PARA ALMACENAR LOS RESULTADOS
65 CarpetaPrincipal = os.path.join(Direccion_Carpeta, nombre_Carpeta1) #Especifica el nombre de la carpeta y la direccion en la cual se crea la
66 if not os.path.isdir(CarpetaPrincipal): #Comprueba que no exista la carpeta
67     os.mkdir(CarpetaPrincipal) #Crea la carpeta
68
69 CarpetaTempFacialPGM = os.path.join(Direccion_Principal, nombre_TempFacialPGM) #Especifica el nombre de la carpeta y la direccion en la cual se crea la
70 if not os.path.isdir(CarpetaTempFacialPGM): #Comprueba que no exista la carpeta
71     os.mkdir(CarpetaTempFacialPGM) #Crea la carpeta
72
73 CarpetaTempFacialJPG = os.path.join(Direccion_Principal, nombre_TempFacialJPG) #Especifica el nombre de la carpeta y la direccion en la cual se crea la
74 if not os.path.isdir(CarpetaTempFacialJPG): #Comprueba que no exista la carpeta
```



```

File Edit Format Run Options Windows Help
75     os.mkdir(CarpetaTempFacialJPG)                #Crea la carpeta
76
77     CarpetaRostroPGM = os.path.join(Direccion_Principal, nombre_RostroPGM)    #Especifica el nombre de la carpeta y la direccion en la cual se creara la
78     if not os.path.isdir(CarpetaRostroPGM):        #Comprueba que no exista la carpeta
79         os.mkdir(CarpetaRostroPGM)                #Crea la carpeta
80
81     CarpetaRostroJPG = os.path.join(Direccion_Principal, nombre_RostroJPG)    #Especifica el nombre de la carpeta y la direccion en la cual se creara la
82     if not os.path.isdir(CarpetaRostroJPG):        #Comprueba que no exista la carpeta
83         os.mkdir(CarpetaRostroJPG)                #Crea la carpeta
84
85     #COMUNICACION SERIAL
86     ser.write('A')                                #Envia el caracter 'A' para recibir la cadena con los datos de la matriz 16x4 de la temperatura facial
87     temp_facial = ser.readline()                   #Lee una linea enviada por el arduino
88     temp_facial = temp_facial.strip()              #Compara los caracteres recibidos de inicio a fin
89     b = len(temp_facial)                           #Obtiene el tamaño de la cadena
90     print b                                        #imprime el tamaño de la cadena
91
92     #INICIO DEL CICLO PARA CAPTAR LA TEMPERATURA FACIAL
93     for frame in camera.capture_continuous(Camara, format="bgr", use_video_port=True): #Captura video en formato RGB
94         key = cv2.waitKey(1)                       #Estable un retardo de un segundo
95         Camara.truncate(0)                          #Utilizar la salida para producir varias matrices
96
97     #DIRECCIONES PARA ALMACENAR LAS IMAGENES
98     TemperaturaPGM = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/TempFacialPGM'
99     TemperaturaJPG = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/TempFacialJPG'
100    RostroPGM = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/RostroPGM'
101    RostroJPG = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/TemperaturaFacial/RostroJPG'
102
103    #INICIALIZACION DE VARIABLES EMPLEADAS
104    texto = ""
105    textom = ""
106    d=0
107    e=5
108    m=0
109    n=5
110    fila= -10
111    columna= 300+mover
112    fila1 = -10

```

num Temperatura.py - /home/pi/RecoleccionDatos/TemperaturaFacial/Temperatura.py (3.4.2)

```
File Edit Format Run Options Windows Help
113     columna1 = 60+mover
114     f=0
115     c=3
116     filatemp = 30
117     filatemp1 = 30
118     columnatemp = 30
119     columnatemp1 = 30
120
121     #PREPROCESAMIENTO DE LA IMAGEN
122     image = frame.array                #Almacena una imagen del video
123     gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY) #Convierte la imagen a escala de grises
124     equ = exposure.equalize_hist(gray)  #Aplica a la imagen el ecualizador del histograma
125
126     #DERECTAR EL ROSTRO
127     Rostros = haar_cascade.detectMultiScale(gray,1.1,5) #Busca las coordenadas de los rostros y guarda su posicion
128     print 'Rostros Detectados: %d' % (len(Rostros))      #Imprime el numero de rostros encontrados en la imagen
129
130     #COMUNICACION SERIAL
131     ser.write('A')                #Envia el caracter 'A' para recibir la cadena con el valor de la matriz 16x4 de la temperatura facial
132     temp_facial = ser.readline()   #Lee una linea enviada por el arduino
133     temp_facial = temp_facial.strip() #Compara los caracteres recibidos de inicio a fin
134     b = len(temp_facial)           #Obtiene el tamaño de la cadena
135
136     #IMPRIME EL VALOR DEL TAMANO DE LA CADENA Y LA CADENA
137     print "Tamano de la cadena Temperatura Objeto"
138     print b
139     print "Cadena de la Temperatura Objeto"
140     print temp_facial
141
142     #CONDICION PARA VERIFICAR EL NUMERO DE CARACTERES RECIBIDOS
143     if (b==320):                    #Condicion para verificar que se han recibido todos los valores de la matriz 16x4
144
145         #SEPARA LA CADENA DE LA TEMPERATURA FACIAL EN UNA MATRIZ 16X4
146         for j in range (64):
147             for i in range(d,e):
148                 texto = texto + temp_facial[i]
149                 entero = int(float(texto))
150                 color = colores.ColoresTemperatura(entero) #Emplea la libreria ColoresTemperatura para asignar un color a cada valor de tempe
```

```
File Edit Format Run Options Windows Help
151
152         fila=fila+35                                #Incrementa la posicion de la fila para dibujar un rectangulo
153         cv2.rectangle(image,(fila,columna),(fila+35,columna+80),color,-10)    #Dibuja un rectangulo sobre la imagen con el color de la t
154
155     #CONDICIONES PARA DIBUJAR UN RECTANGULO EN CADA VALOR DE TEMPERATURA
156     if(j==15):
157         fila= -10
158         columna=220+mover
159     if(j==31):
160         fila= -10
161         columna=140+mover
162     if(j==47):
163         fila= -10
164         columna= 60+mover
165
166     #CONDICION PARA LIMPIAR LA CADENA DE CADA TEMPERATURA DESPUES DE ALMACENAR UN VALOR
167     if (e==d+5):
168         d=e
169         e=d+5
170         texto = ""
171
172     #CICLO PARA ALMACENAR LOS VALORES DE TEMPERATURA EN UNA MATRIZ 16X4
173     for k in range (64):
174         for l in range(m,n):
175             textom = textom + temp_facial[l]          #Ciclo para dividir la cadena en los 64 valores de temperatura
176             matriz[c,f] = float(textom)             #Almacenar a una cadena el valor de cada temperatura
177             f=f+1                                    #Almacenar el valor de tempertarua en la matriz con la cadena convertida a tipo fl
178             #Condiciones para incrementar las columnas de la matriz e inicializar en cero la fila
179             if(k==15):
180                 f=0
181                 c=2
182             if(k==31):
183                 f=0
184                 c=1
185             if(k==47):
186                 f=0
187                 c=0
```

num Temperatura.py - /home/pi/RecoleccionDatos/TemperaturaFacial/Temperatura.py (3.4.2)

```
File Edit Format Run Options Windows Help
188         #Condicion para almacenar cada caracter a una cadena de 5 caracteres
189         if (n==m+5):
190             m=n
191             n=m+5
192             textom = ""
193
194     #IMPRIME LA MATRIZ 16X4 CON LOS DATOS DE LA TEMPERATURA
195     print "Matriz 16x4 con los valores de las temperaturas objeto"
196     print matriz
197
198     #CONDICION PARA VERIFICAR QUE SE HA ENCONTRADO UN ROSTRO
199     if(len(Rostros)>0):
200         VozRostroDetectado = "Rostro del Conductor Detectado" #Mensaje para el audio cuando se ha detectado un rostro
201         #Emplea el sinterizador de texto a voz Festival para leer la instruccion de que se ha detectado un rostro
202         subprocess.call('echo '+VozRostroDetectado+'|festival --tts --language spanish', shell=True)
203
204     #CICLO PARA ASIGNAR LAS POSICIONES DE LAS TEMPERATURAS QUE COINCIDAN CON EL ROSTRO DETECTADO
205     for (x,y,w,h) in Rostros:
206         cv2.rectangle(image,(x,y),(x+w,y+h),(255,0,0),3) #Dibuja un rectangulo con las coordenadas del rostro detectado
207         filaFin = x+w #Asigna el valor para conocer el fin de las filas del rostro
208         columnaFin = y+h #Asigana el valor para conocel el fin de las columnas del rostro
209
210     #Condiciones para verificar que el rostro se encuentre en el rango de la matriz de temperatura
211     if(y >= 61):
212         mover = y - 50
213     if(y <= 60):
214         mover = -y
215
216     filatemp = posicionesMatriz.PosicionX(x,filal) - 1 #Asigna una variable temporal para el inicio de las filas
217     filatemp1 = posicionesMatriz.PosicionFilaFin(filaFin,filal) - 1 #Asigna una variable temporal para el fin de las filas
218
219     #Condicion para verificar asignar el fin de la fila cuando la coordenada del rectangulo del rostro se sale de la matriz
220     if (filatemp1 == 29):
221         filatemp1 = 15
222
223     columnatemp = posicionesMatriz.PosicionY(y,columnal) #Asigna una variable temporal para el inicio de las columnas
224
```

num Temperatura.py - /home/pi/RecoleccionDatos/TemperaturaFacial/Temperatura.py (3.4.2)

```
File Edit Format Run Options Windows Help
225     #Condicion para verificar que se asigna el inicio de la columna
226     if (columnatemp == 30):
227         columnatemp = 0
228
229     columnatemp1 = posicionesMatriz.PosicionColumnaFin(columnaFin,columna1) #Asigna una variable temporal para el fin de las columnas
230
231     numeroveces = numeroveces + 1 #Incrementa el numero de veces que no se ha detectado un rostro
232
233     #Condicion para emitir una instruccion que indique que no se ha podido almacenar la temperatura
234     if numeroveces > 2:
235         VozUbicarse = "No se completo el almacenamiento de la temperatura mirar fijamente a la camara"
236         subprocess.call('echo '+VozUbicarse+'|festival --tts --language spanish', shell=True)
237
238     #CONDICION PARA EXTRAR LOS VALORES DE TEMPERATURA SOLO DEL ROSTRO EN UN VECTOR
239     if(filatemp !=30 and columnatemp != 30 and filatemp1 !=30 and columnatemp1 != 30): #Condicion que confirma que el rostro se e
240         xinicio = filatemp #Variable para almacenar el inicio del vector
241         tam = (filatemp1 - filatemp + 1)*(columnatemp1 - columnatemp + 1) #Variable para conocer el tamaño del vector
242         vectorResultado = np.random.rand(tam) #Crea un vector con valores aleatorios en tipo float
243
244     #Condicion para almacenar los valores de temperatura en el vector creado
245     if(tam != 0):
246         for km in range (tam): #Ciclo para recorrer el vector
247             #Condicion para verificar que no se llegue al fin del vector
248             if(filatemp <= filatemp1):
249                 vectorResultado[km]= matriz[columnatemp,filatemp]#Asignar un valor de temperatura al vector de la
250                 filatemp = filatemp + 1 #Incrementar la posicion de la fila de la matriz
251                 #Condicion para cambiar la columna de la matriz
252                 if((filatemp == (filatemp1 + 1))and (columnatemp < columnatemp1)):
253                     columnatemp = columnatemp + 1 #Incrementa la posicion de la columna de la matriz
254                     filatemp = xinicio #Inicializa la posicion de la fila de la matriz
255
256     #Imprime el vector con los datos de la temperatura facial
257     print "Vector con las temperaturas faciales del conductor"
258     print vectorResultado
259
260     #Devuelve el valor de la temperatura maxima del vector
261     Rostro_tempmax = vectorResultado.max()
262     print "Temperatura facial maxima"
```

num Temperatura.py - /home/pi/RecoleccionDatos/TemperaturaFacial/Temperatura.py (3.4.2)

```
File Edit Format Run Options Windows Help
263         #print Rostro_tempmax
264
265         #Devuelve el valor de la temperatura minima del vector
266         Rostro_tempmin = vectorResultado.min()
267         print "Temperatura facial minima"
268         #print Rostro_tempmin
269
270         #CREA LAS FUNCIONES PARA RECUPERAR LA TEMPERATURA MAXIMA Y MINIMA DEL ROSTRO DEL CONDUCTOR
271         def TempMax():
272             return Rostro_tempmax
273         def TempMin():
274             return Rostro_tempmin
275
276         #Asigna un numero para almacenar las imagenes con los resultados
277         pin= sorted([int(n[:n.find('.')]) for n in os.listdir(CarpetaRostroJPG)
278                     if n[0]!='.' ]+[0])[-1] + 1
279         print pin
280
281         #ALMACENA LAS IMAGENES DE LOS RESULTADOS EN LAS DIRECCIONES ESTABLECIDAS
282         cv2.imwrite('%s/%s.pgm' % (TemperaturaPGM, pin), image)
283         cv2.imwrite('%s/%s.jpg' % (TemperaturaJPG, pin), image)
284         cv2.imwrite('%s/%s.pgm' % (RostroPGM, pin), gray)
285         cv2.imwrite('%s/%s.jpg' % (RostroJPG, pin), gray)
286         cv2.imwrite('/home/pi/RecoleccionDatos/TemperaturaFacial/Liceth/1.jpg', image)
287         cv2.imwrite('/home/pi/RecoleccionDatos/TemperaturaFacial/Liceth/1.jpg', gray)
288         cv2.imshow("Faces", image)
289
290
291         cont = 1         #Asigna el uno a la variable que especifica que se ha almacenado la temperatura facial
292     #         if w < 80:
293     #             print 'Rostro muy lejos: %d' % (w)
294
295     #VISUALIZACION DE LOS RESULTADOS
296     cv2.imshow("Faces", image)         #Muestra la imagen en la pantalla de visualizacion
297     key = cv2.waitKey(1)               #Retardo de un segundo
298     Camara.truncate(0)                 #Utilizar la salida para producir varias matrices
299
```

num Temperatura.py - /home/pi/RecoleccionDatos/TemperaturaFacial/Temperatura.py (3.4.2)

File Edit Format Run Options Windows Help

```
300 #Condicion para salir del ciclo si se presiona la tecla 'q'
301 if key == ord("q"):
302     camera.close() #Cierra el puerto de la camara
303     cv2.destroyAllWindows() #Cierra la ventana de visualizacion
304     break #Sale del ciclo
305
306 #Condicion para mostras en la pantalla la imagen guardada de la temperatura facial
307 if cont == 1:
308     imagenguardada = cv2.imread('/home/pi/RecoleccionDatos/TemperaturaFacial/Liceth/1.jpg') #Lee la imagen en la direccion establecida
309     cv2.imshow('TEMPERATURA FACIAL', imagenguardada) #Muestra la imagen leida
310     VozTempFacial = "Temperatura Facial Almacenada" #Mensaje para indicar que la temperatura facial ha sido almacenada
311     #Emplea el sinterizador de texto a voz Festival para leer la instruccion de que se ha detectado un rostro
312     subprocess.call('echo '+VozTempFacial+'|festival --tts --language spanish', shell=True)
313     key = cv2.waitKey(4000) #Retardo de 4 segundo para mostrar la imagen
314     camera.close() #Cierra el puerto de la camara
315     cv2.destroyAllWindows() #Cierra la ventana de visualizacion
316     break #Sale del ciclo
317
```

Ln: 317 Col: 0

Anexo 5. Programación script para adquirir imagen en formato RGB

```
num adquirirImagen.py - /home/pi/AdquirirImagen/adquirirImagen.py (3.4.2)
File Edit Format Run Options Windows Help
1      #AGREGAR LOS MODULOS NECESARIOS
2  import io                                #Modulo utilizado parqa leer y escribir imagenes en varios formatos
3  import picamera                          #Modulo empleado para interacturar conel modulo de la camara del Raspberry Pi
4  import cv2                              ##Modulo empleado para utilizar las funciones de vision por computador
5  import numpy                             #Modulo empleado para operar con vectores o matrices
6
7  stream=io.BytesIO()                      #Crear una secuencia de memoria para guardar las imagenes
8
9  with picamera.PiCamera() as camera:      #Crea un objeto para llamar a la camara
10     camera.resolution = (1920,1080)      #Define la resolucion
11     camera.capture(stream,format='jpeg')  #Define el formato
12     buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8) #Convierte la imagen en una matriz numpy
13     image=cv2.imdecode(buff,1)           #Crea una imagen empleando OpenCV
14
15
16     cv2.imwrite('AdquisionImg2.jpg',image) #Guarda el resultado
17     img = cv2.imread('AdquisionImg2.jpg') #Leer la imagen
18     cv2.imshow('AdquisionImagen2',img)   #Visualizar la imagen
19
```

Ln: 19 Col: 0

Anexo 6. Programación script para preprocesamiento de imágenes

```
num ecualizar.py - /home/pi/Ecualizar/ecualizar.py (3.4.2)
File Edit Format Run Options Windows Help
1 #Agregar las librerias necesarias
2 import io #Modulo utilizado parqa leer y escribir imagenes en varios formatos
3 import picamera #Modulo empleado para interacturar conel modulo de la camara del Raspberry Pi
4 import cv2 #Modulo empleado para utilizar las funciones de vision por computador
5 import numpy #Modulo empleado para operar con vectores o matrices
6
7 stream=io.BytesIO() #Crear una secuencia de memoria para guardar las imagenes
8
9 with picamera.PiCamera() as camera: #Crea un objeto para llamar a la camara
10     camera.resolution = (1920,1080) #Define la resolucion
11     camera.capture(stream,format='jpeg') #Define el formato
12     buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8) #Convierte la imagen en una matriz numpy
13 image=cv2.imdecode(buff,1) #Crea una imagen empleando OpenCV
14
15 escalaGrises=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY) #Convierte a escala de grises
16
17 equ = cv2.equalizeHist(escalaGrises) #Aplicar equalizacion
18
19 cv2.imwrite('AdquisionImg2.jpg',image) #Guarda el resultado
20 cv2.imwrite('EscalaGrises2.jpg',escalaGrises) #Guarda el resultado
21 cv2.imwrite('Ecualizar2.jpg',equ) #Guarda el resultado
22
Ln: 22 Col: 0
```

Anexo 7. Programación script para detectar rostro

```
num detectarRostro.py - /home/pi/DetectarRostro/detectarRostro.py (3.4.2)
File Edit Format Run Options Windows Help
1 #Agregar las librerias necesarias
2 import io #Modulo utilizado parqa leer y escribir imagenes en varios formatos
3 import picamera #Modulo empleado para interactuar conel modulo de la camara del Raspberry Pi
4 import cv2 #Modulo empleado para utilizar las funciones de vision por computador
5 import numpy #Modulo empleado para operar con vectores o matrices
6
7 stream=io.BytesIO() #Crear una secuencia de memoria para guardar las imagenes
8
9 with picamera.PiCamera() as camera: #Crea un objeto para llamar a la camara
10     camera.resolution = (1920,1080) #Define la resolucion
11     camera.capture(stream,format='jpeg') #Define el formato
12     buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8) #Convierte la imagen en una matriz numpy
13 image=cv2.imdecode(buff,1) #Crea una imagen empleando OpenCV
14
15 face_cascade = cv2.CascadeClassifier('/home/pi/data/haarcascades/haarcascade_frontalface_alt.xml')#Cargar la plantilla
16
17 grises=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY) #Convierte a escala de grises
18
19 rostro = face_cascade.detectMultiScale(grises,1.1,5) #Busca las coordenadas de los rostros y guarda su posicion
20
21 for (x,y,w,h) in rostro: #Ciclo para enmarcar con un rectangulo el rostro deectado
22     cv2.rectangle(image,(x,y),(x+w,y+h),(255,0,0),8) #Dibujar un rectangulo en el contorno del rostro de color azul
23
24 cv2.imwrite('Detect_rostro.jpg',image) #Guarda el resultado
25
```

Ln: 25 Col: 0

Anexo 8. Programación script para detectar ojos

```
num detectarOjos.py - /home/pi/DetectarOjos/detectarOjos.py (3.4.2)
File Edit Format Run Options Windows Help
1 #Agregar las librerias necesarias
2 import io #Modulo utilizado parqa leer y escribir imagenes en varios formatos
3 import picamera #Modulo empleado para interacturar conel modulo de la camara del Raspberry Pi
4 import cv2 #Modulo empleado para utilizar las funciones de vision por computador
5 import numpy #Modulo empleado para operar con vectores o matrices
6 import time #Modulo empleado para trabajar con fechas y/o horas
7
8 stream=io.BytesIO() #Crear una secuencia de memoria para guardar las imagenes
9
10 with picamera.PiCamera() as camara: #Crea un objeto para llamar a la camara
11     camara.resolution = (1920,1080) #Define la resolucion
12     camara.capture(stream,format='jpeg') #Define el formato
13     buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8) #Convierte la imagen en una matriz numpy
14 imagen=cv2.imdecode(buff,1) #Crea una imagen empleando OpenCV
15
16 rostro_cascade = cv2.CascadeClassifier('/home/pi/data/haarcascades/haarcascade_frontalface_alt.xml')#Cargar la plantilla Rostro
17 ojos_cascade = cv2.CascadeClassifier('/home/pi/data/haarcascades/haarcascade_eye_tree_eyeglasses.xml')#Cargar la plantilla Ojos
18 bocas_cascade = cv2.CascadeClassifier('/home/pi/data/haarcascades/Mouth.xml')#Cargar la plantilla Boca
19
20 grises=cv2.cvtColor(imagen,cv2.COLOR_BGR2GRAY) #Convierte a escala de grises
21
22 equ = cv2.equalizeHist(grises) #Aplica ecualizacion del histograma a la imagen
23
24 rostros = face_cascade.detectMultiScale(equ,1.1,5) #Busca las coordenadas de los rostros y guarda su posicion
25 print "Encontrados"+str(len(rostros))+"Rostros(s)" #Devuelve una lista con las coordenadas del rectangulo de la cara detectada
26 time.sleep(1) #Rerardo de un segundo
27
28 for (x,y,w,h) in rostros: #Ciclo para buscar los ojos dentro de las coordenadas del rostro
29     cv2.rectangle(imagen,(x,y),(x+w,y+h),(255,0,0),6) #Dibujar un rectangulo en el contorno del rostro
30     roi_grises = equ [y:y+h,x:x+w] #Variable con la imagen aplicada ecualizacion del histograma
31     roi_color = imagen [y:y+h,x:x+w] #Tener en una variable con la imagen en formato RGB
32     eyes = eye_cascade.detectMultiScale(roi_grises) #Busca las coordenadas de los ojos y guarda su posicion
33     print "Found"+str(len(eyes))+"eye(s)" #Imprime el numero de ojos encontrados en el rostro
34     time.sleep(2) #Retardo de dos segundos
35
```

```
num detectarOjos.py - /home/pi/DetectarOjos/detectarOjos.py (3.4.2)
File Edit Format Run Options Windows Help
36     for (ex,ey,ew,eh) in eyes:                #Ciclo para enmarcar los Ojos
37         cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),4)    #Dibuja un rectangulo de color verde en los ojos
38
39     boca = boca_cascade.detectMultiScale(roi_grises)    #Busca las coordenadas de la boca
40     print "Encontradas"+str(len(boca))+"bocas(s)"    #Imprime el numero de bocas encontradas en el rostro
41     time.sleep(3)    #Retardo de 3 segundos
42
43     for (mx,my,mw,mh) in bocas:                #Ciclo para enmarcar la boca
44         cv2.rectangle(roi_color,(mx,my),(mx+mw,my+mh),(0,0,255),2)    #Dibuja un rectangulo color rojo en la boca
45
46     cv2.imwrite('Detect_rostro_ojos_boca.jpg',imagen)    #Guarda el resultado
47
```

Ln: 47 Col: 0

Anexo 9. Programación script para diámetro pupilas

```
num Ojos.py - /home/pi/RecoleccionDatos/DiametroPupilas/Ojos.py (3.4.2)
File Edit Format Run Options Windows Help
1 #Agregar las librerias necesarias
2 import subprocess #Modulo empleado para los mensajes de audio
3 from picamera.array import PiRGBArray #Biblioteca empleada para construir matrices n-dimensional numpy a partir de la salida de la camara
4 from picamera import PiCamera #Biblioteca empleada para hacer uso de las clases para interactuar con el modulo de la camara del Raspberry Pi
5 import cv2 #Modulo empleado para utilizar las funciones de vision por computador
6 import numpy as np #Modulo empleado para operar con vectores o matrices
7 import os #Modulo empleado para utilizar la funcionalidad dependiente del sistema operativo
8 import time #Modulo empleado para trabajar con fechas y/o horas
9
10 camera = PiCamera() #Crea un objeto para llamar a la camara
11 camera.resolution = (1920, 1080) #Define la resolucion
12 camera.framerate = 30 #Establece la velocidad de fotogramas en la que se ejecutan las capturas de imagen
13 rawCapture = PiRGBArray(camera, size=(1920,1080)) #Capturar video
14
15 #DEFINICION DE VARIABLES
16 size = 4 #Variable para el tamaño de la imagen
17 guardar = 0 #Variable para almacenar los resultados
18 count = 0 #Variable para contar las veces que ha finalizado el ciclo buscando la pupila
19 numeroveces = 0 #Variable para contar las veces que ha finalizado el ciclo buscando la pupila
20
21 #Direccion para almacenar las imagenes adquiridas
22 face_cascade = '/home/pi/data/haarcascades/haarcascade_eye_tree_eyeglasses.xml'
23 face_direccion = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/DiametroPupilas/'
24 datos_direccion = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1'
25 datos_name = "DiametroPupilas"
26 fn_name = "OjosPGM"
27 fn_name1 = "OjosJPG"
28 fn_name2 = "FiltroPGM"
29 fn_name3 = "FiltroJPG"
30 fn_name4 = "PupilaPGM"
31 fn_name5 = "PupilaJPG"
32 OjosPGM = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/DiametroPupilas/OjosPGM'
33 OjosJPG = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/DiametroPupilas/OjosJPG'
34 GrayPGM = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/DiametroPupilas/FiltroPGM'
35 GrayJPG = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/DiametroPupilas/FiltroJPG'
36 PupilaPGM = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/DiametroPupilas/PupilaPGM'
```

```
File Edit Format Run Options Windows Help
37 PupilaJPG = '/home/pi/RecoleccionDatos/DatosEntrenamiento/Datos1/DiametroPupilas/PupilaJPG'
38
39         #CREA CARPETAS PARA ALMACENAR LOS RESULTADOS
40 path = os.path.join(datos_direccion, datos_name) #Especifica el nombre de la carpeta y la direccion en la cual se crea la carpeta
41 if not os.path.isdir(path): #Comprueba que no exista la carpeta
42     os.mkdir(path) #Crea la carpeta
43
44 path = os.path.join(face_direccion, fn_name)
45 if not os.path.isdir(path):
46     os.mkdir(path)
47
48 path = os.path.join(face_direccion, fn_name1)
49 if not os.path.isdir(path):
50     os.mkdir(path)
51
52 path = os.path.join(face_direccion, fn_name2)
53 if not os.path.isdir(path):
54     os.mkdir(path)
55
56 path = os.path.join(face_direccion, fn_name3)
57 if not os.path.isdir(path):
58     os.mkdir(path)
59
60 path = os.path.join(face_direccion, fn_name4)
61 if not os.path.isdir(path):
62     os.mkdir(path)
63
64 path = os.path.join(face_direccion, fn_name5)
65 if not os.path.isdir(path):
66     os.mkdir(path)
67
68 (im_width, im_height) = (112,92) #Define el alto y ancho de la imagen del ojo
```

```

File Edit Format Run Options Windows Help
69 haar_cascade = cv2.CascadeClassifier(face_cascade)      #Cargar plantilla de clasificacion
70 time.sleep(0.5)                                       #Retardo de medio segundo
71
72     #INICIO DEL CICLO PARA CAPTURAR LOS OJOS Y EL DIAMETRO DE LA PUPILA
73 for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
74     im = frame.array                                  #Almacena una imagen del video
75     im = cv2.flip(im, 1, 0)                          #Gira la imagen horizontalmente
76     gris= cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)       #Convierte la imagen a escala de grises
77     mini = cv2.resize(gris, (gris.shape[1] / size, gris.shape[0] / size)) #Recorta la imagen con el alto y ancho establecidos
78     ojo = haar_cascade.detectMultiScale(mini)        #Busca las coordenadas de los ojos y guarda su posicion
79     print "Encontrados"+str(len(ojo))+"ojos(s)"      #Imprime el numero de ojos encontrados
80
81     #Ciclo para dibujar un rectangulo alrededor del ojo
82     for (x,y,w,h) in ojo:
83         cv2.rectangle(im, (x, y), (x+w, y+h), (0,255,255),2) #Dibujar un rectangulo de color cyan en el contorno
84         ojo = sorted(ojo, key=lambda x: x[3])          #Clasifica los ojos encontrados
85
86     #Condicion para redimension las coordenadas de los ojos encontrados
87     if ojo:
88         ojo_i = ojo[0]
89         (x, y, w, h) = [v * size for v in ojo_i]
90
91     #Condicion para restringir el tamaño del ojo
92     if w > 200:
93         #
94         #         VozOjosDetectados = "Ojos del conductor detectados"
95         subprocess.call('echo '+VozOjosDetectados+'|festival --tts --language spanish', shell=True)
96         ojos = gris[y:y + h, x:x + w]                #Variable que almacena la imagen en escala de grises
97         ojos_resize = cv2.resize(ojos, (im_width, im_height)) #Redimensionar el tamaño del ojo
98
99     #Asigna un número para almacenar las imágenes con los resultados
100    pin= sorted([int(n[:n.find('.')]) for n in os.listdir(path)
               if n[0]!='.' ][0])[-1] + 1

```

```
File Edit Format Run Options Windows Help
101
102     #ALMACENA LAS IMAGENES DE LOS RESULTADOS EN LAS DIRECCIONES ESTABLECIDAS
103     cv2.imwrite('%s/%s.pgm' % (OjosPGM, pin), ojos_resize)
104     cv2.imwrite('%s/%s.jpg' % (OjosJPG, pin), ojos_resize)
105     cv2.imwrite('/home/pi/RecoleccionDatos/DiametroPupilas/Liceth/1.pgm', ojos_resize)
106
107
108     cv2.rectangle(im, (x, y), (x + w, y + h), (0, 255, 0), 3)      #Dibuja un rectangulo de color verde
109     # cv2.putText(im, fn_name, (x - 10, y - 10), cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))
110     # count += 1
111     # print count
112
113     time.sleep(0.5)      #Retardo de medio segundo
114
115     #LEER LA IMAGEN ALMACENADA DEL OJO
116     image = cv2.imread('/home/pi/RecoleccionDatos/DiametroPupilas/Liceth/1.pgm')
117     # output = ojos_resize.copy()
118     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)      #Convierte la imagen a escala de grises
119
120     #Aplicar filtro Gaussiano
121     gray = cv2.GaussianBlur(gray, (5,5),0)
122     gray = cv2.medianBlur(gray,5)
123     gray = cv2.adaptiveThreshold(gray,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,3)
124
125     #Crear un vector para la imagen
126     kernel = np.ones((2.6,2.7),np.uint8)
127     gray = cv2.erode(gray,kernel,iterations = 1)      #Aplicar tecnica de erosion a la imagen
128
129     gray = cv2.dilate(gray,kernel,iterations = 1)      #Aplicar tecnica de dilatacion a la imagen
130
131     #Detectar circulos
132     circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1, 200, param1=5, param2=10, minRadius=0, maxRadius=0)
```



```
File Edit Format Run Options Windows Help
133 # print circles
134
135 #CONDICION PARA COMPROBAR QUE SE HAN ENCONTRADOS CIRCULOS EN LA IMAGEN
136 if circles is not None:
137     #Convierte las coordenadas x,y y el radio en variable tipo entero
138     circlesdibujo = np.round(circles[0, :]).astype("int")
139     #Almacena los valores del circulo en un vector
140     circlesvalores = (circles[0, :])
141
142     #Ciclo para retornar el valor del radio
143     for (x1, y1, r1) in circlesvalores:
144 #         print x1
145 #         print y1
146         if r1 > 5 and r1 < 11:
147             #Mensaje para indicar que se ha detectado la pupila
148             VozPupilaDetectada = "Pupila del Conductor Detectada"
149             ##Emplea el sintetizador de texto a voz Festival para leer la instruccion de que se ha detectado la pupila
150             subprocess.call('echo '+VozPupilaDetectada+'|festival --tts --language spanish', shell=True)
151             guardar = 1 #Almacenar la imagen
152 #         print r1
153
154     #FUNCION QUE RETORNA EL VALOR DEL RADIO
155     def RadioOjo():
156         return r1
157
158     #CONDICION PARA DIBUJAR UN CIRCULO CON EN LAS COORDENAS (x,y) Y RADIO r ESTABLECIDOS
159     for (x, y, r) in circlesdibujo:
160         cv2.circle(image, (x, y), r, (0, 255, 0), 1)
161 #         cv2.rectangle(output, (x - 5, y - 5), (x + 5, y + 5), (0, 128, 255), -1)
162         #time.sleep(0.5)
163
```

num Ojos.py - /home/pi/RecoleccionDatos/DiametroPupilas/Ojos.py (3.4.2)

```
File Edit Format Run Options Windows Help
164         #ALMACENAR LAS IMAGENES CON LOS RESULTADOS OBTENIDOS
165         if guardar == 1:
166             cv2.imwrite('/home/pi/RecoleccionDatos/DiametroPupilas/Liceth/2.pgm', gray)
167             cv2.imwrite('/home/pi/RecoleccionDatos/DiametroPupilas/Liceth/3.pgm', image)
168             cv2.imwrite('/home/pi/RecoleccionDatos/DiametroPupilas/Liceth/4.jpg', gray)
169             cv2.imwrite('/home/pi/RecoleccionDatos/DiametroPupilas/Liceth/5.jpg', image)
170             cv2.imwrite('%s/%s.pgm' % (GrayPGM, pin), gray)
171             cv2.imwrite('%s/%s.jpg' % (GrayJPG, pin), gray)
172             cv2.imwrite('%s/%s.pgm' % (PupilaPGM, pin), image)
173             cv2.imwrite('%s/%s.jpg' % (PupilaJPG, pin), image)
174             #cv2.imshow("gray", gray)
175
176             cv2.imshow("output", image)      #Mostrar la imagen del ojo senalada la pupila
177
178             #FUNCION PARA RETORNAR EL VALOR DEL RADIO
179             def RadioOjo():
180                 return r1
181             count = 2          #Variable que indica que se ha completado el almacenamiento
182             numeroveces = 0    #Inicializa la variable para buscar nuevamente la pupila en la imagen del ojo
183             numeroveces = numeroveces + 1    #Incrementa la variable que indica que el ciclo se ha completado sin exito
184
185             #CONDICION QUE INDICA QUE SE HA BUSCADO 3 VECES LA PUPILA SIN RESULTADOS EXITOS Y EMITE UNA INSTRUCCION AUDITIVA
186             if numeroveces > 3:
187                 numeroveces = 0
188                 VozPupilaNoDetectada = "Por favor acerquese a la camara para medir el diametro de la pupila"
189                 subprocess.call('echo '+VozPupilaNoDetectada+'|festival --tts --language spanish', shell=True)
190
191             #VISUALIZACION DE LA PANTALLA
192             key = cv2.waitKey(1)          #Retardo de un segundo
193             rawCapture.truncate(0)        #Utilizar la salida para producir varias matrices
194
```

num *Ojos.py - /home/pi/RecoleccionDatos/DiametroPupilas/Ojos.py (3.4.2)*

File Edit Format Run Options Windows Help

```
195     ##Condicion para salir del ciclo si se presiona la tecla 'q'
196     if key == ord("q"):
197         camera.close()           #Cierra el puerto de la camara
198         cv2.destroyAllWindows()  #Cierra la ventana de visualizacion
199         break                    #Sale del ciclo
200
201     #Condicion para mostrar en la pantalla la imagen guardada de la pupila
202     if count == 2:
203         imagenguardada1 = cv2.imread('/home/pi/RecoleccionDatos/DiametroPupilas/Liceth/4.jpg') #Lee la imagen en la direccion establecida
204         cv2.imshow('DIAMETROPUPILA', imagenguardada1)
205         imagenguardada2 = cv2.imread('/home/pi/RecoleccionDatos/DiametroPupilas/Liceth/5.jpg') #Lee la imagen en la direccion establecida
206         cv2.imshow('DIAMETROPUPILA', imagenguardada2) #Muestra la imagen leida
207         #Emplea el sinterizador de texto a voz Festival para leer la instruccion de que se ha detectado la pupila
208         VozPupila = "Diametro de la pupila almacenado"
209         subprocess.call('echo '+VozPupila+'|festival --tts --language spanish', shell=True)
210         key = cv2.waitKey(1000) #Retardo para mostrar las imagenes
211         camera.close()         #Cierra el puerto de la camara
212         cv2.destroyAllWindows() #Cierra la ventana de visualizacion
213         break                  #Sale del ciclo
214
215
```

Ln: 215 Col: 0

Anexo 10. Programación script para colores de la temperatura facial

```
num colores.py - /home/pi/RecoleccionDatos/TemperaturaFacial/colores.py (3.4.2)
File Edit Format Run Options Windows Help
1 import time #Modulo empleado para trabajar con fechas y/o horas
2 import numpy as np #Modulo empleado para operar con vectores o matrices
3 import os, sys #Modulo empleado para utilizar la funcionalidad dependiente del sistema operativo
4
5 #FUNCION QUE DEVUELVE EL CODIGO RGB DE LOS COLORES SEGUN EL VALOR DE LA TEMPERATURA
6 def ColoresTemperatura(entero):
7     #Temperaturas con tonalidades en azul
8     n10 = (204,0,0)
9     n9 = (255,51,51)
10    n8 = (255,102,102)
11    n7 = (255,153,153)
12    n6 = (255,238,0)
13    n5 = (255,153,51)
14    #Temperaturas con tonalidades celestes
15    n4 = (255,178,102)
16    n3 = (255,204,153)
17    n2 = (255,255,0)
18    n1 = (255,255,51)
19    #Temperaturas con tonalidades cyan
20    cero = (255,255,102)
21    p1 = (255,255,153)
22    p2 = (204,255,153)
23    p3 = (178,255,102)
24    #Temperaturas con tonalidades Aguamarin
25    p4 = (153,255,51)
26    p5 = (128,255,0)
27    p6 = (153,255,153)
28    p7 = (102,204,0)
29    p8 = (0,204,0)
30    #Temperaturas con tonalidades verde
31    p9 = (0,204,102)
32    p10 = (0,214,0)
33    p11 = (0,224,0)
34    p12 = (0,232,0)
--
```

```
num colores.py - /home/pi/RecoleccionDatos/TemperaturaFacial/colores.py (3.4.2)
File Edit Format Run Options Windows Help
35     p13 = (0,244,0)
36     p14 = (0,255,0)
37     p15 = (51,255,51)
38     #Temperaturas con tonalidades verdeamarillo
39     p16 = (102,255,102)
40     p17 = (0,255,128)
41     p18 = (51,255,153)
42     p19 = (102,255,178)
43     #Temperaturas con tonalidades amarillas
44     p20 = (0,235,255)
45     p21 = (0,215,255)
46     p22 = (0,195,255)
47     p23 = (0,175,255)
48     #Temperaturas con tonalidades oro
49     p24 = (0,155,255)
50     p25 = (0,135,255)
51     #Temperaturas con tonalidades naranja
52     p26 = (0,128,255)
53     p27 = (0,120,255)
54     #Temperaturas con tonalidades tomate
55     p28 = (0,110,255)
56     p29 = (0,100,255)
57     #Temperaturas con tonalidades rojas
58     p30 = (0,90,255)
59     p31 = (0,80,255)
60     p32 = (0,70,255)
61     p33 = (0,60,255)
62     p34 = (0,50,255)
63     p35 = (0,40,255)
64     p36 = (0,35,255)
65     p37 = (0,25,255)
66     p38 = (0,15,255)
67     p39 = (0,5,255)
68     p40 = (0,0,255)
```

num colores.py - /home/pi/RecoleccionDatos/TemperaturaFacial/colores.py (3.4.2)

File Edit Format Run Options Windows Help

```
69
70     #CONDICIONES PARA ESTABLECER EL CODIGO RGB
71     if(entero == -10):
72         color = n10
73     if(entero == -9):
74         color = n9
75     if(entero == -8):
76         color = n8
77     if(entero == -7):
78         color = n7
79     if(entero == -6):
80         color = n6
81     if(entero == -5):
82         color = n5
83     if(entero == -4):
84         color = n4
85     if(entero == -3):
86         color = n3
87     if(entero == -2):
88         color = n2
89     if(entero == -1):
90         color = n1
91     if(entero == 0):
92         color = cero
93     if(entero == 1):
94         color = p1
95     if(entero == 2):
96         color = p2
97     if(entero == 3):
98         color = p3
99     if(entero == 4):
100         color = p4
101     if(entero == 5):
102         color = p5
```

```
num colores.py - /home/pi/RecoleccionDatos/TemperaturaFacial/colores.py (3.4.2)
File Edit Format Run Options Windows Help
103     if(entero == 6):
104         color = p6
105     if(entero == 7):
106         color = p7
107     if(entero == 8):
108         color = p8
109     if(entero == 9):
110         color = p9
111     if(entero == 10):
112         color = p10
113     if(entero == 11):
114         color = p11
115     if(entero == 12):
116         color = p12
117     if(entero == 13):
118         color = p13
119     if(entero == 14):
120         color = p14
121     if(entero == 15):
122         color = p15
123     if(entero == 16):
124         color = p16
125     if(entero == 17):
126         color = p17
127     if(entero == 18):
128         color = p18
129     if(entero == 19):
130         color = p19
131     if(entero == 20):
132         color = p20
133     if(entero == 21):
134         color = p21
135     if(entero == 22):
136         color = p22
```

```
num colores.py - /home/pi/RecoleccionDatos/TemperaturaFacial/colores.py (3.4.2)
File Edit Format Run Options Windows Help
137     if(entero == 23):
138         color = p23
139     if(entero == 24):
140         color = p24
141     if(entero == 25):
142         color = p25
143     if(entero == 26):
144         color = p26
145     if(entero == 27):
146         color = p27
147     if(entero == 28):
148         color = p28
149     if(entero == 29):
150         color = p29
151     if(entero == 30):
152         color = p30
153     if(entero == 31):
154         color = p31
155     if(entero == 32):
156         color = p32
157     if(entero == 33):
158         color = p33
159     if(entero == 34):
160         color = p34
161     if(entero == 35):
162         color = p35
163     if(entero == 36):
164         color = p36
165     if(entero == 37):
166         color = p37
167     if(entero == 38):
168         color = p38
169     if(entero == 39):
170         color = p39
```



```
num colores.py - /home/pi/RecoleccionDatos/TemperaturaFacial/colores.py (3.4.2)
File Edit Format Run Options Windows Help
169     if(entero == 39):
170         color = p39
171     if(entero >= 40):
172         color = p40
173
174     return color    #Retornar el valor del color
175
---
```

Ln: 175 Col: 0

Anexo 11. Programación script para posicionesMatriz de la temperatura facial

```
num posicionesMatriz.py - /home/pi/RecoleccionDatos/TemperaturaFacial/posicionesMatriz.py (3.4.2)
File Edit Format Run Options Windows Help
1 import time
2 import numpy as np
3
4 def PosicionX(x, fila1):
5     if(x>=0 and x < (fila1-20)):
6         filatemp = 30
7     if(x>=(fila1 - 20) and x<(fila1 + 35)):
8         filatemp = 0
9     if(x>=(fila1 + 35) and x<(fila1 + 70)):
10        filatemp = 1
11    if(x>=(fila1 + 70) and x<(fila1 + 105)):
12        filatemp = 2
13    if(x>=(fila1 + 105) and x<(fila1 + 140)):
14        filatemp = 3
15    if(x>=(fila1 + 140) and x<(fila1 + 175)):
16        filatemp = 4
17    if(x>=(fila1 + 175) and x<(fila1 + 210)):
18        filatemp = 5
19    if(x>=(fila1 + 210) and x<(fila1 + 245)):
20        filatemp = 6
21    if(x>=(fila1 + 245) and x<(fila1 + 280)):
22        filatemp = 7
23    if(x>=(fila1 + 280) and x<(fila1 + 315)):
24        filatemp = 8
25    if(x>=(fila1 + 315) and x<(fila1 + 350)):
26        filatemp = 9
27    if(x>=(fila1 + 350) and x<(fila1 + 385)):
28        filatemp = 10
29    if(x>=(fila1 + 385) and x<(fila1 + 420)):
30        filatemp = 11
31    if(x>=(fila1 + 420) and x<(fila1 + 455)):
32        filatemp = 12
33    if(x>=(fila1 + 455) and x<(fila1 + 490)):
34        filatemp = 13
--
```

```

File Edit Format Run Options Windows Help
35     if(x>=(fila1 + 490) and x<(fila1 + 525)):
36         filatemp = 14
37     if(x>=(fila1 + 525) and x<(fila1 + 580)):
38         filatemp = 15
39     if(x>=(fila1 + 580) and x < 640):
40         filatemp = 30
41     return filatemp
42
43 def PosicionFilaFin(filaFin, fila1):
44     if(filaFin>=0 and filaFin < (fila1-20)):
45         filatemp1 = 30
46     if(filaFin>=(fila1 - 20) and filaFin<(fila1 + 35)):
47         filatemp1 = 0
48     if(filaFin>=(fila1 + 35) and filaFin<(fila1 + 70)):
49         filatemp1 = 1
50     if(filaFin>=(fila1 + 70) and filaFin<(fila1 + 105)):
51         filatemp1 = 2
52     if(filaFin>=(fila1 + 105) and filaFin<(fila1 + 140)):
53         filatemp1 = 3
54     if(filaFin>=(fila1 + 140) and filaFin<(fila1 + 175)):
55         filatemp1 = 4
56     if(filaFin>=(fila1 + 175) and filaFin<(fila1 + 210)):
57         filatemp1 = 5
58     if(filaFin>=(fila1 + 210) and filaFin<(fila1 + 245)):
59         filatemp1 = 6
60     if(filaFin>=(fila1 + 245) and filaFin<(fila1 + 280)):
61         filatemp1 = 7
62     if(filaFin>=(fila1 + 280) and filaFin<(fila1 + 315)):
63         filatemp1 = 8
64     if(filaFin>=(fila1 + 315) and filaFin<(fila1 + 350)):
65         filatemp1 = 9
66     if(filaFin>=(fila1 + 350) and filaFin<(fila1 + 385)):
67         filatemp1 = 10

```

```

File Edit Format Run Options Windows Help
67         filatemp1 = 10
68         if(filaFin>=(fila1 + 385) and filaFin<(fila1 + 420)):
69             filatemp1 = 11
70         if(filaFin>=(fila1 + 420) and filaFin<(fila1 + 455)):
71             filatemp1 = 12
72         if(filaFin>=(fila1 + 455) and filaFin<(fila1 + 490)):
73             filatemp1 = 13
74         if(filaFin>=(fila1 + 490) and filaFin<(fila1 + 525)):
75             filatemp1 = 14
76         if(filaFin>=(fila1 + 525) and filaFin<(fila1 + 580)):
77             filatemp1 = 15
78         if(filaFin>=(fila1 + 580) and filaFin<640):
79             filatemp1 = 30
80         return filatemp1
81
82     def PosicionY(y,columna1):
83
84         if(y>=0 and y<(columna1-30)):
85             columnatemp = 30
86         if(y>=(columna1 - 30) and y<(columna1 + 40)):
87             columnatemp = 0
88         if(y>=(columna1 + 40) and y<(columna1 + 80)):
89             columnatemp = 1
90         if(y>=(columna1 + 80) and y<(columna1 + 120)):
91             columnatemp = 2
92         if(y>=(columna1 + 120) and y<(columna1 + 210)):
93             columnatemp = 3
94         if(y>=(columna1 + 210) and y<480):
95             columnatemp = 30
96
97         return columnatemp
98
99     def PosicionColumnaFin(columnaFin,columna1):

```

num posicionesMatriz.py - /home/pi/RecoleccionDatos/TemperaturaFacial/posicionesMatriz.py (3.4.2)

```
File Edit Format Run Options Windows Help
99 def PosicionColumnaFin(columnaFin,columna1):
100     if(columnaFin>=0 and columnaFin<(columna1 - 30)):
101         columnatemp = 30
102     if(columnaFin>=(columna1 - 30) and columnaFin<(columna1 + 40)):
103         columnatemp1 = 0
104     if(columnaFin>=(columna1 + 40) and columnaFin<(columna1 + 80)):
105         columnatemp1 = 1
106     if(columnaFin>=(columna1 + 80) and columnaFin<(columna1 + 120)):
107         columnatemp1 = 2
108     if(columnaFin>=(columna1 + 120) and columnaFin<(columna1 + 210)):
109         columnatemp1 = 3
110     if(columnaFin>=(columna1 + 210) and columnaFin<480):
111         columnatemp1 = 3
112
113     return columnatemp1
114
```

Ln: 114 Col: 0

Anexo 12. Datos de Entrenamiento

<datosEntrenamientoPruebas.csv>							
Archivo Editar Buscar Opciones Ayuda							
Nro	Fecha	Hora	Alcohol	TempAmbiente	TempFacialMax	TempFacialMin	RadioPupila,Estado
1	13/05/17	20:48:23	1.06	27.94	27.23	11.749999999999999	3.53553,0
2	13/05/17	20:52:38	1.11	27.83	27.26	12.620000000000001	3.53553,0
3	13/05/17	20:55:12	1.08	27.94	27.49	12.09	3.53553,0
4	13/05/17	20:59:33	1.13	28.17	26.699999999999999	13.01	3.53553,0
5	13/05/17	21:03:23	1.14	28.11	27.15	13.26	3.53553,0
6	13/05/17	21:09:32	1.01	27.42	27.340000000000002	11.54	4.30116,0
7	13/05/17	21:13:42	1.09	28.16	26.81	14.559999999999999	4.30116,0
8	13/05/17	21:16:32	1.15	27.68	28.099999999999999	11.49	4.301196,0
9	13/05/17	21:10:43	1.12	27.69	27.34	12.19	4.301116,0
10	13/05/17	21:23:53	1.59	28.70	28.850000000000002	15.779999999999999	8.7005,1
11	13/05/17	21:26:53	1.62	28.89	28.94	12.360000000000001	8.7005,1
12	13/05/17	21:29:12	1.39	28.65	28.689999999999999	17.35	8.005,1
13	13/05/17	21:31:24	1.45	28.96	28.32	11.940000000000001	9.92472,1
14	13/05/17	21:35:42	1.82	28.56	28.07	15.84	9.92472,1
15	13/05/17	21:37:21	2.15	28.09	28.85	15.159999999999999	9.92472,1
16	13/05/17	21:42:38	1.03	28.42	27.10	16.270000000000002	3.80789,0
17	13/05/17	21:44:56	1.09	27.89	27.26	16.2	3.80789,0
18	13/05/17	21:48:38	1.11	27.99	27.739999999999999	16.099999999999999	3.80789,0
19	13/05/17	21:55:39	1.07	28.27	27.1	16.25	3.53553,0
20	13/05/17	22:02:23	1.09	28.74	27.8	13.26	2.91548,0
21	13/05/17	22:06:31	1.10	28.30	27.17	12.66	3.80789,0
22	13/05/17	22:09:13	0.91	28.69	27.43	15.689999999999999	2.91548,0
23	13/05/17	22:11:15	0.93	28.27	27.8	13.48	4.30116,0
24	13/05/17	22:15:28	0.95	28.21	27.28	12.99	2.91548,0
25	13/05/17	22:18:25	1.04	28.42	27.319999999999999	16.24	3.80789,0
26	13/05/17	22:21:49	1.56	28.39	28.16	13.83	8.63134,0
27	13/05/17	22:24:28	1.64	28.59	28.319999999999999	16.080000000000001	8.74643,1
28	13/05/17	22:26:58	1.74	28.45	28.64	17.14	8.74643,1
29	13/05/17	22:29:42	1.60	29.61	28.65	12.25	9.19239,1
30	13/05/17	22:31:12	1.64	29.17	28.99	16.259999999999999	10.5119,1

<datosEntrenamientoPruebas.csv>							
Archivo Editar Buscar Opciones Ayuda							
31	13/05/17	22:33:05	1.76	28.76	28.839999999999999	18.26	100000000000002.5119,1
32	13/05/17	22:37:12	1.12	27.96	30.819999999999999	12.25	5.80789,0
33	13/05/17	22:39:18	1.09	27.11	25.8	12.36	5.80789,0
34	13/05/17	22:42:42	1.10	27.18	26.93	12.25	5.9045,0
35	13/05/17	22:44:21	0.96	27.32	27.82	12.899999999999999	5.2444,0
36	13/05/17	22:49:42	1.02	27.39	26.359999999999999	17.25	6.3656,0
37	13/05/17	22:52:34	1.11	27.78	27.69	14.269999999999999	7.64853,0
38	13/05/17	22:55:41	0.96	27.51	27.07	14.25	7.63134,0
39	13/05/17	22:59:51	1.05	26.92	28.58	16.24	8.92472,1
40	13/05/17	23:06:02	1.58	26.84	27.83	13.859999999999999	9.19239,1
41	13/05/17	23:09:43	1.79	26.82	28.570000000000001	14.56	8.82344,1
42	13/05/17	23:11:57	1.80	26.98	28.799999999999999	14.26	9.19239,1
43	13/05/17	23:14:21	1.64	26.17	28.410000000000002	18.25	10.5119,1
44	13/05/17	23:17:22	1.86	26.89	27.889999999999999	14.36	9.82344,1
45	13/05/17	23:20:40	1.98	26.52	27.819999999999999	15.23	9.61769,1
46	13/05/17	23:24:21	1.04	27.67	27.12	11.229999999999999	5.5192,0
47	13/05/17	23:28:17	1.12	27.79	26.69	15.240000000000001	5.70088,0
48	13/05/17	23:33:42	0.96	27.74	26.60	15.04	6.14782,0
49	13/05/17	23:35:22	1.06	27.80	26.98	17.01	6.14782,0
50	13/05/17	23:38:39	1.07	27.97	26.259999999999999	16.25	5.52278,0
51	13/05/17	23:41:12	1.54	28.78	27.129999999999999	18.26	8.27647,1
52	13/05/17	23:44:24	1.67	28.96	28.399999999999999	19.26	8.27647,1
53	13/05/17	23:46:37	1.90	28.80	28.690000000000001	17.259999999999999	9.82344,1
54	13/05/17	23:48:47	1.64	28.89	28.32	16.28	9.82344,1
55	13/05/17	23:52:12	2.09	28.05	28.680000000000002	16.28	9.1242,1
56	13/05/17	23:54:12	1.01	28.12	26.79	12.360000000000001	6.5192,0
57	13/05/17	23:57:24	0.91	27.96	26.49	17.28	5.70088,0
58	14/05/17	00:07:01	1.05	27.96	26.98	12.369999999999999	6.04152,0
59	14/05/17	00:11:24	0.96	27.59	26.27	16.25	6.04152,0
60	14/05/17	00:16:53	1.10	27.69	26.599999999999999	18.14	5.52268,0
61	14/05/17	00:21:37	1.04	27.87	26.289999999999999	16.259999999999999	5.52268,0

Archivo Editar Buscar Opciones Ayuda

```

62,14/05/17,00:25:27,0,26.81,26.4,14.26,5.14782,0
63,14/05/17,00:32:10,0,26.78,26.749999999999999,14.25,6.67083,0
64,14/05/17,00:36:23,0,26.96,26.94,15.36,5.5192,0
65,14/05/17,00:40:55,0,27.75,27.39,17.819999999999999,5.5192,0
66,14/05/17,00:42:19,0,27.61,26.140000000000002,16.99,5.7008,0
67,14/05/17,00:47:21,0,27.12,26.77,17.850000000000001,5.7008,0
68,14/05/17,00:49:27,0,27.89,27.25,16.24,5.52268,0
69,14/05/17,00:14:46,0,27.93,27.39,16.459999999999999,6.5192,0
70,14/05/17,00:55:43,0,27.96,26.31,15.750000000000002,6.5192,0
71,14/05/17,01:02:27,1.56,28.66,26.929999999999999,16.239999999999999,8.74643,1
72,14/05/17,01:08:23,2.02,28.48,26.48,17.98,9.82344,1
73,14/05/17,01:13:13,1.93,28.87,28.149999999999999,15.68,9.19239,1
74,14/05/17,01:19:24,1.99,28.89,29.24,19.24,100000000000001.1242,1
75,14/05/17,01:25:58,2.74,28.88,29.37,18.26,10.5119,1
76,14/05/17,01:32:42,0,28.09,26.34,1700000000000001.25,5.7008,0
77,14/05/17,01:39:15,0,28.16,26.48,19.289999999999999,5.7008,0
78,14/05/17,01:43:26,0,28.17,26.94,17.28,5.5192,0
79,14/05/17,01:48:33,0,28.12,26.379999999999999,16.98,5.5192,0
80,14/05/17,01:54:53,0,28.19,25.35,16.259999999999999,5.5192,0
81,09/09/17,07:50:58,0,27.73,28.109999999999999,17.84,7.64853,0
82,09/09/17,08:02:38,0,28.73,28.57,16.550000000000001,5.70088,0
83,09/09/17,08:04:42,0,28.94,20.34,17.879999999999999,5.70088,0
84,09/09/17,08:05:31,0,28.97,28.98,17.960000000000001,5.14782,0
85,09/09/17,08:12:12,0,29.41,27.780000000000001,17.82,7.77817,0
86,09/09/17,08:13:04,0,29.42,28.23,18.09,5.70088,0
87,09/09/17,08:15:18,0,29.16,28.77,18.510000000000002,7.51665,0
88,09/09/17,09:11:34,0,28.58,29.010000000000002,19.0,6.67083,0
89,09/09/17,09:13:41,0,28.65,27.629999999999999,17.579999999999998,7.10634,0
90,09/09/17,09:15:01,0,28.80,28.100000000000001,17.629999999999999,6.67083,0
91,09/09/17,09:15:54,0,28.89,27.77,17.57,7.10634,0
92,09/09/17,09:17:53,0,28.85,28.260000000000002,17.800000000000001,7.64853,0

```

<datosEntrenamientoPruebas.csv>

Archivo Editar Buscar Opciones Ayuda

```

93,09/09/17,10:11:49,1.32,28.46,28.170000000000002,14.73,5.52268,0
94,09/09/17,10:13:17,1.32,28.55,26.719999999999999,17.5,6.5192,0
95,09/09/17,19:52:01,1.15,28.00,26.81,13.41,5.70088,0
96,09/09/17,19:53:45,1.23,28.49,24.23,15.91,6.36396,0
97,09/09/17,19:56:06,1.19,29.43,28.509999999999999,8.509999999999998,7.64853,0
98,09/09/17,19:57:35,1.17,29.51,24.17,8.419999999999999,7.61769,0
99,09/09/17,20:27:51,1.15,28.07,26.550000000000001,15.17,5.70088,0
100,09/09/17,20:29:49,1.19,28.64,28.350000000000001,16.140000000000001,7.51665,0
101,09/09/17,20:30:52,1.16,28.80,24.06,12.970000000000001,6.5192,0
102,09/09/17,20:31:45,0.91,28.60,22.880000000000001,12.619999999999999,3.53553,0
103,09/09/17,20:33:16,0.93,28.27,23.24,11.609999999999999,6.04152,0
104,09/09/17,20:34:08,0.95,28.21,26.760000000000002,16.670000000000002,8.63134,0
105,09/09/17,20:35:26,1.24,28.32,27.329999999999998,16.530000000000001,6.04152,0
106,09/09/17,20:36:52,1.01,28.38,27.289999999999999,16.0,9.82344,0
107,09/09/17,20:38:08,1.18,28.39,27.870000000000001,16.43,9.82344,0
108,09/09/17,20:39:28,1.14,28.44,27.41,16.969999999999999,7.5192,0
109,09/09/17,20:44:23,0.99,29.51,26.510000000000002,18.920000000000002,7.52268,0
110,09/09/17,20:46:06,1.01,29.10,27.550000000000001,18.829999999999998,8.63134,0
111,09/09/17,20:54:45,1.97,27.86,27.84,16.690000000000001,7.90569,1
112,09/09/17,20:56:05,1.53,27.97,28.050000000000001,16.82,8.7005,1
113,09/09/17,20:57:11,1.58,28.01,28.34,19.579999999999998,9.61769,1
114,09/09/17,20:57:58,1.50,28.14,29.140000000000001,17.350000000000001,5.52268,1
115,09/09/17,20:58:52,1.58,28.22,26.969999999999999,17.129999999999999,7.64853,1
116,09/09/17,20:59:49,1.77,28.32,29.109999999999999,17.739999999999998,9.19239,1
117,09/09/17,21:02:11,1.53,28.28,27.73,17.129999999999999,9.19239,1
118,09/09/17,21:27:16,1.26,27.58,25.710000000000001,16.510000000000002,8.1242,0
119,09/09/17,21:28:29,1.20,27.72,26.780000000000001,16.620000000000001,8.51469,0
120,09/09/17,21:29:21,1.12,27.84,28.449999999999999,17.600000000000001,7.64853,0
121,09/09/17,21:30:25,1.12,27.92,28.050000000000001,18.32,10.5119,0
122,09/09/17,21:31:35,1.19,27.92,28.699999999999999,20.010000000000002,9.1242,0
123,09/09/17,21:54:08,1.76,27.13,28.359999999999999,14.98,10.7005,1

```

<datosEntrenamientoPruebas.csv>

Archivo Editar Buscar Opciones Ayuda

124,09/09/17,21:55:11,1.53,27.25,30.579999999999998,19.870000000000001,9.92472,1
125,09/09/17,21:56:25,1.49,27.55,28.420000000000002,16.199999999999999,10.5119,1
126,09/09/17,21:57:33,1.64,27.63,28.800000000000001,17.600000000000001,10.7935,1
127,09/09/17,21:58:33,1.52,27.74,28.18,16.949999999999999,10.1242,1
128,09/09/17,22:00:54,1.78,27.84,28.579999999999998,16.93,8.67083,1
129,09/09/17,22:02:29,1.56,27.90,28.059999999999999,17.93,8.63134,1
130,09/09/17,22:14:43,1.57,27.67,28.350000000000001,17.129999999999999,10.1242,1
131,09/09/17,22:16:01,1.81,27.78,29.5,17.219999999999999,11.3358,1
132,09/09/17,22:17:49,1.88,27.86,29.050000000000001,14.279999999999999,10.9772,1
133,09/09/17,22:19:03,1.90,27.90,29.300000000000001,17.100000000000001,8.86001,1
134,09/09/17,22:20:45,1.64,27.78,28.690000000000001,18.120000000000001,13.2853,1
135,09/09/17,22:23:19,2.09,28.11,29.710000000000001,16.100000000000001,13.5831,1
136,09/09/17,22:26:20,1.51,28.21,26.91,25.969999999999999,11.3358,1
137,09/09/17,22:41:27,2.11,27.80,27.030000000000001,15.609999999999999,10.7935,1
138,09/09/17,22:42:35,1.60,27.60,28.129999999999999,16.190000000000001,8.63134,1
139,09/09/17,22:43:31,1.59,27.49,28.780000000000001,16.620000000000001,11.068,1
140,09/09/17,22:44:42,1.66,27.19,25.68,16.399999999999999,9.92472,1
141,09/09/17,22:45:57,1.99,27.17,29.609999999999999,16.440000000000001,10.1242,1
142,09/09/17,22:46:57,1.50,26.83,26.309999999999999,16.079999999999998,9.30054,1
143,09/09/17,22:48:04,1.73,26.51,26.120000000000001,14.81,9.19239,1
144,09/09/17,22:49:56,1.69,26.20,27.920000000000002,16.789999999999999,9.19239,1
145,09/09/17,23:09:39,1.89,27.35,29.25,16.91,9.82344,1
146,09/09/17,23:10:45,1.71,27.51,29.050000000000001,27.780000000000001,11.068,1
147,09/09/17,23:11:45,1.91,27.62,26.0,16.879999999999999,11.068,1
148,09/09/17,23:13:52,2.54,27.69,29.219999999999999,18.57,10.6066,1
149,09/09/17,23:14:53,2.20,27.93,27.489999999999998,17.010000000000002,9.19239,1
150,09/09/17,23:15:47,1.79,28.06,28.789999999999999,16.52,11.3358,1
151,09/09/17,23:17:02,2.06,28.26,27.199999999999999,16.140000000000001,12.5897,1
152,09/09/17,23:18:10,2.02,28.45,27.370000000000001,15.83,12.5897,1
153,09/09/17,23:49:36,1.93,27.84,29.010000000000002,19.07,8.74643,1
154,09/09/17,23:50:30,1.99,27.86,26.640000000000001,16.57,12.3491,1

<datosEntrenamientoPruebas.csv>

Archivo Editar Buscar Opciones Ayuda

155,09/09/17,23:51:56,2.74,27.86,29.579999999999998,14.92,5.14789,1
156,09/09/17,23:53:02,2.73,28.05,27.370000000000001,17.109999999999999,9.82344,1
157,09/09/17,23:54:14,2.81,28.17,29.199999999999999,15.800000000000001,9.61769,1
158,09/09/17,23:56:12,2.62,28.13,27.359999999999999,16.98,9.61769,1
159,10/09/17,00:06:47,1.82,28.11,29.07,15.41,9.36396,1
160,10/09/17,00:07:41,1.91,28.14,26.739999999999998,15.17,9.61769,1
161,10/09/17,00:12:12,2.15,27.94,28.379999999999999,25.23,9.61769,1
162,10/09/17,00:13:36,2.45,28.04,27.34,15.1,9.27647,1
163,10/09/17,00:15:16,2.76,28.21,27.690000000000001,15.49,10.7935,1
164,10/09/17,00:17:19,1.93,28.48,25.82,20.960000000000001,10.5119,1
165,10/09/17,00:18:17,1.93,28.38,26.300000000000001,15.890000000000001,8.63134,1
166,10/09/17,00:19:33,1.84,28.38,26.68,20.920000000000002,8.74643,1
167,10/09/17,00:20:49,2.46,28.45,29.469999999999999,17.170000000000002,9.52268,1
168,10/09/17,00:21:40,1.95,28.34,28.309999999999999,16.699999999999999,9.82344,1
169,10/09/17,00:44:12,3.2,27.49,24.969999999999999,14.23,10.9772,1
170,10/09/17,00:45:38,3.2,27.64,27.539999999999999,16.690000000000001,12.51,1
171,10/09/17,00:47:27,4.2,27.84,24.809999999999999,15.0,11.068,1
172,10/09/17,00:50:09,3.2,28.01,27.34,14.890000000000001,9.82344,1
173,10/09/17,00:52:16,4.05,28.10,27.949999999999999,15.550000000000001,8.63134,1
174,10/09/17,00:54:36,3.13,28.19,29.41,17.460000000000001,9.5192,1
175,10/09/17,00:55:52,3.33,28.30,29.359999999999999,16.59,8.64853,1
176,10/09/17,00:58:05,4.42,28.26,27.899999999999999,16.530000000000001,12.51,1
177,10/09/17,01:00:25,4.28,28.26,25.809999999999999,14.9,8.64853,1
178,10/09/17,01:09:10,3.49,28.30,27.510000000000002,17.129999999999999,12.5897,1
179,10/09/17,01:11:25,3.58,28.40,25.440000000000001,17.210000000000001,8.63134,1
180,10/09/17,01:12:47,3.12,28.31,28.41,15.699999999999999,10.5119,1
181,10/09/17,01:13:58,4.55,28.18,27.949999999999999,15.08,8.90569,1
182,10/09/17,01:15:03,2.79,27.91,27.789999999999999,15.08,12.51,1
183,10/09/17,01:36:45,5.09,25.72,29.73,13.94,10.9772,1
184,10/09/17,01:38:05,2.54,25.76,26.52,15.41,11.068,1
185,10/09/17,01:39:09,3.15,25.83,26.359999999999999,24.52,10.38241,1


```

<datosEntrenamientoPruebas.csv>
Archivo Editar Buscar Opciones Ayuda
186,10/09/17,01:40:51,4.5599999999999996,25.81,17.620000000000001,15.0,13.5831,1
187,10/09/17,01:42:34,3.9699999999999998,29.14,23.949999999999999,14.789999999999999
188,10/09/17,01:44:09,4.26,26.43,27.379999999999999,15.630000000000001,11.068,1
189,10/09/17,01:50:22,5.66,27.18,29.84,16.329999999999998,9.82567,1
190,10/09/17,01:56:56,4.41,27.62,29.52,15.039999999999999,10.5119,1
191,10/09/17,01:58:28,3.76,27.74,29.059999999999999,14.99,10.1242,1
192,10/09/17,02:00:01,5.44,27.89,27.350000000000001,14.65,12.3491,1
193,10/09/17,02:06:10,4.15,27.90,26.399999999999999,15.029999999999999,8.90569,1
194,10/09/17,02:07:34,4.91,27.83,29.559999999999999,14.65,13.5831,1
195,10/09/17,02:09:01,3.96,27.85,27.73,21.309999999999999,9.51315,1
196,10/09/17,02:10:01,2.19,27.98,27.390000000000001,17.0,11.36396,1
197,10/09/17,02:11:21,3.67,27.96,29.32,14.35,10.5119,1
198,10/09/17,02:12:48,4.0,27.51,26.32,14.69,10.1242,1
199,10/09/17,02:17:21,4.08,27.58,26.530000000000001,16.41,9.82344,1
200,10/09/17,02:22:35,3.42,27.77,27.260000000000002,17.780000000000001,9.92472,1
201,10/09/17,02:25:38,3.9400000000000004,27.72,25.27,12.869999999999999,9.51315,1
202,10/09/17,02:27:21,1.7999999999999998,27.84,26.239999999999998,15.32,12.5897,1
203,10/09/17,02:28:51,1.68,27.93,26.719999999999999,16.390000000000001,8.63134,1
204,10/09/17,02:30:32,3.58,27.39,28.170000000000002,12.539999999999999,10.5119,1
205,10/09/17,01:50:48,2.76,27.14,24.479999999999999,14.46,6.5192,0
206,23/09/17,13:50:48,2.76,26.89,26.84,14.71,6.5192,0
207,06/10/17,15:41:28,2.76,27.09,25.919999999999999,15.919999999999999,6.5192,0
208,06/10/17,15:42:58,2.76,26.29,27.38,18.11,7.90569,0
209,06/10/17,15:43:16,1.12,26.98,27.22,11.649999999999999,6.5192,0
210,06/10/17,15:44:34,1.02,27.12,27.429999999999999,11.08,4.80789,0
211,06/10/17,15:44:12,0.99,25.87,27.26,11.81,4.53553,0
212,06/10/17,15:45:35,1.03,26.42,27.49,18.17,499999999999999.74342,0
213,06/10/17,15:45:24,1.04,27.07,27.739999999999999,11.2,5.04152,0
214,06/10/17,15:46:28,1.17,26.47,26.380000000000001,13.03,5.04152,0
215,07/10/17,22:48:23,1.10,26.43,27.35,16.55,5.53553,0
216,07/10/17,22:53:19,0.97,24.79,25.34,13.500000000000001,5.53553,0

```

```

<datosEntrenamientoPruebas.csv>
Archivo Editar Buscar Opciones Ayuda
217,07/10/17,22:57:12,0.99,25.89,26.67,21.039999999999999,7.90569,0
218,07/10/17,23:32:33,0.98,27.04,25.520000000000001,17.19,8.63134,0
219,07/10/17,23:34:23,1.03,25.9,27.19,11.579999999999999,8.63134,0
220,15/10/17,11:09:12,1.08,26.29,27.570000000000001,15.49,7.64853,0
221,15/10/17,11:11:23,1.04,25.78,27.689999999999999,10.28,8.51469,0
222,15/10/17,11:39:34,1.12,24.98,27.5,14.21,8.27647,0
223,15/10/17,11:40:54,1.11,25.86,26.08,10.890000000000001,8.2853,0
224,15/10/17,11:46:01,1.11,26.38,24.219999999999999,12.399999999999999,7.90569,0
225,15/10/17,11:46:59,0.98,24.85,26.54,13.5,8.51469,0
226,15/10/17,11:49:12,0.95,24.94,27.360000000000001,16.23,8.51469,0
227,15/10/17,12:52:11,0.93,24.84,25.949999999999999,11.94,8.2853,0
228,15/10/17,13:00:24,0.94,25.98,27.479999999999999,11.860000000000001,7.6485,0
229,15/10/17,13:05:25,0.95,26.4,26.789999999999999,11.42,10.1242,0
230,15/10/17,13:15:46,0.96,25.65,26.83,13.23,799999999999999.90569,0
231,15/10/17,13:26:45,0.97,26.45,26.579999999999999,11.18,8.82344,0
232,15/10/17,23:15:27,0.98,27.01,26.740000000000001,15.6,8.82344,0
233,16/10/17,18:15:32,0.96,25.86,24.679999999999999,13.84,6.67083,0
234,16/10/17,18:25:12,1.11,26.39,26.999999999999999,11.920000000000002,8.74643,0
235,16/10/17,19:16:02,1.00,26.47,26.79,12.1,8.63134,0
236,16/10/17,19:27:34,1.03,24.92,26.300000000000001,14.139999999999999,8.61769,0
237,16/10/17,19:35:53,1.11,25.98,26.799999999999999,16.48,8.82344,0
238,16/10/17,19:35:21,0.97,24.78,27.899999999999999,18.2,5.04152,0
239,16/10/17,19:37:41,0.99,26.89,27.68,20.150000000000001,7.5119,0
240,16/10/17,19:38:49,1.12,25.82,27.4,18.99,10.5119,0
241,16/10/17,19:38:23,1.00,24.92,27.320000000000001,17.92,10.7935,0
242,16/10/17,19:41:37,1.02,24.88,27.759999999999999,19.02,10.7935,0
243,16/10/17,19:42:34,1.10,24.99,27.399999999999999,16.680000000000001,8.82344,0
244,16/10/17,19:44:25,0.96,24.89,27.179999999999999,16.83,8.82344,0
245,16/10/17,20:28:12,0.96,26.31,27.98,15.82,3.53553,0
246,16/10/17,20:29:54,0.97,26.27,27.259999999999999,16.369999999999999,5.04152,0
247,16/10/17,20:30:12,0.98,24.79,26.149999999999999,17.329999999999999,8.63134,0

```

```

<datosEntrenamientoPruebas.csv>
Archivo Editar Buscar Opciones Ayuda
248,16/10/17,20:31:34,0.97,24.98,26.280000000000001,14.82,8.61769,0
249,16/10/17,20:34:23,0.97,24.78,26.719999999999999,13.02,5.70088,0
250,16/10/17,20:38:34,1.10,25.92,27.860000000000001,17.18,5.04152,0
251,16/10/17,20:46:56,1.09,25.85,27.129999999999999,16.16,5.04152,0
252,16/10/17,20:47:32,1.04,25.82,26.18,16.759999999999999,6.5192,0
253,16/10/17,20:48:12,1.11,26.51,27.249999999999999,19.68,6.67083,0
254,16/10/17,20:52:11,1.12,26.32,27.26,18.720000000000001,6.5192,0
255,16/10/17,20:54:24,0.99,26.34,27.29,17.939999999999999,6.5192,0
256,16/10/17,20:56:28,0.98,25.94,27.379999999999999,16.170000000000001,6.04152,0
257,16/10/17,20:57:27,0.95,25.82,26.78,17.19,6.5192,0
258,16/10/17,20:59:01,1.10,25.87,25.949999999999999,17.21,5.70088,0
259,16/10/17,21:11:38,0.97,24.99,25.839999999999999,18.33,5.52268,0
260,16/10/17,21:16:27,1.03,24.89,24.91,20.210000000000002,8.51469,0
261,16/10/17,21:19:21,1.02,26.23,26.479999999999999,22.19,8.86002,0
262,16/10/17,21:21:12,1.11,24.79,27.170000000000001,17.15,5.70088,0
263,16/10/17,21:24:15,0.98,25.86,26.809999999999999,11.08,6.36396,0
264,16/10/17,21:27:17,0.97,25.95,25.99,17.17,5.70088,0
265,16/10/17,21:28:25,0.96,25.92,26.179999999999999,13.19,5.70088,0
266,16/10/17,21:29:42,0.97,26.45,27.649999999999999,16.219999999999999,6.5192,0
267,16/10/17,21:32:31,1.12,26.37,27.17,18.43,6.5192,0
268,16/10/17,21:34:24,1.07,26.22,26.19,17.520000000000001,8.51469,0
269,16/10/17,21:38:12,1.10,25.71,27.17,14.639999999999999,8.86002,0
270,16/10/17,21:42:22,1.11,25.83,28.209999999999999,14.139999999999999,8.86002,0
271,16/10/17,21:48:45,0.91,26.29,25.88,15.479999999999999,9.5109,0
272,13/11/17,16:06:32,1.11,27.2,24.79,17.870000000000002,8.30054,0
273,13/11/17,16:17:22,0.97,26.45,27.190000000000001,19.18,8.51469,0
274,13/11/17,16:28:15,0.98,25.82,27.219999999999999,17.05,6.5192,0
275,13/11/17,16:30:46,0.99,25.93,26.759999999999999,16.31,10.7005,0
276,13/11/17,16:37:35,0.97,26.23,27.559999999999999,16.21,7.90569,0
277,13/11/17,16:42:21,1.12,25.92,27.18,17.280000000000001,8.30054,0
278,13/11/17,16:54:31,1.09,26.5,28.87,14.569999999999999,6.5192,0

```

```

<datosEntrenamientoPruebas.csv>
Archivo Editar Buscar Opciones Ayuda
279,13/11/17,16:59:45,1.07,26.42,28.599999999999999,15.260000000000002,6.5192,0
280,13/11/17,17:16:37,1.06,26.37,27.170000000000001,17.21,6.67083,0
281,13/11/17,17:27:39,1.12,25.98,26.18,16.279999999999999,11.5109,0
282,13/11/17,17:37:11,0.98,26.32,26.19,15.550000000000001,5.70088,0
283,13/11/17,17:48:28,1.98,26.43,28.979999999999999,16.650000000000001,9.30064,1
284,13/11/17,17:52:21,2.12,25.82,28.17,14.67,9.30054,1
285,13/11/17,17:55:02,0.97,25.86,26.179999999999999,13.710000000000001,5.70088,0
286,13/11/17,17:58:18,0.96,25.98,27.71,170000000000001,17.700000000000001,5.1665,0
287,13/11/17,18:11:04,1.12,26.34,27.249999999999999,17.08,7.51665,0
288,13/11/17,18:23:23,1.01,26.13,27.369999999999999,15.020000000000001,8.5109,0
289,13/11/17,18:25:12,1.11,25.82,27.29,16.77,8.0168,0
290,13/11/17,18:37:16,0.95,25.82,28.529999999999999,14.82,8.1242,0
291,13/11/17,18:39:45,0.98,26.53,27.79,13.860000000000001,7.64853,0
292,13/11/17,18:47:28,1.08,24.23,27.670000000000001,17.16,5.04152,0
293,13/11/17,18:52:24,1.05,25.9,27.919999999999999,15.040000000000002,10.7935,0
294,13/11/17,18:55:21,0.91,26.43,28.29,17.320000000000001,8.51469,0
295,13/11/17,19:08:41,1.10,26.45,27.899999999999999,19.67,8.63134,0
296,13/11/17,19:20:17,0.96,26.3,27.77,15.850000000000001,5.70088,0
297,13/11/17,19:27:12,0.95,25.99,27.849999999999999,15.82,5.04152,0
298,13/11/17,19:32:54,1.08,25.92,27.47,16.31,6.67083,0
299,13/11/17,19:39:21,1.12,25.87,27.560000000000001,17.28,8.63134,0
300,13/11/17,19:43:37,0.98,25.81,27.15,16.879999999999999,6.67083,0
301,13/11/17,19:52:29,2.67,26.54,29.760000000000001,19.26,10.1242,1
302,13/11/17,20:10:21,1.98,26.41,29.46,17.589999999999999,10.1342,1
303,13/11/17,20:19:24,1.03,26.37,27.42,18.94,6.67083,0
304,13/11/17,20:28:52,1.06,24.81,27.700000000000001,18.330000000000002,7.51665,0
305,13/11/17,20:30:18,1.11,24.93,29.89,17.469999999999999,7.51665,0
306,13/11/17,20:37:19,1.06,24.93,27.11,17.24,9.1224,0
307,13/11/17,20:43:24,0.96,25.96,27.640000000000001,17.589999999999999,9.1242,0
308,13/11/17,20:45:13,0.97,27.12,30.8,17.24,8.86002,0
309,13/11/17,20:52:52,0.99,27.16,27.499999999999999,18.529999999999999,8.63134,0

```

<datosEntrenamientoPruebas.csv>

Archivo Editar Buscar Opciones Ayuda

310	13/11/17	20:55:46	0.96	25.95	26.49	17.53	5.70088	0
311	13/11/17	21:07:42	0.97	26.58	29.87	9999999999999999	17.1	8.7935,0
312	13/11/17	21:19:12	1.08	27.28	26.86	18.02	00000000000001	8.51469,0
313	13/11/17	21:28:43	0.96	25.82	26.95	16.71	9999999999999999	7.64853,0
314	13/11/17	21:31:21	1.10	25.89	27.85	14.55	00000000000001	6.04152,0
315	13/11/17	21:33:22	1.08	26.3	27.75	9999999999999999	15.23	0000000000002,5.70088,0
316	13/11/17	21:42:56	1.11	25.33	27.07	15.14	00000000000001	8.86002,0
317	13/11/17	21:50:28	0.96	26.35	28.93	16.24	9999999999999999	7.64853,0
318	13/11/17	21:59:33	0.97	27.16	28.63	15.84	5.70088	0
319	13/11/17	22:17:51	0.98	27.27	28.05	00000000000001	16.16	9999999999999999,8.1242,0
320	13/11/17	22:21:22	1.98	27.39	29.8	16.8	10.1242	1
321	13/11/17	22:36:35	1.05	27.23	27.66	15.85	00000000000001	5.70088,0
322	13/11/17	22:38:37	1.10	25.89	27.24	16.27	9999999999999999	7.6485,0
323	13/11/17	22:42:16	0.98	26.32	28.37	00000000000001	16.2	7.6485,0
324	13/11/17	22:44:23	0.97	25.45	27.73	00000000000001	14.83	0000000000002,12.3491,0
325	13/11/17	22:59:45	0.97	26.43	28.58	9999999999999999	16.37	10.1242,0
326	17/11/17	15:12:23	1.76	24.78	26.05	14.84	8.04152	1
327	17/11/17	15:14:34	1.67	26.32	28.05	9999999999999999	14.84	8.63134,1
328	17/11/17	15:16:36	1.85	25.87	28.53	9999999999999999	14.77	8.6485,1
329	17/11/17	15:19:21	1.73	24.98	28.01	9999999999999999	17.04	9999999999999999,8.70088,1
330	17/11/17	15:22:28	1.67	27.46	28.18	16.09	8.70088	1
331	17/11/17	15:25:47	1.75	25.65	28.32	9999999999999999	15.68	00000000000001,8.53553,1
332	17/11/17	15:28:51	1.97	27.12	28.59	13.26	9.04152	1
333	17/11/17	15:31:23	0.94	24.92	27.84	9999999999999999	13.83	00000000000002,8.70088,0
334	17/11/17	15:35:30	0.98	26.36	28.87	00000000000001	13.31	8.04152,0
335	17/11/17	15:37:21	1.76	24.87	27.16	12.66	8.53553	1
336	17/11/17	15:39:18	1.69	26.43	27.57	9999999999999999	11.92	00000000000002,12.1037,1
337	17/11/17	15:42:22	1.85	26.74	27.69	11.22	00000000000001	8.53553,1
338	17/11/17	15:44:45	1.76	26.34	28.4	11.43	9.6485	1
339	17/11/17	15:49:12	1.79	26.54	29.59	9999999999999999	17.12	9.63134,1
340	17/11/17	16:01:17	1.78	26.45	27.4	11.43	00000000000001	8.6485,1

<datosEntrenamientoPruebas.csv>

Archivo Editar Buscar Opciones Ayuda

341	17/11/17	16:05:04	2.14	24.98	27.60	00000000000001	11.92	12.1037,1
342	17/11/17	16:11:23	1.79	26.54	28.8	13.33	00000000000002	8.53553,1
343	17/11/17	16:14:15	2.11	25.87	28.05	9999999999999999	12.99	9.82344,1
344	17/11/17	16:17:47	2.24	24.91	28.31	12.52	00000000000001	9.61769,1
345	17/11/17	16:21:35	2.12	25.89	29.17	18.17	8.04152	1
346	17/11/17	16:25:50	2.25	26.34	28.51	9999999999999999	13.59	8.63134,1
347	17/11/17	16:27:42	2.15	24.93	28.36	12.43	8.53553	1
348	17/11/17	16:32:44	2.01	24.95	28.67	00000000000001	18.08	8.53553,1
349	17/11/17	16:36:51	1.90	26.56	28.75	16.89	00000000000001	8.52268,1
350	17/11/17	16:38:30	1.87	24.92	28.76	9999999999999999	15.05	00000000000002,8.53553,1
351	17/11/17	16:42:22	1.76	25.83	28.01	9.0	8.52268	1
352	17/11/17	16:44:19	0.99	25.83	26.51	13.16	9999999999999999	6.04152,0
353	17/11/17	16:47:06	0.96	25.87	26.3	13.19	9999999999999999	6.04152,0
354	17/11/17	16:49:23	1.92	26.43	25.80	00000000000001	10.7	8.53553,1
355	17/11/17	16:55:38	1.73	25.89	25.24	11.02	00000000000002	12.1037,1
356	17/11/17	17:05:20	1.99	26.34	28.46	9999999999999999	11.11	00000000000001,8.53553,1
357	17/11/17	17:14:17	1.84	26.45	28.94	13.59	8.65083	1
358	17/11/17	17:16:58	1.76	26.32	28.88	10.13	9999999999999999	8.63134,1
359	17/11/17	17:19:36	2.36	26.45	28.81	00000000000001	10.79	8.74643,1
360	17/11/17	17:22:29	1.98	26.34	29.42	12.82	9.19239	1
361	17/11/17	17:24:12	2.24	26.32	28.32	10.88	9999999999999999	8.04152,1
362	17/11/17	17:28:16	2.16	26.45	29.57	12.77	00000000000001	9.82344,1
363	17/11/17	17:32:02	2.24	27.12	28.82	00000000000002	12.58	9.61769,1
364	17/11/17	17:36:11	2.33	27.43	27.00	00000000000001	12.42	8.04152,1
365	17/11/17	17:41:34	2.26	25.89	27.86	12.73	9999999999999999	8.63134,1
366	17/11/17	17:46:39	2.31	25.89	28.17	11.97	00000000000001	8.04152,1
367	17/11/17	17:49:21	2.23	26.45	27.68	9999999999999999	17.47	00000000000002,8.04152,1
368	17/11/17	17:52:34	1.75	27.32	27.76	00000000000001	17.2	8.04152,1
369	17/11/17	17:58:25	1.84	25.78	27.32	9999999999999999	13.80	00000000000002,8.53553,1
370	17/11/17	18:06:34	1.90	25.78	27.84	8.11	00000000000001	8.04152,1
371	17/11/17	18:14:15	0.99	25.98	27.12	9999999999999999	12.08	5.70088,0

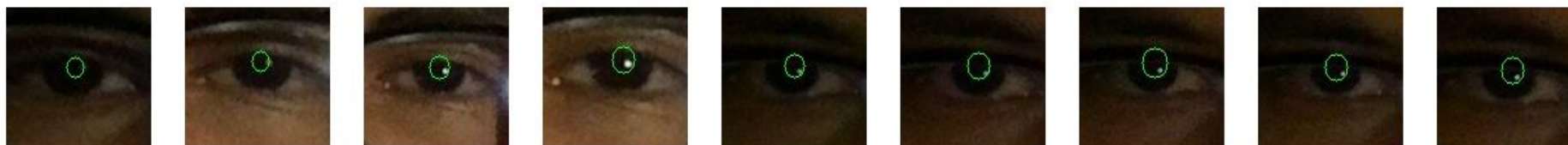
372,17/11/17,18:19:09,0,26.54,26.429999999999999,12.38,6.04152,0
373,17/11/17,18:24:23,1.89,26.75,28.240000000000001,8.57,8.04152,1
374,17/11/17,18:28:28,2.57,26.54,28.64,10.129999999999999,12.1037,1|
375,17/11/17,18:34:31,2.34,25.89,28.9,11.380000000000001,8.04152,1
376,17/11/17,18:38:45,3.11,25.89,29.36,12.559999999999999,8.04152,1
377,17/11/17,18:43:36,2.36,26.55,28.43,10.08,8.63134,1
378,17/11/17,18:47:52,2.32,26.66,28.17,11.499999999999999,6.04152,1
379,17/11/17,18:50:12,1.98,26.67,28.800000000000001,11.520000000000001,5.04152,1
380,17/11/17,18:52:11,2.63,27.23,28.079999999999999,12.26,8.53553,1
381,17/11/17,18:58:55,1.98,28.01,28.64,11.75,9.82344,1
382,17/11/17,19:12:24,1.09,26.56,28.699999999999999,12.200000000000002,9.61769,1
383,20/11/17,15:31:56,0,26.65,25.62,9.46,7.648531,0
384,24/11/17,15:36:22,0,26.67,26.859999999999999,10.61,5.14782,0
385,24/11/17,16:01:49,0,27.21,27.27,11.86,5.04152,0
386,24/11/17,16:13:25,0,27.34,27.270000000000001,11.699999999999999,5.04152,0
387,24/11/17,16:18:40,0,27.23,27.680000000000001,12.17,4.30116,0
388,24/11/17,16:35:17,0,28.02,27.52,12.499999999999999,7.64853,0
389,24/11/17,16:41:06,0,26.13,26.939999999999999,10.370000000000002,5.70088,0
390,24/11/17,16:50:35,0,27.23,26.57,12.259999999999999,5.52268,0
391,25/04/18,06:46:01,0,26.21,53,23.760000000000002,17.329999999999998,5.70088,0
392,25/04/18,06:49:43,1.22,23.62,28.5,16.48,4.74342,1
393,25/04/18,06:58:53,0,27.24,81,28.199999999999999,16.949999999999999,4.74342,0
394,25/04/18,07:02:46,1.95,25.73,28.100000000000001,16.710000000000001,4.74342,1
395,25/04/18,07:20:46,1.8400000000000001,26.48,28.84,19.010000000000002,3.53553,1
396,25/04/18,07:38:46,0,25999999999999999,29.47,30.16,21.329999999999998,4.74342,0
397,25/04/18,07:45:05,1.385,30.02,29.510000000000002,28.710000000000001,4.52769,1
398,25/04/18,07:54:25,0,96,30.64,30.5,24.010000000000002,6.5192,1
399,25/04/18,07:59:34,0,28,30.72,30.23,21.219999999999999,4.52769,0
400,25/04/18,08:23:29,1.93,30.99,30.449999999999999,21.789999999999999,4.52769,1
401,25/04/18,08:32:37,1.87,26.88,28.550000000000001,19.34,6.04152,1
401,25/04/18,08:32:37,1.87,26.88,28.550000000000001,19.34,6.04152,1
402,25/04/18,08:37:27,0,28,29.42,29.579999999999998,20.449999999999999,4.74342,0
403,25/04/18,08:45:40,1.26000000000000001,30.85,29.829999999999998,21.039999999999999
404,25/04/18,08:50:46,0,26,33.32,30.300000000000001,22.649999999999999,4.74342,1
405,25/04/18,09:24:24,1.69,34.39,30.449999999999999,22.109999999999999,4.74342,1
406,25/04/18,16:55:19,0,25999999999999999,27.76,27.300000000000001,17.190000000000000
407,25/04/18,17:00:07,0,17000000000000001,27.81,27.489999999999998,16.91,6.04152,0
408,25/04/18,17:04:10,0,26999999999999999,28.03,27.620000000000001,17.27,10.5119,0
409,25/04/18,17:07:20,1.43,28.10,27.75,17.120000000000001,9.92472,1
410,25/04/18,17:10:33,0,25,28.11,27.940000000000001,16.329999999999998,10.5119,0
411,25/04/18,17:13:10,0,19000000000000001,28.10,28.329999999999998,16.690000000000000
412,25/04/18,17:17:07,1.38,28.34,27.870000000000001,17.879999999999999,9.61769,1
413,25/04/18,17:19:55,0,25,28.37,27.460000000000001,20.600000000000001,10.5119,1
414,25/04/18,17:23:00,0,26000000000000001,28.25,27.149999999999999,17.030000000000000
415,25/04/18,17:25:50,0,18000000000000001,28.14,27.640000000000001,16.199999999999999
416,25/04/18,17:28:39,1.96,28.03,27.539999999999999,17.07,9.61769,1
417,25/04/18,17:31:14,0,23999999999999999,27.92,28.039999999999999,15.98,5.14782,0
418,25/04/18,17:33:59,0,26,27.80,28.16,16.190000000000001,8.63134,0
419,25/04/18,17:39:12,1.77,27.76,27.960000000000001,17.420000000000002,5.52268,1
420,25/04/18,17:42:42,1.58,27.76,27.420000000000002,16.190000000000001,8.63134,1
421,25/04/18,17:45:16,1.77,27.73,27.149999999999999,16.0,10.7005,1
422,25/04/18,17:50:44,0,56,27.71,27.390000000000001,15.83,8.63134,1
423,25/04/18,17:53:28,1.46,27.67,27.059999999999999,15.65,8.86002,1
424,25/04/18,17:56:09,0,25000000000000007,27.78,27.149999999999999,15.869999999999999
425,25/04/18,17:59:02,0,20000000000000006,27.83,26.879999999999999,15.51,8.27647,1
426,25/04/18,18:04:23,0,27,27.62,27.09,15.07,9.61769,0
427,25/04/18,18:07:37,1.59,27.69,27.010000000000002,13.890000000000001,7.64853,1
428,25/04/18,18:10:30,1.35,27.69,26.969999999999999,15.51,6.36396,1
429,25/04/18,18:13:07,1.53,27.53,27.489999999999998,16.77,6.96419,1
430,25/04/18,18:17:45,2.74,27.47,26.66,15.15,10.9772,1
431,25/04/18,18:21:19,2.68,27.37,26.460000000000001,15.66,10.5119,1
432,25/04/18,18:23:47,2.49,27.31,26.52,15.85,9.61769,1
433,25/04/18,18:26:16,1.67,27.36,26.879999999999999,16.140000000000001,8.63134,1
434,25/04/18,18:29:49,1.85,27.37,26.739999999999998,16.09,6.5192,1
435,25/04/18,18:34:28,1.83,27.36,27.13,17.23,6.5192,1

Anexo 13. Imágenes diámetro de pupilas de los datos de entrenamiento





28.jpg 29.jpg 30.jpg 31.jpg 32.jpg 33.jpg 34.jpg 35.jpg 36.jpg



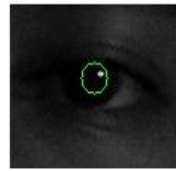
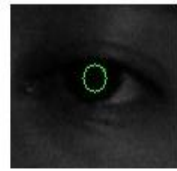
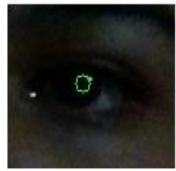
37.jpg 38.jpg 39.jpg 40.jpg 41.jpg 42.jpg 43.jpg 44.jpg 45.jpg



46.jpg 47.jpg 48.jpg 49.jpg 50.jpg 51.jpg 52.jpg 53.jpg 54.jpg



55.jpg 56.jpg 57.jpg 58.jpg 59.jpg 60.jpg 61.jpg 62.jpg 63.jpg



64.jpg

65.jpg

66.jpg

67.jpg

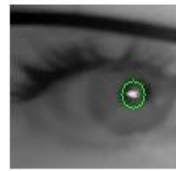
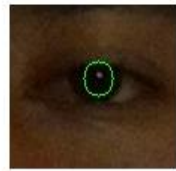
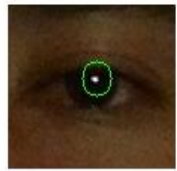
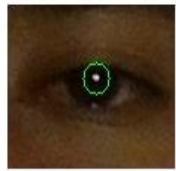
68.jpg

69.jpg

70.jpg

71.jpg

72.jpg



73.jpg

74.jpg

75.jpg

76.jpg

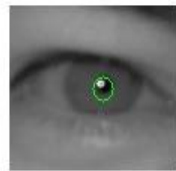
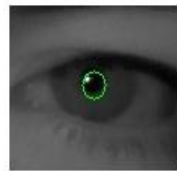
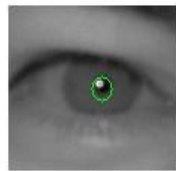
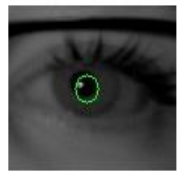
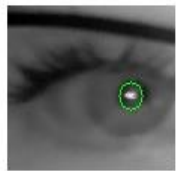
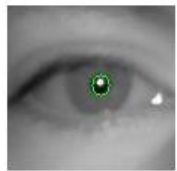
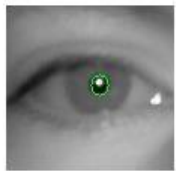
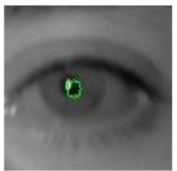
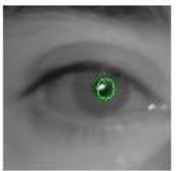
77.jpg

78.jpg

79.jpg

80.jpg

81.jpg



82.jpg

83.jpg

84.jpg

85.jpg

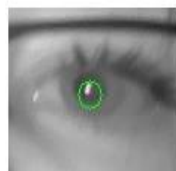
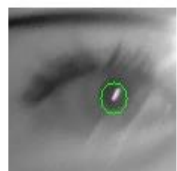
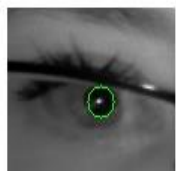
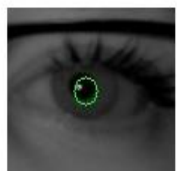
86.jpg

87.jpg

88.jpg

89.jpg

90.jpg



91.jpg

92.jpg

93.jpg

94.jpg

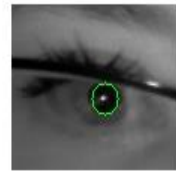
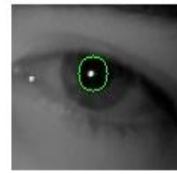
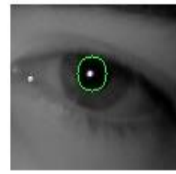
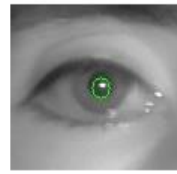
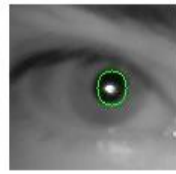
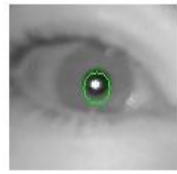
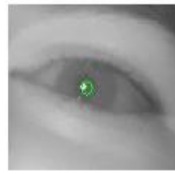
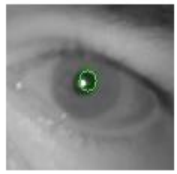
95.jpg

96.jpg

97.jpg

98.jpg

99.jpg



100.jpg

101.jpg

102.jpg

103.jpg

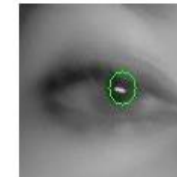
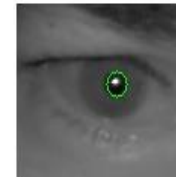
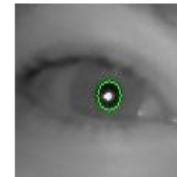
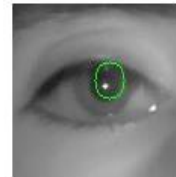
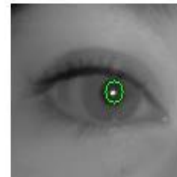
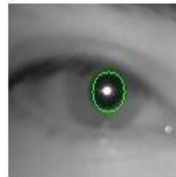
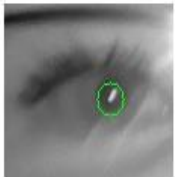
104.jpg

105.jpg

106.jpg

107.jpg

108.jpg



109.jpg

110.jpg

111.jpg

112.jpg

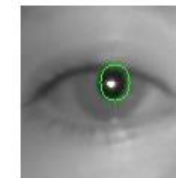
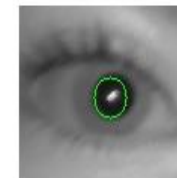
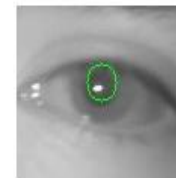
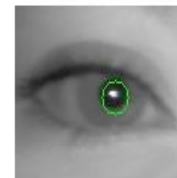
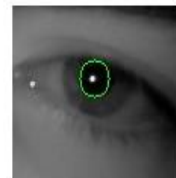
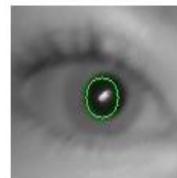
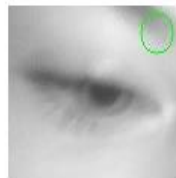
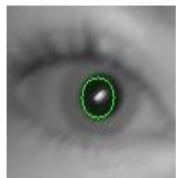
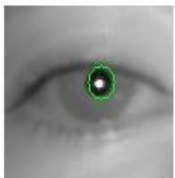
113.jpg

114.jpg

115.jpg

116.jpg

117.jpg



118.jpg

119.jpg

120.jpg

121.jpg

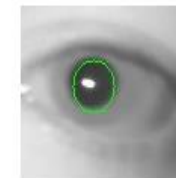
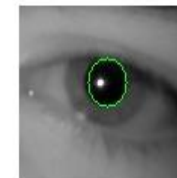
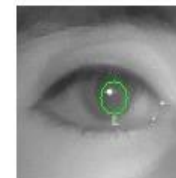
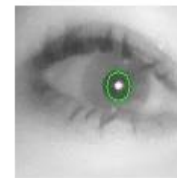
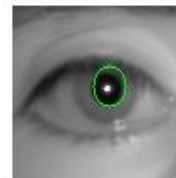
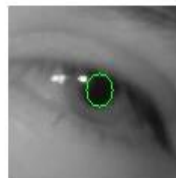
122.jpg

123.jpg

124.jpg

125.jpg

126.jpg



127.jpg

128.jpg

129.jpg

130.jpg

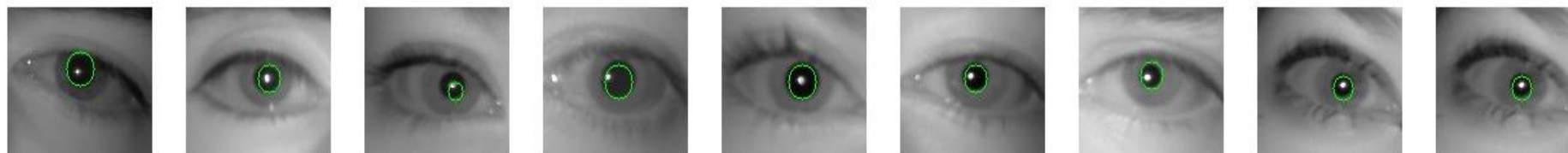
131.jpg

132.jpg

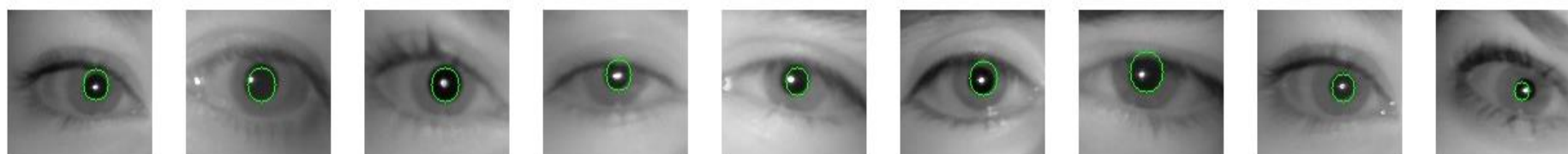
133.jpg

134.jpg

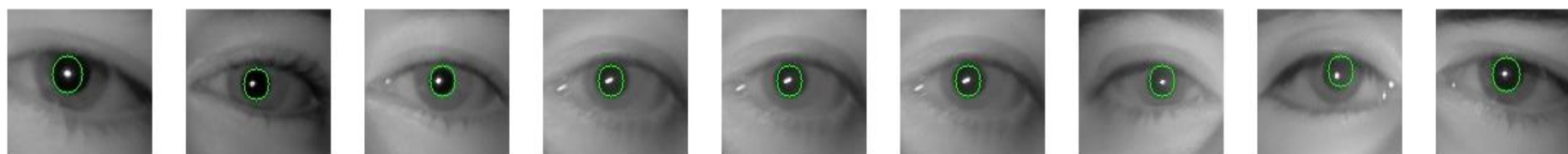
135.jpg



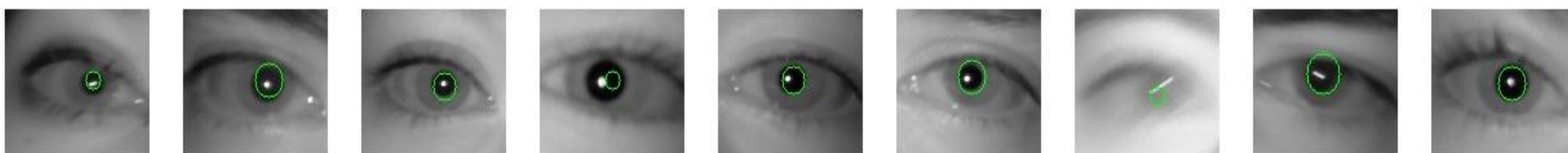
136.jpg 137.jpg 138.jpg 139.jpg 140.jpg 141.jpg 142.jpg 143.jpg 144.jpg



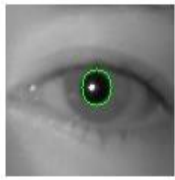
145.jpg 146.jpg 147.jpg 148.jpg 149.jpg 150.jpg 151.jpg 152.jpg 153.jpg



154.jpg 155.jpg 156.jpg 157.jpg 158.jpg 159.jpg 160.jpg 161.jpg 162.jpg



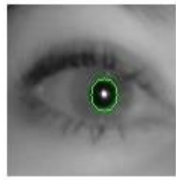
163.jpg 164.jpg 165.jpg 166.jpg 167.jpg 168.jpg 169.jpg 170.jpg 171.jpg



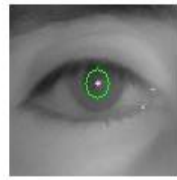
172.jpg



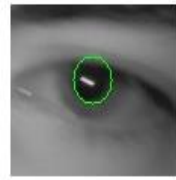
173.jpg



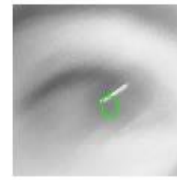
174.jpg



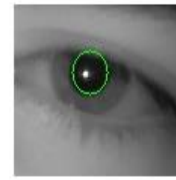
175.jpg



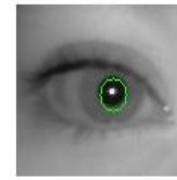
176.jpg



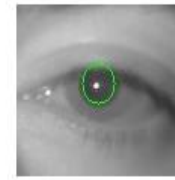
177.jpg



178.jpg



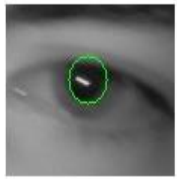
179.jpg



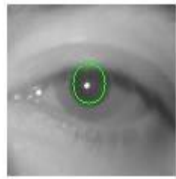
180.jpg



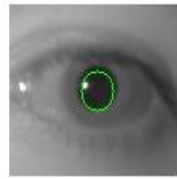
181.jpg



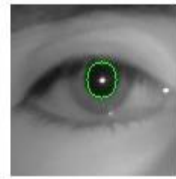
182.jpg



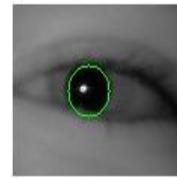
183.jpg



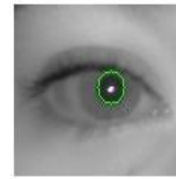
184.jpg



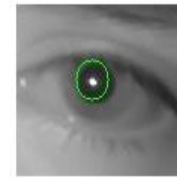
185.jpg



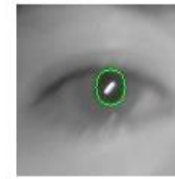
186.jpg



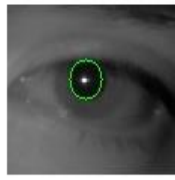
187.jpg



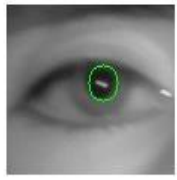
188.jpg



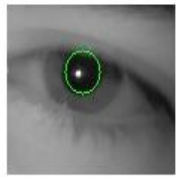
189.jpg



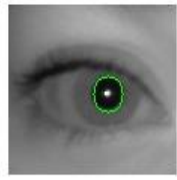
190.jpg



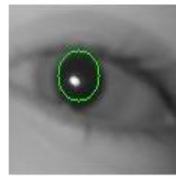
191.jpg



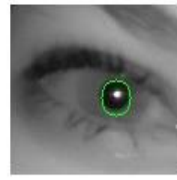
192.jpg



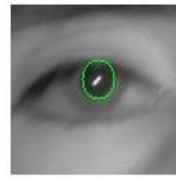
193.jpg



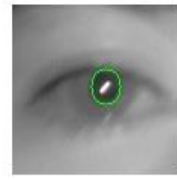
194.jpg



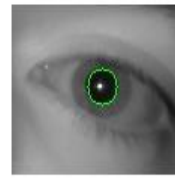
195.jpg



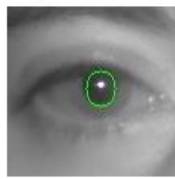
196.jpg



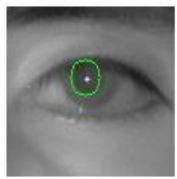
197.jpg



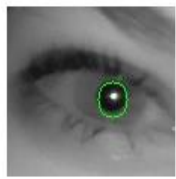
198.jpg



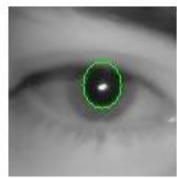
199.jpg



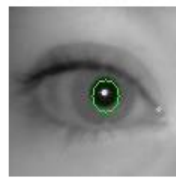
200.jpg



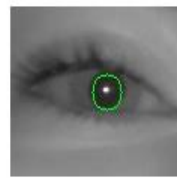
201.jpg



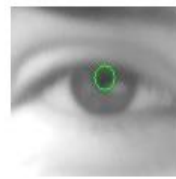
202.jpg



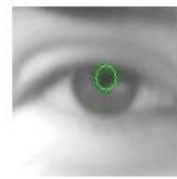
203.jpg



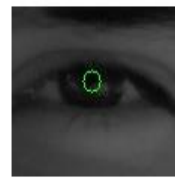
204.jpg



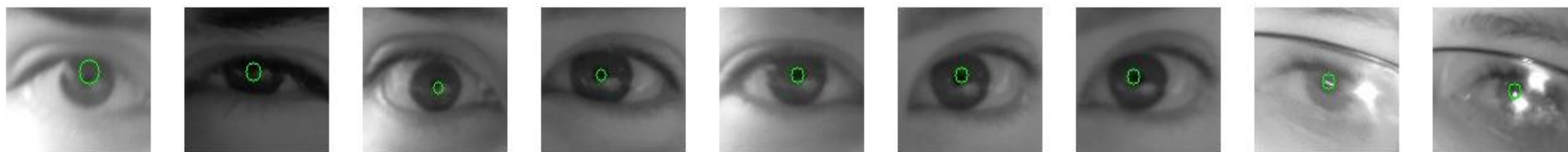
205.jpg



206.jpg



207.jpg



208.jpg

209.jpg

210.jpg

211.jpg

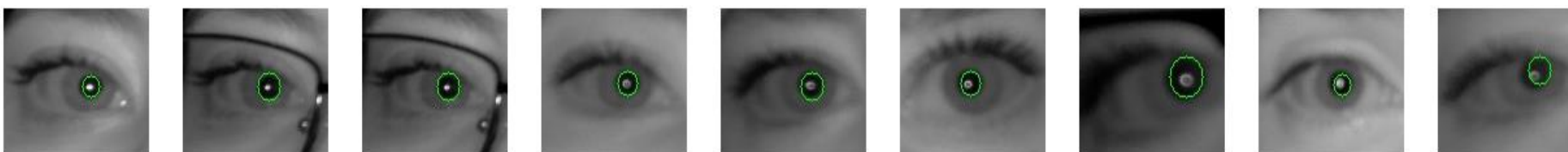
212.jpg

213.jpg

214.jpg

215.jpg

216.jpg



217.jpg

218.jpg

219.jpg

220.jpg

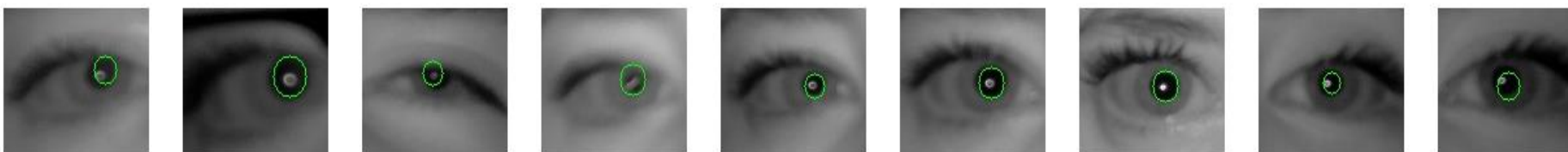
221.jpg

222.jpg

223.jpg

224.jpg

225.jpg



226.jpg

227.jpg

228.jpg

229.jpg

230.jpg

231.jpg

232.jpg

233.jpg

234.jpg



235.jpg

236.jpg

237.jpg

238.jpg

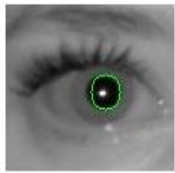
239.jpg

240.jpg

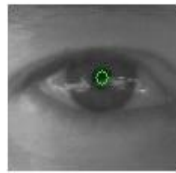
241.jpg

242.jpg

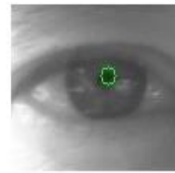
243.jpg



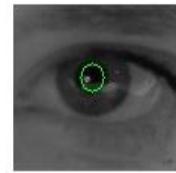
244.jpg



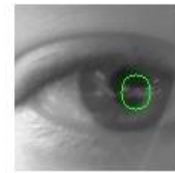
245.jpg



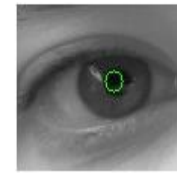
246.jpg



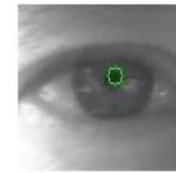
247.jpg



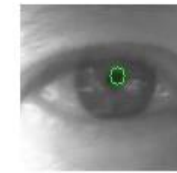
248.jpg



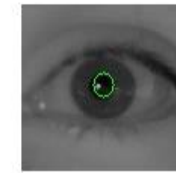
249.jpg



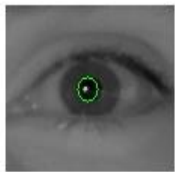
250.jpg



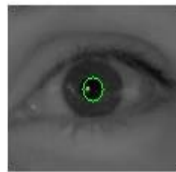
251.jpg



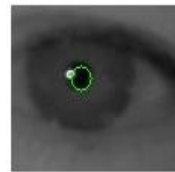
252.jpg



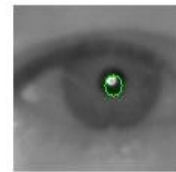
253.jpg



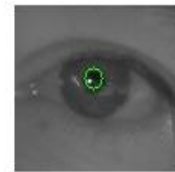
254.jpg



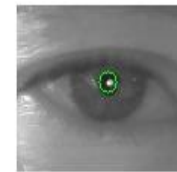
255.jpg



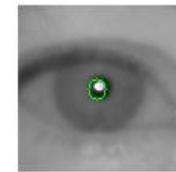
256.jpg



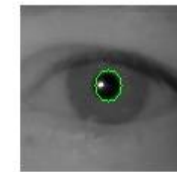
257.jpg



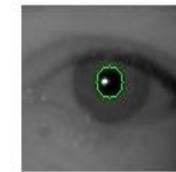
258.jpg



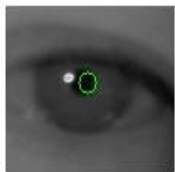
259.jpg



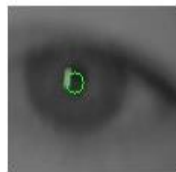
260.jpg



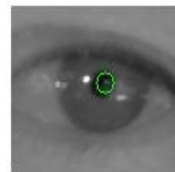
261.jpg



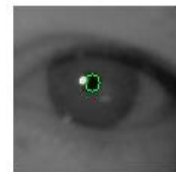
262.jpg



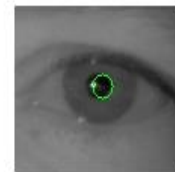
263.jpg



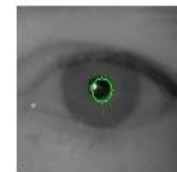
264.jpg



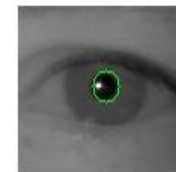
265.jpg



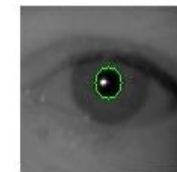
266.jpg



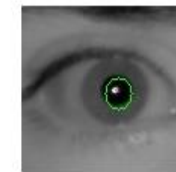
267.jpg



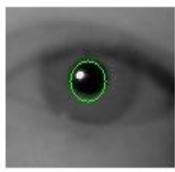
268.jpg



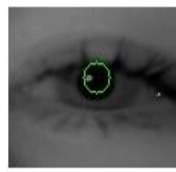
269.jpg



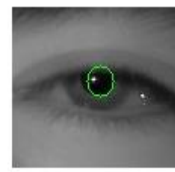
270.jpg



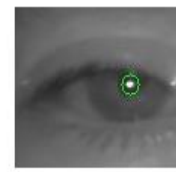
271.jpg



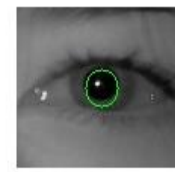
272.jpg



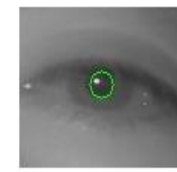
273.jpg



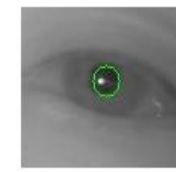
274.jpg



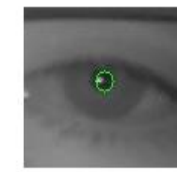
275.jpg



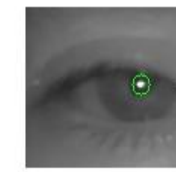
276.jpg



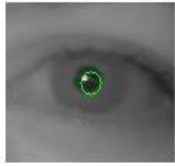
277.jpg



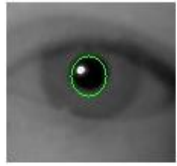
278.jpg



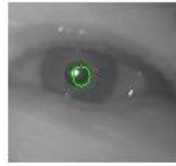
279.jpg



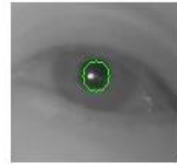
280.jpg



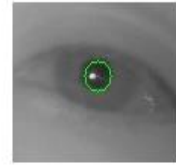
281.jpg



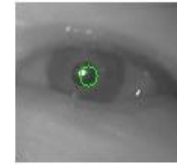
282.jpg



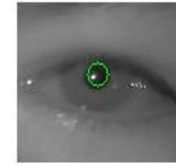
283.jpg



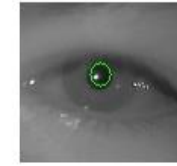
284.jpg



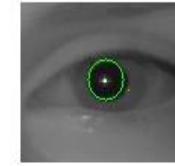
285.jpg



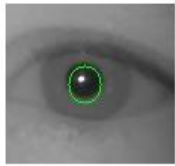
286.jpg



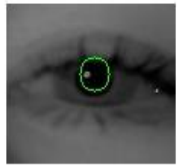
287.jpg



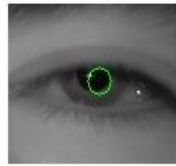
288.jpg



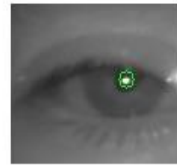
289.jpg



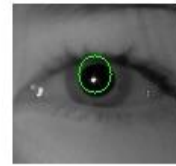
290.jpg



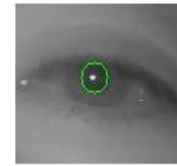
291.jpg



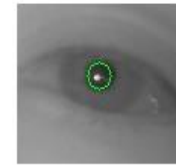
292.jpg



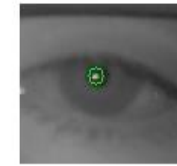
293.jpg



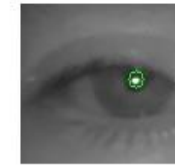
294.jpg



295.jpg



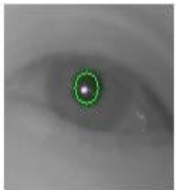
296.jpg



297.jpg



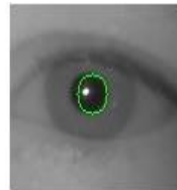
298.jpg



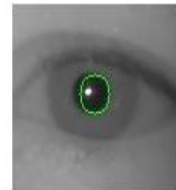
299.jpg



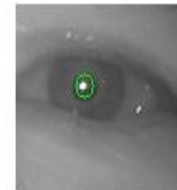
300.jpg



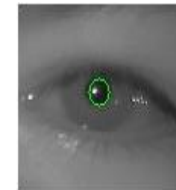
301.jpg



302.jpg



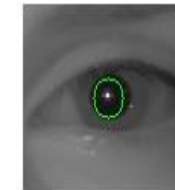
303.jpg



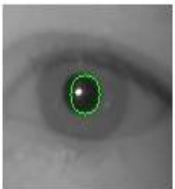
304.jpg



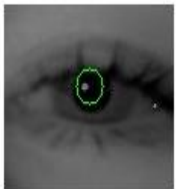
305.jpg



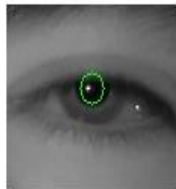
306.jpg



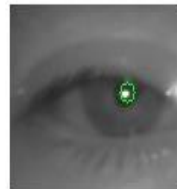
307.jpg



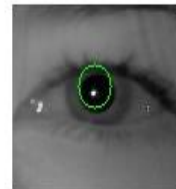
308.jpg



309.jpg



310.jpg



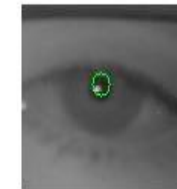
311.jpg



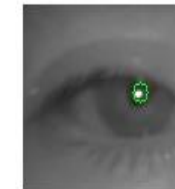
312.jpg



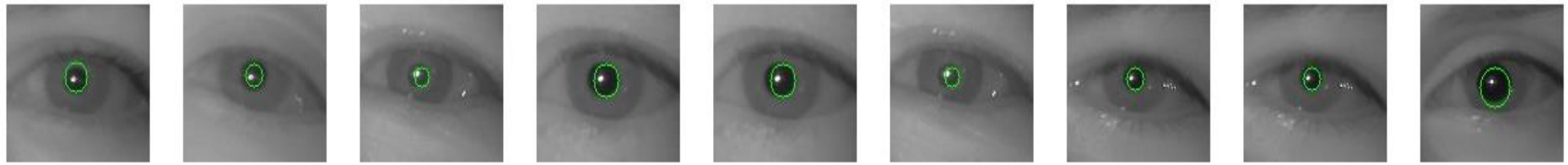
313.jpg



314.jpg



315.jpg



316.jpg

317.jpg

318.jpg

319.jpg

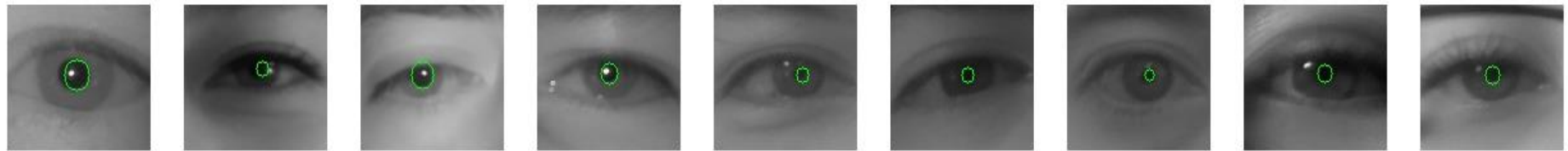
320.jpg

321.jpg

322.jpg

323.jpg

324.jpg



325.jpg

326.jpg

327.jpg

328.jpg

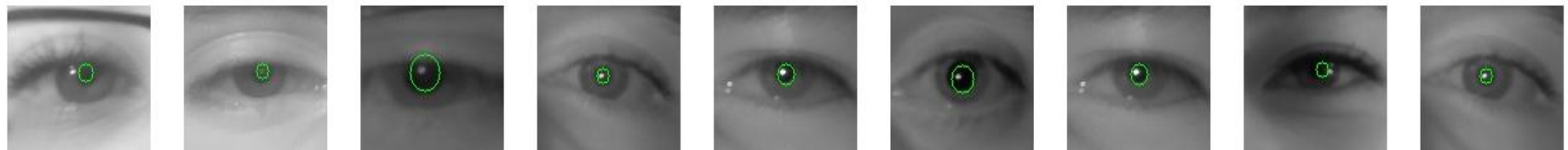
329.jpg

330.jpg

331.jpg

332.jpg

333.jpg



334.jpg

335.jpg

336.jpg

337.jpg

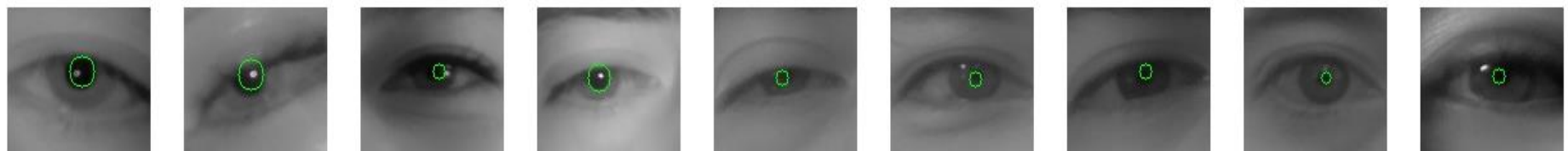
338.jpg

339.jpg

340.jpg

341.jpg

342.jpg



343.jpg

344.jpg

345.jpg

346.jpg

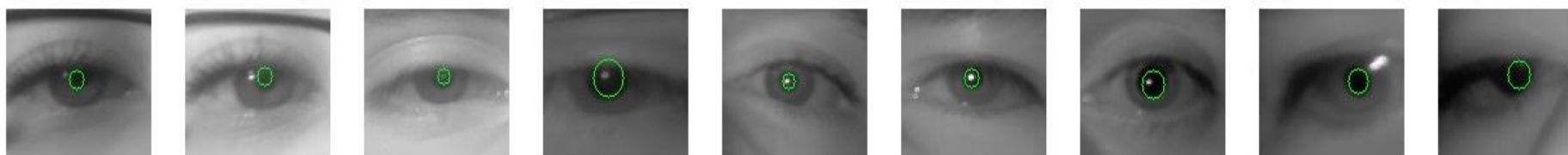
347.jpg

348.jpg

349.jpg

350.jpg

351.jpg



352.jpg

353.jpg

354.jpg

355.jpg

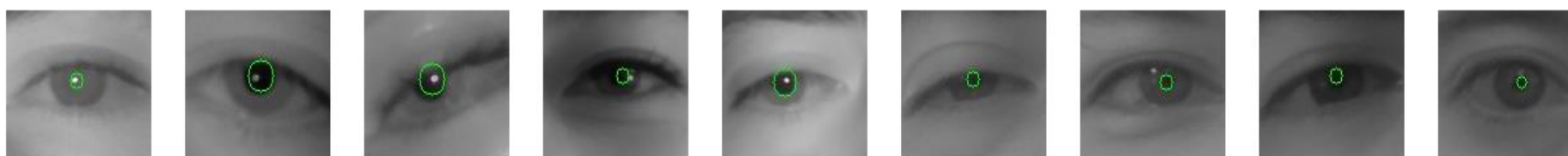
356.jpg

357.jpg

358.jpg

359.jpg

360.jpg



361.jpg

362.jpg

363.jpg

364.jpg

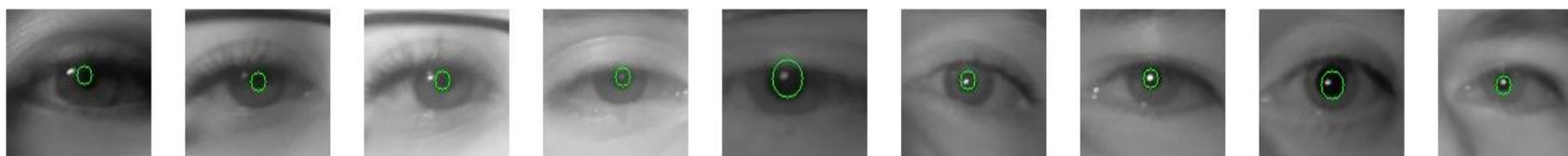
365.jpg

366.jpg

367.jpg

368.jpg

369.jpg



370.jpg

371.jpg

372.jpg

373.jpg

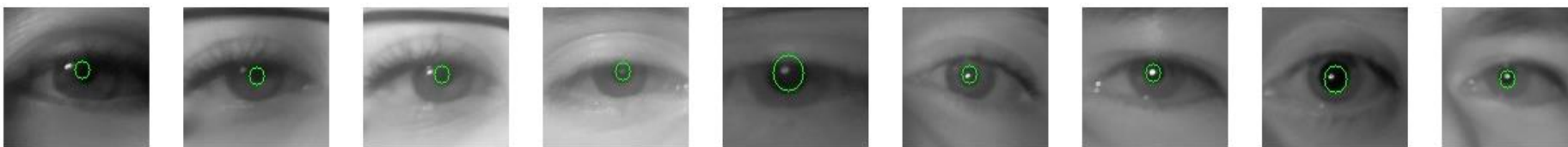
374.jpg

375.jpg

376.jpg

377.jpg

378.jpg



370.jpg

371.jpg

372.jpg

373.jpg

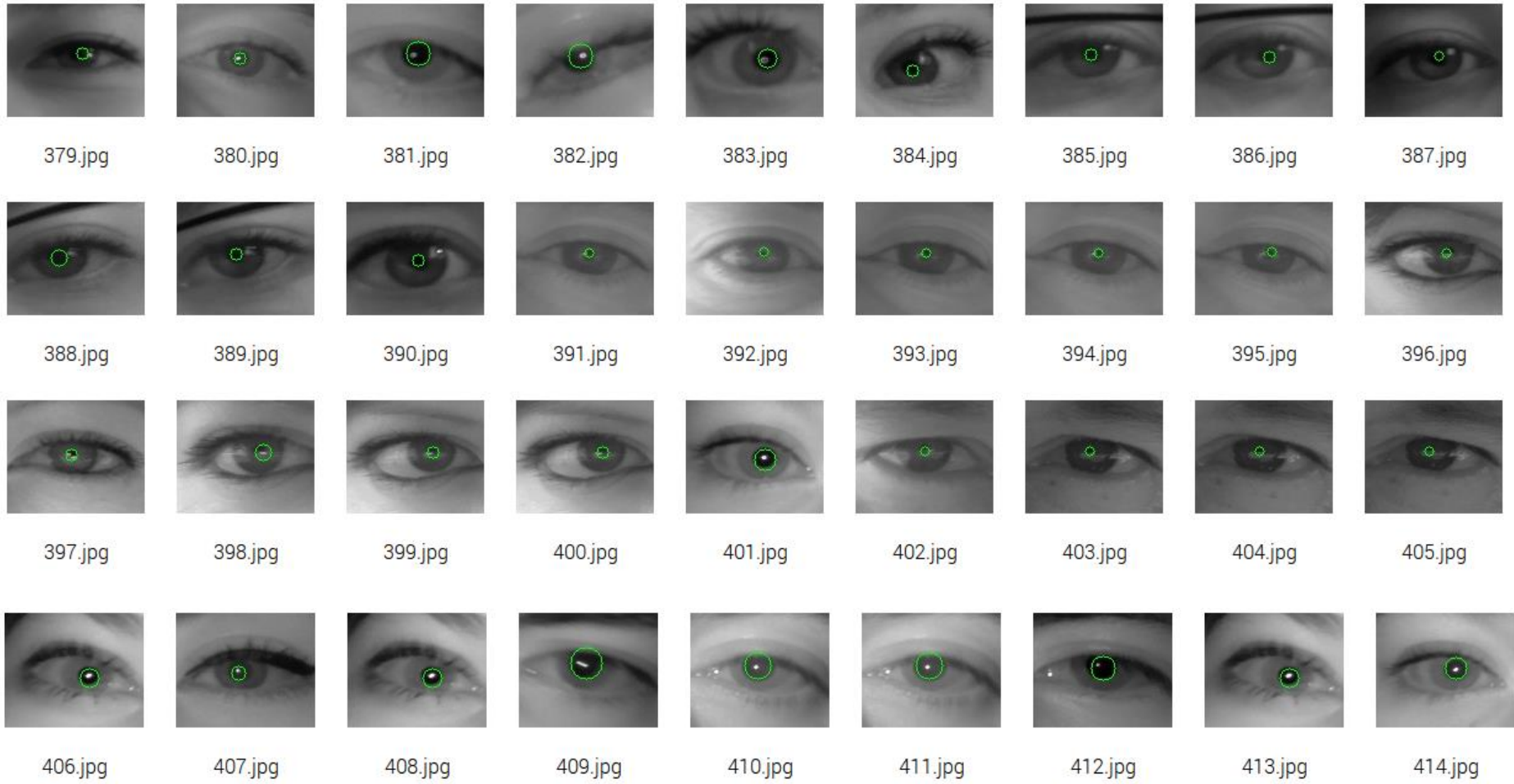
374.jpg

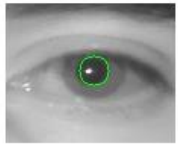
375.jpg

376.jpg

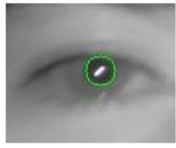
377.jpg

378.jpg

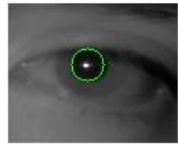




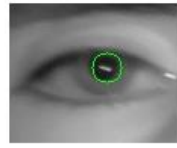
415.jpg



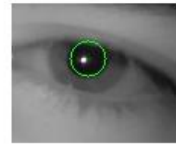
416.jpg



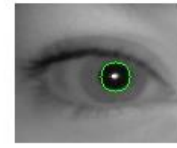
417.jpg



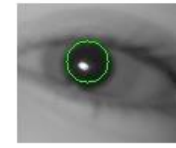
418.jpg



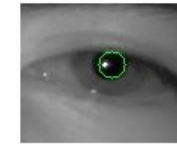
419.jpg



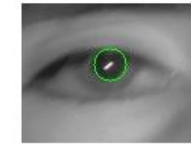
420.jpg



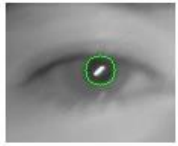
421.jpg



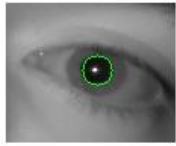
422.jpg



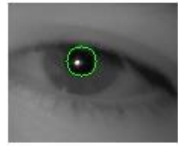
423.jpg



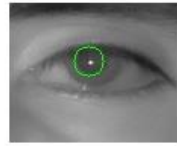
424.jpg



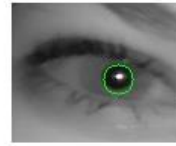
425.jpg



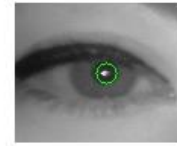
426.jpg



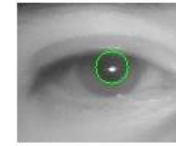
427.jpg



428.jpg



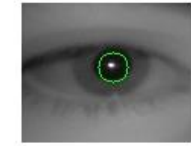
429.jpg



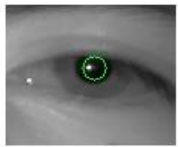
430.jpg



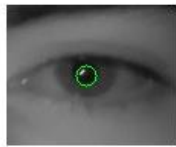
431.jpg



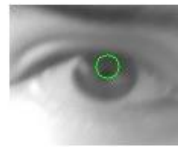
432.jpg



433.jpg

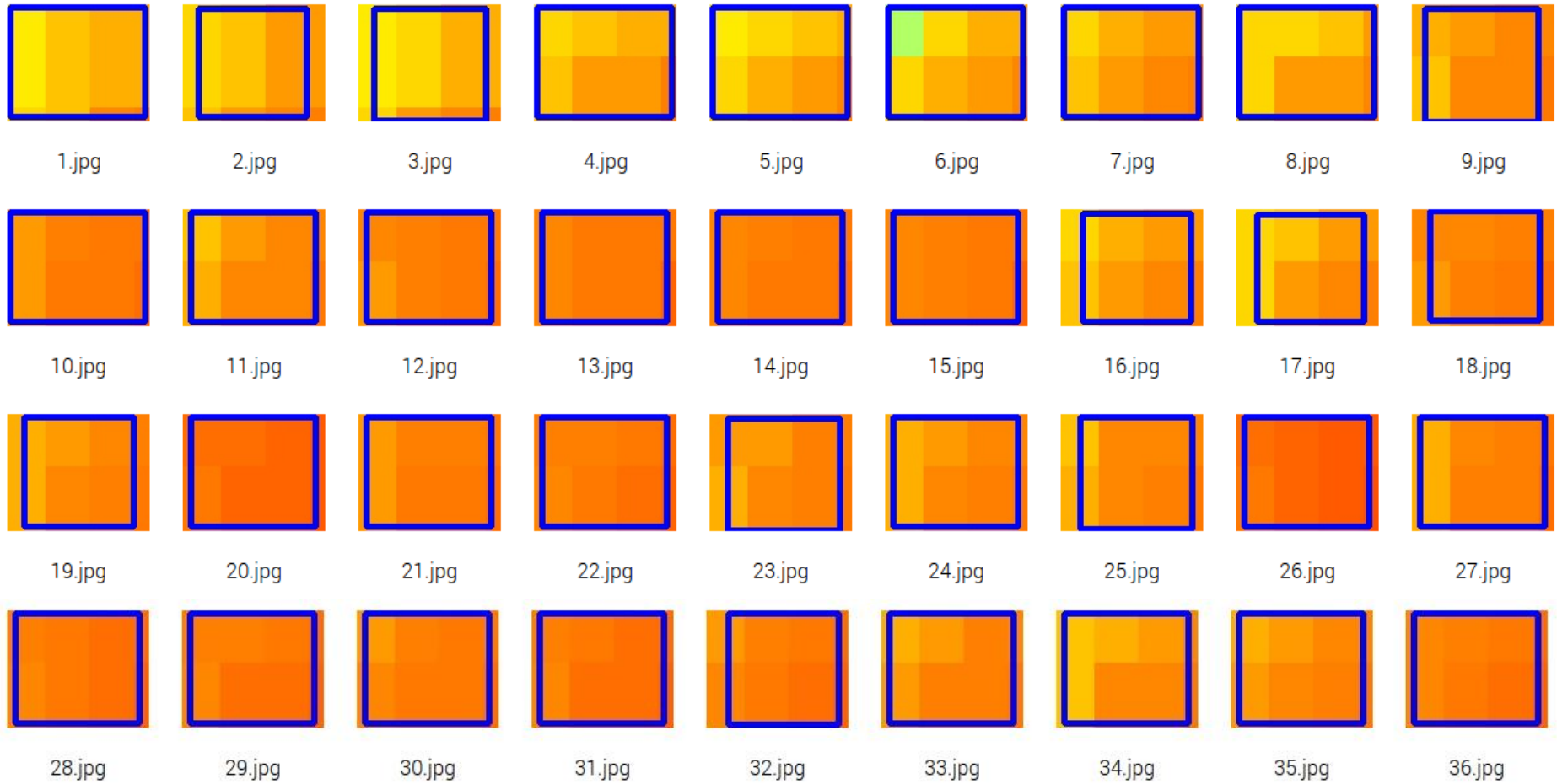


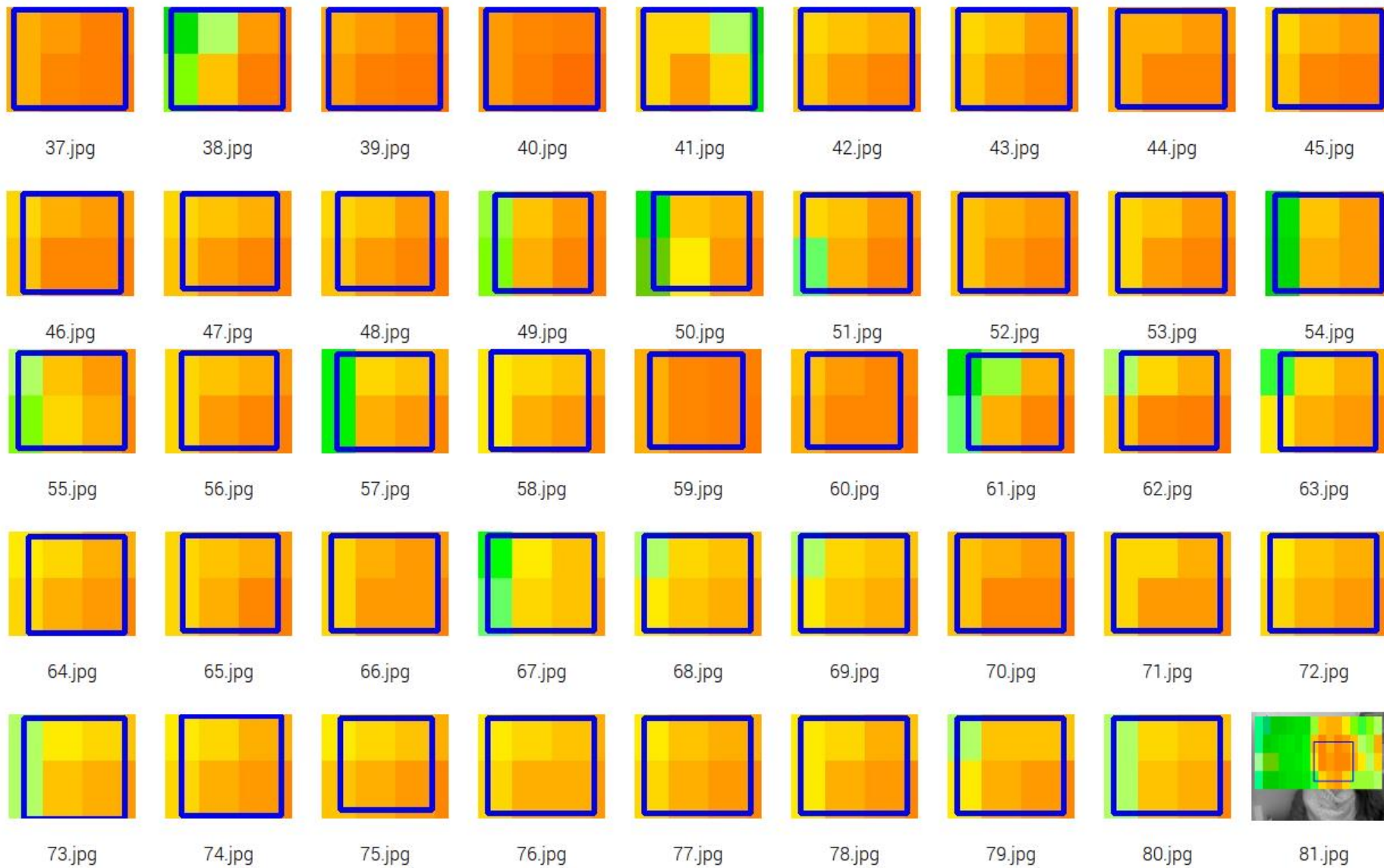
434.jpg



435.jpg

Anexo 14. Imágenes temperatura facial de los datos de entrenamiento







82.jpg

83.jpg

84.jpg

85.jpg

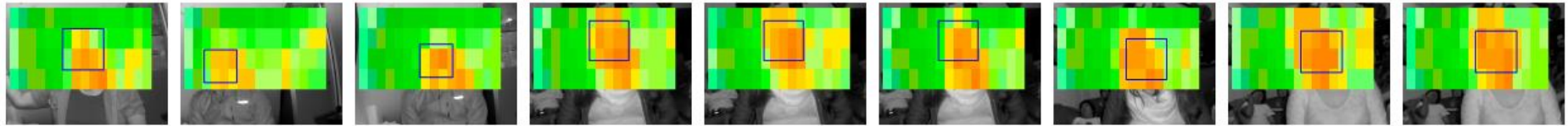
86.jpg

87.jpg

88.jpg

89.jpg

90.jpg



91.jpg

92.jpg

93.jpg

94.jpg

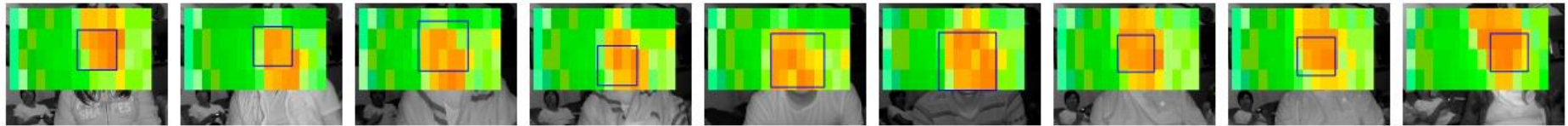
95.jpg

96.jpg

97.jpg

98.jpg

99.jpg



100.jpg

101.jpg

102.jpg

103.jpg

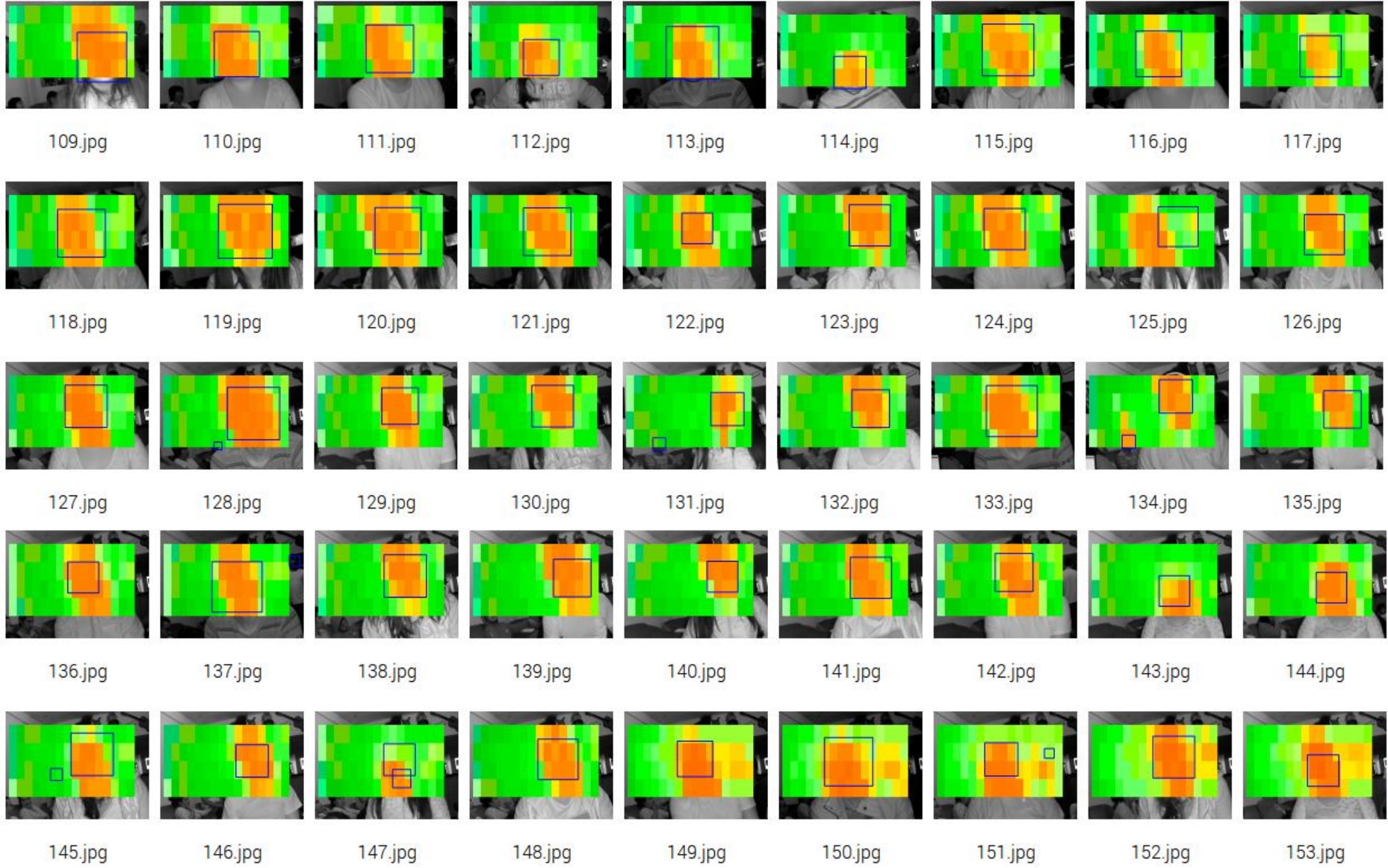
104.jpg

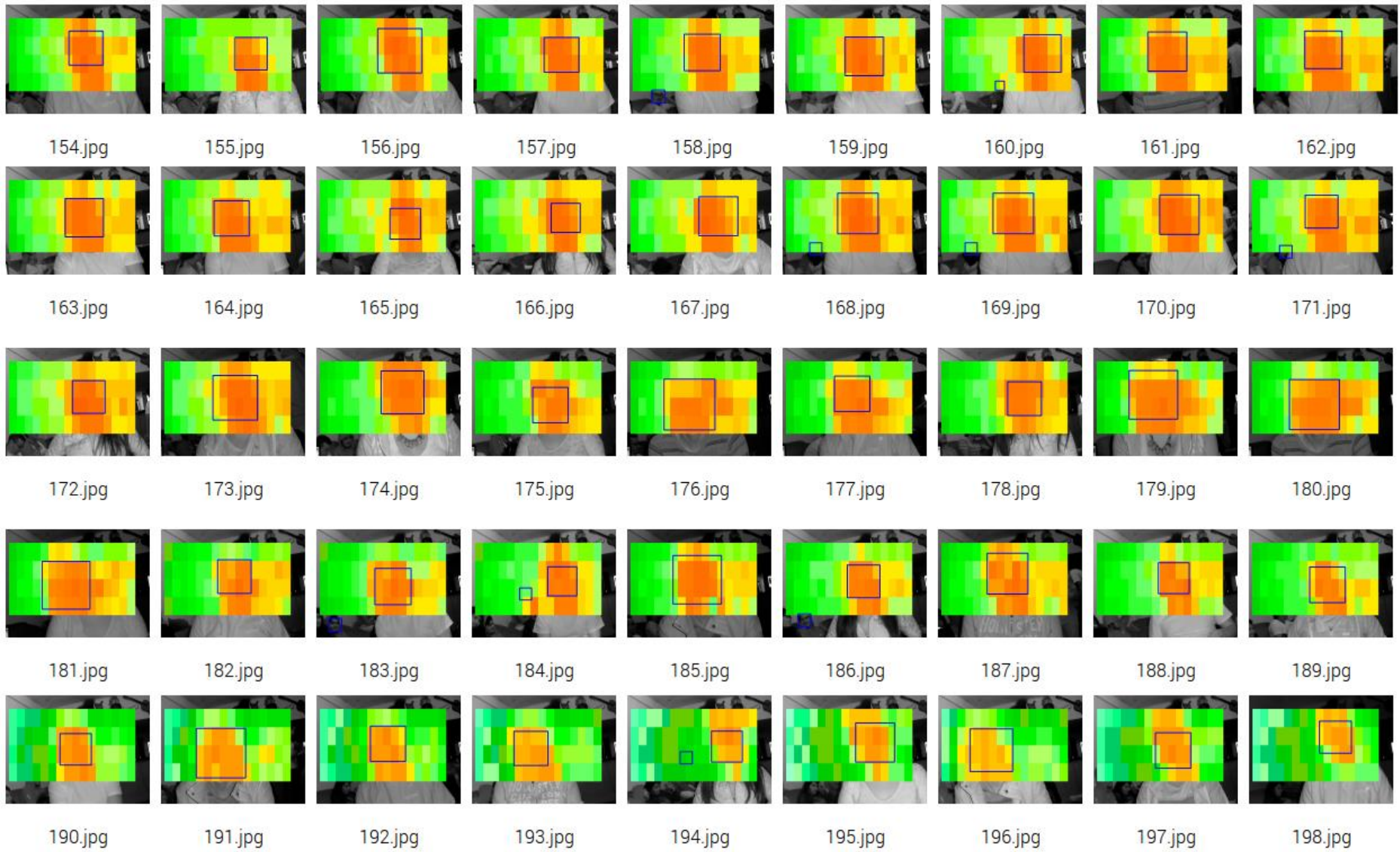
105.jpg

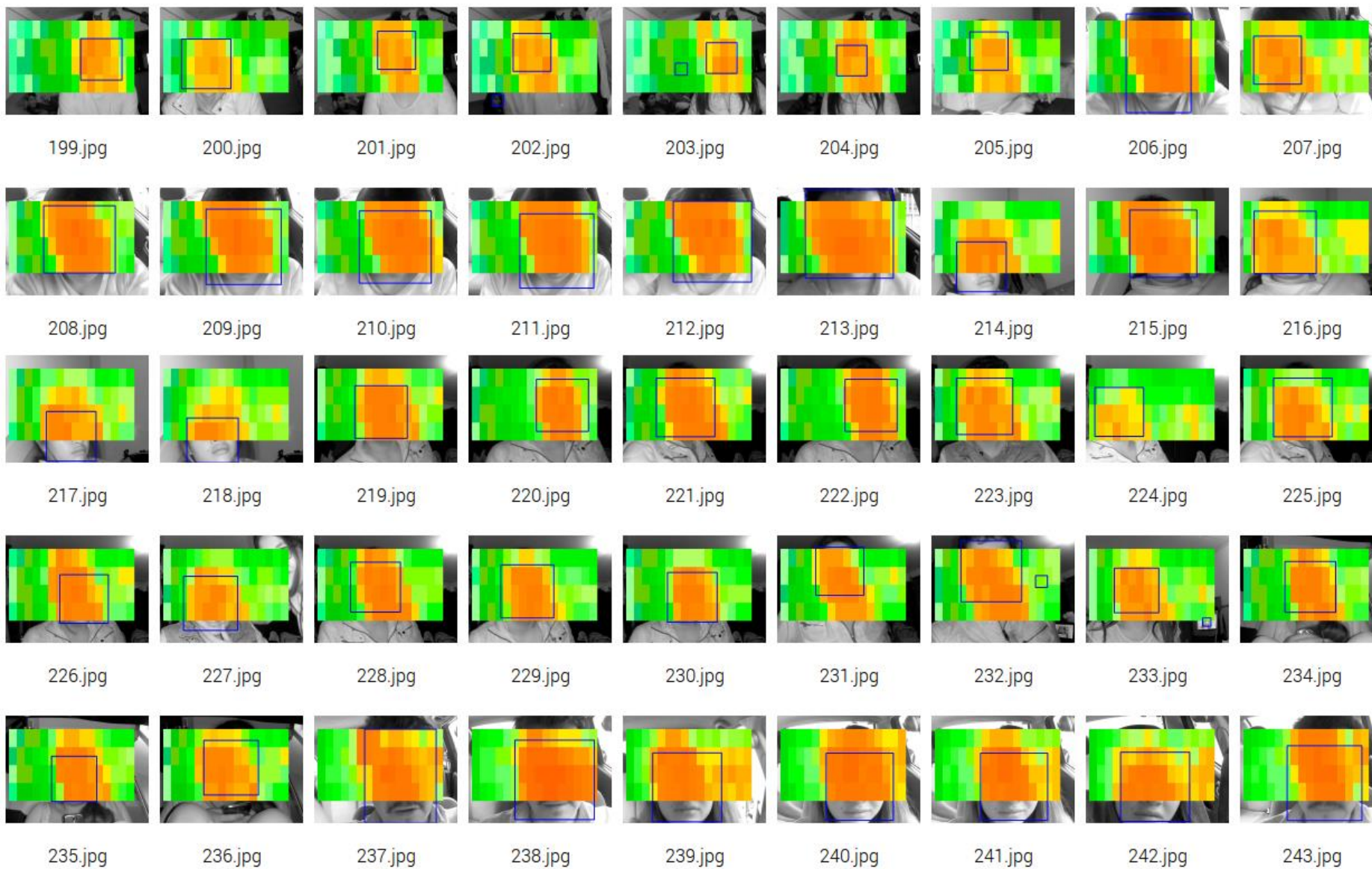
106.jpg

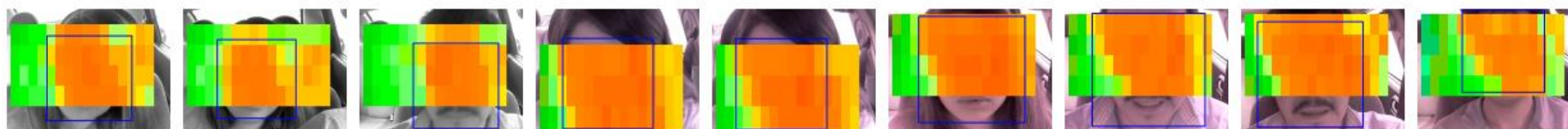
107.jpg

108.jpg

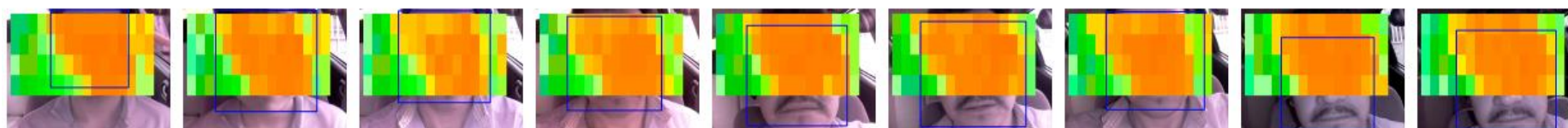




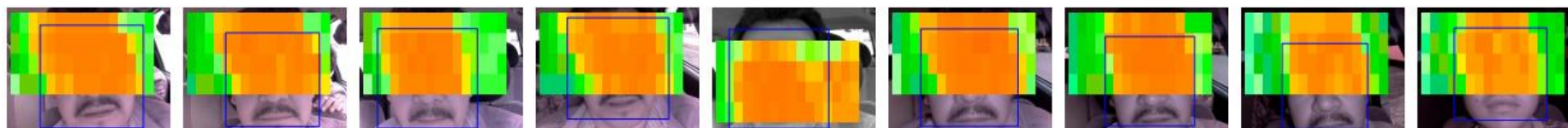




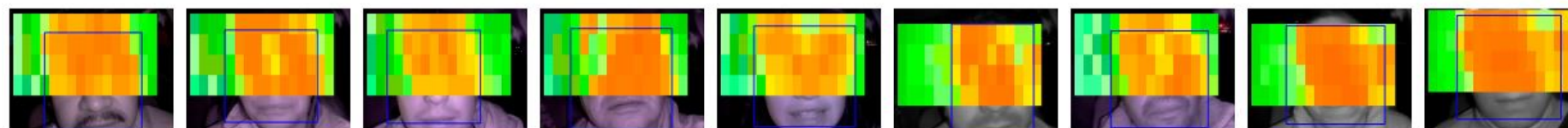
244.jpg 245.jpg 246.jpg 247.jpg 248.jpg 249.jpg 250.jpg 251.jpg 252.jpg



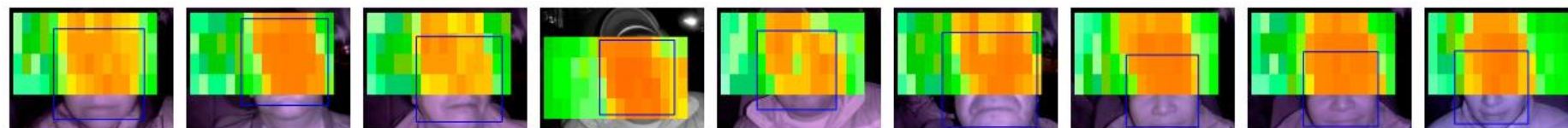
253.jpg 254.jpg 255.jpg 256.jpg 257.jpg 258.jpg 259.jpg 260.jpg 261.jpg



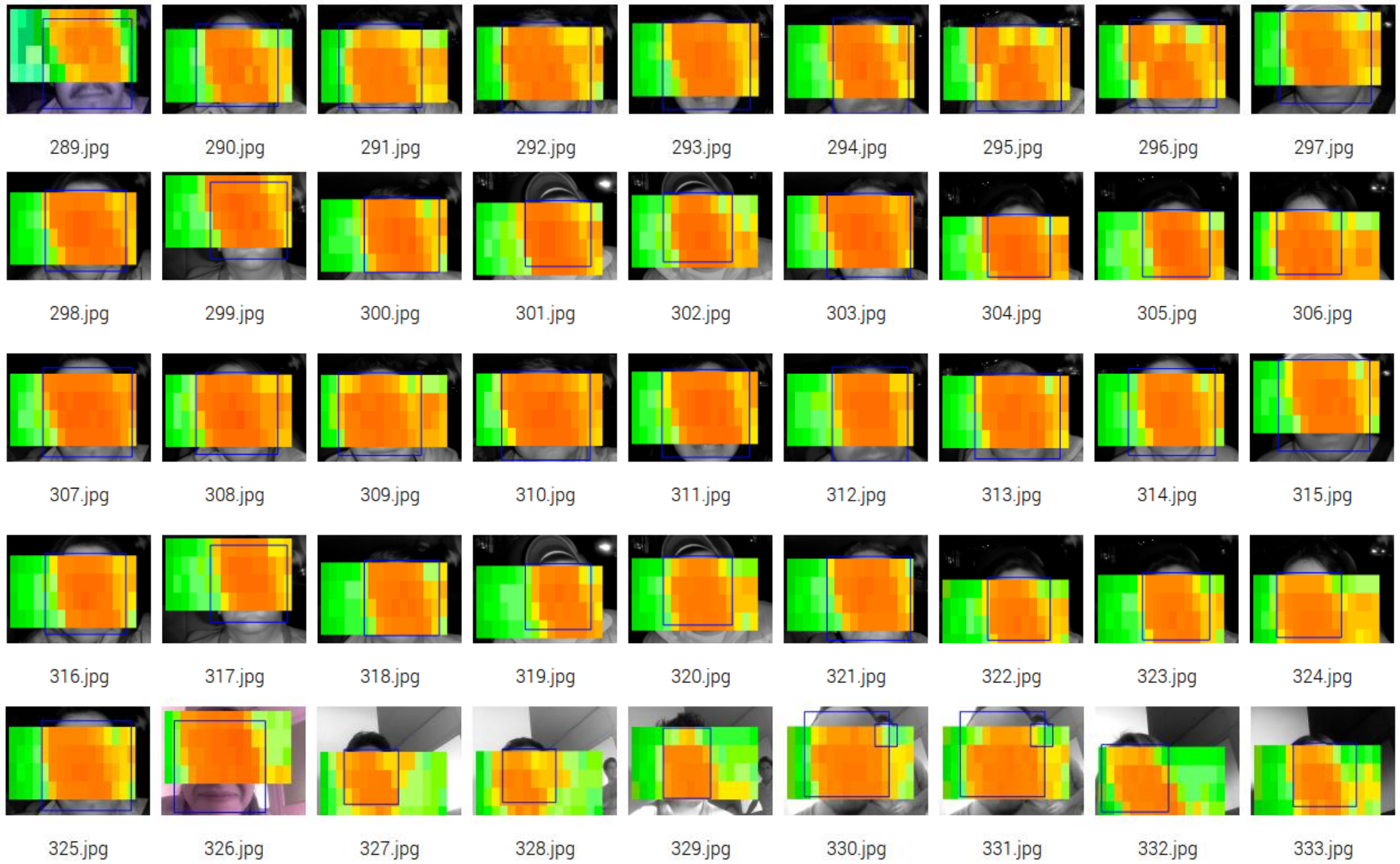
262.jpg 263.jpg 264.jpg 265.jpg 266.jpg 267.jpg 268.jpg 269.jpg 270.jpg

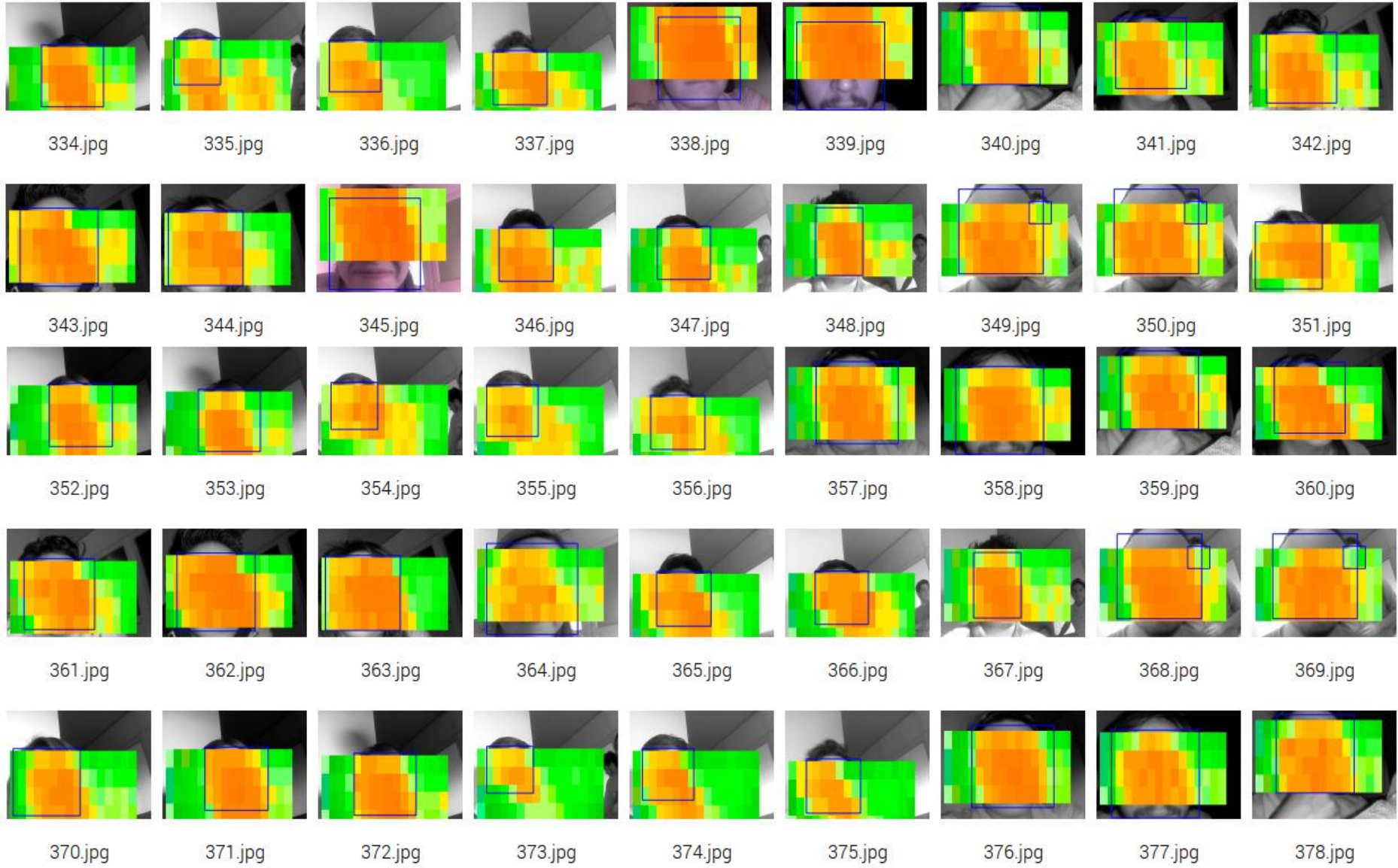


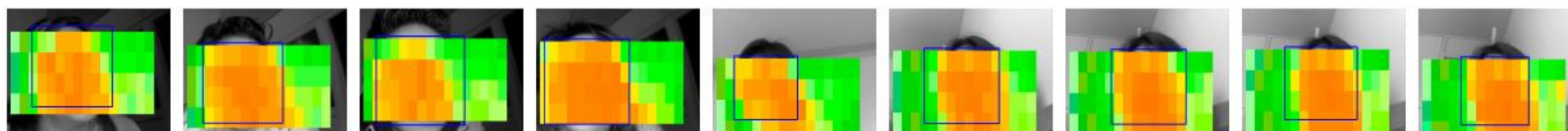
271.jpg 272.jpg 273.jpg 274.jpg 275.jpg 276.jpg 277.jpg 278.jpg 279.jpg



280.jpg 281.jpg 282.jpg 283.jpg 284.jpg 285.jpg 286.jpg 287.jpg 288.jpg







379.jpg

380.jpg

381.jpg

382.jpg

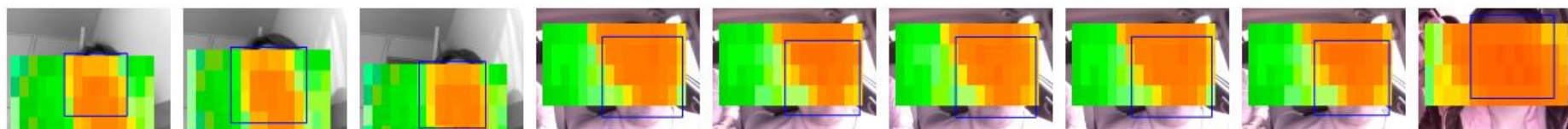
383.jpg

384.jpg

385.jpg

386.jpg

387.jpg



388.jpg

389.jpg

390.jpg

391.jpg

392.jpg

393.jpg

394.jpg

395.jpg

396.jpg



397.jpg

398.jpg

399.jpg

400.jpg

401.jpg

402.jpg

403.jpg

404.jpg

405.jpg



406.jpg

407.jpg

408.jpg

409.jpg

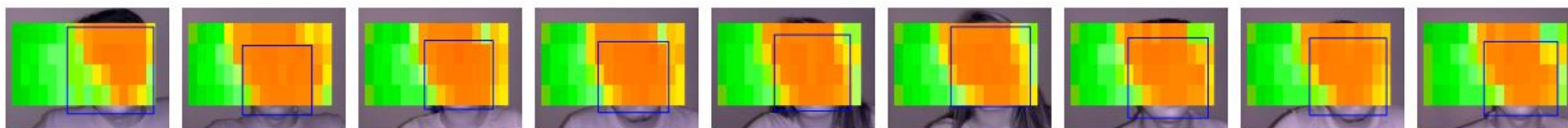
410.jpg

411.jpg

412.jpg

413.jpg

414.jpg



415.jpg

416.jpg

417.jpg

418.jpg

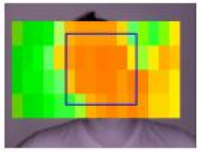
419.jpg

420.jpg

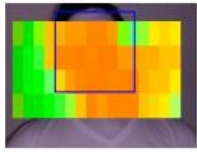
421.jpg

422.jpg

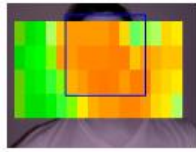
423.jpg



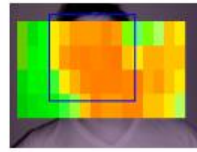
424.jpg



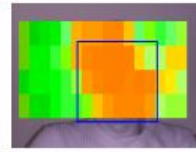
425.jpg



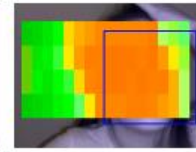
426.jpg



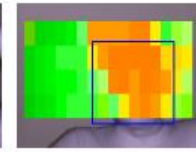
427.jpg



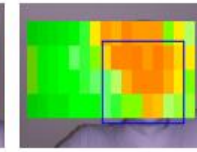
428.jpg



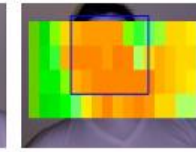
429.jpg



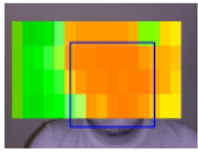
430.jpg



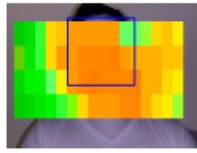
431.jpg



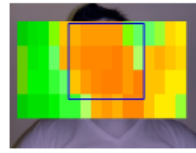
432.jpg



433.jpg



434.jpg



435.jpg

Anexo 15. Programación script para iniciar el sistema

```
num iniciar.py - /home/pi/RecoleccionDatos/iniciar.py (3.4.2)
File Edit Format Run Options Windows Help
1 import subprocess      #Modulo empleado para los mensajes de audio
2 import serial          #Modulo empleado para establecer la comunicacion con la plataforma ARDUINO
3 import time            #Modulo empleado para trabajar con fechas y/o horas
4 import cv2            #Modulo empleado para utilizar las funciones de vision por computador
5
6     #COMUNICACION SERIAL
7 ser = serial.Serial('/dev/ttyUSB0', 9600)      #Define la variable que identifica el puerto al cual esta conectado el Arduino UNOR3 y la velocidad
8 ser.close()      #Cerrar el puerto serial
9 ser.open()       #Abrir el puerto serial
10 time.sleep(1)   #Retardo de un segundo
11 ser.setDTR(level=0) #Reset manual del arduino para evitar lecturas de datos erroneos cada vez que el arduino abra el puerto serial
12 time.sleep(0.5) #Retardo de medio segundo
13 time.sleep(1)   #Retardo de un segundo
14 ser.write('K')  #Envia el caracter 'K' para recibir la cadena con el estado del rele que activa o desactiva la bomba de combustible
15 bomba = ser.readline() #Lee una linea enviada por el arduino
16 bomba = bomba.strip() #Compara los caracteres recibidos de inicio a fin#Obtiene el tamaño de la cadena
17 b = len(bomba)      #Obtiene el tamaño de la cadena
18 bomba = (int(bomba)) #Convierte a tipo entero la cadena recibida
19 print bomba        #Imprime la variable que identifica el estado de la bomba de combustible 1 ACTIVADA y 0 DESACTIVADA
20
21 cont = 0           #Variable para contar
22
23     #DEFINIR LAS DIRECCIONES DE LAS IMAGENES A MOSTRAR AL INICIAR EL SISTEMA
24 direccion = '/home/pi/RecoleccionDatos/MENSAJES/ENCENDIDO VEHICULO BLOQUEADO.jpg'
25 direccion1 = '/home/pi/RecoleccionDatos/MENSAJES/PRESIONE EL BOTON.jpg'
26 image = cv2.imread(direccion)      #Leer la imagen en la direccion establecida
27 cv2.imshow('VEHICULO BLOQUEADO', image) #Mostrar la imagen en la direccion establecida
28 key = cv2.waitKey(4000)            #Retardo de 4 segundos para mostrar la imagen
29
30 #Emplea el sinterizador de texto a voz Festival para leer la instruccion de que indica que el encendido del vehiculo esta bloqueado
31 VozInformacion = '"El encendido del automovil se encuentra bloqueado realice la prueba de alcoholemia para desbloquear"'
32 subprocess.call('echo '+VozInformacion+'|festival --tts --language spanish', shell=True)
33
34 image1 = cv2.imread(direccion1)      #Leer la imagen en la direccion establecida
35 cv2.imshow('PRESIONAR PULSADOR', image1) #Mostrar la imagen en la direccion establecida
36 key = cv2.waitKey(4000)            #Retardo de 4 segundos para mostrar la imagen
37
```

```
num iniciar.py - /home/pi/RecoleccionDatos/iniciar.py (3.4.2)
File Edit Format Run Options Windows Help
37
38 #Emplea el sinterizador de texto a voz Festival para leer la instruccion de que indica que debe precionar el boton para iniciar el sistema
39 VozIndicaciones = "Precione el boton para iniciar la prueba de alcoholemia"
40 subprocess.call('echo '+VozIndicaciones+'|festival --tts --language spanish', shell=True)
41 print "esperar boton"
42
43 #CICLO PARA RECOLECTAR LOS DATOS DE ENTRENAMIENTO
44 while True:
45     key = cv2.waitKey(1)          #Retardo de un segundo
46     inicio = ser.readline()      #Lee una linea enviada por el arduino
47     print inicio                 #imprime el tamaño de la cadena
48     inicio = inicio.strip()      #Compara los caracteres recibidos de inicio a fin
49     b = len(inicio)              #Obtiene el tamaño de la cadena
50     inicio = (int(inicio))       #Convierte a tipo entero la cadena recibida
51
52 #CONDICION PARA TOMAR LOS DATOS UNA VEZ PRECIONADO EL BOTON
53 if inicio == 1:
54     #Emplea el sinterizador de texto a voz Festival para leer la instruccion de que esta iniciando la prueba de alcoholemia
55     VozInicio = "Iniciando prueba de alcoholemia"
56     subprocess.call('echo '+VozInicio+'|festival --tts --language spanish', shell=True)
57
58     import datos                #Llama al modulo datos en el cual se encuentran las funciones para almacenar los datos de entrenamiento
59
60     cv2.destroyAllWindows()      #Cierra la ventana de visualizacion
61     break                       #Sale del ciclo
62
63     key = cv2.waitKey(1)          #Retardo de un segundo
64
65 #Condicion para salir del ciclo si se presiona la tecla 'q'
66 if key == ord("q"):
67     cv2.destroyAllWindows()      #Cierra la ventana de visualizacion
68     break                       #Sale del ciclo
69
Ln: 69 Col: 0
```

num datos.py - /home/pi/RecoleccionDatos/datos.py (3.4.2)

```
File Edit Format Run Options Windows Help
1 import time
2 time.sleep(2)
3 import subprocess
4 import csv
5 import matplotlib
6 import serial
7 import cv2
8 ser = serial.Serial('/dev/ttyUSB0', 9600)
9 ser.close()
10 ser.open()
11
12 matplotlib.use("Agg") #Added to plot graphs without running X server.
13 import matplotlib.pyplot as plt
14 import numpy as np
15 gas = 10
16
17 hora = time.strftime("%H:%M:%S") #Formato 24 Horas
18 print "Hora "+hora
19
20 fecha = time.strftime("%d/%m/%y")
21 print "Fecha "+fecha
22
23 vozTemperaturaFacial = "Mire al frente para proceder a medir la temperatura facial"
24 subprocess.call('echo '+vozTemperaturaFacial+'|festival --tts --language spanish', shell=True)
25 time.sleep(1)
26
27 from ReconocimientoFacial import CapturaConductor
28 time.sleep(1)
29
30
31 #time.sleep(1)
32 from TemperaturaFacial import Temperatura
33 tempMax = Temperatura.TempMax()
34 tempMin = Temperatura.TempMin()
35 print "Temp Facial Max "
36 print tempMax
37 print "Temp Facial Min "
38 print tempMin
39
40 from SensorGas import nivelalcohol
41 gas = nivelalcohol.ALCOHOL()
42 print gas
```

```
File Edit Format Run Options Windows Help
43 nivel=0
44
45
46 print nivel
47
48 from TemperaturaAmbiente import temperaturaAmbiente
49 temp_amb = temperaturaAmbiente.TempAmb()
50 print "Temperatura Ambiental"
51 print temp_amb
52 #time.sleep(1)
53
54
55 vozPupila = "Mirar fijamente a la camara para medir el diametro de la pupila"
56 subprocess.call('echo '+vozPupila+'|festival --tts --language spanish', shell=True)
57 time.sleep(1)
58
59
60 from DiametroPupilas import Ojos
61 radio = Ojos.RadioOjo()
62 print "Pupila "
63 print radio
64 #time.sleep(1)
65
66 direccion = '/home/pi/RecoleccionDatos/MENSAJES/CONDUCTOR SOBRIJO.jpg'
67 direccion1 = '/home/pi/RecoleccionDatos/MENSAJES/CONDUCTOR EBRIJO.jpg'
68
69 vozAlcohol = "Sople para medir su nivel de alcohol"
70 subprocess.call('echo '+vozAlcohol+'|festival --tts --language spanish', shell=True)
71
72 from SensorHumedad import Humedad
73 humedad = Humedad.Humedad()
74 print humedad
75 if humedad != 300:
76     from SensorGas import sensorGas
77     alcohol = sensorGas.Gas()
78     print alcohol
79
80     print "alcohol"
81     print alcohol
82 else:
83     vozNoSoplo = "Automovil bloqueado realice la prueba de alcoholemia correctamente"
84     subprocess.call('echo '+vozNoSoplo+'|festival --tts --lanuqaqe spanish', shell=True)
```


num datos.py - /home/pi/RecoleccionDatos/datos.py (3.4.2)

```
File Edit Format Run Options Windows Help
43 nivel=0
44 if gas >= 1.30:
45     nivel= gas-1.25
46     print nivel
47
48 from TemperaturaAmbiente import temperaturaAmbiente
49 temp_amb = temperaturaAmbiente.TempAmb()
50 print "Temperatura Ambiental"
51 print temp_amb
52 #time.sleep(1)
53
54
55 VozPupila = "Mirar fijamente a la camara para medir el diametro de la pupila"
56 subprocess.call('echo '+VozPupila+'|festival --tts --language spanish', shell=True)
57 time.sleep(1)
58
59
60 from DiametroPupilas import Ojos
61 radio = Ojos.RadioOjo()
62 print "Pupila "
63 print radio
64 #time.sleep(1)
65
66 direccion = '/home/pi/RecoleccionDatos/MENSAJES/CONDUCTOR SOBRIO.jpg'
67 direccion1 = '/home/pi/RecoleccionDatos/MENSAJES/CONDUCTOR EBRIIO.jpg'
68
69 VozAlcohol = "Sople para medir su nivel de alcohol"
70 subprocess.call('echo '+VozAlcohol+'|festival --tts --language spanish', shell=True)
71
72 from SensorHumedad import Humedad
73 humedad = Humedad.Humedad()
74 print humedad
75 if humedad != 300:
76     from SensorGas import sensorGas
77     alcohol = sensorGas.Gas()
78     print alcohol
79     alcohol = alcohol-nivel
80     print "alcohol"
81     print alcohol
82 else:
83     VozNoSoplo = "Automovil bloqueado realice la prueba de alcoholemia correctamente"
84     subprocess.call('echo '+VozNoSoplo+'|festival --tts --language spanish', shell=True)
```

```

85 #         time.sleep(1)
86
87
88 import IPython                #Modulo empleado para agregar funcionalidades extras al modo interactivo
89 import sklearn as sk         #Modulo para emplear las funciones de aprendizaje automatico
90 from sklearn import preprocessing #Biblioteca que permite cambiar el vector de las características en una representacion que sea mas adecuada para los estimadores
91 from sklearn import tree     #Biblioteca para emplear el clasificador de arbol
92 import numpy as np          #Modulo empleado para operar con vectores o matrices
93 import matplotlib           #Modulo empleado para realizar graficas en 2 dimensiones
94 import matplotlib.pyplot as plt #Modulo empleado para realizar graficas en 2 dimensiones con funcionalidades en el modo interactivo
95 import csv                  #Modulo empleado para la importacion o exportacion de hojas de calculo y bases de datos
96 import pyparsing           #Modulo para crear y ejecutar gramaticas simples
97 from sklearn.model_selection import train_test_split #Biblioteca de aprendizaje automatico para dividir los datos de entrenamiento en prueba y validacion
98 from sklearn.preprocessing import StandardScaler   #Biblioteca de aprendizaje automatico para calcular la media y desviacion estandar en un conjunto de entrenamiento
99 from sklearn.tree import DecisionTreeClassifier   #Biblioteca para emplear el clasificador de decision
100 from sklearn.tree import export_graphviz         #Biblioteca que permite exportar un arbol de decision en formato DOT
101 from graphviz import Digraph                   #Biblioteca para representar graficos en la que los nodos representan variables y las ramas representan conexiones
102 from graphviz import Source                    #Biblioteca para crear nuevos graficos de flujo
103 from graphviz import render                   #Biblioteca para leer graficos de arbol de decision en formato DOT
104 from graphviz import view                     #Biblioteca para ver los graficos
105
106         #carga el conjunto de datos en una matriz numpy
107 with open('/home/pi/RecoleccionDatos/AprendizajeAutomatico/datosEntrenamiento00K0K.csv', 'rb') as csvfile: #carga el conjunto de datos en una matriz numpy
108     datosEntrenamiento_reader = csv.reader(csvfile, delimiter=',')
109
110     #Encabezado que continene los nombres de cada atributo
111     row = datosEntrenamiento_reader.next()
112     Estado_Conductor = np.array(row)
113 #     print row[8]
114     #Cargar el conjunto de datos y las clases de destino
115     datosEntrenamiento_X, datosEntrenamiento_y = [], []
116     for row in datosEntrenamiento_reader:
117         datosEntrenamiento_X.append(row)
118         datosEntrenamiento_y.append(row[8]) #el atributo objetivo es 'Estado'
119     datosEntrenamiento_X = np.array(datosEntrenamiento_X)
120     datosEntrenamiento_y = np.array(datosEntrenamiento_y)
121
122 #print Estado_Conductor, datosEntrenamiento_X[0], datosEntrenamiento_y[0] # Imprimir los nombres de las clases y valores de la fila 1
123
124     # Para el estudio se analizan las clases Grado de alcohol, Temperatura Maxima y Radio de la Pupila
125     datosEntrenamiento_X = datosEntrenamiento_X[:, [3, 5, 7]] #Posiciones de las clases 3=Alcohol, 5=TempFacialMax y 7=RadioPupila
126     Estado_Conductor = Estado_Conductor[[3, 5, 7]]

```

```
File Edit Format Run Options Windows Help
127 Etiqueta_Estado = np.array(['0','1'])
128 #print Estado_Conductor #Imprimir los nombres de las clases.
129 #print datosEntrenamiento_X[0], datosEntrenamiento_y[0] #Imprimir los valores de las clases que se encuentran en la fila 2
130
131 # Convertir a valores numericos
132 datosEntrenamiento_x = datosEntrenamiento_x.astype(float)
133 datosEntrenamiento_y = datosEntrenamiento_y.astype(float)
134
135 #print 'Nuevo nombre para las características:',Estado_Conductor
136 #print 'Valores:',datosEntrenamiento_x[1]
137
138
139 #Dividir los datos en entrenamiento y prueba
140 #los datos necesarios para dividir la data son las características especificadas como datosEntrenamiento_x y las etiquetas definidas como datosEntrenamiento_y
141 X_train, X_test, y_train, y_test = train_test_split(datosEntrenamiento_x, datosEntrenamiento_y, test_size=0.20, random_state=33) #Separar la data en datos de entren
142
143 #print X_train.shape, y_train.shape #Muestra los elementos para entrenas y los elementos para pruebas de los datos de entrenamiento
144
145 #ARBOL DE DECISION
146 #arbol = tree.DecisionTreeClassifier(criterion='entropy', max_depth=3,min_samples_leaf=5)
147 arbol = tree.DecisionTreeClassifier(max_depth=7)
148 arbol = arbol.fit(X_train,y_train)
149
150 #verificar la exactitud de prediccion sobre los mismos datos de entrenamiento
151 #Si existe un 100% de exito esto manifiesta que existe sobreajuste
152
153 exactitudArbol = arbol.score(X_test, y_test)
154 print "////////Exactitud con el clasificador Arbol de Decision////////"
155 print exactitudArbol
156
157 exactitudArbol2 = arbol.score(X_train, y_train)
158 print "////////Exactitud con el clasificador Arbol de Decision2////////"
159 print exactitudArbol2
160
161 # PREDECIR UN DATO
162 datosPredecir = np.array([[alcohol,tempMax,radio]])
163 print "//////////DATOS A PREDECIR//////////"
164 print datosPredecir
165 prediction = arbol.predict(datosPredecir)
166 print "////////CLASE A LA QUE PERTENECE////////"
167 print prediction
168 predecir = prediction[0]
```

num datos.py - /home/pi/RecoleccionDatos/datos.py (3.4.2)

```
File Edit Format Run Options Windows Help
169 print predecir
170
171 if predecir == 0.0:
172     time.sleep(1)
173     ser.write('G')
174     estado_bomba = ser.readline()
175     print estado_bomba
176
177     image = cv2.imread(direccion)
178     cv2.imshow('CONDUCTOR SOBRIO', image)
179     key = cv2.waitKey(4000)
180     from ReconocimientoFacial import PruebaReconocimiento
181 if predecir == 1.0:
182     VozEstadoEbrio = "Conductor Ebrio por favor cambie de conductor"
183     subprocess.call('echo '+VozEstadoEbrio+'|festival --tts --language spanish', shell=True)
184     image1 = cv2.imread(direccion1)
185     cv2.imshow('CONDUCTOR EBRIO', image1)
186     key = cv2.waitKey(4000)
```

Ln: 176 Col: 0

Anexo 16. Programación preprocesamiento de los datos de entrenamiento

```
num preprocesamientoDatos.py - /home/pi/RecoleccionDatos/AprendizajeAutomatico/preprocesamientoDatos.py (3.4.2)
File Edit Format Run Options Windows Help
1 import csv #Modulo empleado para la importacion o exportacion de hojas de calculo y bases de datos
2 from sklearn import preprocessing #Modulo que permite cambiar el vector de las características en una representacion que sea mas adecu
3 import numpy as np #Modulo empleado para operar con vectores o matrices
4 from sklearn.model_selection import train_test_split #Modulo para representar la proporcion del conjunto de datos para incluir en la division de prueba
5
6 #carga el conjunto de datos en una matriz numpy
7 with open('/home/pi/RecoleccionDatos/AprendizajeAutomatico/datosEntrenamientoPruebas.csv', 'rb') as csvfile: #carga el conjunto de datos en una matriz nu
8     datosEntrenamiento = csv.reader(csvfile, delimiter=',') #Leer el archivo y limitar con la coma
9
10 # Encabezado que continene los nombres de cada atributo
11 filas = datosEntrenamiento.next()
12 Estado_Conductor = np.array(filas)
13
14 # Cargar el conjunto de datos y las clases de destino
15 datosEntrenamiento_X, datosEntrenamiento_y = [], []
16 for filas in datosEntrenamiento:
17     datosEntrenamiento_X.append(filas)
18     datosEntrenamiento_y.append(filas[8]) #el atributo objetivo es 'Estado'
19 datosEntrenamiento_X = np.array(datosEntrenamiento_X)
20 datosEntrenamiento_y = np.array(datosEntrenamiento_y)
21
22 #print "/////////Nombres de los atributos del conjunto de datos/////////"
23 #print Estado_Conductor #Imprime los nombres de los atributos del conjunto de datos
24 #print "/////////Atributos/////////"
25 #print datosEntrenamiento_X #Imprime los atributos del conjunto de datos
26 #print "/////////NOMBRES DE LAS CLASES/////////"
27 Etiqueta_Estado = np.array(['Estado']) #Vector que contiene el nombre de la clase estado del concuctor
28 #print Etiqueta_Estado
29 #print "/////////Clases/////////"
30 #print datosEntrenamiento_y #Imprime las clases del conjunto de datos
31
32 # Para el estudio se analian los atributos Grado de alcohol, Temperatura Maxima y Radio de la Pupila
33 datosEntrenamiento_X = datosEntrenamiento_X[:, [3, 5, 7]] #Posiciones de las clases 3=Alcohol, 5=TempFacialMax y 7=RadioPupila
```

num preprocesamientoDatos.py - /home/pi/RecoleccionDatos/AprendizajeAutomatico/preprocesamientoDatos.py (3.4.2)

```
File Edit Format Run Options Windows Help
34 Estado_Conductor = Estado_Conductor[[3, 5, 7]] #Vector para los atributos
35 Etiqueta_Conductor = np.array(['0','1']) #Vector con las clases que existen
36
37 # Convertir a valores numericos
38 datosEntrenamiento_X = datosEntrenamiento_X.astype(float)
39 datosEntrenamiento_y = datosEntrenamiento_y.astype(float)
40
41
42 #print "/////////Nombres de los atributos Seleccionados/////////"
43 #print Estado_Conductor #Imprimir los nombres de los atributos.
44 #print "/////////ATRIBUTOS SELECCIONADOS/////////"
45 #print datosEntrenamiento_X #Imprimir los valores de los atributos que se encuentran en la fila 2
46
47 # Convertir a valores numericos
48 datosEntrenamiento_X = datosEntrenamiento_X.astype(float)
49 datosEntrenamiento_y = datosEntrenamiento_y.astype(float)
50
51 #print 'Nuevo nombre para las características:', Estado_Conductor
52 #print 'Valores:', datosEntrenamiento_X[1]
53
54 #Dividir los datos en entrenamiento y prueba
55 #los datos necesarios para dividir la data son las características especificadas como datosEntrenamiento_X y las etiquetas definidas como datosEntrenamiento_y
56 X_train, X_test, y_train, y_test = train_test_split(datosEntrenamiento_X, datosEntrenamiento_y, test_size=0.20, random_state=33) #Separar la data en entrenamiento y prueba
57 print "/////////X_ENTRENAMIENTO/////////"
58 print X_train.shape
59 print "/////////y_ENTRENAMIENTO/////////"
60 print y_train.shape #Muestra los elementos para entrenas y los elementos para pruebas de los datos de entrenamiento
61
62 print "/////////X_PRUEBA/////////"
63 print X_test.shape
64 print "/////////y_PRUEBA/////////"
65 print y_test.shape #Muestra los elementos para entrenas y los elementos para pruebas de los datos de entrenamiento
66
--
```

Ln: 66 Col: 0

Anexo 17. Programación clasificador k-vecinos más cercanos

```
num clasificacionKNN.py - /home/pi/RecoleccionDatos/AprendizajeAutomatico/clasificacionKNN.py (3.4.2)
File Edit Format Run Options Windows Help
1 from mpl_toolkits.mplot3d import axes3d #Biblioteca para realizar graficas en tres dimensiones
2 import IPython #Modulo empleado para agregar funcionalidades extras al modo interactivo
3 import sklearn as sk #Modulo para emplear las funciones de aprendizaje automatico
4 from sklearn import preprocessing, neighbors #Biblioteca para emplear el clasificador de vecinos mas cercanos
5 import numpy as np #Modulo empleado para operar con vectores o matrices
6 import matplotlib #Modulo empleado para realizar graficas en 2 dimensiones
7 import matplotlib.pyplot as plt #Modulo empleado para realizar graficas en 2 dimensiones con funcionalidades en el modo interactivo
8 import csv #Modulo empleado para la importacion o exportacion de hojas de calculo y bases de datos
9 import pyparsing #Modulo para crear y ejecutar gramaticas simples
10 from sklearn.model_selection import train_test_split #Biblioteca de aprendizaje automatico para dividir los datos de entrenamiento en prueba y validacion
11 from sklearn.preprocessing import StandardScaler #Biblioteca de aprendizaje automatico para calcular la media y desviacion estandar en un conjunto de
12
13 #carga el conjunto de datos en una matriz numpy
14 with open('/home/pi/RecoleccionDatos/AprendizajeAutomatico/datosEntrenamientoPruebas.csv', 'rb') as csvfile: #carga el conjunto de datos en una matriz numpy
15     datosEntrenamiento_reader = csv.reader(csvfile, delimiter=',')
16
17 #Encabezado que continene los nombres de cada atributo
18 row = datosEntrenamiento_reader.next()
19 Estado_Conductor = np.array(row)
20 #print row[8]
21
22 #Cargar el conjunto de datos y las clases de destino
23 datosEntrenamiento_X, datosEntrenamiento_y = [], []
24 for row in datosEntrenamiento_reader:
25     datosEntrenamiento_X.append(row)
26     datosEntrenamiento_y.append(row[8]) #el atributo objetivo es 'Estado'
27 datosEntrenamiento_X = np.array(datosEntrenamiento_X)
28 datosEntrenamiento_y = np.array(datosEntrenamiento_y)
29
30 #print Estado_Conductor, datosEntrenamiento_X[0], datosEntrenamiento_y[0] # Imprimir los nombres de las clases y valores de la fila 1
31
32 # Para el estudio se analizan las clases Grado de alcohol, Temperatura Maxima y Radio de la Pupila
33 datosEntrenamiento_X = datosEntrenamiento_X[:, [3, 5, 7]] #Posiciones de las clases 3=Alcohol, 5=TempFacialMax y 7=RadioPupila
```

```
num clasificacionKNN.py - /home/pi/RecoleccionDatos/AprendizajeAutomatico/clasificacionKNN.py (3.4.2)
File Edit Format Run Options Windows Help
34 Estado_Conductor = Estado_Conductor[[3, 5, 7]]
35 Etiqueta_Conductor = np.array(['Sobrio','Ebrio'])
36 #print Estado_Conductor #Imprimir los nombres de las clases.
37 #print datosEntrenamiento_X[0], datosEntrenamiento_y[0] #Imprimir los valores de las clases que se encuentran en la fila 2
38
39 # Convertir a valores numericos
40 datosEntrenamiento_X = datosEntrenamiento_X.astype(float)
41 datosEntrenamiento_y = datosEntrenamiento_y.astype(float)
42
43 #print 'Nuevo nombre para las características:',Estado_Conductor
44 #print 'Valores:',datosEntrenamiento_X[1]
45
46 #Dividir los datos en entrenamiento y prueba
47 #los datos necesarios para dividir la data son las características especificadas como datosEntrenamiento_X y las etiquetas definidas como datosEntrenamiento_y
48 X_train, X_test, y_train, y_test = train_test_split(datosEntrenamiento_X, datosEntrenamiento_y, test_size=0.20, random_state=33) #Separar la data en entrenamiento y prueba
49 #print X_train.shape, y_train.shape #Muestra los elementos para entrenas y los elementos para pruebas de los datos de entrenamiento
50
51 #VECINOS CERCANOS considerando 7 vecinos mas cercanos por defecto N=4
52 knn = neighbors.KNeighborsClassifier(n_neighbors=7) #Cuando se le da un nuevo registro y se lo clasifica mediante K vecinos cercanos, al punto se le asigna la clase de los K vecinos
53 knn.fit(X_train, y_train) #Entrenar el algoritmo
54 exactitudVecinos = knn.score(X_test, y_test) #Porcentaje que indica que tan bien aprendio el algoritmo
55 #print "Exactitud con el clasificador Vecinos Cercanos"
56 #print exactitudVecinos
57
58 #PREDECIR UN DATO
59 datosPredecir = np.array([[1.61,27.33,9.5]])
60 print "//////////DATOS A PREDECIR//////////"
61 print datosPredecir
62 prediction = knn.predict(datosPredecir)
63 print "//////////CLASE A LA QUE PERTENECE//////////"
64 print prediction
65
66
```



```
File Edit Format Run Options Windows Help
67     #ESCALAR LAS CARACTERISTICAS
68     scaler = StandardScaler().fit(X_train)
69     X_train = scaler.transform(X_train)
70     X_test = scaler.transform(X_test)
71
72
73     #Graficar los atributos
74     fig = plt.figure()
75     ax1 = fig.add_subplot(111,projection='3d')
76
77     colors = ['red', 'greenyellow', 'yellow']
78     for i in xrange(len(colors)):
79         px = X_train[:, 0][y_train == i]
80         py = X_train[:, 1][y_train == i]
81         pz = X_train[:, 2][y_train == i]
82         plt.scatter(px, py, pz, c=colors[i])
83         ax1.plot_wireframe(px, py, pz)
84         ax1.scatter(px,py,pz, c=colors[i])
85     plt.legend(Etiqueta_Conductor)
86     ax1.set_xlabel('Alcohol')
87     ax1.set_ylabel('Temperatura Maxima')
88     ax1.set_zlabel('Diametro Pupila')
89     plt.show()
90
```

Anexo 18. Programación clasificador árbol de decisión

```
num arboldecision.py - /home/pi/RecoleccionDatos/AprendizajeAutomatico/arboldecision.py (3.4.2)
File Edit Format Run Options Windows Help
1 import IPython #Modulo empleado para agregar funcionalidades extras al modo interactivo
2 import sklearn as sk #Modulo para emplear las funciones de aprendizaje automatico
3 from sklearn import preprocessing #Biblioteca que permite cambiar el vector de las características en una representacion que sea mas adecuada para los
4 from sklearn import tree #Biblioteca para emplear el clasificador de arbol
5 import numpy as np #Modulo empleado para operar con vectores o matrices
6 import matplotlib #Modulo empleado para realizar graficas en 2 dimensiones
7 import matplotlib.pyplot as plt #Modulo empleado para realizar graficas en 2 dimensiones con funcionalidades en el modo interactivo
8 import csv #Modulo empleado para la importacion o exportacion de hojas de calculo y bases de datos
9 import parsing #Modulo para crear y ejecutar gramaticas simples
10 from sklearn.model_selection import train_test_split #Biblioteca de aprendizaje automatico para dividir los datos de entrenamiento en prueba y validacion
11 from sklearn.preprocessing import StandardScaler #Biblioteca de aprendizaje automatico para calcular la media y desviacion estandar en un conjunto de
12 from sklearn.tree import DecisionTreeClassifier #Biblioteca para emplear el clasificador de decision
13 from sklearn.tree import export_graphviz #Biblioteca que permite exportar un arbol de decision en formato DOT
14 from graphviz import Digraph #Biblioteca para representar graficos en la que los nodos representan variables y las ramas representan
15 from graphviz import Source #Biblioteca para crear nuevos graficos de flujo
16 from graphviz import render #Biblioteca para leer graficos de arbol de decision en formato DOT
17 from graphviz import view #Biblioteca para ver los graficos
18
19 #carga el conjunto de datos en una matriz numpy
20 with open('/home/pi/RecoleccionDatos/AprendizajeAutomatico/datosEntrenamientoPruebas.csv', 'rb') as csvfile: #carga el conjunto de datos en una matriz numpy
21     datosEntrenamiento_reader = csv.reader(csvfile, delimiter=',')
22
23     #Encabezado que contiene los nombres de cada atributo
24     row = datosEntrenamiento_reader.next()
25     Estado_Conductor = np.array(row)
26 # print row[8]
27     #Cargar el conjunto de datos y las clases de destino
28     datosEntrenamiento_X, datosEntrenamiento_y = [], []
29     for row in datosEntrenamiento_reader:
30         datosEntrenamiento_X.append(row)
31         datosEntrenamiento_y.append(row[8]) #el atributo objetivo es 'Estado'
32     datosEntrenamiento_X = np.array(datosEntrenamiento_X)
33     datosEntrenamiento_y = np.array(datosEntrenamiento_y)
34
```

```
File Edit Format Run Options Windows Help
35 #print Estado_Conductor, datosEntrenamiento_X[0], datosEntrenamiento_y[0] # Imprimir los nombres de las clases y valores de la fila 1
36
37     # Para el estudio se analizan las clases Grado de alcohol, Temperatura Maxima y Radio de la Pupila
38 datosEntrenamiento_X = datosEntrenamiento_X[:, [3, 5, 7]] #Posiciones de las clases 3=Alcohol, 5=TempFacialMax y 7=RadioPupila
39 Estado_Conductor = Estado_Conductor[[3, 5, 7]]
40 Etiqueta_Estado = np.array(['0','1'])
41 #print Estado_Conductor #Imprimir los nombres de las clases.
42 #print datosEntrenamiento_X[0], datosEntrenamiento_y[0] #Imprimir los valores de las clases que se encuentran en la fila 2
43
44     # Convertir a valores numericos
45 datosEntrenamiento_X = datosEntrenamiento_X.astype(float)
46 datosEntrenamiento_y = datosEntrenamiento_y.astype(float)
47
48 #print 'Nuevo nombre para las características:',Estado_Conductor
49 #print 'Valores:',datosEntrenamiento_X[1]
50
51
52     #Dividir los datos en entrenamiento y prueba
53 #los datos necesarios para dividir la data son las características especificadas como datosEntrenamiento_X y las etiquetas definidas como datosEntrenamiento_y
54 X_train, X_test, y_train, y_test = train_test_split(datosEntrenamiento_X, datosEntrenamiento_y, test_size=0.20, random_state=33) #Separar la data en (
55
56 #print X_train.shape, y_train.shape #Muestra los elementos para entrenas y los elementos para pruebas de los datos de entrenamiento
57
58     #ESCALAR LAS CARACTERISTICAS
59 scaler = StandardScaler().fit(X_train)
60 X_train = scaler.transform(X_train)
61 X_test = scaler.transform(X_test)
62
63     #ARBOL DE DECISION
64 #arbol = tree.DecisionTreeClassifier(criterion='entropy', max_depth=3,min_samples_leaf=5)
65 arbol = tree.DecisionTreeClassifier()
66 arbol = arbol.fit(X_train,y_train)
67
```

num arboldecision.py - /home/pi/RecoleccionDatos/AprendizajeAutomatico/arboldecision.py (3.4.2)

```
File Edit Format Run Options Windows Help
68         #Verificar la exactitud de prediccion sobre los mismos datos de entrenamiento
69 #Si existe un 100% de exito esto manifiesta que existe sobreajuste
70
71 exactitudArbol = arbol.score(X_test, y_test)
72 #print "////////Exactitud con el clasificador Arbol de Decision////////"
73 #print exactitudArbol
74
75 #GENERAR UN ARCHIVO .dot para almacenar la informacion del arbol de decisiones
76 export_graphviz(arbol, out_file='/home/pi/RecoleccionDatos/AprendizajeAutomatico/arbol.dot', feature_names=Estado_Conductor, class_names=Etiqueta_Estado, impur:
77 #tree.export_graphviz(arbol, out_file='/home/pi/RecoleccionDatos/AprendizajeAutomatico/tree.dot')
78 with open ('/home/pi/RecoleccionDatos/AprendizajeAutomatico/arbol.dot') as f:
79     dot_graph = f.read()
80
81 render('dot', 'png', '/home/pi/RecoleccionDatos/AprendizajeAutomatico/arbol.dot')           #Visualizar el grafico del arbol de decision
82
83         #Graficar la importancia de los atributos del arbol de decision
84 caract=datosEntrenamiento_X.shape[1]
85 plt.barh(range(caract), arbol.feature_importances_)
86 plt.yticks(np.arange(caract), Estado_Conductor)
87 plt.xlabel('Importancia de las características')
88 plt.ylabel('Características')
89 plt.show()
90
91
92
93 #     PREDECIR UN DATO
94 datosPredecir = np.array([[1.41, 23.33, 7.5]])
95 print "//////////DATOS A PREDECIR//////////"
96 print datosPredecir
97 prediction = arbol.predict(datosPredecir)
98 print "//////////CLASE A LA QUE PERTENECE//////////"
99 print prediction
100
```

Ln: 100 Col: 0

Anexo 19. Programación del script de Captura del Conductor

```
num CapturaConductor.py - /home/pi/ReconocimientoFacial/CapturaConductor.py (3.4.2)
File Edit Format Run Options Windows Help
1 #Agregar las librerias necesarias
2 from picamera.array import PiRGBArray #Biblioteca empleada para construir matrices n-dimensional numpy a partir de la salida de la camara
3 from picamera import PiCamera #Biblioteca empleada para hacer uso de las clases para interactuar con el modulo de la camara del Raspberry Pi
4 import cv2 #Modulo empleado para utilizar las funciones de vision por computador
5 import sys #Modulo empleado para utilizar la funcionalidad dependiente del sistema operativo
6 import numpy #Modulo empleado para operar con vectores o matrices
7 import os #Modulo empleado para utilizar la funcionalidad dependiente del sistema operativo
8 import time #Modulo empleado para trabajar con fechas y/o horas
9
10 camera = PiCamera() #Crea un objeto para llamar a la camara
11 camera.resolution = (1920, 1080) #Define la resolucioin
12 camera.framerate = 30 #Establece la velocidad de fotogramas en la que se ejecutan las capturas de imagen
13 rawCapture = PiRGBArray(camera, size=(1920,1080)) #Capturar video
14
15 def Capturar_Conductor():
16
17     display_window = cv2.namedWindow("Faces") #Crea una ventana para cargar la imagen mas tarde
18     size = 4 #Define el tamaño
19     face_cascade = '/home/pi/data/haarcascades/haarcascade_frontalface_alt.xml' #Cargar la plantilla Rostro
20     face_direccion = '/home/pi/ReconocimientoFacial/att_faces' #Direccion para almacenar los rostros
21     fn_name = "Conductor" #Nombre de la carpeta para el conductor
22
23     #CREA CARPETAS PARA ALMACENAR LOS RESULTADOS
24     path = os.path.join(face_direccion, fn_name) #Especifica el nombre de la carpeta y la direccion en la cual se crea la carpeta
25     if not os.path.isdir(path): #Comprueba que no exista la carpeta
26         os.mkdir(path) #Crea la carpeta
27     (im_width, im_height) = (112,92) #Variables que especifican el alto y ancho de la imagen
28     haar_cascade = cv2.CascadeClassifier(face_cascade) #Cargar plantilla de clasificacion
29     time.sleep(0.5) #Retardo de medio segundo
30     count = 0 #Variable que inicializa en conteo de los rostros
31
```

```
File Edit Format Run Options Windows Help
32 #INICIO DEL CICLO PARA ALMACENAR LOS ROSTROS
33 for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
34     im = frame.array #Almacena una imagen del video
35     im = cv2.flip(im, 1, 0) #Gira la imagen horizontalmente
36     gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY) #Convierte la imagen a escala de grises
37     mini = cv2.resize(gray, (gray.shape[1] / size, gray.shape[0] / size)) #Recorta la imagen con el alto y ancho establecidos
38     faces = haar_cascade.detectMultiScale(mini) #Busca las coordenadas del rostro y guarda su posicon
39     print "Encontrados"+str(len(faces)+"Rostros(s)" #Imprime el numero de rostros encontrados
40     for (x,y,w,h) in faces:
41         cv2.rectangle(im, (x,y), (x+w,y+h), (0,255,0),2) #Dibujar un rectangulo en el contorno
42
43     #Condicion para redimencion las coordenadas del rostro encontrado
44     faces = sorted(faces, key=lambda x: x[3])
45     if faces:
46         face_i = faces[0]
47         (x, y, w, h) = [v * size for v in face_i]
48         face = gray[y:y + h, x:x + w] #Variable que almacena la imagen en escala de grises
49         face_resize = cv2.resize(face, (im_width, im_height)) #Redimensionar el tamaño del ojo
50
51     #Asigna un numero para almacenar las imagenes con los resultados
52     pin=sorted([int(n[:n.find('.')]) for n in os.listdir(path)
53               if n[0]!='.' ][+0])[-1] + 1
54
55     #ALMACENA LAS IMAGENES DE LOS RESULTADOS EN LAS DIRECCIONES ESTABLECIDAS
56     cv2.imwrite('%s/%s.pgm' % (path, pin), face_resize)
57
58     cv2.rectangle(im, (x, y), (x + w, y + h), (0, 255, 0), 3) #Dibujar un rectangulo
59     cv2.putText(im, fn_name, (x - 10, y - 10), cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0)) #Escribir conductor en la imagen
60     count += 1 #Incrementar el numero de rostros almacenados
61     print count #Imprimir el numero de rostros
62
```

```
63     #VISUALIZACION DE LA PANTALLA
64     cv2.imshow("Faces", im)         #Mostrar la imagen
65     key = cv2.waitKey(1)           #Retardo de un segundo
66     rawCapture.truncate(0)         #Utilizar la salida para producir varias matrices
67
68     #Condicion para salir del ciclo si se presiona la tecla 'q'
69     if key == ord("q"):
70         camera.close()             #Cierra el puerto de la camara
71         cv2.destroyAllWindows()    #Cierra la ventana de visualizacion
72         break                       #Sale del ciclo
73
74     #Condicion para salir del ciclo si ya se almacenan los 100 rostros
75     if count == 100:
76         camera.close()             #Cierra el puerto de la camara
77         cv2.destroyAllWindows()    #Cierra la ventana de visualizacion
78         break                       #Sale del ciclo
79
80 #Llamar la funcion
81 Capturar_Conductor()
82
83
```

Ln: 83 Col: 0

Anexo 20. Programación del script de Reconocimiento del Conductor

```
nur PruebaReconocimiento.py - /home/pi/RecoleccionDatos/ReconocimientoFacial/PruebaReconocimiento.py (3.4.2)
File Edit Format Run Options Windows Help
1 #Agregar las librerias necesarias
2 from picamera.array import PiRGBArray #Biblioteca empleada para construir matrices n-dimensional numpy a partir de la salida de la camara
3 from picamera import PiCamera #Biblioteca empleada para hacer uso de las clases para interactuar con el modulo de la camara del Raspberry Pi
4 import cv2 #Modulo empleado para utilizar las funciones de vision por computador
5 import cv2.face #Modulo empleado para utilizar las funciones de recocimiento facial de vision por computador
6 import numpy #Modulo empleado para operar con vectores o matrices
7 import serial #Modulo empleado para la comunicacion a traves del puerto serial
8 import os #Modulo empleado para utilizar la funcionalidad dependiente del sistema operativo
9 import time #Modulo empleado para trabajar con fechas y/o horas
10 import subprocess #Modulo empleado para abrir subprocessos del sistema operativo
11
12 ser = serial.Serial('/dev/ttyUSB0', 9600) #Define la variable que identifica el puerto al cual esta conectado el Arduino UNOR3 y la velocidad
13 ser.close() #Cierra la variable para la comunicacion serial
14 ser.open() #Abre la variable para iniciar la comunicacion serial con los parametros establecidos
15 print "iniciando"
16 time.sleep(1) #Retardo de un segundo
17 ser.setDTR(level=0) #Reset manual del arduino para evitar lectura de datos erroneos cada vez que el arduino a$
18 time.sleep(0.5) #Retardo de medio segundo
19 print "envia"
20 time.sleep(1) #Retardo de un segundo
21 ser.write('G') #Enviar el caracter 'G' para activar la bomba de combustible
22 estado_bomba = ser.readline() #Lee una linea enviada por el arduino
23 print estado_bomba
24 #Mensaje auditivo indicando que ya puede encender el automovil
25 VozEstadoSobrio = "Conductor Sobrio ya puede encender el automovil"
26 subprocess.call('echo '+VozEstadoSobrio+'|festival --tts --language spanish', shell=True)
27 Advertencia = "Advertencia en caso de que se detecte alcohol en el aire se procedera a comunicar al ECU novecientos once y el automovil sera apagado"
28 subprocess.call('echo '+Advertencia+'|festival --tts --language spanish', shell=True)
29
30 #Definir variables empleadas en el script
31 numeroveces = 0 #variable que cuenta el numero de ciclos recorridos
32 cerrar = 0 #variable que cuenta antes de cerrar el script
33 sinrostro = 0 #variable que cuenta el numero de veces que no se ha detectado ningun rostro
34 intentosrostro = 0 #variable que cuenta los rostros desconocidos
35 nombre = "" #variable para almacenar el nombre del conductor
36 size = 4 #tamano de la imagen
37 suplantar = 0 #variable que cuenta el numero de rostros desconocidos
38 apagar = 0 #variable para apagar el vehiculo en caso de que el conductor ha sido suplantado
39
40 camera = PiCamera() #Crea un objeto para llamar a la camara
```



```

File Edit Format Run Options Windows Help
41 camera.resolution = (620, 440) #Define la resolucio
42 camera.framerate = 30 #Establece la velocidad de fotogramas en la que se ejecutan las capturas de imagen
43 rawCapture = PiRGBArray(camera, size=(620, 440)) #Capturar video
44 display_window = cv2.namedWindow("Faces") #Crea una ventana para cargar la imagen mas tarde
45
46 fn_haar = '/home/pi/data/haarcascades/haarcascade_frontalface_alt.xml' #Cargar la plantilla Rostro
47 fn_dir = '/home/pi/RecoleccionDatos/ReconocimientoFacial/att_faces' #Direccion en la que se almacenaron los rostros
48
49 # Part 1: Creando fisherRecognizer
50 print('Formando...')
51 # Crear una lista de imagenes y una lista de nombres correspondientes
52 (images, lables, names, id) = ([], [], {}, 0)
53 for (subdirs, dirs, files) in os.walk(fn_dir):
54     for subdir in dirs:
55         names[id] = subdir
56         subjectpath = os.path.join(fn_dir, subdir)
57         for filename in os.listdir(subjectpath):
58             path = subjectpath + '/' + filename
59             lable = id
60             images.append(cv2.imread(path, 0))
61             lables.append(int(lable))
62         id += 1
63 (im_width, im_height) = (112, 92)
64
65 # Crear una matriz Numpy de las dos listas anteriores
66 (images, lables) = [numpy.array(lis) for lis in [images, lables]]
67
68 # OpenCV entrena un modelo a partir de las imagenes
69 #model = cv2.face.createLBPHFaceRecognizer()
70 model = cv2.face.createFisherFaceRecognizer()
71 model.train(images, lables)
72
73 # Part 2: Utilizar fisherRecognizer en funcionamiento la camara
74 haar_cascade = cv2.CascadeClassifier(fn_haar)
75
76 time.sleep(0.5)
77
78 for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
79     # count = 0
80     key = cv2.waitKey(1) #Estable un retardo de un segundo

```

```

File Edit Format Run Options Windows Help
81 rawCapture.truncate(0)
82 frame = frame.array
83 frame=cv2.flip(frame,1,0)
84 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
85 mini = cv2.resize(gray, (gray.shape[1] / size, gray.shape[0] / size))
86 faces = haar_cascade.detectMultiScale(mini)
87
88 for i in range(len(faces)):
89     face_i = faces[i]
90     (x, y, w, h) = [v * size for v in face_i]
91     face = gray[y:y + h, x:x + w]
92     face_resize = cv2.resize(face, (im_width, im_height))
93
94     # Intentado reconocer la cara
95     prediction = model.predict(face_resize)
96     cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 3)
97
98 # Escribiendo el nombre de la cara reconocida
99 # [1]
100     if prediction[1]<500:
101         if names[prediction[0]] != "Conductor Designado":
102             #apagar = 0
103             font = cv2.FONT_HERSHEY_SIMPLEX
104             cv2.putText(frame,'DESCONOCIDO',(x-10, y-10), font,1,(0, 255, 0))
105             print names[prediction[0]]
106             #Emplea el sinterizador de texto a voz Festival para leer la instruccion de que indica que el conductor es suplantado
107             # VozSuplantacion = "Mire al frente para realizar el reconocimiento facial"
108             # subprocess.call('echo '+vozSuplantacion+'|festival --tts --language spanish', shell=True)
109             suplantar = suplantar + 1
110             numeroveces = numeroveces + 1
111             intentosrostro = 0
112             sinrostro = 0
113             #
114             #
115             #
116             #
117             #
118             #
119             #
120             #
121             #
122             #
123             #
124             #
125             #
126             #
127             #
128             #
129             #
130             #
131             #
132             #
133             #
134             #
135             #
136             #
137             #
138             #
139             #
140             #
141             #
142             #
143             #
144             #
145             #
146             #
147             #
148             #
149             #
150             #
151             #
152             #
153             #
154             #
155             #
156             #
157             #
158             #
159             #
160             #
161             #
162             #
163             #
164             #
165             #
166             #
167             #
168             #
169             #
170             #
171             #
172             #
173             #
174             #
175             #
176             #
177             #
178             #
179             #
180             #
181             #
182             #
183             #
184             #
185             #
186             #
187             #
188             #
189             #
190             #
191             #
192             #
193             #
194             #
195             #
196             #
197             #
198             #
199             #
200             #
201             #
202             #
203             #
204             #
205             #
206             #
207             #
208             #
209             #
210             #
211             #
212             #
213             #
214             #
215             #
216             #
217             #
218             #
219             #
220             #
221             #
222             #
223             #
224             #
225             #
226             #
227             #
228             #
229             #
230             #
231             #
232             #
233             #
234             #
235             #
236             #
237             #
238             #
239             #
240             #
241             #
242             #
243             #
244             #
245             #
246             #
247             #
248             #
249             #
250             #
251             #
252             #
253             #
254             #
255             #
256             #
257             #
258             #
259             #
260             #
261             #
262             #
263             #
264             #
265             #
266             #
267             #
268             #
269             #
270             #
271             #
272             #
273             #
274             #
275             #
276             #
277             #
278             #
279             #
280             #
281             #
282             #
283             #
284             #
285             #
286             #
287             #
288             #
289             #
290             #
291             #
292             #
293             #
294             #
295             #
296             #
297             #
298             #
299             #
300             #
301             #
302             #
303             #
304             #
305             #
306             #
307             #
308             #
309             #
310             #
311             #
312             #
313             #
314             #
315             #
316             #
317             #
318             #
319             #
320             #
321             #
322             #
323             #
324             #
325             #
326             #
327             #
328             #
329             #
330             #
331             #
332             #
333             #
334             #
335             #
336             #
337             #
338             #
339             #
340             #
341             #
342             #
343             #
344             #
345             #
346             #
347             #
348             #
349             #
350             #
351             #
352             #
353             #
354             #
355             #
356             #
357             #
358             #
359             #
360             #
361             #
362             #
363             #
364             #
365             #
366             #
367             #
368             #
369             #
370             #
371             #
372             #
373             #
374             #
375             #
376             #
377             #
378             #
379             #
380             #
381             #
382             #
383             #
384             #
385             #
386             #
387             #
388             #
389             #
390             #
391             #
392             #
393             #
394             #
395             #
396             #
397             #
398             #
399             #
400             #
401             #
402             #
403             #
404             #
405             #
406             #
407             #
408             #
409             #
410             #
411             #
412             #
413             #
414             #
415             #
416             #
417             #
418             #
419             #
420             #
421             #
422             #
423             #
424             #
425             #
426             #
427             #
428             #
429             #
430             #
431             #
432             #
433             #
434             #
435             #
436             #
437             #
438             #
439             #
440             #
441             #
442             #
443             #
444             #
445             #
446             #
447             #
448             #
449             #
450             #
451             #
452             #
453             #
454             #
455             #
456             #
457             #
458             #
459             #
460             #
461             #
462             #
463             #
464             #
465             #
466             #
467             #
468             #
469             #
470             #
471             #
472             #
473             #
474             #
475             #
476             #
477             #
478             #
479             #
480             #
481             #
482             #
483             #
484             #
485             #
486             #
487             #
488             #
489             #
490             #
491             #
492             #
493             #
494             #
495             #
496             #
497             #
498             #
499             #
500             #
501             #
502             #
503             #
504             #
505             #
506             #
507             #
508             #
509             #
510             #
511             #
512             #
513             #
514             #
515             #
516             #
517             #
518             #
519             #
520             #
521             #
522             #
523             #
524             #
525             #
526             #
527             #
528             #
529             #
530             #
531             #
532             #
533             #
534             #
535             #
536             #
537             #
538             #
539             #
540             #
541             #
542             #
543             #
544             #
545             #
546             #
547             #
548             #
549             #
550             #
551             #
552             #
553             #
554             #
555             #
556             #
557             #
558             #
559             #
560             #
561             #
562             #
563             #
564             #
565             #
566             #
567             #
568             #
569             #
570             #
571             #
572             #
573             #
574             #
575             #
576             #
577             #
578             #
579             #
580             #
581             #
582             #
583             #
584             #
585             #
586             #
587             #
588             #
589             #
590             #
591             #
592             #
593             #
594             #
595             #
596             #
597             #
598             #
599             #
600             #
601             #
602             #
603             #
604             #
605             #
606             #
607             #
608             #
609             #
610             #
611             #
612             #
613             #
614             #
615             #
616             #
617             #
618             #
619             #
620             #
621             #
622             #
623             #
624             #
625             #
626             #
627             #
628             #
629             #
630             #
631             #
632             #
633             #
634             #
635             #
636             #
637             #
638             #
639             #
640             #
641             #
642             #
643             #
644             #
645             #
646             #
647             #
648             #
649             #
650             #
651             #
652             #
653             #
654             #
655             #
656             #
657             #
658             #
659             #
660             #
661             #
662             #
663             #
664             #
665             #
666             #
667             #
668             #
669             #
670             #
671             #
672             #
673             #
674             #
675             #
676             #
677             #
678             #
679             #
680             #
681             #
682             #
683             #
684             #
685             #
686             #
687             #
688             #
689             #
690             #
691             #
692             #
693             #
694             #
695             #
696             #
697             #
698             #
699             #
700             #
701             #
702             #
703             #
704             #
705             #
706             #
707             #
708             #
709             #
710             #
711             #
712             #
713             #
714             #
715             #
716             #
717             #
718             #
719             #
720             #
721             #
722             #
723             #
724             #
725             #
726             #
727             #
728             #
729             #
730             #
731             #
732             #
733             #
734             #
735             #
736             #
737             #
738             #
739             #
740             #
741             #
742             #
743             #
744             #
745             #
746             #
747             #
748             #
749             #
750             #
751             #
752             #
753             #
754             #
755             #
756             #
757             #
758             #
759             #
760             #
761             #
762             #
763             #
764             #
765             #
766             #
767             #
768             #
769             #
770             #
771             #
772             #
773             #
774             #
775             #
776             #
777             #
778             #
779             #
780             #
781             #
782             #
783             #
784             #
785             #
786             #
787             #
788             #
789             #
790             #
791             #
792             #
793             #
794             #
795             #
796             #
797             #
798             #
799             #
800             #
801             #
802             #
803             #
804             #
805             #
806             #
807             #
808             #
809             #
810             #
811             #
812             #
813             #
814             #
815             #
816             #
817             #
818             #
819             #
820             #
821             #
822             #
823             #
824             #
825             #
826             #
827             #
828             #
829             #
830             #
831             #
832             #
833             #
834             #
835             #
836             #
837             #
838             #
839             #
840             #
841             #
842             #
843             #
844             #
845             #
846             #
847             #
848             #
849             #
850             #
851             #
852             #
853             #
854             #
855             #
856             #
857             #
858             #
859             #
860             #
861             #
862             #
863             #
864             #
865             #
866             #
867             #
868             #
869             #
870             #
871             #
872             #
873             #
874             #
875             #
876             #
877             #
878             #
879             #
880             #
881             #
882             #
883             #
884             #
885             #
886             #
887             #
888             #
889             #
890             #
891             #
892             #
893             #
894             #
895             #
896             #
897             #
898             #
899             #
900             #
901             #
902             #
903             #
904             #
905             #
906             #
907             #
908             #
909             #
910             #
911             #
912             #
913             #
914             #
915             #
916             #
917             #
918             #
919             #
920             #
921             #
922             #
923             #
924             #
925             #
926             #
927             #
928             #
929             #
930             #
931             #
932             #
933             #
934             #
935             #
936             #
937             #
938             #
939             #
940             #
941             #
942             #
943             #
944             #
945             #
946             #
947             #
948             #
949             #
950             #
951             #
952             #
953             #
954             #
955             #
956             #
957             #
958             #
959             #
960             #
961             #
962             #
963             #
964             #
965             #
966             #
967             #
968             #
969             #
970             #
971             #
972             #
973             #
974             #
975             #
976             #
977             #
978             #
979             #
980             #
981             #
982             #
983             #
984             #
985             #
986             #
987             #
988             #
989             #
990             #
991             #
992             #
993             #
994             #
995             #
996             #
997             #
998             #
999             #
1000            #

```

num *PruebaReconocimiento.py - /home/pi/RecoleccionDatos/ReconocimientoFacial/PruebaReconocimiento.py (3.4.2)*

```
File Edit Format Run Options Windows Help
121             intentosrostro = 0
122             sinrostro = 0
123             apagar = 0
124         else:
125             font = cv2.FONT_HERSHEY_SIMPLEX
126             cv2.putText(frame, 'Desconocido', (x-10, y-10), font,1,(0, 255, 0))
127 #Emplea el sinterizador de texto a voz Festival para leer la instruccion de que indica que el ence
128 #     vozSuplantacion = 'Mire al frente para realizar el reconocimiento facial'
129 #     subprocess.call('echo '+vozSuplantacion+'|festival --tts --language spanish', shell=True)
130             suplantar = suplantar + 1
131             numeroveces = numeroveces + 1
132             intentosrostro = 0
133             sinrostro = 0
134         print suplantar
135         print numeroveces
136         if numeroveces == 10:
137             time.sleep(1)
138             ser.write('R')
139             print "enviado"
140             inicio = ser.readline()           #Lee una linea enviada por el arduino
141             print inicio                     #imprime el tamaño de la cadena
142             print "recibido"
143             inicio = inicio.strip()          #Compara los caracteres recibidos de inicio a fin
144             b = len(inicio)                 #Obtiene el tamaño de la cadena
145             inicio = (int(inicio))         #Convierte a tipo entero la cadena recibida
146             numeroveces = 0
147             time.sleep(1)
148         #CONDICION PARA TOMAR LOS DATOS UNA VEZ PRECIONADO EL BOTON
149         if inicio == 9:
150 #             Emplea el sinterizador de texto a voz Festival para leer la instruccion de que esta iniciando la pru$
151             vozAlcoholAmbiente = "Se ha detectado alcohol en el ambiente se procedera a comunicar al ECU novecientos once y se ap
152             subprocess.call('echo '+vozAlcoholAmbiente+'|festival --tts --language spanish', shell=True)
153             time.sleep(1)
154             time.sleep(1)
155             ser.write('Z')
156             cerrar = 1
157
158         sinrostro = sinrostro + 1
159         print sinrostro
160         print intentosrostro
```

```

File Edit Format Run Options Windows Help
161     if sinrostro == 100:
162         intentosrostro = intentosrostro + 1
163         intentos = "No se ha detectado el rostro mire al frente"
164         subprocess.call('echo '+intentos+'|festival --tts --language spanish', shell=True)
165         sinrostro = 0
166     if intentosrostro == 3:
167         Vozsinrostro = "Imposible detectar el rostro se procedera a apagar el automovil"
168         subprocess.call('echo '+Vozsinrostro+'|festival --tts --language spanish', shell=True)
169         cerrar = 1
170     print "apagar"
171     print apagar
172     if apagar == 5:
173         ConductorSuplantado = "El conductor ha sido suplantado el automovil se apagara en pocos segundos"
174         subprocess.call('echo '+ConductorSuplantado+'|festival --tts --language spanish', shell=True)
175         ser.write('F')
176         estado_bomba = ser.readline()
177         print estado_bomba
178         camera.close()
179         cv2.destroyAllWindows()
180         break
181
182     #DISPLAY TO WINDOW
183     cv2.imshow("Faces", frame)
184     key = cv2.waitKey(1)
185
186     rawCapture.truncate(0)
187
188     # if the `q` key was pressed, break from the loop
189     if key == ord("q"):
190         camera.close()
191         cv2.destroyAllWindows()
192         break
193     if cerrar == 1:
194         time.sleep(1)
195         ser.write('F')
196         camera.close()
197         cv2.destroyAllWindows()
198         break
199     if suplantar == 6:
200         VozSuplantacion = "Mire al frente para realizar el reconocimiento facial"

```

num *PruebaReconocimiento.py - /home/pi/RecoleccionDatos/ReconocimientoFacial/PruebaReconocimiento.py (3.4.2)*

```
File Edit Format Run Options Windows Help
161     if sinrostro == 100:
162         intentosrostro = intentosrostro + 1
163         intentos = "No se ha detectado el rostro mire al frente"
164         subprocess.call('echo '+intentos+'|festival --tts --language spanish', shell=True)
165         sinrostro = 0
166     if intentosrostro == 3:
167         vozsinrostro = "Imposible detectar el rostro se procedera a apagar el automovil"
168         subprocess.call('echo '+vozsinrostro+'|festival --tts --language spanish', shell=True)
169         cerrar = 1
170     print "apagar"
171     print apagar
172     if apagar == 5:
173         ConductorSuplantado = "El conductor ha sido suplantado el automovil se apagara en pocos segundos"
174         subprocess.call('echo '+ConductorSuplantado+'|festival --tts --language spanish', shell=True)
175         ser.write('F')
176         estado_bomba = ser.readline()
177         print estado_bomba
178         camera.close()
179         cv2.destroyAllWindows()
180         break
181
182     #DISPLAY TO WINDOW
183     cv2.imshow("Faces", frame)
184     key = cv2.waitKey(1)
185
186     rawCapture.truncate(0)
187
188     # if the `q` key was pressed, break from the loop
189     if key == ord("q"):
190         camera.close()
191         cv2.destroyAllWindows()
192         break
193     if cerrar == 1:
194         time.sleep(1)
195         ser.write('F')
196         camera.close()
197         cv2.destroyAllWindows()
198         break
199     if suplantar == 6:
200         vozsuplantacion = "Mire al frente para realizar el reconocimiento facial"
201         subprocess.call('echo '+vozsuplantacion+'|festival --tts --language spanish', shell=True)
202     #     ConductorSuplantado = "El conductor ha sido suplantado el automovil se apaqara en pocos segundos"
```

Ln: 202 Col: 102

Anexo 21. Programación del script recibir dato de Humedad de la placa Arduino

```
num Humedad.py - /home/pi/RecoleccionDatos/SensorHumedad/Humedad.py (3.4.2)
File Edit Format Run Options Windows Help
1 import subprocess #Modulo empleado para los mensajes de audio
2 import serial #Modulo empleado para establecer la comunicacion con la plataforma ARDUINO
3 import cv2 #Modulo empleado para utilizar las funciones de vision por computador
4 import time #Modulo empleado para trabajar con fechas y/o horas
5
6 ser = serial.Serial('/dev/ttyUSB0', 9600) #Define la variable que identifica el puerto al cual esta conectado el Arduino UNOR3 y la velocidad
7
8 ser.close() #Cierra la variable para la comunicacion serial
9 ser.open() #Abre la variable para iniciar la comunicacion serial con los parametros establecidos
10
11 time.sleep(1) #Retardo de un segundo
12 ser.setDTR(level=0) #Reset manual del arduino para evitar lectura de datos erroneos cada vez que el arduino abra el puerto serial
13 time.sleep(0.5) #Retardo de medio segundo
14 cont = 0 #Variable que cuenta el numero de veces que se ha terminado el ciclo
15
16 #Iniciar un ciclo hasta recibir el valor de la Humedad
17 while True:
18     time.sleep(3) #Retardo de 3 segundos
19     ser.write('D') #Enviar el caracter 'D' para recibir el dato de la Humedad de la placa Arduino
20     time.sleep(2) #Retardo de 2 segundos
21     humedad = ser.readline() #Lee una linea enviada por el arduino
22     humedad = humedad.strip() #Compara los caracteres de inicio y fin
23     b = len(humedad) #Obtener el tamaño de la cadena
24     humedad = (float(humedad)) #Convierte la variable string a tipo float
25     #print humedad #Escribir el valor de la humedad
26
27     #Condicion para conocer si el conductor ha soplado
28     if humedad <= 74.0:
29         VozHumedad = "Por favor Sople Nuevamente"
30         subprocess.call('echo '+VozHumedad+'|festival --tts --language spanish', shell=True)
31         cont = cont + 1
32     if humedad > 75.0:
33         def Humedad():
34             return humedad
35         cv2.destroyAllWindows()
36         break
37     if cont == 3:
38         VozNoSoplo = "Imposible medir el nivel de alcohol"
39         subprocess.call('echo '+VozNoSoplo+'|festival --tts --language spanish', shell=True)
40         humedad = 300
41         cont = 0
42
```

Ln: 42 Col: 16

Anexo 22. Oficio con la Autorización para activar el botón de pánico



Oficio No. 003-TS-ANT-ECU911T-2018

Tulcán; 09 de Abril de 2018

Asunto: Atención Oficio No. 71 Cooperativa de Transportes Atahualpa.

Señor

Sr. Álvaro Cano

GERENTE COOPERATIVA DE TAXIS "ATAHUALPA"

De mi consideración:

Mediante el presente reciban un cordial y atento saludo por parte de Transporte Seguro - Agencia Nacional de Tránsito, a la vez desear el mejor de los éxitos en las funciones encomendadas.

En atención al oficio No. 071 de Cooperativa de Taxis "Atahualpa" suscrito por su distinguida autoridad, el cual dice textualmente "solicitarle muy comedidamente autorice a la Señorita Liceth Geovana Portilla Arteaga portadora de la C.I. 0401724125, estudiante del décimo semestre de la carrera de Ingeniería en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte para activar el botón de auxilio implementado por el ECU911 en el taxi N. 029 con placas CBO0627 de la institución, con la finalidad de dar alerta al ECU911 cuando el conductor se encuentre en estado de ebriedad, a través del trabajo de titulación denominado DETECTOR DE ALCOHOLEMIA PARA CONDUCTORES QUE ANALIZA VARIABLES FACIALES Y AMBIENTALES DEL AUTOMOVIL MEDIANTE EL APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA REDUCCIÓN DE ACCIDENTES DE TRÁNSITO [...]"

Por tal razón, tengo a bien informar que una vez realizado la inspección física del sistema "Detector de Alcoholemia para Conductores" el cual no afecta el sistema de kit de seguridad de Proyecto Transporte Seguro, se autoriza realizar la prueba de activación de botón de pánico por el presunto conductor en estado embriaguez, a fin de evidenciar lo requerido en coordinación con las instituciones de control articuladas a SIS ECU911. Actividad se la realizara de manera oportuna previa notificación.

Saludos cordiales



Ing. Danny Ontaneda
**SUPERVISOR DE MONITOREO
PROYECTO TRANSPORTE SEGURO
AGENCIA NACIONAL DE TRÁNSITO**

Proyecto Transporte Seguro – ECU911
Av. Veintimilla y Alejandro R. Mera
0988823903
Tulcán - Ecuador

Anexo 23. Autorización para realizar las pruebas de funcionamiento del sistema.



COOPERATIVA DE TRANSPORTES "ATAHUALPA"

Filial de la Federación Nacional de Cooperativas de Transporte de Taxis
Dirección: Edificio Atahualpa, Av. Veintimilla y Pasaje Atahualpa Telf.: 2980 700 - Telefax: 2980-550
Institución Jurídica por Acuerdo Ministerial No. 2414 del 17 de Septiembre de 1964

TULCAN - ECUADOR



Tulcán, a 26 de Diciembre del 2017

Srta.
Liceth Portilla
ESTUDIANTE DE LA UNIVERSIDAD TÉCNICA DEL NORTE.
Presente.-

De mis consideraciones:

A nombre y en representación de Cooperativa de Transporte de Pasajeros en Taxis "Atahualpa", reciba un cordial y atento saludo, a la vez que auguramos éxitos en las funciones que realiza.

Dando contestación a la solicitud requerida por usted, para realizar las pruebas de funcionamiento de su trabajo de titulación denominado: DETECTOR DE ALCOHOLEMIA PARA CONDUCTORES QUE ANALIZA VARIABLES FACIALES Y AMBIENTALES DEL AUTOMÓVIL MEDIANTE EL APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA REDUCCIÓN DE ACCIDENTES DE TRÁNSITO, le autorizo para que lo haga en la unidad N. 29 que pertenece al Sr. José Luis Portilla Menza Socio Activo de la Institución

Atentamente:

"Por la Libertad del Obrero del Volante"

SR. ALVARÓ CANO MONTENEGRO
GERENTE DE COOP. DE TAXIS "ATAHUALPA"



coop_atahualpa46@hotmail.com

Telf.: 2980700 / 2980550 / Cel.: 0980245082

Anexo 24. Formato de encuesta aplicada a los señores conductores



UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERIA EN ELECTRÓNICA Y REDES DE LA COMUNICACIÓN

ENCUESTA DIRIGIDA A TAXISTAS DE LA COOPERATIVA DE TAXIS ATAHUALPA EN LA CIUDAD DE TULCÁN

Esta encuesta va dirigida a los señores conductores de la Cooperativa de taxis Atahualpa con la finalidad de determinar los resultados obtenidos y la aceptación de los conductores del “DETECTOR DE ALCOHOLEMIA PARA CONDUCTORES QUE ANALIZA VARIABLES FACIALES Y AMBIENTALES DEL AUTOMÓVIL MEDIANTE EL APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA REDUCCIÓN DE ACCIDENTES DE TRÁNSITO”. Agradecemos el tiempo que se le dedica a esta encuesta.

NOMBRE: **CÉDULA**.....

1. ¿En qué porcentaje considera que la implementación de un detector de alcoholemia en un vehículo ayuda a disminuir los accidentes de tránsito?
 - a. Entre el 50 y 80%. ()
 - b. Entre el 20 y 50%. ()
 - c. Menor al 20%. ()
 - d. No ayuda a disminuir los accidentes de tránsito. ()
2. Después de haber realizado la prueba de alcoholemia, cómo fueron las instrucciones que proporciona el sistema para verificar el estado en que se encuentra el conductor. Seleccione una.
 - a. Muy buenas. ()
 - b. Buenas. ()
 - c. Regulares. ()
 - d. Malas. ()
3. Que tan complicado le resultó realizar la prueba de alcoholemia para proceder a desbloquear el encendido del vehículo. Seleccione una.
 - a. Muy complicado (las instrucciones no se entienden). ()
 - b. De mediana dificultad. ()
 - c. Lo realizo sin problemas. ()

4. De las variables que se considera para establecer el estado de ebriedad o sobriedad del conductor. Seleccione las que considere necesarias:
 - a. La temperatura facial. ()
 - b. La temperatura ambiental. ()
 - c. El diámetro de la pupila. ()
 - d. El nivel de alcohol en el aliento. ()
 - e. Ninguna. ()

5. La ubicación del detector de alcoholemia le impide visualizar correctamente la carretera. Seleccione según sus experiencias las que considere.
 - a. No se observa bien, el sistema obstaculiza la visión. ()
 - b. Tengo que girarme o moverme para poder visualizar la carretera. ()
 - c. El detector de alcoholemia no obstaculiza la visión. ()

6. De los mensajes de alerta que emite el del detector de alcoholemia cuales le resultaron más adecuados. Seleccione una.
 - a. Los mensajes de alerta visuales. ()
 - b. Los mensajes de alerta auditivos. ()
 - c. Los dos mensajes de alerta. ()
 - d. No es necesario ningún mensaje de alerta. ()

7. ¿Ésta usted en acuerdo o desacuerdo con bloquear o desbloquear el encendido del vehículo según el estado en que se encuentre el conductor? Justifique su respuesta.
 - a. Estoy totalmente de acuerdo. ()
 - b. Estoy parcialmente de acuerdo, existen otras medidas que se pueden adoptar ()
 - c. Estoy en desacuerdo. ()

JUSTIFICACIÓN:.....

8. ¿Ésta usted en acuerdo o desacuerdo con la activación del botón de pánico del ECU 911, en el caso de que el conductor del vehículo proceda a ingerir bebidas alcohólicas después de haber desbloqueado el encendido del automotor? Justifique su respuesta.
- a. Estoy totalmente de acuerdo. ()
 - b. Estoy parcialmente de acuerdo, existen otras medidas que se pueden adoptar ()
 - c. Estoy en desacuerdo. ()

JUSTIFICACIÓN:.....

9. ¿Ésta usted en acuerdo o desacuerdo con la implementación del detector de alcoholemia en un taxi, considerando los riesgos que conllevan manejar en estado de ebriedad, como son las pérdidas materiales, personas heridas, la muerte del conductor o los tripulantes, además de arriesgar la vida de terceras personas? Justifique su respuesta.
- a. Estoy de acuerdo. ()
 - b. No estoy de acuerdo. ()
 - c. La implementación de un detector de alcoholemia me parece indiferente. ()

JUSTIFICACIÓN:.....

.....

FIRMA

Anexo 25: Señores conductores del grupo C de la Cooperativa de taxis Atahualpa de la ciudad de Tulcán que realizaron la encuesta























Anexo 26: Tabulación de resultados de encuestas aplicadas los 58 señores conductores del grupo C de la Cooperativa de taxis Atahualpa de la ciudad de Tulcán

En la primera pregunta realizada se solicitó a los encuestados que identifiquen en que porcentaje consideran que la implementación de un detector de alcoholemia en un vehículo ayuda a disminuir los accidentes de tránsito, en este caso el 69% de los encuestados manifiestan que disminuiría entre un 50 y 80 %, seguido por el 24% que indica que ayudaría a disminuir entre el 20 y 50%, el 10% de la muestra establece que la disminución sería menor al 7%, mientras que ningún encuestado afirma que la implementación de un detector no ayudaría a disminuir accidentes de tránsito. La Figura 111 muestra la tabulación de los resultados a la primera pregunta.

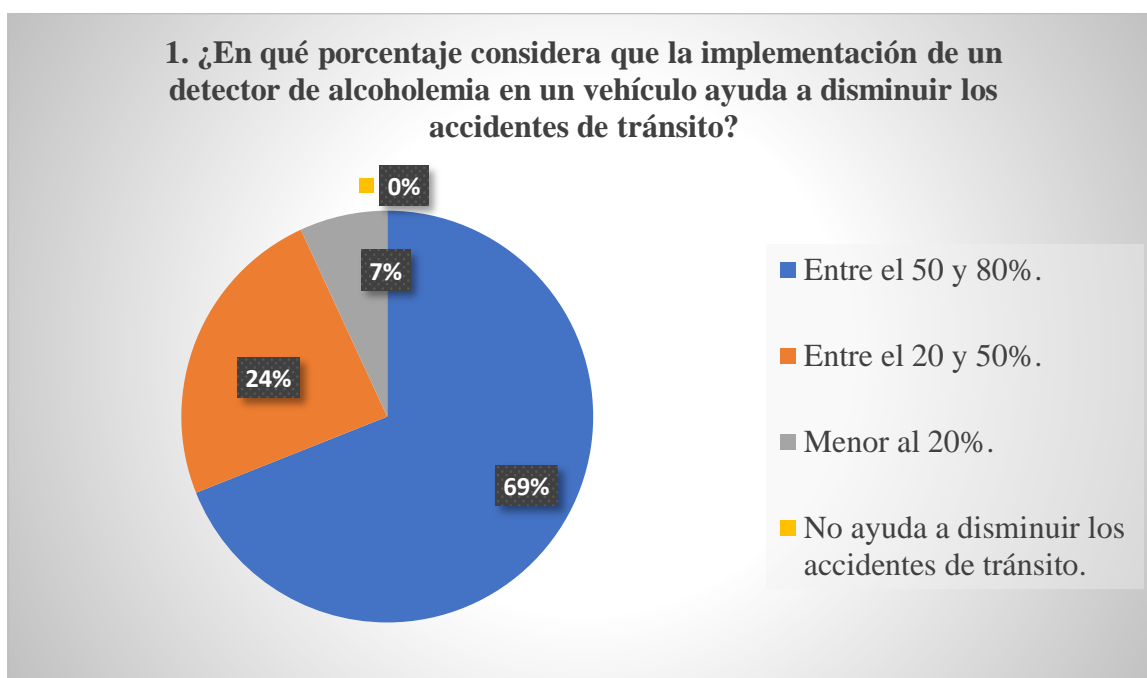


Figura 111. Primera pregunta de la encuesta.

Fuente: Autoría.

En la segunda pregunta después de que cada usuario realizó la prueba de alcoholemia se solicitó a los señores conductores que indiquen cómo fueron las instrucciones que

proporciona el sistema para verificar el estado en que se encuentra el conductor, la finalidad de esta pregunta es para proceder a realizar las correcciones respectivas de las instrucciones que posee el proyecto; Los resultados que se obtuvieron fueron los siguientes: El 76% de los usuarios afirma que las instrucciones son muy buenas, el 24% manifiesta que son buenas, no existen conductores que indiquen que las instrucciones fueron ni regulares ni malas. La Figura 112 muestra la tabulación de los resultados a la segunda pregunta.



Figura 112. Segunda pregunta de la encuesta.

Fuente: Autoría.

Continuando con la encuesta los señores taxistas indicaron que tan complicado les resultó realizar la prueba de alcoholemia para proceder a desbloquear el encendido del vehículo: En donde en este caso el 84% de los encuestados afirman no tuvieron dificultades al realizar la prueba de alcoholemia, el 16% considera que la prueba fue de mediana dificultad y ningún usuario manifiesta que la prueba es muy complicada. En la Figura 113 se indica la tabulación de los resultados a la tercera pregunta.



Figura 113. Tercera pregunta de la encuesta.

Fuente: Autoría.

Para la tabulación de la siguiente pregunta en la que se solicitó a los usuarios establecer las variables que se considera para establecer el estado de ebriedad o sobriedad del conductor, se procede a agrupar dependiendo de cuantas de las opciones seleccionó cada encuestado: En esta ocasión el 28% de los usuarios considera que solo se debería medir el nivel de alcohol en el aliento, seguido de un 21% que establecen que se debería tomar en cuenta todas las variables, a continuación se encuentra medir la temperatura facial el 15% de los usuarios consideraron esta opción, el diámetro de la pupila y el nivel de alcohol en el aliento manifiesta un 11% que solo se deben medir estos valores, otro 11% de los encuestados afirman que solo se debería analizar la temperatura facial y el nivel de alcohol, mientras que el 14% restante se encuentra dividido en que se debería medir la temperatura facial y el diámetro de la pupila, solo la temperatura facial, o solo la temperatura y ninguna respuesta anterior. La Figura 114 muestra la tabulación de los resultados a la cuarta pregunta.

4. De las variables que se considera para establecer el estado de ebriedad o sobriedad del conductor.

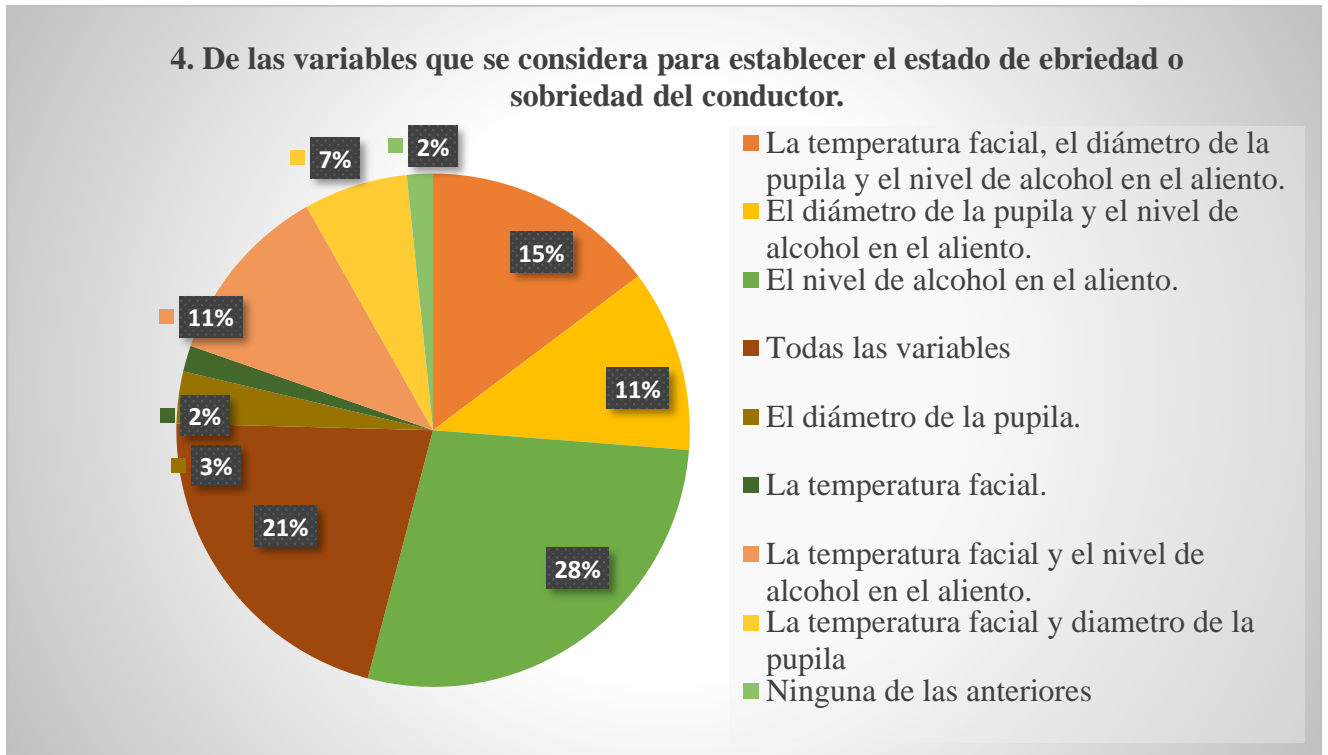


Figura 114. Cuarta pregunta de la encuesta.

Fuente: Autoría.

En la siguiente pregunta en la que se requiere conocer si la ubicación del detector de alcoholemia le impide visualizar correctamente la carretera, el 86% de los señores encuestados afirmaron que el detector de alcoholemia no obstaculiza la visión, el 12% revelo que no se observa bien el sistema obstaculiza la visión, mientras que el 2% afirmo que debe de girarse o moverse para poder visualizar la carretera. La Figura 115 muestra la tabulación de los resultados a la quinta pregunta.

5. La ubicación del detector de alcoholemia le impide visualizar correctamente la carretera. Seleccione según sus experiencias las que considere.

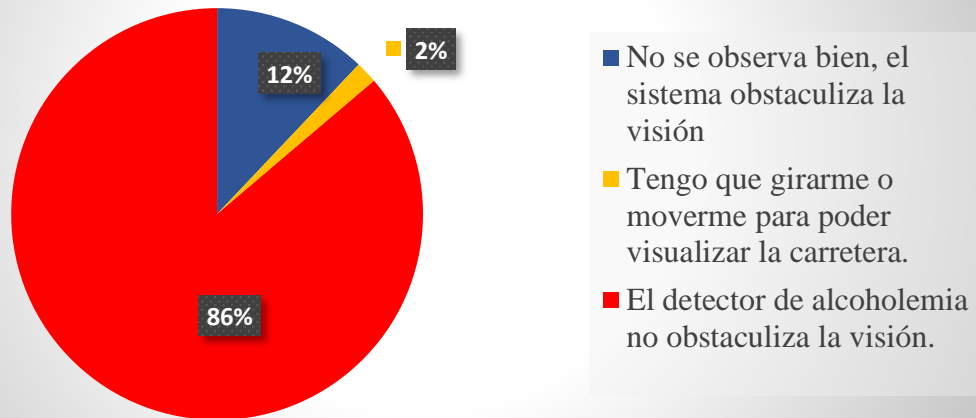


Figura 115. Quinta pregunta de la encuesta.

Fuente: Autoría.

Una vez examinada la ubicación del proyecto, se procede a indagar acerca de cuáles de los mensajes de alerta que emite el del detector de alcoholemia resultaron más adecuados, obteniendo como resultado que el 59% de la muestra afirma que los dos mensajes de alerta son necesarios, el 22% los mensajes de alerta auditivos, el 19% los mensajes de alerta visuales y ningún conductor manifestó que no es necesario ningún mensaje de alerta. La Figura 1 16 indica la tabulación de los resultados a la sexta pregunta.

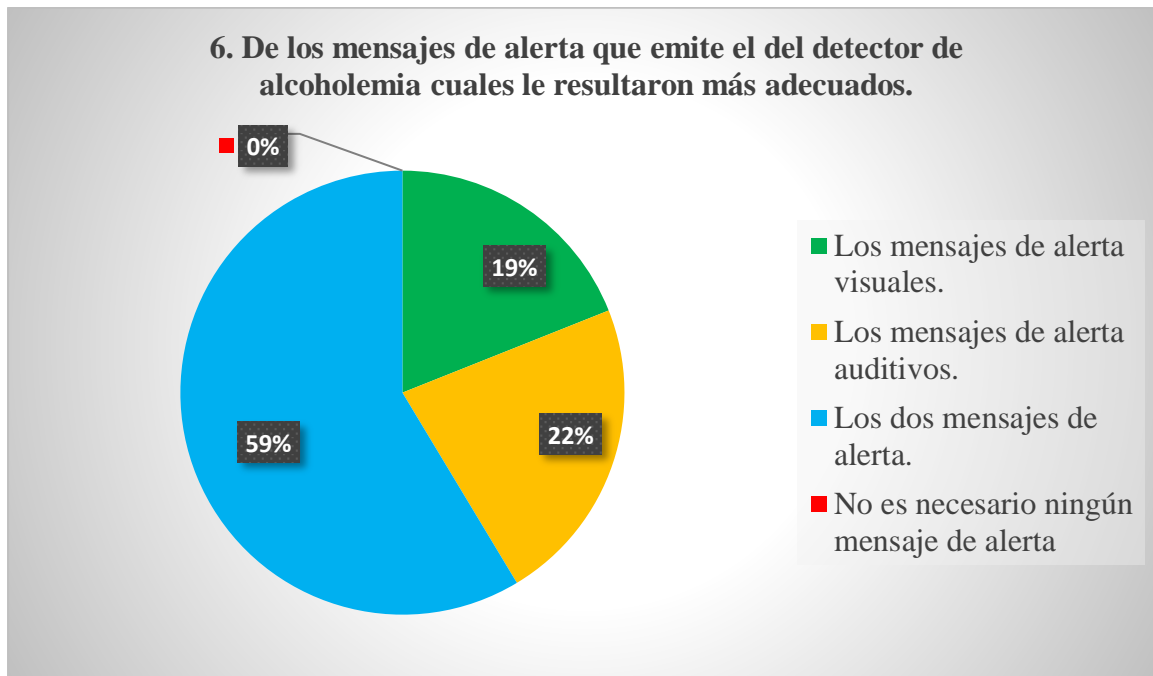


Figura 116. Sexta pregunta de la encuesta.

Fuente: Autoría.

Con respecto a la pregunta en la que se investiga si el conductor está en acuerdo o desacuerdo con bloquear o desbloquear el encendido del vehículo según el estado en que se encuentre el conductor los resultados se muestran en Figura 117 en donde el 95% afirmaron que está de acuerdo, el 3% está parcialmente de acuerdo y el otro 2% restante manifestó que no está de acuerdo, las razón por las cuales están parcialmente de acuerdo se detallan en la Figura 119; mientras que en la Figura 118 se detallan las razones por las cuales el 95% están de acuerdo, en este caso de estudio las razones con mayor porcentaje se encuentran con el 31% cuya finalidad es Prevenir o evitar accidentes de tránsito, seguido de un 13% Evitar conductores necios e irresponsables y un 9% manifiesta que es necesario bloquear para prevenir accidentes y salvar vidas; En la Figura 119 se indican las razones por las cuales están parcialmente de acuerdo los conductores en donde el 67% manifiesta que se debe comunicar a las autoridades competentes y el 33% restante avisar al Ecu911.

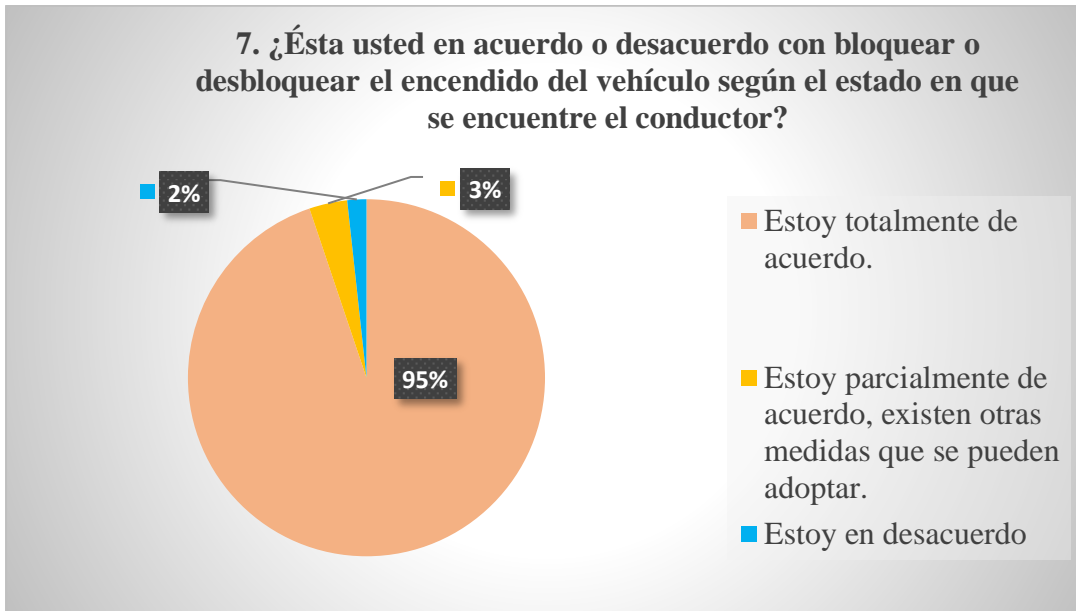


Figura 117. Séptima pregunta de la encuesta.

Fuente: Autoría.

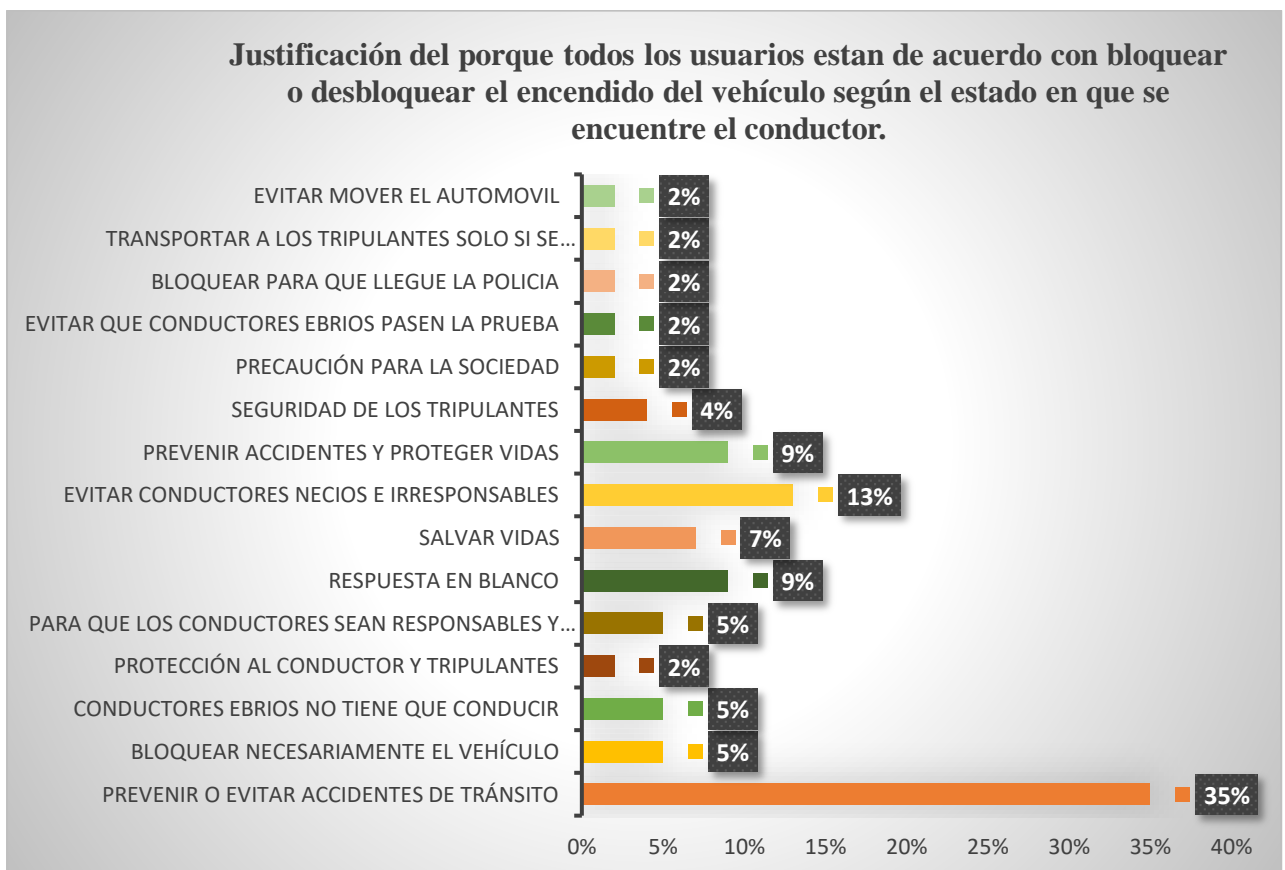


Figura 118. Justificación séptima pregunta de la encuesta en la que los usuarios están de acuerdo en bloquear el encendido del vehículo.

Fuente: Autoría.

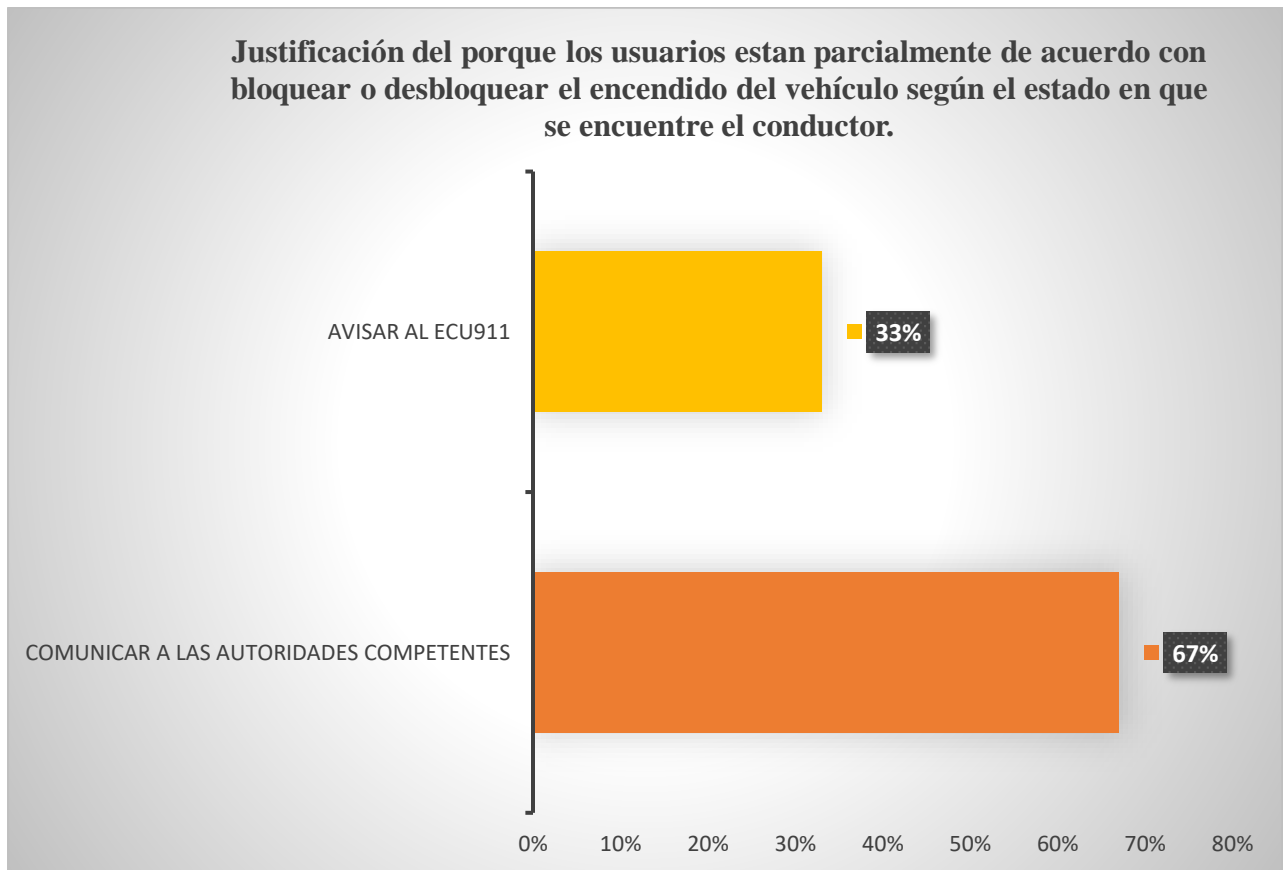


Figura 119. Justificación séptima pregunta de la encuesta con la opción de que están parcialmente de acuerdo.

Fuente: Autoría.

Continuando con la encuesta se procede a averiguar si los usuarios están en acuerdo o desacuerdo con la activación del botón de pánico del ECU 911, en el caso de que el conductor del vehículo proceda a ingerir bebidas alcohólicas después de haber desbloqueado el encendido del automotor, el 93% está totalmente de acuerdo por lo que en la Figura 120 se muestran las razones por las cuales están de acuerdo, el 5% está parcialmente de acuerdo y las justificaciones que dieron estos usuarios son: el bloqueo debe ser inmediato para evitar accidentes, las autoridades competentes no acuden a tiempo y evitar que el conductor sea sancionado; el 2% restante de la muestra manifiesta no estar de acuerdo y la justificación que presenta es que no se conoce las actividades que se realizarán en el día. De la Figura 121 las razones con mayor

porcentaje son: el 22% poner en conocimiento a autoridades competentes, el 19% prevenir accidentes, 17% sancionar a conductores irresponsables y el 11% salvar vidas.

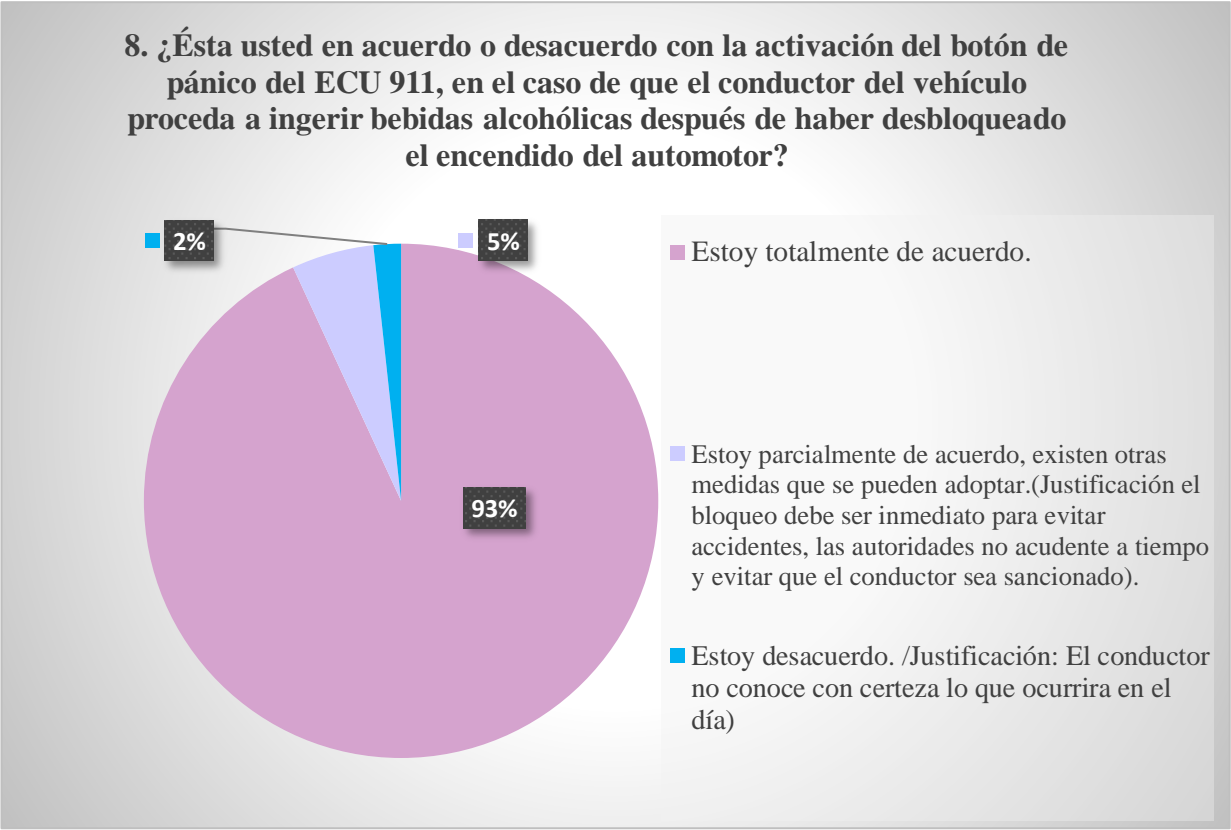


Figura 120. Octava pregunta de la encuesta,
Fuente: Autoría.

Justificación del porque todos los usuarios están en acuerdo con la activación del botón de pánico del ECU 911, en el caso de que el conductor del vehículo proceda a ingerir bebidas alcohólicas después de haber desbloqueado el encendido del automotor

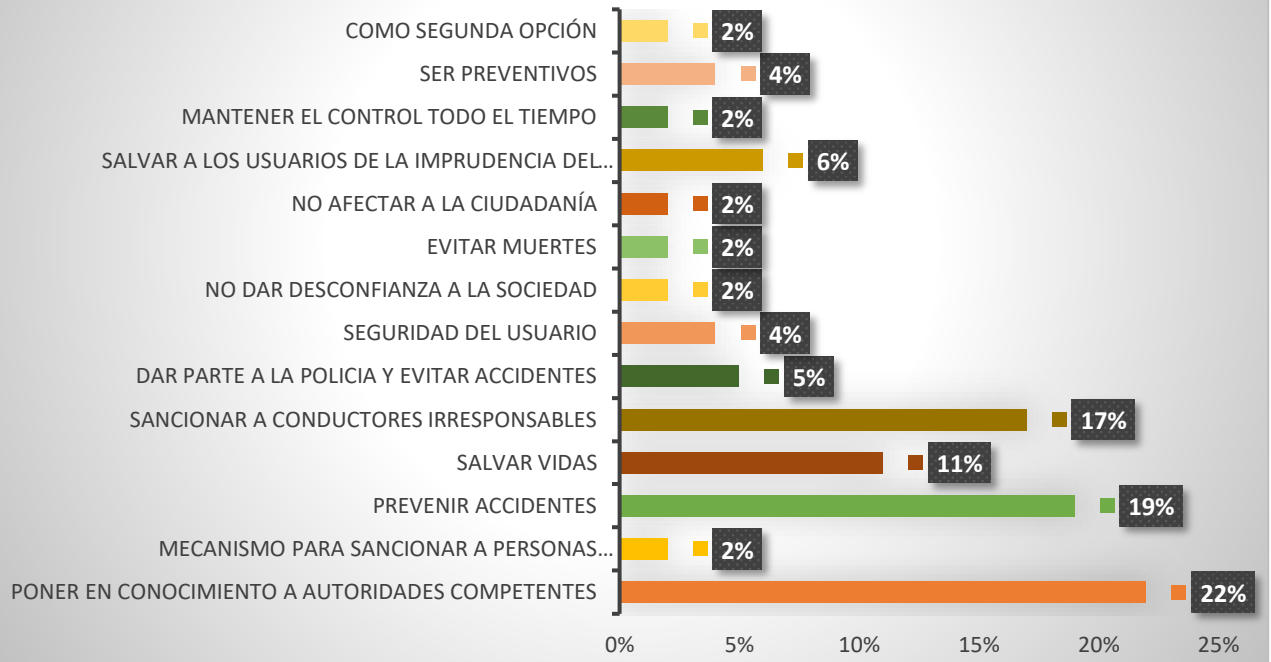


Figura 121. Justificación Octava pregunta de la encuesta por la que están de acuerdo con la activación del botón de pánico.

Fuente: Autoría.

Para finalizar la encuesta se procede a indagar si los usuarios están en acuerdo o desacuerdo con la implementación del detector de alcoholemia en un taxi, considerando los riesgos que conllevan manejar en estado de ebriedad, como son las pérdidas materiales, personas heridas, la muerte del conductor o los tripulantes, además de arriesgar la vida de terceras personas, en este caso el 98% de la muestra manifiesta estar de acuerdo como se puede apreciar en la Figura 122 y solo el 2% considera que no se debería instalar el detector de alcoholemia debido a que para desbloquear el encendido del vehículo toma mucho tiempo para un taxista. En la Figura 123 se muestran las razones por las cuales están de acuerdo que sea implementado, en donde el 30% indica que es necesario para evitar accidentes y el 18% para salvar vidas.

9. ¿Ésta usted en acuerdo o desacuerdo con la implementación del detector de alcoholemia en un taxi, considerando los riesgos que conllevan manejar en estado de ebriedad, como son las pérdidas materiales, personas heridas, la muerte del conductor o los tr

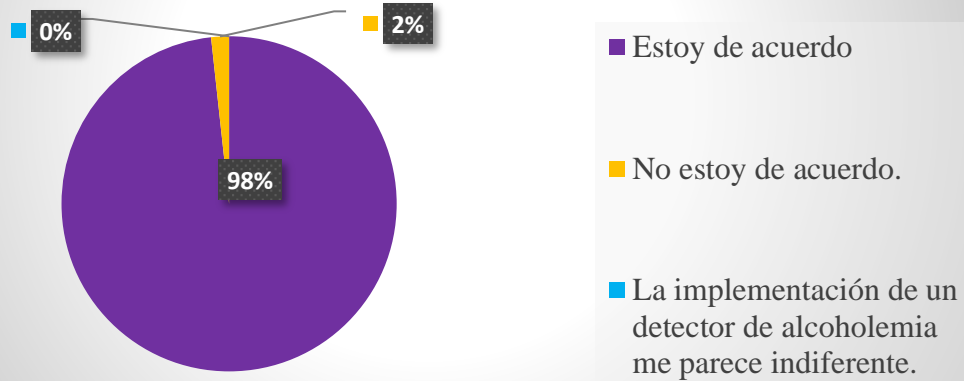


Figura 122. Novena pregunta de la encuesta,

Fuente: Autoría.

Justificación del porque todos los usuarios están en acuerdo con la implementación del detector de alcoholemia en un taxi

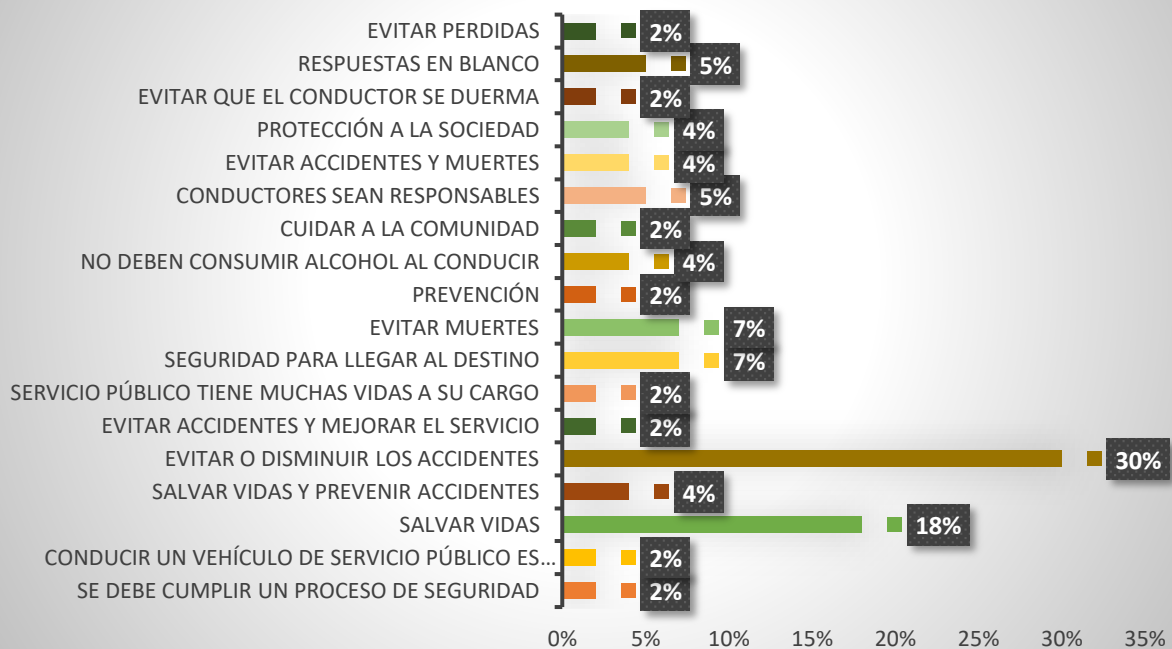


Figura 123. Justificación Novena pregunta de la encuesta con las razones porque están de acuerdo

Fuente: Autoría.

Anexo 27. Certificado de las pruebas de funcionamiento del sistema.



**COOPERATIVA DE TRANSPORTES
"ATAHUALPA"**

Filial de la Federación Nacional de Cooperativas de Transporte de Taxis
Dirección: Edificio Atahualpa, Av. Veintimilla y Pasaje Atahualpa Telf.: 2980 700 - Telefax: 2980-550
Institución Jurídica por Acuerdo Ministerial No. 2414 del 17 de Septiembre de 1964

TULCAN - ECUADOR



Tulcán, 24 de Enero del 2018.

**SR. ALVARO HERNAN CANO
GERENTE COOP. DE TRANSPORTES "ATAHUALPA"**

A petición verbal de los interesados:

CERTIFICO:

Que los resultados obtenidos de las pruebas de funcionamiento del trabajo de titulación denominado **DETECTOR DE ALCOHOLEMIA PARA CONDUCTORES QUE ANALIZA VARIABLES FACIALES Y AMBIENTALES DEL AUTOMOVIL MEDIANTE EL APRENDIZAJE AUTOMÁTICO SUPERVISADO PARA LA REDUCCIÓN DE ACCIDENTES DE TRANSITO.**

Realizado por la Señorita Liceth Geovana Portilla Arteaga con cedula de identidad N. 0401724125 son los esperados en el diseño del proyecto.

Es todo cuanto puedo certificar en honor a la verdad, facultando a los interesados hacer uso del presente certificado cuando lo crean necesario.

Lo Certifico:

**SR. ALVARO CANO MONTENEGRO
GERENTE COOP. DE TRANSPORTES "ATAHUALPA"**

