

**SISTEMA BASADO EN LA DETECCIÓN Y NOTIFICACIÓN DE SOMNOLENCIA
PARA CONDUCTORES DE AUTOS**

EDUARDO LUIS BENAVIDES MUÑOZ

MAURICIO MIGUEL MEDINA MENDEZ



UNIVERSIDAD DE CÓRDOBA

FACULTAD DE INGENIERIAS

INGENIERÍA DE SISTEMAS

MONTERÍA

2015

**SISTEMA BASADO EN LA DETECCIÓN Y NOTIFICACIÓN DE SOMNOLENCIA
PARA CONDUCTORES DE AUTOS**

EDUARDO LUIS BENAVIDES MUÑOZ

MAURICIO MIGUEL MEDINA MENDEZ

Asesor

Ing. Pedro Guevara Salgado

UNIVERSIDAD DE CORDOBA

FACULTAD DE INGENIERIAS

INGENIERÍA DE SISTEMAS

MONTERÍA

2015

NOTAS DE ACEPTACIÓN

FIRMA DEL ASESOR

FIRMA DEL JURADO

FIRMA DEL JURADO

DEDICATORIA

Dedico esta tesis a mis amigos Omar Moreno, Harold Aparicio, Samuel Gomez, Carlos Urueta y Francisco Hernandez quienes fueron gran apoyo emocional durante el tiempo en que la escribía.

A mi compañero Eduardo Benavides, sin su conocimiento éste proyecto no sería posible. A su familia, en especial a su padre Oscar Benavides por prestarnos su vehículo y su tiempo.

A mis padres Liney Mendez y Jose Medina quienes me apoyaron todo el tiempo y a mi hermano Ricardo Medina.

A mis maestros quienes nunca desistieron al enseñarme, aun sin importar que muchas veces no prestaba atención en clase, a ellos que continuaron depositando su confianza en mí.

A los ingenieros Samir Castaño, Javier Peniche y en especial a nuestro director Pedro Guevara por su tiempo y ayuda en el desarrollo de esta tesis.

A todos los que nos ayudaron y apoyaron, para ellos es esta dedicatoria, pues es a ellos a quienes se las debo por su apoyo incondicional.

Mauricio Miguel Medina Mendez

DEDICATORIA

A Dios por haberme dado la vida y por ser mi compañía en todos los momentos más difíciles de la vida.

A mis padres por siempre estar a mi lado animándome a estudiar para seguir adelante. A mis hermanos por darme todo su apoyo en el transcurso de este proyecto.

A todos los amigos que me acompañaron en este proceso de aprendizaje y en especial a mi compañero Mauricio Medina debido a que con su dedicación este proyecto se hizo posible.

A los ingenieros Pedro Guevara y Samir castaño por habernos guiado en la realización de esta tesis.

Eduardo Luis Benavides Muñoz

CONTENIDO

1. OBJETIVOS	1
1.1. General	1
1.2. Específicos.....	1
2. INTRODUCCIÓN.....	2
3. MARCO TEORICO Y MARCO CONCEPTUAL	11
3.1. Marco Teórico	11
3.1.1. Somnolencia.....	11
3.1.1.1. Factores que causan somnolencia	12
3.1.1.2. Características visuales que indican somnolencia.....	13
3.1.1.3. ¿Cómo afecta la somnolencia en la conducción?	13
3.1.1.4. Recomendaciones para evitar la somnolencia al volante.	14
3.2. Marco conceptual	15
3.2.1. Visión artificial.....	15
3.2.2. Detección facial	15
3.2.3. Reconocimiento facial.....	15
3.2.4. Herramientas de reconocimiento de objetos	16
3.2.4.1. Hardware	16
3.2.4.1.1. Kinect	16
3.2.4.1.2. Intel RealSense Camera (F200)	17
3.2.4.1.3. ASUS Xtion PRO	17
3.2.4.1.4. Sony EyeToy	18
3.2.4.2. Software	19
3.2.4.2.1. Kinect Para Windows SDK	19

3.2.4.2.2. SDK Intel RealSense para Windows.....	19
3.2.4.2.3. OpenCV.....	20
3.2.4.2.4. Algoritmo de Viola-Jones.....	21
3.2.4.2.5. EmguCV	21
3.2.4.2.6. OpenCvSharp.....	22
3.2.4.2.7. LuxandFace SDK	22
3.2.4.2.8. Otras librerías.....	23
4. METODÓLOGIA.....	26
4.1. Tipo de investigación.....	26
4.2. Línea de investigación.....	26
4.3. Población objetivo	26
4.4. Fases del proyecto	27
4.4.1. Fase I: Análisis	27
4.4.2. Fase II: Diseño	27
4.4.3. Fase III: Desarrollo	28
4.4.4. Fase IV: Pruebas.....	28
4.5. Metodología de desarrollo	28
5. DESARROLLO.....	30
5.1. Arquitectura del sistema	30
5.2. Diseño del sistema	32
5.2.1. Especificación de requisitos.....	32
5.2.1.1. Objetivos del sistema	32
5.2.1.2. Requisitos de información	34
5.2.1.3. Requisitos no funcionales	35
5.2.2. Modelo relacional	36

5.2.3.	Diagrama de clases	37
5.2.4.	Documentación de actores	44
5.2.5.	Diagramas de casos de uso	45
5.2.6.	Diagramas de secuencia	55
5.2.7.	Diagramas de actividad	60
5.2.8.	Diagrama de flujo	65
5.3.	Desarrollo del proyecto.....	66
5.3.1.	Desarrollo de algoritmos.....	70
5.3.1.1.	Sistema nocturno	70
5.3.1.1.1.	OpenCvSharp.....	70
5.3.1.1.2.	Luxand con EmguCV	76
5.3.1.1.	Sistema diurno	78
5.3.1.1.1.	Kinect vs Luxand	78
5.3.1.3.	Notificación.....	78
5.3.2.	Ubicación de dispositivos	79
5.3.3.	Pruebas y análisis	84
5.3.3.1.	Pruebas.....	84
5.3.3.2.	Análisis.....	106
6.	CONCLUSIONES.....	109
	BIBLIOGRAFIA	111
	ANEXOS.....	115
1.	Mortalidad por siniestros de tránsito	116
2.	Tasas de mortalidad	116
3.	Manual de instalación	117
4.	Manual de usuario	120

5. De cámara RGB a cámara IR.....	126
-----------------------------------	-----

Índice de figuras

Ilustración 1 Microsoft Kinect para Xbox 360 Fuente: Xbox	16
Ilustración 2 Intel RealSense Camara F200 Fuente: Intel	17
Ilustración 3 ASUS Xtion PRO Fuente: ASUS.....	18
Ilustración 4 Camara USB EyeToy Fuente: Play Station	18
Ilustración 5 Metodología de desarrollo.....	29
Ilustración 6 Arquitectura del sistema.....	30
Ilustración 7 Modelo relacional BD Vigilant	36
Ilustración 8 Diagrama de clases registrarse.....	37
Ilustración 9 Diagrama de clases iniciar sesión	38
Ilustración 10 Diagrama de clases Kinect.....	39
Ilustración 11 Diagrama de clases cámara.....	40
Ilustración 12 Diagrama de clases sistema	41
Ilustración 13 Diagrama de clases alarma.....	42
Ilustración 14 Diagrama de clases sistema completo	43
Ilustración 15 Caso de uso general	45
Ilustración 16 Caso de uso gestionar usuario.....	46
Ilustración 17 Caso de uso gestionar procesos	49
Ilustración 18 Caso de uso gestionar notificación.....	53
Ilustración 19 Diagrama de secuencia registrarse	55
Ilustración 20 Diagrama de secuencia iniciar sesión con reconocimiento facial	56
Ilustración 21 Diagrama de secuencia iniciar sesión manualmente.....	57
Ilustración 22 Diagrama de secuencia procesamiento de Kinect.....	58
Ilustración 23 Diagrama de secuencia procesamiento cámara IR	58
Ilustración 24 Diagrama de secuencia alerta.....	59
Ilustración 25 Diagrama de secuencia actualizar datos	59
Ilustración 26 Diagrama de secuencia cerrar sesión	60
Ilustración 27 Diagrama de actividad registrarse.....	60
Ilustración 28 Diagrama de actividad iniciar sesión	61
Ilustración 29 Diagrama de actividad análisis facial	62
Ilustración 30 Diagrama de actividad Conexión TCP.....	63

Ilustración 31 Diagrama de actividad actualizar datos	64
Ilustración 32 Diagrama de flujo Vigilant	65
Ilustración 33 Yuv Fps 15	66
Ilustración 34 Rgb Fps 30	66
Ilustración 35 IR Fps 30	66
Ilustración 36 Error.....	66
Ilustración 37 Función 1	67
Ilustración 38 Función 2	67
Ilustración 39 Imagen IR Kinect	68
Ilustración 40 Imagen IR Cámara.....	68
Ilustración 41 Imagen obtenida de la función 2	69
Ilustración 42 Imagen obtenida de la función con el proyector medianamente obstruido	69
Ilustración 43 Detección de cara OpenCvSharp.....	71
Ilustración 44 Dimensiones de los ojos y la boca	72
Ilustración 45 Ojos y boca en un rostro real	72
Ilustración 46 Imagen binaria (derecha) con Thresload 50.....	73
Ilustración 47 Imagen binaria (derecha) con Thresload 20.....	73
Ilustración 48 Imagen binaria ojos abiertos	74
Ilustración 49 imagen binaria ojos cerrados	74
Ilustración 50 Imagen binaria (derecha) boca cerrada.....	75
Ilustración 51 Imagen binaria (derecha) boca abierta.....	75
Ilustración 52 Vigilant.Apk en ejecución	79
Ilustración 53 Vehículo de prueba	79
Ilustración 54 ubicación laptop	80
Ilustración 55 ubicación Kinect.....	81
Ilustración 56 Dimensiones Vehiculo.....	81
Ilustración 57 Conexión de Kinect al Vehículo.....	82
Ilustración 58 Ubicación cámara para OpenCvSharp	83
Ilustración 59 Ubicación cámara para Luxand.....	84
Ilustración 60 Mensaje de bostezo Kinect SAC – Día.....	85

Ilustración 61 Mensaje Cabeceo Kinect SAC – Día.....	85
Ilustración 62 Bostezo Luxand SAC – Día.....	86
Ilustración 63 Cabeceo Luxand SAC – Día	87
Ilustración 64 Bostezo OpenCvSharp SAC – Noche	88
Ilustración 65 Ojos cerrados OpenCvSharp SAC – Noche	88
Ilustración 66 Cabeceo Luxand SAC – Noche.....	89
Ilustración 67 Bostezo Luxand SAC – Noche	90
Ilustración 68 Ojos cerrados Luxand SAC – Noche.....	90
Ilustración 69 Bostezo Kinect SVDG – Día.....	91
Ilustración 70 Cabeceo Kinect SVDG - Día	92
Ilustración 71 Bostezo Luxand SVDG – Día	93
Ilustración 72 Cabeceo Luxand SVDG - Día	93
Ilustración 73 Ojos cerrados OpenCvSharp SVDG – Noche	94
Ilustración 74 Bostezo OpenCvSharp SVDG - Noche	95
Ilustración 75 Cabeceo Luxand SVDG – Noche.....	96
Ilustración 76 Ojos Cerrados Luxand SVDG – Noche	96
Ilustración 77 Cabeceo Luxand SVDG - Noche.....	97
Ilustración 78 Cabeceo Kinect VCR – Día.....	98
Ilustración 79 Bostezo Kinect VCR - Día	98
Ilustración 80 Cabeceo Luxand VCR - Día.....	99
Ilustración 81 Bostezo Luxand VCR - Día	100
Ilustración 82 Ojos cerrados OpenCvSharp VCR – Noche.....	101
Ilustración 83 Bostezo OpenCvSharp VCR – Noche	101
Ilustración 84 Bostezo Luxand VCR – Noche.....	102
Ilustración 85 Cabeceo Luxand VCR - Noche	103
Ilustración 86 Ojos cerrados ERROR Luxand VCR - Noche.....	103
Ilustración 87 FaceTracking no puede detectar el rostro	104
Ilustración 88 Vista trasera de la ubicación del dispositivo en el taxi	105
Ilustración 89 Falso positivo ojos cerrados	105
Ilustración 90 Falso positivo bostezo.....	106
Ilustración 91 Mortalidad por siniestros de tránsito. Colombia 1995-2012 Py.....	116

Ilustración 92 Tasas de Mortalidad Colombia.....	116
Ilustración 93 Pagina de descarga de XAMPP	117
Ilustración 94 Panel de control XAMPP.....	118
Ilustración 95 Página de inicio XAMPP	118
Ilustración 96 Página de phpMyAdmin	119
Ilustración 97 Importación de la base de datos	119
Ilustración 98 Ventana Índice.....	120
Ilustración 99 Ventana Registrar Datos	120
Ilustración 100 Erros en los Datos.....	121
Ilustración 101 Ventana Capturar fotograma del rostro	121
Ilustración 102 Frame Capturado	122
Ilustración 103 Ventana de Confirmación de Datos.....	122
Ilustración 104 Ventana principal	123
Ilustración 105 Ventana de iniciar sesión con reconocimiento facial	123
Ilustración 106 Ventana de inicio de sesión manual.....	124
Ilustración 107 Opciones para configurar dispositivos.....	124
Ilustración 108 Menú de configuraciones	124
Ilustración 109 Ventana Configuraciones	125
Ilustración 110 Ventana Ojos Ilustración 111 Ventana Boca.....	125
Ilustración 112 Cámara Normal.....	126
Ilustración 113 LEDs Infrarrojos	126
Ilustración 114 Cámara desarmada	127
Ilustración 115 LEDs Infrarrojos instalados	128
Ilustración 116 Filtro que se debe remover	128
Ilustración 117 Cámara IR terminada.....	129

Índice de tablas

Tabla 1 Objetivo gestionar usuario	32
Tabla 2 Objetivo gestionar procesos	33
Tabla 3 Objetivo gestionar usuario	33
Tabla 4 Información acerca del usuario.....	34
Tabla 5 Datos de configuración de dispositivos.....	34
Tabla 6 Requisitos no funcionales.....	35
Tabla 7 Documentación de actores	44
Tabla 8 Documentación caso de uso registro de usuario	47
Tabla 9 Documentación caso de uso iniciar sesión manualmente	47
Tabla 10 Documentación caso de uso iniciar sesión con reconocimiento facial	48
Tabla 11 Documentación caso de uso actualizar configuración de dispositivos	49
Tabla 12 Documentación caso de uso ver proceso del ojo	50
Tabla 13 Documentación caso de uso ver proceso de la boca.....	51
Tabla 14 Documentación caso de uso modo de operación diurno	51
Tabla 15 Documentación caso de uso modo de operación nocturno	52
Tabla 16 Documentación caso de uso iniciar detección	52
Tabla 17 Documentación caso de uso iniciar servidor.....	53
Tabla 18 Documentación caso de uso recibir mensaje	54
Tabla 19 Documentación caso de uso activar alarma	54
Tabla 20 Resultados SAC Kinect	86
Tabla 21 Resultados SAC Luxand - Dia	87
Tabla 22 Resultados SAC OpenCvSharp Noche	89
Tabla 23 Resultados SAC Luxand Noche	91
Tabla 24 Resultados SVDG Kinect	92
Tabla 25 Resultados SVDG Luxand Dia	94
Tabla 26 Resultados SVDG OpenCvSharp Noche.....	95
Tabla 27 Resultados SVDG Luxand Noche	97
Tabla 28 Resultados VCR Kinect	99
Tabla 29 Resultados VCR Luxand Día.....	100
Tabla 30 Resultados VCR OpenCvSharp Noche	102

Tabla 31 Resultados VCR Luxand Noche	104
--	-----

SISTEMA BASADO EN LA DETECCIÓN Y NOTIFICACIÓN DE SOMNOLENCIA EN CONDUCTORES DE AUTOS

1. OBJETIVOS

1.1. General

Desarrollar un sistema de alerta utilizando visión artificial para la detección de somnolencia en conductores.

1.2. Específicos

- Investigar sobre las causas o efectos que produce la somnolencia para reconocer los rasgos y características físicas que identifican a una persona que la padece.
- Diseñar el sistema para detección de somnolencia utilizando la tecnología de reconocimiento de objetos Kinect y la librería OpenCV para el procesamiento de imágenes
- Diseñar una App móvil para la notificación de somnolencia utilizando una conexión socket tipo TCP.
- Realizar pruebas al sistema incorporando los dispositivos en un vehículo.

2. INTRODUCCIÓN

Diariamente las personas necesitan transportarse con un fin específico sea para ir a casa, al trabajo, a los supermercados, entre otros. A este hecho se afirma que el medio de transporte preferido para las personas es el automóvil, que puede ser particular o de servicio público. Se estima que en el año 2014 se vendieron 328.526 unidades en Colombia superando las ventas del 2013 (Suarez, 2015) a pesar de estos datos, resulta un enigma saber a ciencia cierta cuantos autos circulan en el país (Marín, 2009). Desafortunadamente debido al alto porcentaje de vehículos que circulan en las calles, de algunas personas inescrupulosas que los conducen, muchas veces se presentan accidentes en los cuales la mayoría de las ocasiones hay víctimas, de hecho en los últimos años los accidentes de tránsito son una de las diez primeras causas de muerte en Colombia. Para el año 2010 ocuparon la novena causa de muerte con una tasa de 13,27 muertes por 100.000 habitantes según las cifras del DANE. Adicionalmente, son responsables de 3.722 AVISAS¹ de mortalidad y 0,064 de discapacidad. Para el 2011, el Instituto de Medicina Legal reportó 5.792 muertes por accidentes de tránsito, lo que significa un incremento de 1,5% ante el número de casos registrados para el 2010, y se presentaron 40806 víctimas con algún tipo de lesión para un aumento de 3,6% en relación con el año inmediatamente anterior (Seguimiento al sector salud en Colombia, s.f.). Aproximadamente el 35% de las muertes por accidentes de tránsito ocurren en carreteras, la mortalidad por accidentes de tránsito para el 2012 fue de 5693 según la fuente suministrada por el instituto de medicina legal el 24 de enero de 2013. Con estos datos se puede afirmar que muchos de los accidentes de tránsito presentados en Colombia tienen una alta mortalidad y en ocasiones un gran número de víctimas sobreviven a las lesiones, pero como consecuencia deben vivir sobrellevando algún tipo de discapacidad (Corporación Fondo de Prevención vial, s.f.).

¹AVISAS: Indicador de salud instrumento de evaluación que puede determinar directa o indirectamente modificaciones dando así una idea del estado de situación de una condición. Atribuciones científicas : – valido – confiable – sensible – específico

Para muchas personas el final del día es el fin de su jornada de trabajo lo cual significa un descanso, en cambio para otras las noches son una batalla contra el cansancio debido a que su jornada es nocturna. Una persona que presente agotamiento mientras conduce es un riesgo, debido a que en tan solo un descuido puede provocar un accidente. En ocasiones estos accidentes no siempre son culpa de aquel que va al volante, cuando se trata de conductores de servicio público él o ella solo quieren cumplir con su trabajo a veces sin darse cuenta el riesgo, entonces la culpa esta en los malos manejos que tiene la administración de la empresa por no tener estrategias que eviten la fatiga de los trabajadores.

Aunque la jornada nocturna es donde los conductores son más inmunes a la presencia de fatiga, en las jornadas diurnas hay ciertas horas en las que el conductor puede verse inmerso en pequeños estados de somnolencia. También existen personas que presentan cansancio ocasionado por el consumo de medicamentos, alguna sustancia o el padecimiento de una enfermedad. En estos casos la hora del día es completamente irrelevante para el cuerpo.

Según las estadísticas de la CFPV se incrementó aproximadamente en 6.000.000 el parque automotor desde 1994 a 2012, a pesar de este incremento considerable de vehículos el número de personas muertas por siniestros de tránsito disminuye en promedio 128 por año (ver anexo 1). Esta disminución se dio gracias a las políticas de seguridad vial implementadas por el ministerio de transporte y a las campañas de la CFPV como “inteligencia vial úsala”.

De acuerdo al Instituto Nacional de Medicina Legal, en el periodo comprendido entre los años 2007 y 2012 la tasa de muertes por cada 10.000 vehículos se reduce en promedio 0.72% anualmente (ver anexo 2), esto a pesar de ser alentador no deja de ser preocupante debido a que son cifras de mortalidad y lo indispensable seria tener una tasa lo más baja posible (Corporación Fondo de Prevención vial, s.f.).

Desde hace 37 años los accidentes de tránsito se consideran un problema de salud pública mundial, pero solo 16 años atrás se realizan campañas a nivel nacional en Colombia con el fin de prevenirlos. Durante este tiempo se han abordado diferentes ejes temáticos que aportan a disminuir la accidentalidad y resguardar la integridad de todas las personas que transitan las calles y carreteras del país (Seguimiento al sector salud en Colombia, s.f.).

Como principales causas de accidentes de tránsito se pueden encontrar:

- Exceso de velocidad.
- Irrespeto por las señales de tránsito.
- Maniobrar de forma peligrosa.
- Malas condiciones de las vías.
- Distracciones.
- Fatiga o sueño.
- Consumir alcohol o drogas.

Al analizar algunas de las causas de los accidentes se puede decir que el conductor en la mayoría de las ocasiones es el responsable ya sea de forma directa o indirecta (Sura, s.f.).

Durante los recorridos por las vías la somnolencia en los conductores producto de la fatiga o cansancio es una causa importante de accidentes de tránsito, principalmente en los choques sucedidos en las grandes autopistas. Sin embargo, el peso otorgado y como ésta contribuye a los accidentes es poco comprensible. Muchos accidentes son ocasionados por conductores que se quedan dormidos en la carretera mientras conducen su vehículo ya sea por un alto grado de alcohol, por algún tipo de alucinógeno o por que cometen errores relacionados con las consecuencias de la somnolencia sobre el estado de alerta o con el juicio al majear (Clinica del ronquido y trastornos de sueño, s.f.).

Alguien que esta electroencefalográficamente² dormido no puede manejar apropiadamente un vehículo automotor. Se ha determinado que se pueden desarrollar problemas serios durante el intervalo de tiempo que existe entre un periodo de alerta total y el inicio de un ataque de sueño electroencefalográficamente definido. Por ejemplo, los estudios basados en pruebas de atención mediante el uso de un ordenador arrojan como resultado que los individuos somnolientos pueden ejecutar las pruebas correctamente por periodos largos de tiempo con 1 o 2 episodios de total aislamiento. Si estos episodios de fallas en la ejecución psicomotora llegasen a ocurrir en situaciones peligrosas como lo es una curva de una autopista o cerca a otro carro, pueden inducir u ocasionar un accidente.

Los errores en la ejecución psicomotora y la somnolencia diurna alteran el juicio de una persona. Los accidentes de tránsito especialmente ocurren cuando las personas tienden a desarrollar una mala percepción del riesgo al conducir y por lo general son personas que se encuentran privadas de su sueño.

Existen dos tipos de individuos, aquellos que presentan somnolencia crónica y otros que son privados de su sueño por lo cual perciben la somnolencia de forma más grave. Justamente la forma en que éstos perciben la somnolencia se convierte en una grave incapacidad para el conductor extenuado y debe ser un tema evaluado más prudentemente por los encargados de investigar los accidentes (Colombia, U. N, s.f.).

(Rosales & de Castro, 2010) Una persona somnolienta en su estado físico presenta características como bostezar en periodos cortos de tiempo, actividad espontánea oculomotora, frecuencia del cierre ocular comúnmente conocido como “pestañeo”, expresiones faciales y movimiento de la cabeza conocido como “cabeceo”. En su estado mental presenta cambios en el estado de ánimo, dificultad con la memoria

² Electroencefalografía (EEG): es una exploración neurofisiológica que se basa en el registro de la actividad bioeléctrica cerebral en condiciones basales de reposo, en vigilia o sueño, y durante diversas activaciones.

(pensar, hablar, comprender, escribir o leer), fatiga, energía baja, malestar o letargo (Somnolencia, s.f.).

El 11 de Abril del 2013 ocurrió un accidente ocasionado por un conductor que transportaba frutas, los agentes de tránsito del municipio de pasto manifestaron que el conductor invadió el carril contrario tras haberse quedado dormido (Burbano, 2013). En estados unidos se registró este año un conductor de bus ocasiono un grave accidente involucrando otros vehículos debido a que el conductor se quedó dormido por unos segundos (Caracol Noticias, 2015).

La corporación FICOSA en conjunto con el IBEC (Instituto de Bioingeniería de Cataluña) y la Universidad de Barcelona desarrollaron en 2011 SomnoAlert, un software portable que detecta la somnolencia al volante utilizando dos módulos, el primero analiza la posición del vehículo en el carril mediante los datos proporcionados por un sistema de reconocimiento de líneas (SRC), el segundo módulo considera la dirección, pedales y otros datos del CAN (Controlled Area Network) con el fin de identificar la degradación de la conducción, además integra un algoritmo diseñado con una configuración de 11 parámetros para adaptarse a las preferencias del cliente. El sistema alerta al conductor utilizado 3 tipos de alarma que pueden ser: acústica, visual o háptica (vibración del asiento) (Ficosa).

Flores (2009) en su tesis doctoral desarrolló un sistema avanzado de asistencia a la conducción mediante visión por computador para la detección de la somnolencia, el cual consiste en analizar los ojos para medir el nivel de sueño y estado de distracción que presenta el conductor examinando la orientación del rostro. El sistema se adapta a cualquier persona y trabaja en tiempo real bajo condiciones variables de iluminación, para este último fue necesario dividir el sistema en dos partes, una parte que trabaja durante el día y otra durante la noche, la primera manipula una cámara para capturar imágenes con luz natural y la segunda utiliza un sistema de visión basado en luz infrarroja para resaltar las propiedades espectrales de las pupilas de los ojos en ambientes de baja iluminación (como lo es la noche). Al detectarse un

posible estado de adormecimiento o distracción se emite una alerta para percatar al conductor del estado en que se encuentra.

García, Rogado, Barea, Bergasa, López, Ocaña, & Schleicher, D. (2008) desarrollaron un sistema que tenía como objetivo detectar la somnolencia mientras se conduce a través del estudio del ECG, para ello se centraron en la variable de HRV (variabilidad del ritmo cardíaco) cuya medición se logró calculando el tiempo de ondas R consecutivas del complejo QRS (respuesta galvánica de la piel). La fuerza con la que el conductor agarra el volante y la temperatura corporal del mismo utilizando diversos sensores fueron clave para la adquisición y análisis del HRV. Se observó que cuando hay presencia de fatiga el ritmo cardíaco disminuye paulatinamente y si se presenta cabeceo éste disminuye en consideración. Sin embargo se dedujo que conversar o reír produce resultados similares por lo que es necesario el estudio de otras variables.

(OPTALERT, s.f.) Es una empresa Australiana que desarrollo unos lentes para la detección de somnolencia los cuales utilizan un LED invisible integrado en el marco de las gafas capaz de medir la velocidad del parpadeo 500 veces por segundo. El sistema de detección recoge las señales de advertencia fisiológicas mucho antes de que el operador este sintiendo los efectos. Las gafas envían la información a un procesador industrial Eagle que cuenta con una pantalla grande fácil de leer y una función de alarmas auditivas para prevenir a un operador cuando ha alcanzado niveles de alto riesgo.

Espinola Gonzales, Lopez, A., Maximiliano E. & Rodriguez Sabino, V. G. (2011) diseñaron un sistema de visión artificial capaz de detectar el estado de somnolencia. El proceso consiste en la obtención de imágenes de la zona de los ojos del conductor mediante una cámara de video, luego el sistema las procesa usando un algoritmo matemático implementado en el software MatLab. Si se detecta que los ojos se cierran frecuentemente durante un periodo de tiempo se emite una alarma. Para conocer la posición de los ojos el algoritmo utilizado se basa en la curvatura que

tienen éstos, si esta hacia arriba los ojos están abiertos, de lo contrario están cerrados.

Jiménez (2011) diseño un sistema para la detección del nivel de cansancio en los conductores basado en el procesamiento de imágenes e inteligencia artificial utilizando una cámara de video. Para ello utilizo algoritmos especializados con el fin de identificar movimiento de la cabeza, apertura de la y amplitud de apertura de los ojos. Para medir estas variables se utilizó tres métodos de clasificación:

- Tabla binaria.
- Entrenamiento de una red neuronal.
- Red neuronal realimentada.

Cada uno de los métodos genero tres estados, de no fatiga, alerta y crítico. A pesar que el sistema presento algunas limitantes como la utilización de gafas y movimiento de la cabeza por fuera de los ángulos establecidos, el método más preciso de clasificación es el tercero con un 91 % de precisión en la detección del estado.

Estudiantes de la Universidad de los Andes diseñaron un sistema que permite identificar la condición de sueño en conductores utilizando la cámara de un dispositivo móvil con sistema operativo Android. El sistema fue diseñado utilizando la librería OpenCV y su funcionalidad consiste en adquirir la imagen en escala de grises, luego detectar los bordes Canny y Sobel, aplicar un threshold Gaussiano y por ultimo mostrar la imagen ecualizada. El objetivo del proyecto era monitorear si los ojos del usuario estaban abiertos o cerrados. Sin embargo existen muchas limitantes como la iluminación debido a que es una variable que afectaba el procesamiento de la imagen (Técnicas Tecnogías de Diseño Electrónico, 2012).

Villamizar, Suarez, Villanueva, Borja & De Los Rios (2014) en su proyecto Design and Implementation of Sleep Monitoring System Using Electrooculographs Signals desarrollaron un dispositivo con el objetivo de vigilar la somnolencia presente en los

conductores que trabajan durante largas horas. El sistema de seguimiento diseñado está compuesto por dos etapas, una fase que se encarga del acondicionamiento y adquisición de señales electrooculographs (EOG) y la etapa de control dedicada principalmente a la administración de la alarma. Las señales obtenidas se visualizan en una interfaz de usuario desarrollada en Matlab. Las señales EOG fueron lo suficientemente confiables para establecer los niveles de umbral para la activación de la alarma.

La somnolencia deteriora variables funcionales psicomotoras y neurocognoscitivas como el tiempo de reacción, la capacidad de vigilancia, juicio y atención (Liendo, Castro, & de Castro., 2010). También es importante vigilar a los conductores de vehículos de transporte, principalmente a los que prestan servicios de bus debido a que el trabajo de estos transportistas es en muchas ocasiones monótono, con largos periodos de tiempo y constantes cambios de horarios diurnos y nocturnos. Esto combinado con pocas horas dormidas puede ser un gran riesgo al momento de ejercer su actividad laboral (de Castro, Gallo & Loureiro, 2004).

Según cifras preliminares del Instituto Nacional de Medicina Legal y Ciencias Forenses, al cierre de 2012 hubo 5.693 muertes en siniestros de tránsito en Colombia (Corporación Fondo de Prevención vial, s.f.). Uno de los principales factores que provocan accidentes de tránsito es la somnolencia, por lo cual se hace necesario utilizar ciertos mecanismos que informe a los choferes que padecen de este síntoma y que deben tomar un descanso.

La gran cantidad y funcionalidad de diferentes tipos de sensores, el nuevo concepto emergente como Smart City y las nuevas capacidades tecnológicas que presentan los automóviles, es posible ofrecer diferentes servicios como ubicación geográfica y soporte técnico remoto. Aparte de esto muchas empresas de transporte han optado por utilizar cámaras dentro de los autos con el propósito de registrar actos delictivos, bajo la premisa de este problema se puede combinar la utilización de estas cámaras

con otras tecnologías como la visión artificial para extraer algunas características de imagen y poder tomar decisiones.

De todo lo mencionado anteriormente, en este trabajo se plantea el desarrollo de un sistema que permita alertar a los conductores de vehículos la presencia de sueño al volante utilizando procesamiento de imágenes obtenidas por una cámara, con el fin de evitar o prevenir un posible accidente de tránsito. Lo cual puede contribuir a la seguridad vial tanto para los choferes como para los peatones. Con éste también se pretende incentivar el desarrollo de nuevos mecanismos para mayor control en las vías.

3. MARCO TEORICO Y MARCO CONCEPTUAL

3.1. Marco Teórico

3.1.1. Somnolencia

Según la Real Academia Española (s.f). La somnolencia es la pesadez y torpeza de los sentidos motivados por el sueño. Según investigadores y clínicos del área, la somnolencia al igual que el hambre o la sed es una necesidad fisiológica básica. Una definición operacional es la tendencia a quedarse dormido. La intensidad de ésta se ve afectada por la rapidez con que se inicia el sueño, que tan fácilmente se perturba y cuánto tiempo se permanece dormido. La somnolencia se ha clasificado de dos formas, normal y patológica, el primer caso es secuela del ritmo cardiaco y el segundo es provocado por un sueño alterado. Algunos autores hablan sobre la somnolencia optativa (facilidad de quedarse dormido en un momento aceptable) y excesiva (ocurre en el tiempo que el individuo debería estar despierto) (Rosales et al. 2010).

Es importante enfatizar en la diferencia entre la somnolencia y los conceptos de fatiga. La fatiga es un fenómeno muy complejo que comprende un número de procesos psicosociales y del comportamiento. La fatiga es la molestia ocasionada por un esfuerzo más o menos prolongado o por otras causas y que se manifiesta en la respiración frecuente o difícil (Real Academia Española, s.f.).

En la medicina del sueño el término es usado para hacer referencia a la sensación de agotamiento como incapacidad de ejecutar una actividad física. Los pacientes con fatiga habitualmente no se quejan de quedarse dormidos y sus síntomas prácticamente no son causados por trastornos del sueño. Al contrario la somnolencia no involucra ejercicio físico (Rosales et al. 2010).

3.1.1.1. Factores que causan somnolencia

Generalmente todo aquello que reduzca la cantidad o calidad de sueño nocturno, provocara somnolencia el día siguiente. (Dirección General de Tráfico (DTG), Gobierno de España) Algunos factores comunes son los siguientes:

La privación del sueño: los médicos recomiendan dormir en promedio 7 y 9 horas para tener un óptimo desempeño. Sin embargo el número de horas varía por factores como el trabajo o la edad de la persona. Dormir poco ocasiona una fuerte somnolencia que altera la capacidad para conducir, además los efectos de privar el sueño son acumulativos. Dormir cada día 1 o más horas menos que la noche anterior puede generar una deuda de sueño que llevara a un adormecimiento crónico con el tiempo.

El sueño fragmentado: a pesar de dormir el número de horas adecuadas, despertarse constantemente por la noche produce un sueño poco eficaz. Este factor se produce comúnmente por el consumo de alcohol, los ruidos nocturnos, preocupaciones, la luz, la ansiedad o ciertas enfermedades. Todo ello aumentara la somnolencia e implicará en mayores riesgos a la hora de conducir un vehículo.

Cambios en el horario del sueño: como su nombre lo indica, variar con frecuencia el horario de sueño ocasiona periodos fuertes de somnolencia durante el día, habitualmente le sucede a los trabajadores con turnos nocturnos. Dormir por el día no es completamente reparador (por el ruido, la luz, entre otros). Alterar el ciclo sueño-vigilia puede tener consecuencias para la seguridad en el tráfico.

Sustancias con efectos sedantes: sustancias como el alcohol y ciertos medicamentos favorecen la aparición de somnolencia. Se pueden destacar los antihistamínicos, algunos antidepresivos especialmente algunos fármacos recetados para la ansiedad.

Trastornos del sueño: éstos alteran directamente el ciclo sueño-vigilia. Influyen negativamente en la conducción y son peligrosos para la seguridad vial. De tal manera que trastornos como la narcolepsia impiden legalmente conducir.

3.1.1.2. Características visuales que indican somnolencia.

Las características visuales comunes características de una persona somnolienta son las siguientes (Rosales et al. 2010):

- Bostezo.
- Actividad espontanea oculomotora.
- Frecuencia del cierre ocular comúnmente conocido como pestañeo.
- Expresiones faciales.
- Movimiento de la cabeza también llamado cabeceo.

3.1.1.3. ¿Cómo afecta la somnolencia en la conducción?

Una de las causas de accidentes de tránsito es la somnolencia, ésta a diferencia del alcoholismo y el exceso de velocidad no permite mayor control sobre el vehículo debido a que si el conductor se duerme no reaccionara a tiempo ante algún imprevisto. Por esta razón se piensa que la somnolencia es un factor de riesgo tan grave como los antes mencionados (Dirección General de Tráfico (DTG), Gobierno de España).

Las alteraciones producidas por la somnolencia y que afectan la conducción son:

- Incremento del tiempo de reacción.
- Menor concentración y más distracciones.
- Toma de decisiones más lentas y más errores.

- Alteraciones motoras.
- Movimientos más automatizados.
- Aparición de micro-sueños.
- Alteración de las funciones sensoriales.
- Alteraciones en la percepción.
- Cambios en el comportamiento.

3.1.1.4. Recomendaciones para evitar la somnolencia al volante.

La mejor opción para evitar la somnolencia es guardar buenos hábitos de sueño. Sin embargo esto no siempre es posible por tal motivo se deben tener en cuenta las siguientes recomendaciones (Dirección General de Tráfico (DTG), Gobierno de España):

- Si el sueño aparece mientras se conduce, lo mejor es parar a descansar en un lugar adecuado y dormir unos 20 o 30 minutos.
- Tener cuidado al manejar en las horas más peligrosas que son entre las 3 - 5 de la mañana y entre 2 - 4 de la tarde.
- Evitar trayectos largos, especialmente si es por la noche.
- Descansar 20-30 minutos cada 2 horas o 200 km.
- Sujetar con firmeza el volante y no adoptar posturas relajantes.
- Ventilar adecuadamente el interior del vehículo.
- No escuchar música relajante.
- No ingerir alcohol ni fumar.
- No se debe manejar si se a echo algún tipo de esfuerzo físico (fatiga).

3.2. Marco conceptual

3.2.1. Visión artificial

También llamada como visión por computador, es un subcampo de la inteligencia artificial. Tal como los humanos usamos nuestros ojos y cerebros para comprender el mundo, la visión artificial tiene como propósito programar un computador para que logre “entender” el mismo tipo de escenas a través de una imagen o secuencia de estas, ya sea para producir información numérica o simbólica (Wikipedia, s.f).

3.2.2. Detección facial

Un caso muy particular de la detección de objetos. Es un proceso en el cual la computadora puede ubicar y/o rastrear los rostros presentes en una imagen/video. Pero puede suponer algunos problemas según las condiciones de iluminación o como se encuentre las caras (Wikipedia, s,f).

3.2.3. Reconocimiento facial

Es una técnica aplicada en la visión artificial para identificar automáticamente a una persona en una imagen digital o video. Se logra a través del análisis de las características faciales del individuo extraídas de un fotograma y comparándolas con una base de datos. Sistema de reconocimiento facial (Wikipedia, 2015).

3.2.4. Herramientas de reconocimiento de objetos

3.2.4.1. Hardware

3.2.4.1.1. Kinect

Es un controlador de juego libre diseñado en 2010 para la Xbox 360 con el propósito de interactuar con la consola mediante una interfaz natural de usuario que reconoce gestos, comandos por voz y objetos e imágenes.

La unidad dispone de una cámara RGB, un sensor de profundidad, un micrófono y un procesador encargado de ejecutar el software. También incluye un mecanismo de inclinación motorizado. El sensor de profundidad es la unión de dos dispositivos, un proyector de infrarrojos y una cámara CMOS permitiendo a Kinect ver la habitación en 3D en cualquier condición de luz ambiental (Wikipedia, s.f).



KINECT™
for  XBOX 360.

Ilustración 1 Microsoft Kinect para Xbox 360 Fuente: Xbox

3.2.4.1.2. Intel RealSense Camera (F200)

Es un dispositivo creado por Intel con el propósito de utilizar el reconocimiento facial, de gestos y de voz para permitir que la interacción con el sistema sea más natural. Equipada con una cámara a color, una cámara de profundidad, un proyector infrarrojo, un conjunto de micrófonos, interfaz USB 3.0 y además cuenta con un alcance de 1,2 metros aproximadamente (Intel, s.f.)



Ilustración 2 Intel RealSense Cámara F200 Fuente: Intel

3.2.4.1.3. ASUS Xtion PRO

Es una solución para el desarrollo de aplicaciones y juegos utilizando la detección de gestos. El kit de desarrollo es ampliamente abierto, se pueden crear aplicaciones para el negocio o para hacer la vida de las personas más cómoda. El dispositivo utiliza sensores infrarrojos, detección de profundidad, detección de imágenes en color y flujo de audio (ASUS, s.f.).



Ilustración 3 ASUS Xtion PRO Fuente: ASUS

3.2.4.1.4. Sony EyeToy

Es un periférico que contiene una cámara con conexión USB, sensores de luz y un micrófono para grabar y detectar audio. Fue lanzado para la Play Station 2 y es capaz de detectar los movimientos corporales y proyectar la imagen en la pantalla del televisor (Play Station, s.f.).



Ilustración 4 Cámara USB EyeToy Fuente: Play Station

3.2.4.2. Software

3.2.4.2.1. Kinect Para Windows SDK

El software de desarrollo para kinect permite a los desarrolladores crear aplicaciones capaces de reconocer gestos y comandos de voz utilizando los sensores integrados en el dispositivo en máquinas con SO. Windows (Microsoft, s.f.)

Para que Kinect funcione bajo plataformas Windows, el ordenador debe contar con las siguientes especificaciones:

- Sistema operativo.
- Windows 7, Windows 8, Windows 8.1, Windows Embedded Standard 7
- Requisitos de Hardware.
- SO de 32 bits (x86) o 64 bits (x64)
- Procesador doble núcleo de 2,66 GHz o más rápido
- Bus 2.0 USB dedicado
- Memoria RAM de 2 GB

Requisitos de Software.

- Visual Studio 2010 o Visual Studio 2012
- .NET Framework 4 o .NET Framework 4.5

3.2.4.2.2. SDK Intel RealSense para Windows

El software de desarrollo para la cámara RealSense que permite a los desarrolladores crear aplicaciones y juegos mediante el reconocimiento de la cabeza y las manos en 3 dimensiones. Compatible con Microsoft Windows (Intel, s.f.)

Requisitos de hardware

- USB 3.0
- Procesador Intel® Core™ de cuarta generación (o posterior)
- SO Windows 8.1* (o superior) (64 bits)

Requisitos de Software

- Microsoft Visual Studio o Eclipse.

3.2.4.2.3. OpenCV

Es una biblioteca de visión artificial, desarrollada por Intel y de uso libre, utilizada en infinidad desde aplicaciones desde sistemas de seguridad para la detección de movimientos hasta aplicaciones para el reconocimiento de objetos. OpenCV se puede utilizar para propósitos tanto comerciales como de investigación gracias a su licencia BSD. Además cuenta con la ventaja de ser multiplataforma, existiendo versiones para GNU/Linux, Mac OS X, Windows y también Android. Cuenta con un gran número de funciones que permiten:

- Reconocimiento de objetos (reconocimiento facial)
- Calibración de cámaras
- Visión estérea
- Visión robótica

Es fácil de utilizar y altamente eficiente gracias a su programación en C y C++ optimizados (Wikipedia, s.f)

3.2.4.2.4. Algoritmo de Viola-Jones

Viola & Jones (2004) implementaron un método que es capaz de detectar rostros mediante el procesamiento de imágenes con gran rapidez. Este método presenta las tres siguientes contribuciones:

Esta primera contribución consiste en la introducción de una nueva forma de representación de imagen denominada “Integral Image” permitiendo que las características del detector se conmuten de forma muy rápida.

La segunda es un clasificador simple pero muy eficiente, diseñado con el algoritmo de aprendizaje AdaBoost para seleccionar de un conjunto de características potenciales un pequeño número de características visuales.

La tercera contribución es el método que combina clasificadores en cascada que permiten descartar el fondo de una imagen.

3.2.4.2.5. EmguCV

Es un wrapper de OpenCV diseñado para ser utilizado en lenguajes .NET como C#, VB entre otros. Puede ser compilado en Visual Studio, Xamarin Studio y Unity. Se puede ejecutar en plataformas Windows, Linux, Mac OS X, iOS, Android y Windows Phone (EmguCV, s.f.)

3.2.4.2.6. OpenCvSharp

Es otro wrapper de OpenCV para C#. Con unos métodos de detección más rápidos que EmguCV utilizando los mismos clasificadores en cascada pero con una implementación completamente distinta.

3.2.4.2.7. LuxandFace SDK

FaceSDK es un sistema multi-plataforma de reconocimiento facial. Puede ser utilizado por cualquier desarrollador de software. FaceSDK es una herramienta para desarrollar aplicaciones Web, de escritorio y aplicaciones móviles con la autenticación de usuario basada en el rostro, es decir detección automática de la cara y el reconocimiento del usuario.

Compatible con entornos de escritorio de 32, 64 bits y plataformas móviles, incluyendo iOS y Android. FaceSDK es fácil de integrar con proyectos nuevos o existentes, lo que permite a los desarrolladores crear una amplia variedad de aplicaciones. El sistema detecta los rostros enteros y rasgos faciales individuales, reconoce caras en imágenes fijas y secuencias de vídeo en tiempo real, y permite la creación de una amplia gama de aplicaciones, desde simples herramientas hasta soluciones de acceso biométricos.

El innovador sistema de reconocimiento basado en movimiento permite el reconocimiento sin fisuras y la identificación de los sujetos que aparecen en las secuencias de vídeo, permitiendo a los sistemas de control de asistencia de construcción, seguridad y vigilancia (Luxand, s.f.).

3.2.4.2.8. Otras librerías

En la web es posible encontrar una gran cantidad de herramientas para reconcomiendo facial, algunas son mencionadas a continuación (Ismael, 2013):

Face Recognition

Proporciona el reconocimiento de rostros, detección facial, posición de los ojos, la posición de la nariz, la posición de la boca, y la clasificación de género.

Animetrics Face Recognition

La API Animetrics se puede utilizar para detectar rostros humanos en fotografías. La información sobre los rasgos faciales o puntos de referencia se devuelto a través de coordenadas en la imagen. Animetrics Face Recognition también es capaz de detectar y devolver la orientación.

Face++

Utiliza tecnología de vanguardia de visión por computador y minería de datos para ofrecer 3 servicios centrales (detección, reconocimiento y análisis). Esta API proporciona detección y análisis de Landmark (23 puntos), Landmark (81 puntos), atributos tales como la edad y el género entre otros.

FaceMark

Es una poderosa API para la detección de características faciales. Encuentra 68 puntos por un rostro visto de frente y 35 puntos para un rostro de perfil. FaceMark detecta puntos de referencia para caras en imágenes especificadas por URL o cargados como un archivo y produce una salida JSON con un vector de puntos de referencia faciales y orientación para cada rostro hallado.

EmoVu by Eyeris

Es un software de reconocimiento de emociones que utiliza inteligencia artificial para permitir a las cámaras leer un grupo de expresiones, sexo y edad de la persona en tiempo real.

FaceRect

Es una API libre de detección de rostros. Capaz de encontrar caras (frontales o de perfil) en una imagen especificada por URL o cargada como archivo, es capaz de encontrar múltiples caras en una sola foto y produce una salida JSON con la información relacionada.

EyeFace

Proporciona detección facial, landmark (puntos de referencia), estimación de edad y reconocimiento de género.

Kairos

Es una API de detección y reconocimiento facial de 2D a 3D.

Libface

Es una biblioteca destinada a llevar la tecnología de reconocimiento facial a la comunidad de código abierto. Esta escribe en C y utiliza la biblioteca OpenCV.

FaceReader

Capaz de analizar expresiones faciales y proporciona a los usuarios una evaluación objetiva de la emoción de la persona.

También se encuentran: Betaface API, Skybiometry Face Detection and Recognition, CCV, OpenBR, FlandMark, Bio ID, BetaFace API, fdlib, Visage, Stupeflix, KeyLemon. La lista de APIs es demasiado extensa, solo se han mencionado algunas. Algunas herramientas son flexibles debido a que proporcionan una licencia gratuita o ya sea por 30 días, otras no lo son tanto debido a que se hace necesario contar con una conexión a internet. Las demás son herramientas que hay que comprar si se desea desarrollar con ellas.

4. METODÓLOGIA

4.1. Tipo de investigación

EL proyecto se clasifica como una investigación científica de tipo descriptiva, en la cual inicialmente se recolecto información de fuentes tales como: páginas web, revistas científicas, tesis y artículos. La información recolectada fue clasificada para 2 fines, el primero para identificar las características del problema objeto de estudio y el segundo con el fin de seleccionar las tecnologías necesarias en el marco del presupuesto base para el proyecto.

4.2. Línea de investigación

Este proyecto está enmarcado en la línea de investigación de la Visión Artificial debido a que tiene como propósito elaborar un software de computador que permita a éste analizar y entender escenas en tiempo real a través de secuencia de imágenes con el fin de obtener datos que puedan ser procesados para tomar decisiones.

4.3. Población objetivo

El sistema se encuentra dirigido hacia los conductores de auto, ya sean estos de servicio público o particular, que conduzcan en el día o en la noche. Con el propósito de ayudar a prevenir posibles accidente causados por somnolencia.

4.4. Fases del proyecto

4.4.1. Fase I: Análisis

Esta primera etapa consistió en recolectar información de diversas fuentes confiables. Fue de vital importancia encontrar aquellos proyectos similares a los objetos de este, para tener un visión clara de lo que se iba a hacer, como hacerlo y que herramientas eran necesarias para lograrlo.

A esta fase la preceden las siguientes actividades:

Obtención de información

En esta etapa primero se buscó información acerca del índice de accidentes de forma general ayudándonos de organismos como el Fondo de Prevención Vial (FPV) y el Instituto de Medicina Legal en Colombia, luego se procedió a investigar porcentajes de accidentalidad causados por somnolencia en la carretera utilizando páginas y documentos extranjeros ya que en Colombia existe muy poca información relaciona a accidentes provocados por somnolencia.

Establecer Requerimientos

Se procedió a determinar los requerimientos funcionales y no funcionales del sistema, de esta manera se facilitó la especificación de las funciones y comportamientos de éste.

4.4.2. Fase II: Diseño

En esta fase se visualizó cómo funcionaría el sistema mediante la construcción de diagramas UML tales como, casos de uso, clases, secuencia, actividades y flujo.

Luego se procedió a seleccionar las tecnologías que cumplieran con ciertos requerimientos con los que se pudiera trabajar basándonos en las especificaciones técnicas de los dispositivos además del costo y forma de obtenerlos, para así poder elaborar el proyecto basándonos en la investigación realizada en la fase 1.

4.4.3. Fase III: Desarrollo

Con los modelos construidos se comienza a desarrollar el sistema utilizando las especificaciones establecidas. Esta es la etapa de programación en la cual se desarrollaron los algoritmos concernientes a la detección de somnolencia y se desarrolló el sistema para la alarma enviada al dispositivo móvil, al igual que la implementación del algoritmo de reconocimiento facial.

4.4.4. Fase IV: Pruebas

Luego de montar el sistema en el vehículo se realizaron las pruebas de funcionamiento de los algoritmos, Se evaluó la respuesta del sistema bajo distintas situaciones, se recopilaron datos y se estableció un margen de eficiencia.

4.5. Metodología de desarrollo

El proyecto fue desarrollado bajo un modelo en cascada, mediante 4 fases muy bien establecidas, donde la finalización de una significaba el comienzo de otra, con un tiempo de inicio y de culminación de las mismas evitando así retrasos en el diseño del sistema. El modelo se observa a continuación.

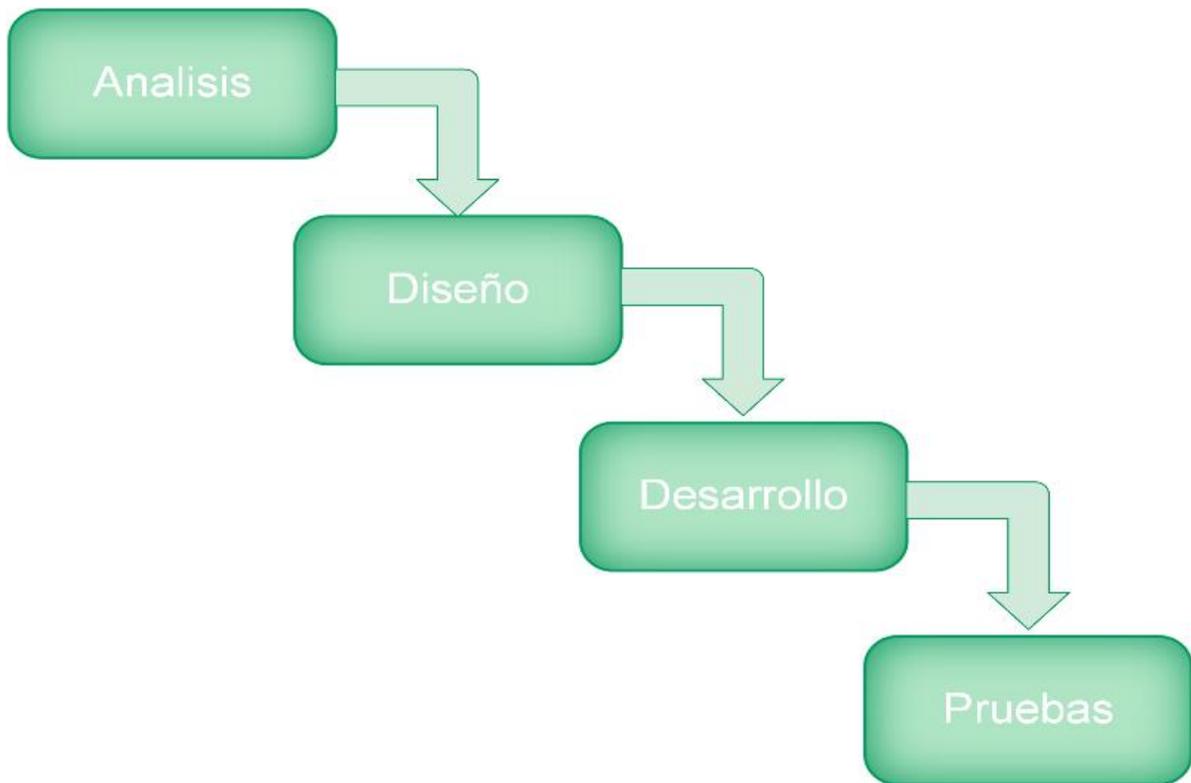


Ilustración 5 Metodología de desarrollo

5. DESARROLLO

5.1. Arquitectura del sistema

La arquitectura del sistema representa una vista global del mismo, de igual forma muestra sus componentes de manera individual y en pequeños subsistemas interactuando entre sí. Lo anterior se puede observar en la siguiente figura.

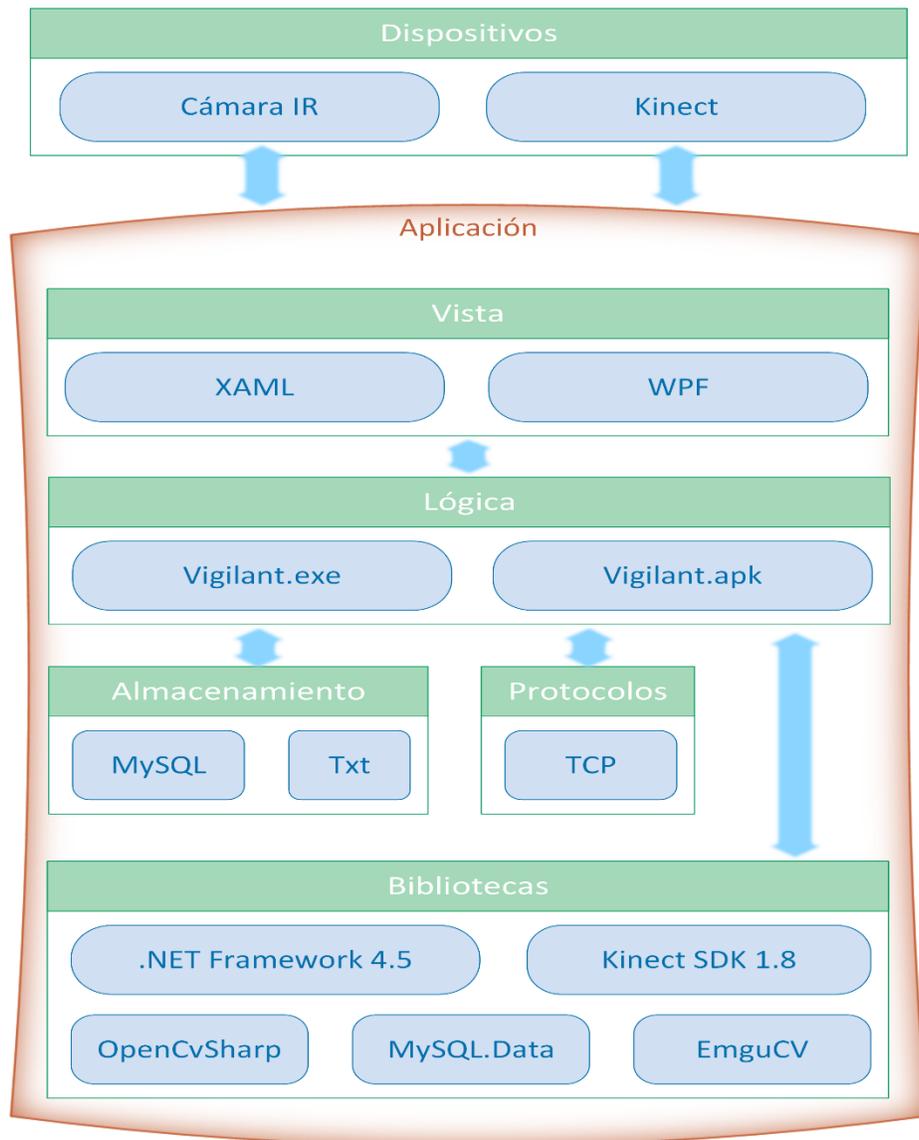


Ilustración 6 Arquitectura del sistema

En su parte física el sistema consta de 2 dispositivos encargados de capturar la información y enviarla al pc.

La cámara IR es un sencilla cámara web modificada para capturar el rostro, visualizado mediante las luces infrarrojas instaladas en dicha cámara. Kinect es el sensor encargado de capturar las posiciones de las partes de la cara, lo cual es de gran importancia para determinar los estados del sujeto.

La capa de vista o GUI es con la cual interactúa el usuario, en ésta se solicitan los datos. Es sencilla e intuitiva y esta desarrolla bajo el sistema de presentación Windows Presentation Foundation (WPF). Esta capa se comunica con la capa lógica.

La capa lógica consta de 2 partes, Vigilant.apk es la App instalada en el móvil Android diseñada para recibir y notificar alertas. Vigilant.exe consta de todas las clases que hacen funcionar al sistema, aquí se procesa la información capturada por los dispositivos y se toman decisiones, ésta se comunica con la capa de almacenamiento, con la capa protocolos y con la capa bibliotecas donde se encuentran los ensamblados externos necesarios.

La capa de almacenamiento es utilizada por la capa lógica para almacenar información en la base de datos creada en MySQL. También se utiliza un archivo .txt directamente relacionado con el inicio de sesión automático (ver diagrama de secuencia, descripción de casos de uso).

En la capa protocolos se encuentra establecido el protocolo a través del cual se comunica el software con la aplicación móvil. Debido a que era necesario un medio confiable que garantizara que la información necesaria llegara a su destino, se optó por utilizar el protocolo de comunicación TCP.

La capa bibliotecas es donde se encuentran todos los ensamblados externos que utiliza la capa lógica para que funcionen tanto los dispositivos como los algoritmos

implementados. En dicha capa se encuentran los siguientes ensamblados: .Net Framework, Kinect SDK 1.8, OpenCvSharp, MySQL.Data y EmguCV.

Kinect necesita .Net Framework y Kinect SDK 1.8 para su funcionamiento. OpenCvSharp y EmguCV son utilizados por la cámara IR y MySQL.Data es quien permite que la información pueda ser almacenada en la base de datos.

5.2. Diseño del sistema

5.2.1. Especificación de requisitos

5.2.1.1. Objetivos del sistema

OBJ-01 GESTIONAR USUARIO	
Descripción	El sistema permitirá crear usuarios y actualizar datos de configuración de los dispositivos utilizados.
Comentarios	Para el registro el sistema deberá capturar un fotograma del rostro del usuario seguido de la información personal, de esta manera el usuario tendrá 2 opciones de iniciar sesión, la primera de estas es a través del reconocimiento de la persona utilizando la cámara IR y la segunda manera es manualmente donde el usuario podrá ingresar al sistema utilizando la cedula y contraseña registrada.

Tabla 1 Objetivo gestionar usuario

OBJ-02 Gestionar procesos	
Descripción	El sistema deberá detectar al usuario e iniciar el proceso de monitoreo de expresiones faciales para extraer características de síntomas de somnolencia
Comentarios	El sistema será capaz de identificar que algoritmo utilizar debido a que éste contiene 2 subsistemas independientes, uno es el encargado de utilizar kinect cuya función es el monitoreo del usuario durante el día. El otro proceso autoriza la cámara IR para el monitoreo nocturno. El sistema también deberá iniciar el servidor TCP el cual permitirá enviar alertas de presencia de somnolencia al usuario a través de un cliente móvil Android. Durante la noche el usuario podrá si así lo desea, observar cómo se está analizando el estado de su ojo y/o estado de su boca

Tabla 2 Objetivo gestionar procesos

OBJ-02 Gestionar notificación	
Descripción	El sistema móvil deberá recibir mensajes del servidor para generar alarmas que alerten al usuario.
Comentarios	Luego de que el sistema detecte presencia de somnolencia enviará un mensaje vía TCP al celular para que éste active un sonido que pueda ser escuchado por el usuario.

Tabla 3 Objetivo gestionar usuario

5.2.1.2. Requisitos de información

RI- 01 INFORMACIÓN ACERCA DEL USUARIO	
Objetivos asociados	OBJ -01
Descripción	El sistema tendrá la capacidad para almacenar información sobre los usuarios que deseen utilizar el sistema.
Información necesaria	Los datos que el sistema solicitará son: nombre, apellido, cedula, contraseña y un fotograma del rostro del usuario.

Tabla 4 Información acerca del usuario

RI-02 DATOS DE CONFIGURACIÓN DE LOS DISPOSITIVOS	
Objetivos asociados	OBJ -01
Descripción	El sistema podrá almacenar información sobre las configuraciones de los dispositivos utilizados.
Información necesaria	Kinect: posición de la cabeza, posición del motor Cámara: coordenadas X y Y del rectángulo del ojo, ancho y alto de rectángulo del ojo, coordenadas X y Y del rectángulo de la boca, ancho y alto de rectángulo de la boca.

Tabla 5 Datos de configuración de dispositivos

5.2.1.3. Requisitos no funcionales

Nombre	Descripción
Interfaz	El software de monitorio VIGILANT deberá contar con una interfaz sencilla y amigable a la vista del usuario.
Usabilidad	La sencilla interfaz del software harán a VIGILANT un sistema intuitivo para el usuario, con una fácil manipulación de los componentes, en donde solo será cuestión de minutos para comprender completamente el manejo de la aplicación
Sistema no invasivo	El usuario no deberá tener ningún tipo de accesorio sobre su cuerpo para que el sistema funcione. Los dispositivos serán ubicados de tal forma que no afecten o incomoden la visibilidad del usuario mientras este conduce. La alerta de notificación será de tipo auditiva para que el usuario no distraiga su mirada de la carretera.

Tabla 6 Requisitos no funcionales

5.2.2. Modelo relacional

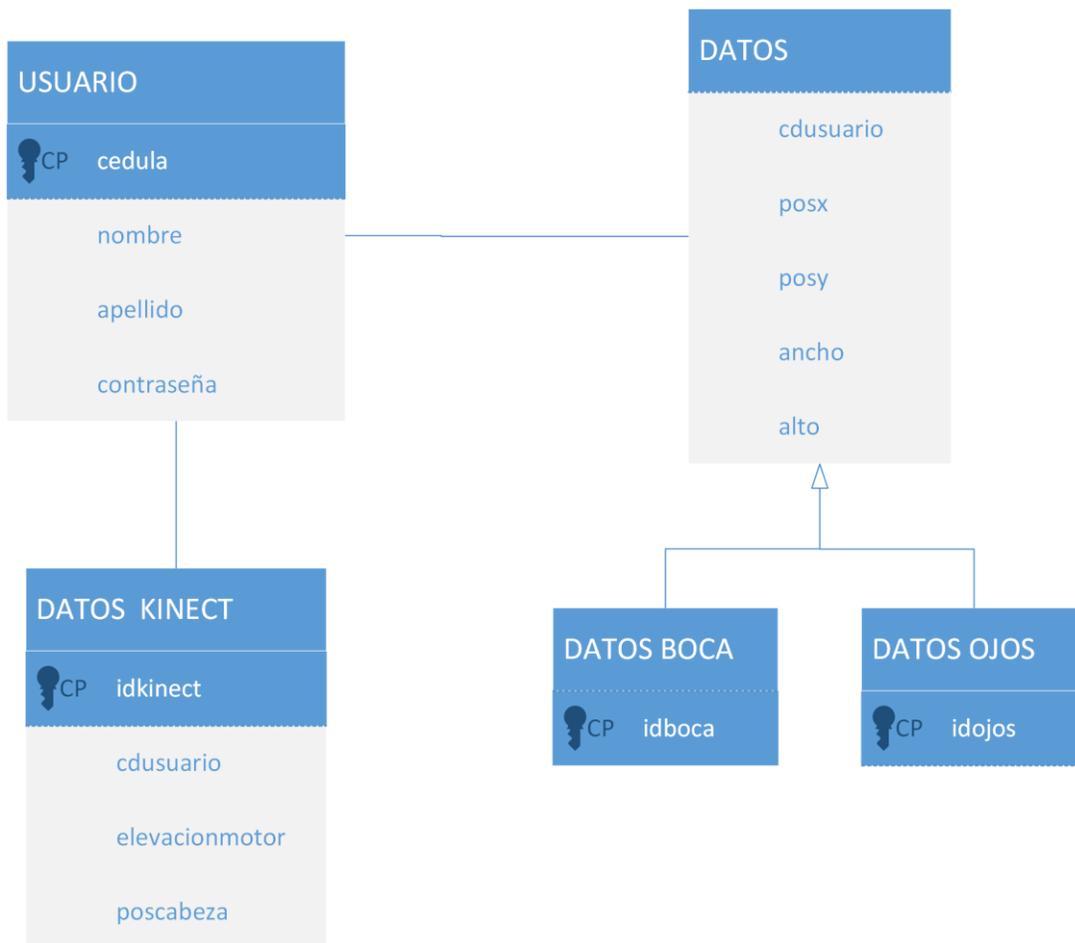


Ilustración 7 Modelo relacional BD Vigilant

5.2.3. Diagrama de clases

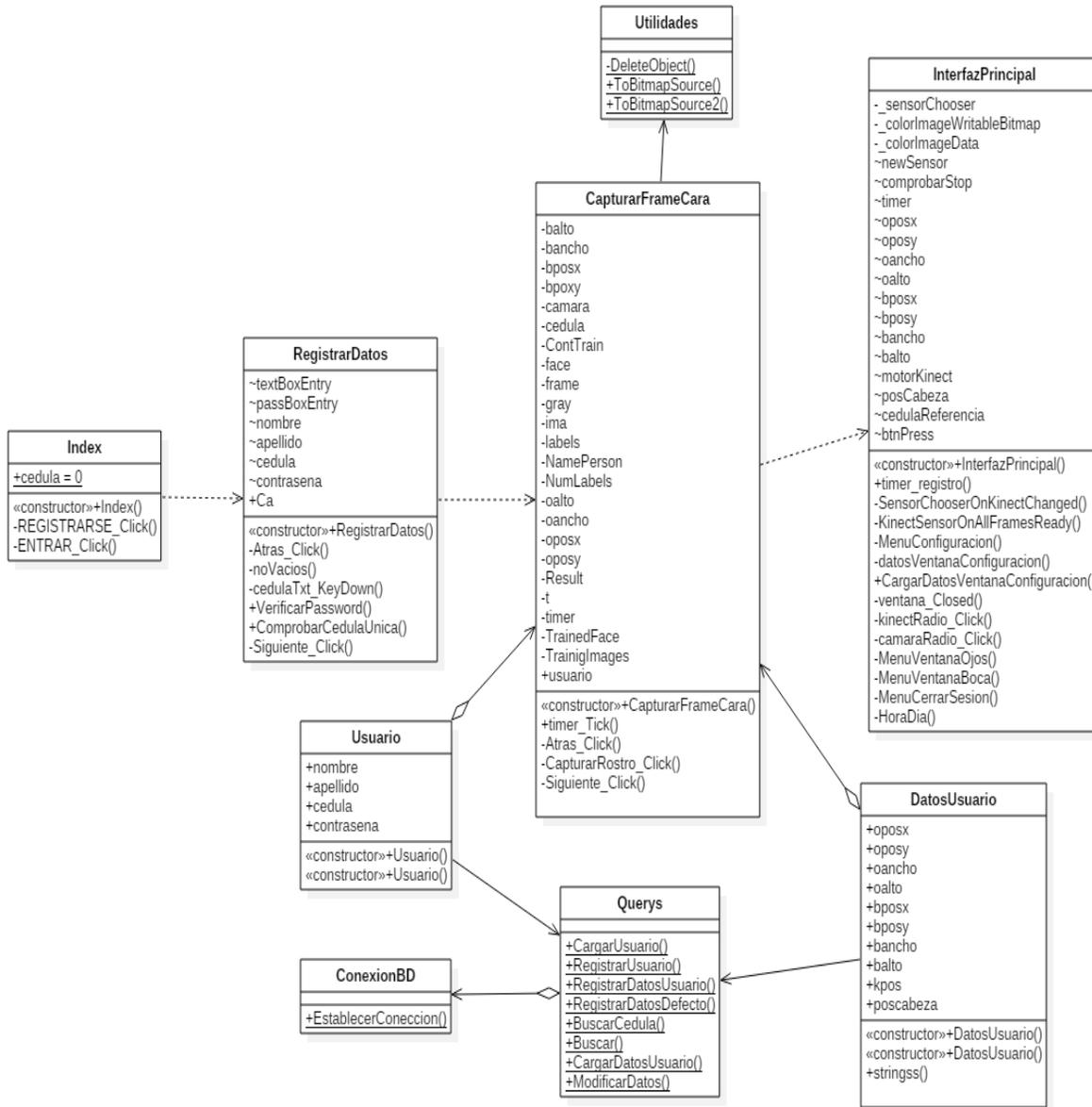


Ilustración 8 Diagrama de clases registrarse

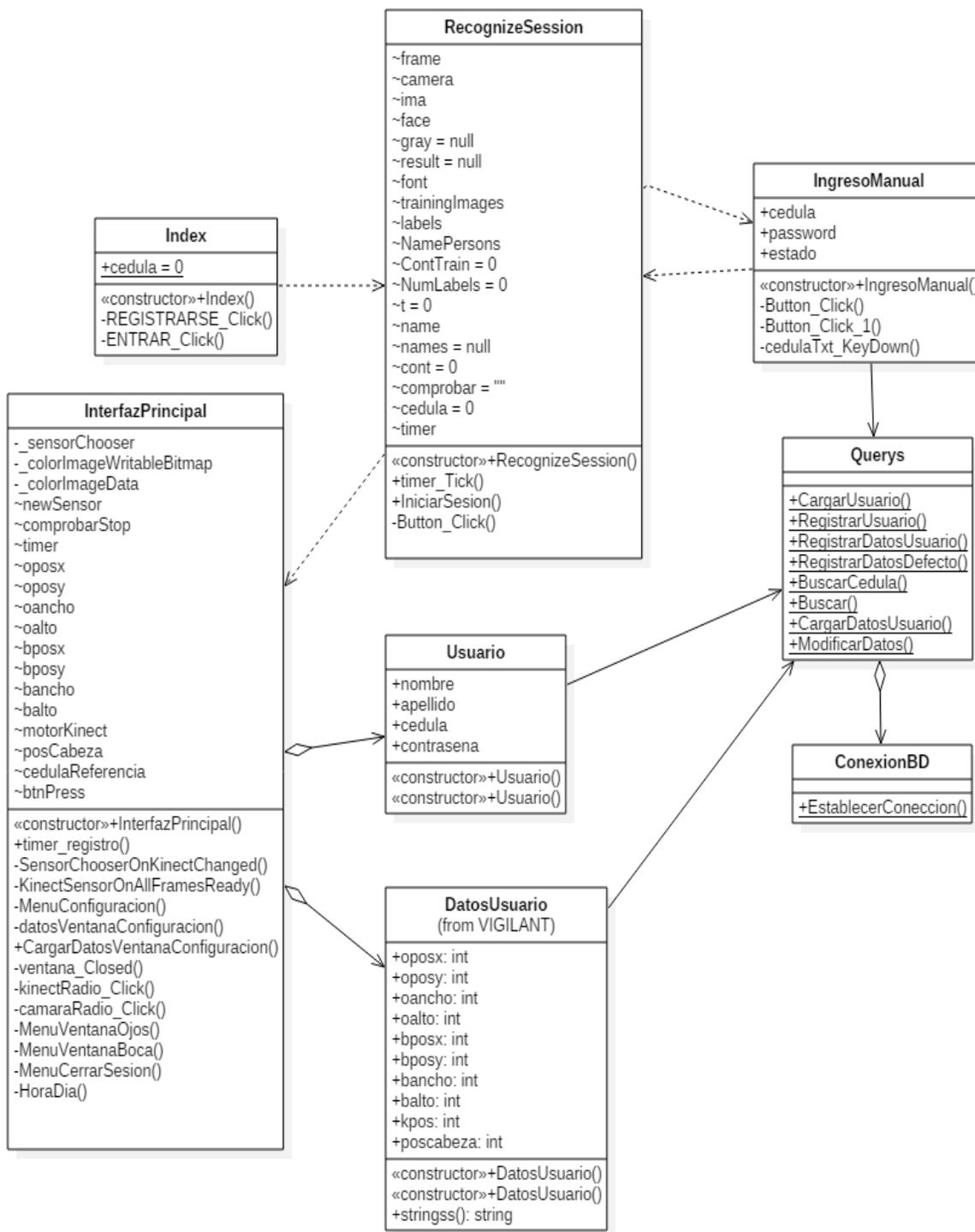


Ilustración 9 Diagrama de clases iniciar sesión

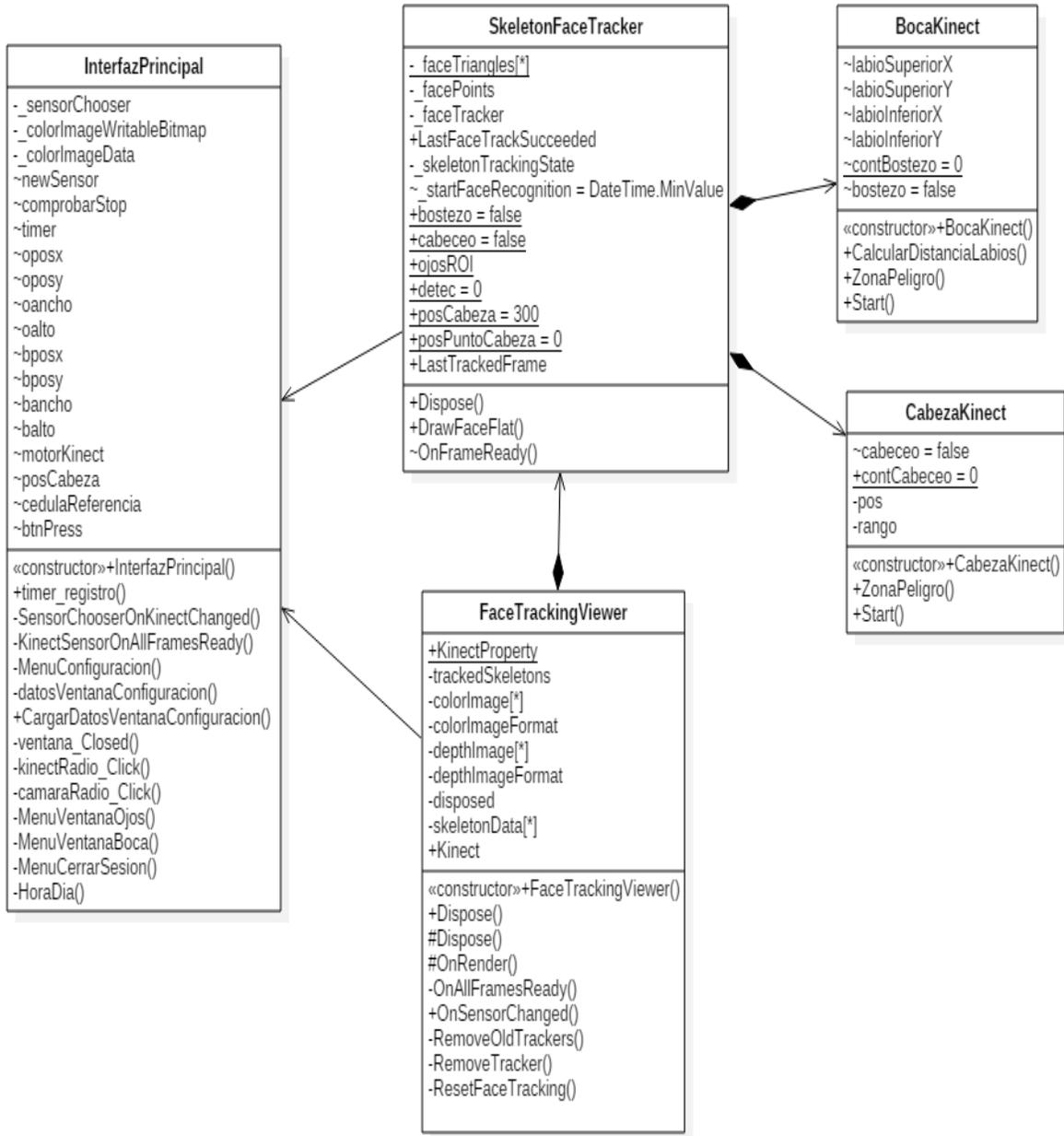


Ilustración 10 Diagrama de clases Kinect

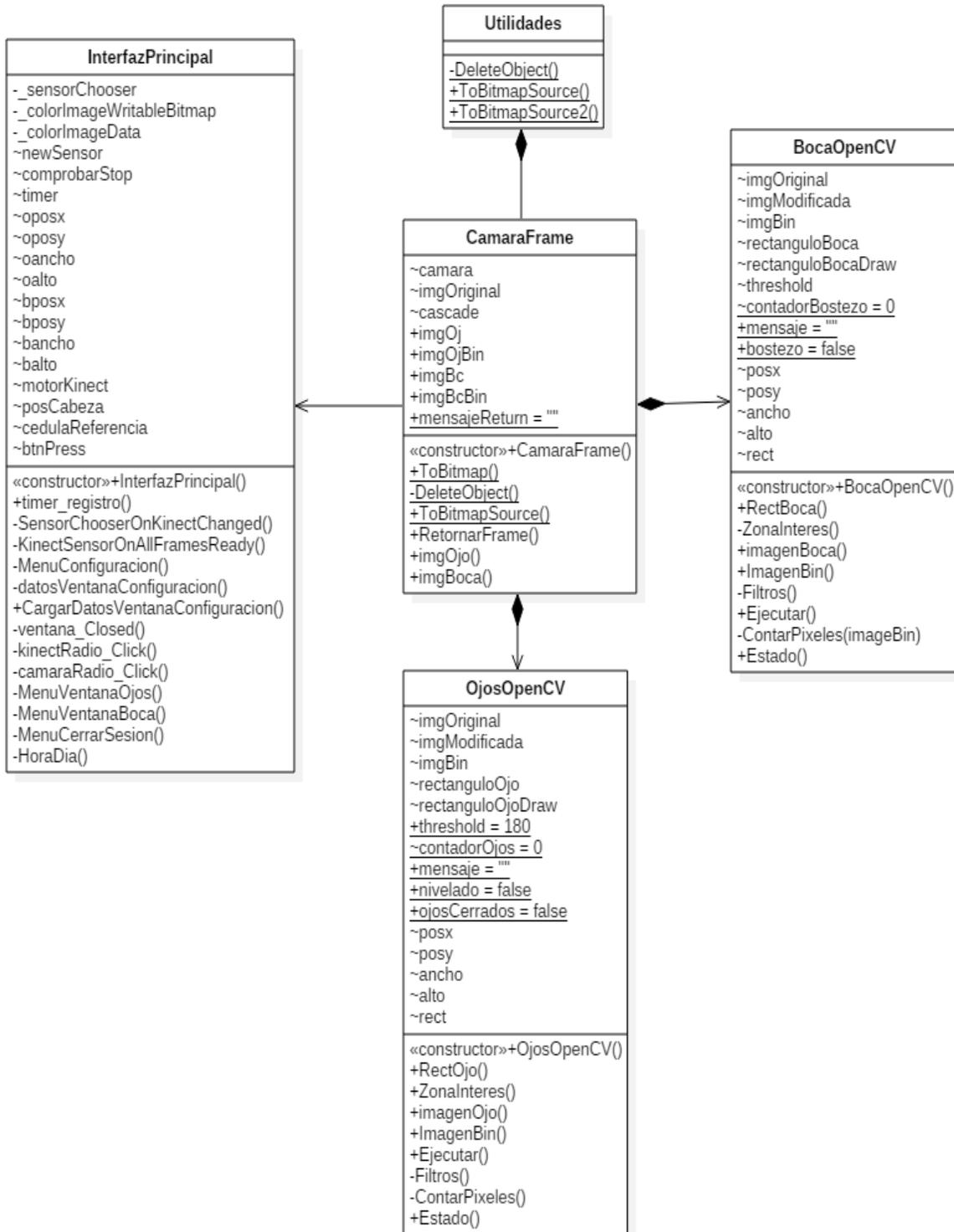


Ilustración 11 Diagrama de clases cámara

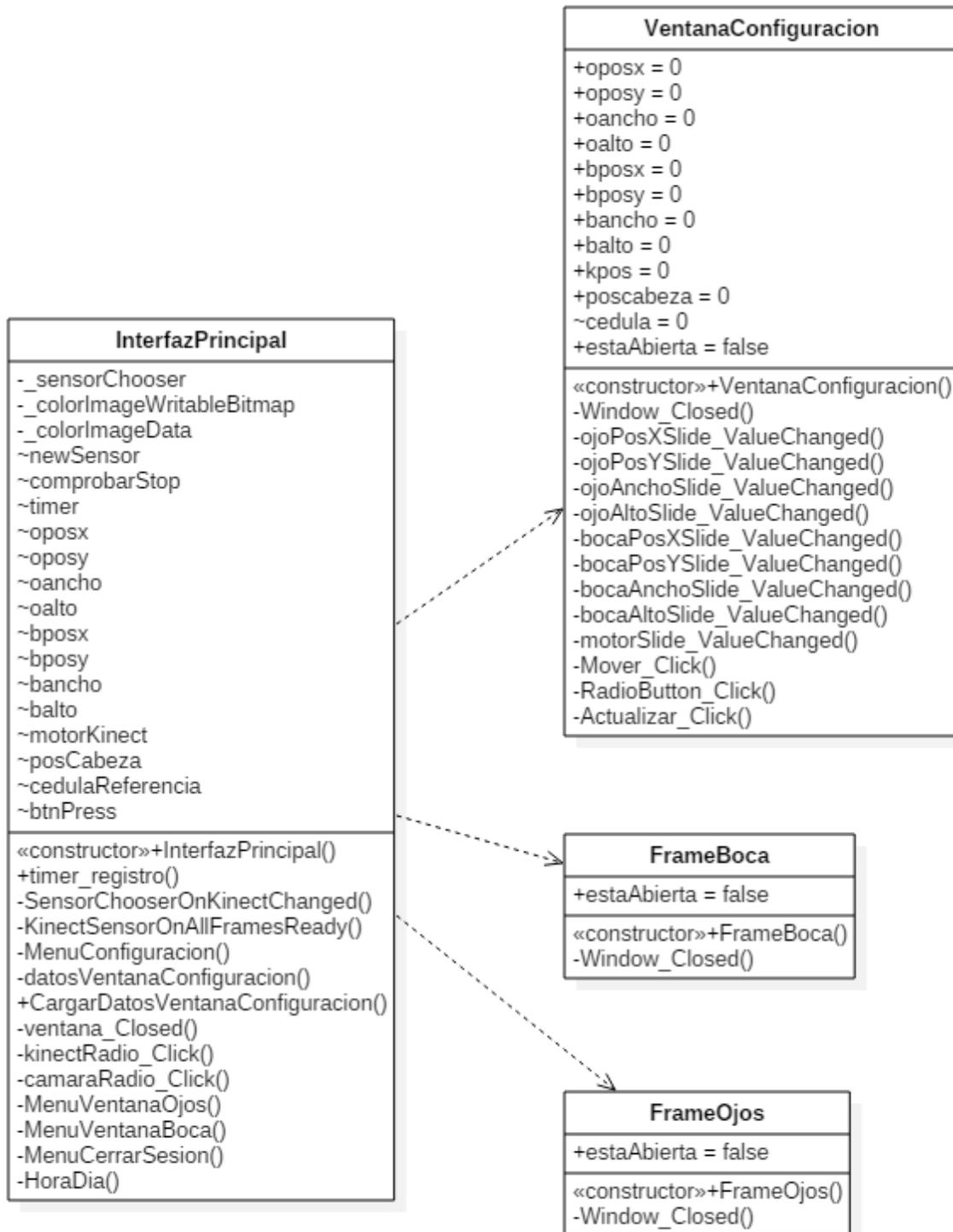


Ilustración 12 Diagrama de clases sistema

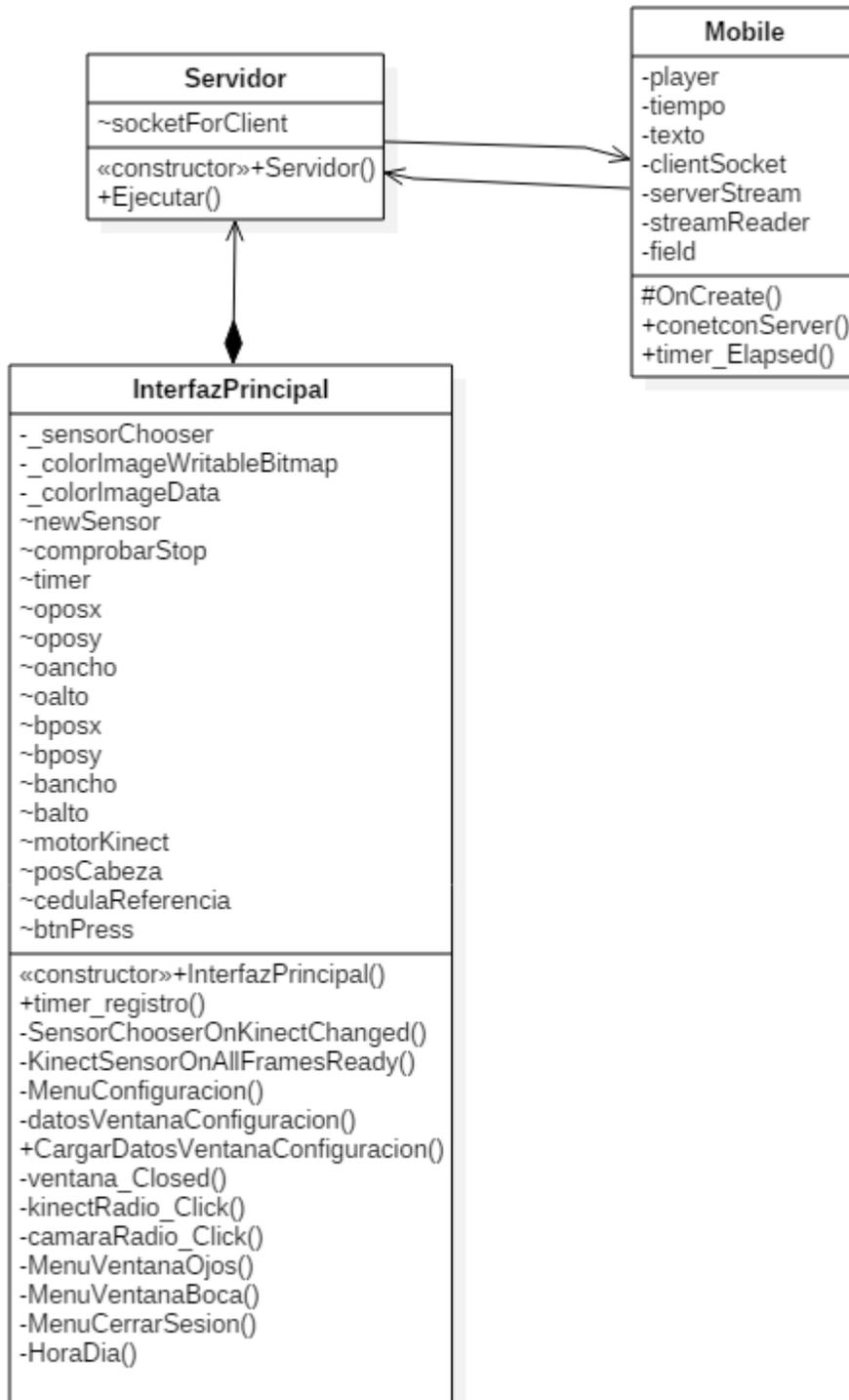


Ilustración 13 Diagrama de clases alarma

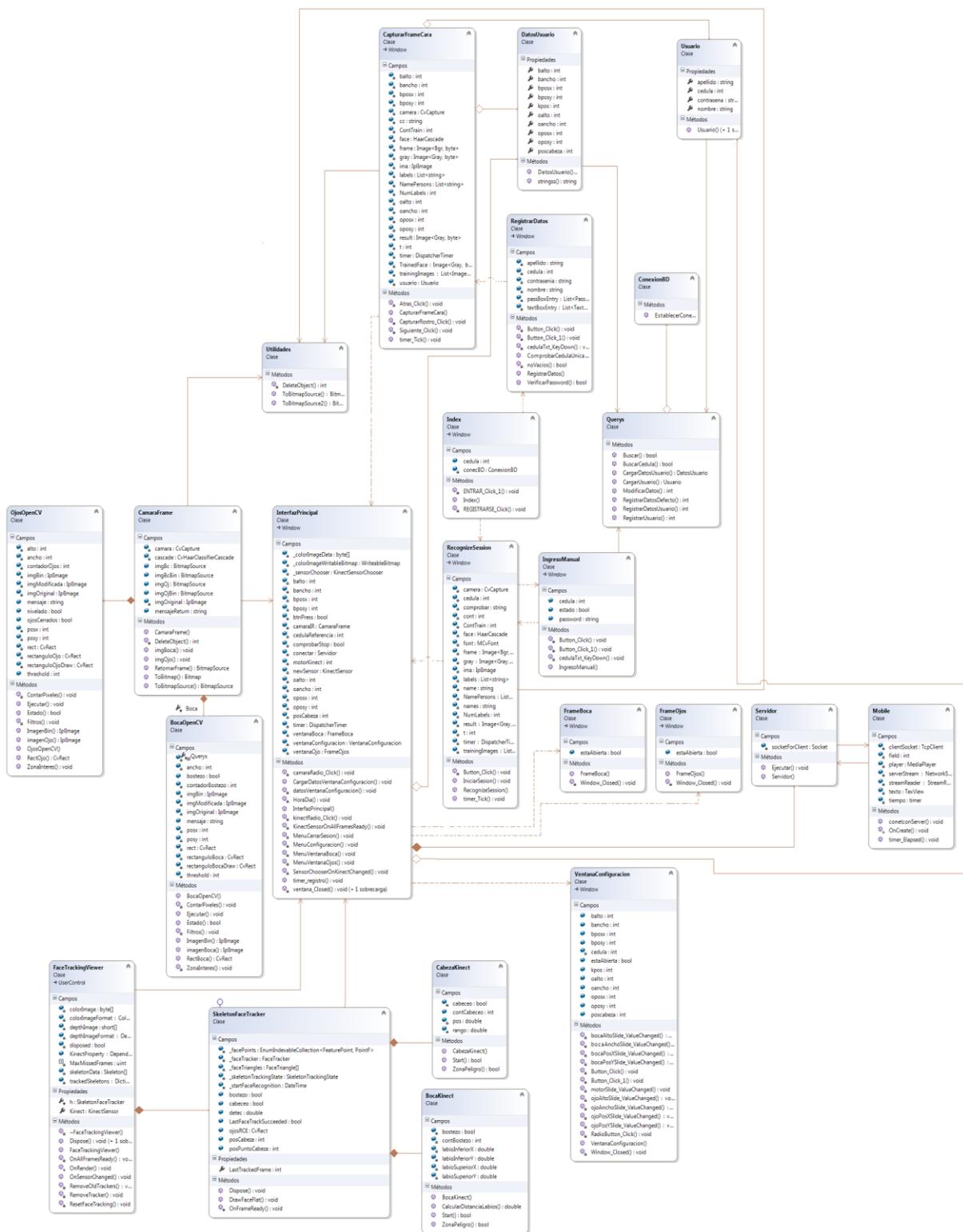


Ilustración 14 Diagrama de clases sistema completo

5.2.4. Documentación de actores

ACTORES	CASOS DE USO	DESCRIPCIÓN
Usuario	Registro de usuario, iniciar sesión manualmente, iniciar sesión con reconocimiento facial, actualizar datos de configuración de dispositivos, ver proceso del ojo, ver proceso de la boca.	Este actor representa a la persona encargada de administrar el sistema, no necesariamente se hace referencia al dueño del vehículo si no a quien lo desee manejar. Cualquier usuario posee la facultad de registrarse en el sistema, pero una vez registrado, éste no será eliminado. Así mismo podrá acceder al sistema con uno de los 2 métodos existentes para el inicio de sesión. Luego a haber ingresado al sistema si no se encuentra a gusto con las opciones predeterminadas de configuración de los dispositivos podrá modificarlas a su antojo.

Tabla 7 Documentación de actores

5.2.5. Diagramas de casos de uso

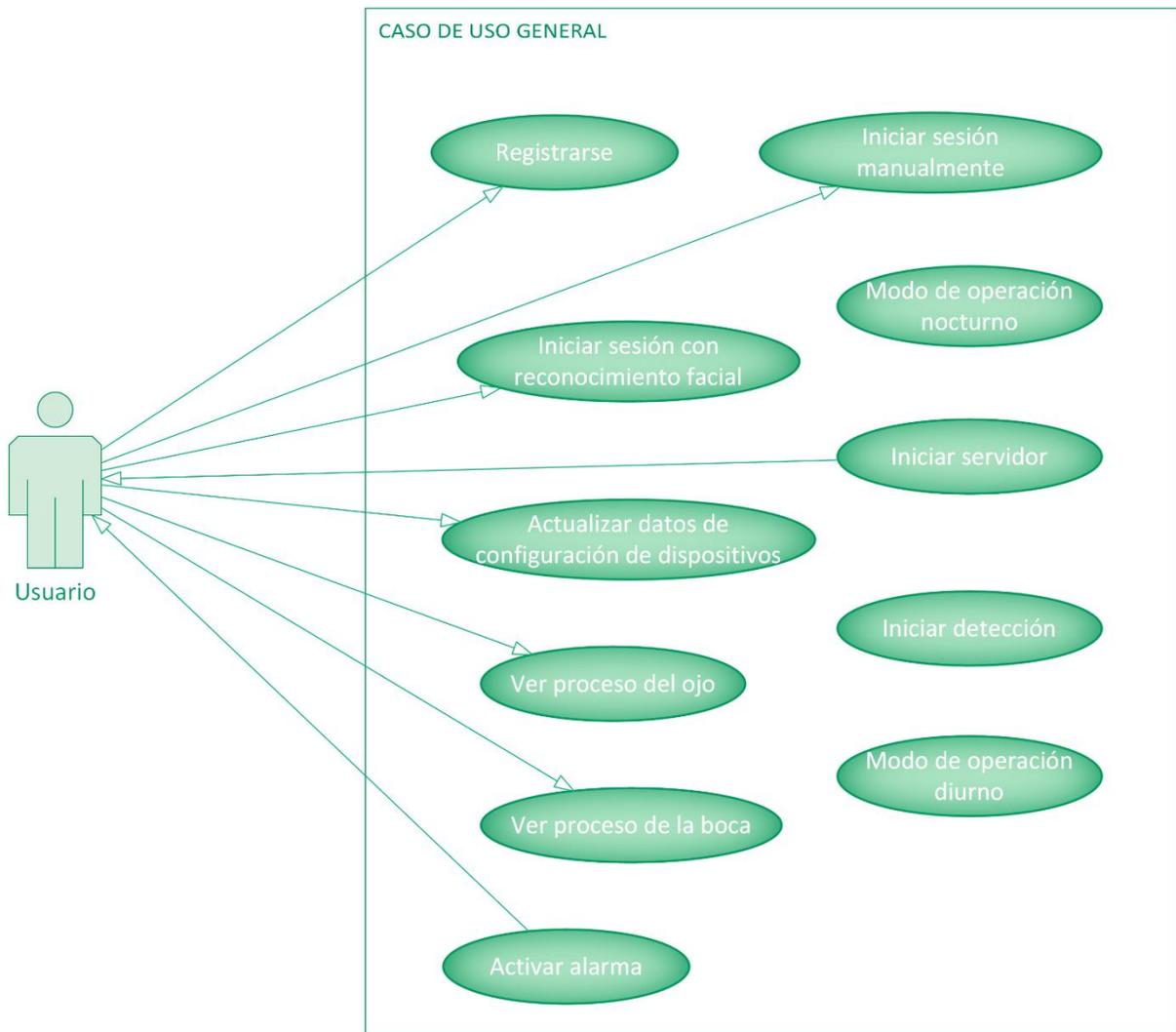


Ilustración 15 Caso de uso general

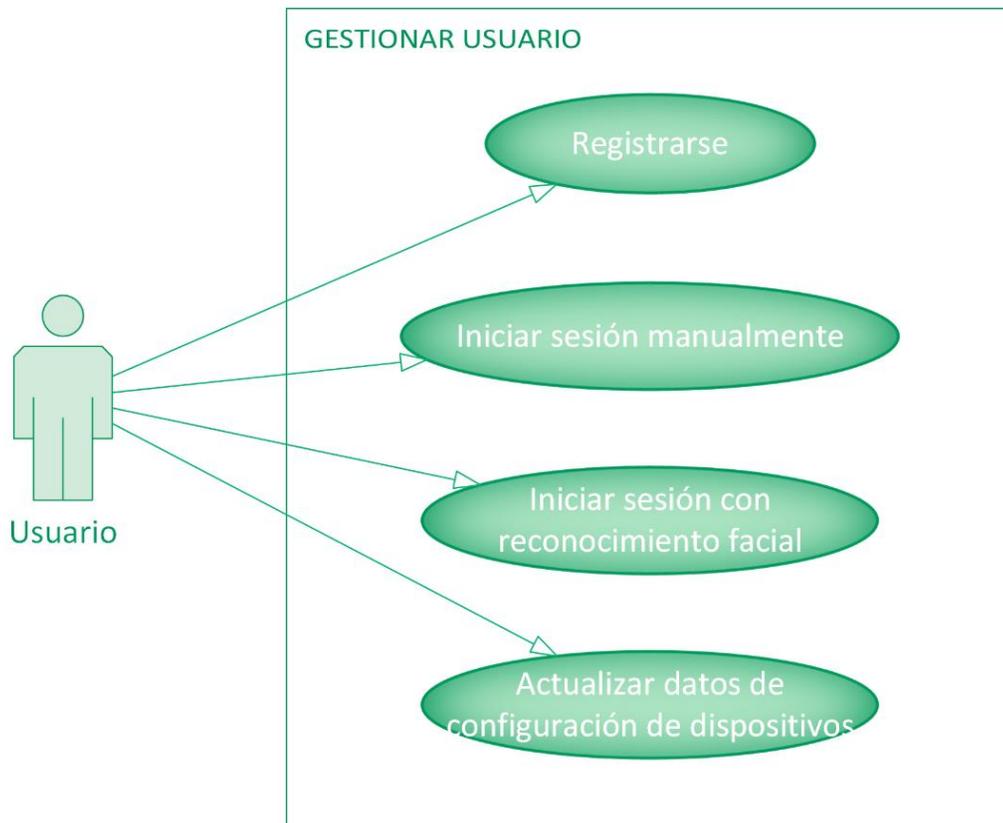


Ilustración 16 Caso de uso gestionar usuario

DOCUMENTACIÓN CASO DE USO – 01 REGISTRO DE USUARIO	
Objetivos vinculados	OBJ – 01
Requisitos de información vinculados	RI – 01 RI – 02
Descripción del caso de uso	El sistema permitirá a los usuarios que deseen conducir el vehículo registrarse con su información personal y de manera obligatoria también es necesario capturar un fotograma del rostro.
Actores	Usuario
Acciones	El usuario debe ingresa los datos solicitados. El sistema valida que la información sea correcta.

	<p>El sistema valida coincidencias en la base de datos.</p> <p>El sistema solicitara un fotograma del rostro del usuario.</p> <p>El sistema guardara los datos.</p> <p>Se mostrará un mensaje de aceptación.</p>
--	--

Tabla 8 Documentación caso de uso registro de usuario

DOCUMENTACIÓN CASO DE USO – 02 INICIAR SESIÓN MANUALMENTE	
Objetivos vinculados	OBJ – 01
Requisitos de información vinculados	RI – 01
Descripción del caso de uso	El sistema permitirá al usuario acceder a éste a través de su cedula y contraseña registrada en caso de que el inicio de sesión utilizando reconocimiento fracase o por decisión propia.
Actores	Usuario
Acciones	<p>El usuario seleccionara la opción de iniciar manualmente</p> <p>El usuario suministrara los datos solicitados por el sistema</p> <p>El sistema validara en base de datos si datos son correctos</p> <p>El sistema devolverá respuesta al usuario</p>

Tabla 9 Documentación caso de uso iniciar sesión manualmente

DOCUMENTACIÓN CASO DE USO – 03 INICIAR SESIÓN CON RECONOCIMIENTO FACIAL	
Objetivos vinculados	OBJ – 01
Requisitos de información vinculados	Ninguno
Descripción del caso de uso	El sistema permitirá al usuario iniciar sesión con tan solo enfocar su mirada hacia la cámara, ésta será capaz de identificar quien es la persona que se encuentra al volante y cargara sus datos y configuraciones desde la base de datos.
Actores	Usuario
Acciones	No seleccionar el modo de ingresar manualmente, por defecto el sistema iniciará con reconocimiento facial. El sistema buscara los fotogramas guardados y los comparará con el rostro de la persona. Las condiciones de iluminación deben ser las mismas o similares a las registro, para evitar falsos positivos.

Tabla 10 Documentación caso de uso iniciar sesión con reconocimiento facial

DOCUMENTACIÓN CASO DE USO – 04 ACTUALIZAR CONFIGURACIÓN DE DISPOSITIVOS	
Objetivos vinculados	OBJ – 01
Requisitos de información vinculados	RI – 02
Descripción del caso de uso	El usuario podrá modificar los datos asociados a las configuraciones de dispositivos en caso de que los

	proporcionados por el sistema no cumplan con las especificaciones faciales de la persona.
Actores	Usuario
Acciones	<p>El usuario abrirá la ventana de configuraciones</p> <p>El usuario modificará los datos que necesite</p> <p>El usuario enviará los datos modificados</p> <p>El sistema procesará los cambios en la base de datos</p> <p>El sistema notificará la actualización</p>

Tabla 11 Documentación caso de uso actualizar configuración de dispositivos

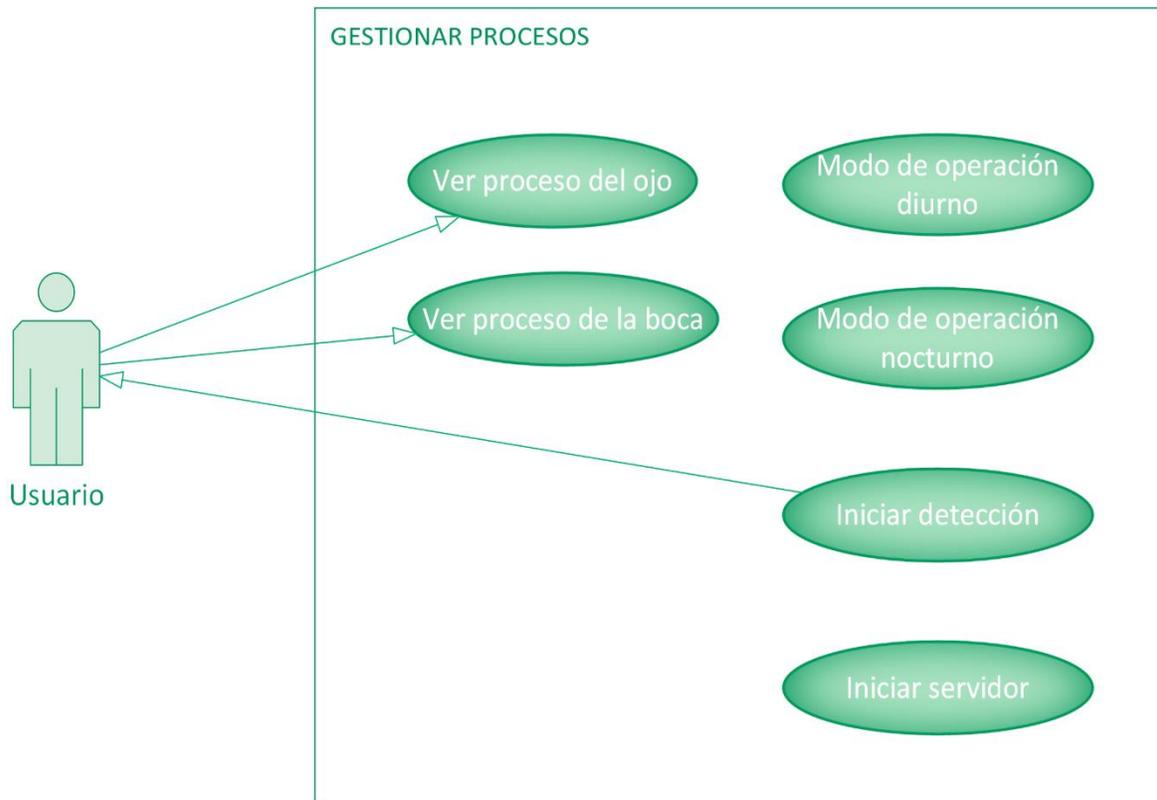


Ilustración 17 Caso de uso gestionar procesos

DOCUMENTACIÓN CASO DE USO – 05 VER PROCESO DEL OJO	
Objetivos vinculados	OBJ – 02
Requisitos de información vinculados	Ninguno
Descripción del caso de uso	El usuario podrá observar la forma el que se monitorea el estado de su ojo durante la noche, de esta manera podrá determinar si las configuraciones predeterminadas no son las óptimas para su ojo, es decir, si éste se encuentra por fuera de los márgenes del rectángulo que lo contiene.
Actores	Usuario
Acciones	El usuario deberá estar logueado en el sistema El usuario tendrá que acceder a la opción de menú y seleccionar mostrar ojo.

Tabla 12 Documentación caso de uso ver proceso del ojo

DOCUMENTACIÓN CASO DE USO – 06 VER PROCESO DE LA BOCA	
Objetivos vinculados	OBJ – 02
Requisitos de información vinculados	Ninguno
Descripción del caso de uso	El usuario podrá observar la forma el que se monitorea el estado de su boca durante la noche, de esta manera podrá determinar si las configuraciones predeterminadas no son las óptimas para su boca, es decir, si ésta se encuentra por fuera de los márgenes del rectángulo que la contiene.

Actores	Usuario
Acciones	El usuario deberá estar logueado en el sistema El usuario tendrá que acceder a la opción de menú y seleccionar mostrar boca.

Tabla 13 Documentación caso de uso ver proceso de la boca

DCU – 07 MODO DE OPERACIÓN DIURNO	
Objetivos vinculados	OBJ – 02
Requisitos de información vinculados	RI – 01 RI – 02
Descripción del caso de uso	El sistema iniciará el Kinect y procesará los algoritmos asociados a este dispositivo.
Actores	Sistema
Acciones	El sistema verificará que el horario sea superior a las 7:00 de la mañana El sistema iniciará el monitoreo de estado de somnolencia

Tabla 14 Documentación caso de uso modo de operación diurno

DCU – 08 MODO DE OPERACIÓN NOCTURNO	
Objetivos vinculados	OBJ – 02
Requisitos de información vinculados	RI – 01 RI – 02
Descripción del caso de uso	El sistema iniciará la cámara IR y procesara los algoritmos asociados a este dispositivo.
Actores	Sistema

Acciones	<p>El sistema verificará que el horario sea superior a las 6:00 de la noche</p> <p>El sistema iniciará el monitoreo de estado de somnolencia.</p>
-----------------	---

Tabla 15 Documentación caso de uso modo de operación nocturno

DCU – 09 INICIAR DETECCIÓN	
Objetivos vinculados	OBJ – 02
Requisitos de información vinculados	RI – 02
Descripción del caso de uso	El sistema detectara el rostro del usuario independiente de cuál de los dispositivo este trabajando
Actores	Sistema
Acciones	<p>El sistema iniciara el monitoreo con el dispositivo indicado de acuerdo al horario</p> <p>El sistema detectara el rostro de la persona</p> <p>El sistema aplicara los algoritmos para verificar estados de somnolencia.</p>

Tabla 16 Documentación caso de uso iniciar detección

DCU – 10 INICIAR SERVIDOR	
Objetivos vinculados	OBJ – 02
Requisitos de información vinculados	RI – 02
Descripción del caso de uso	Cuando el usuario entre a la interfaz que maneja los proceso principales del sistema, automáticamente

	iniciara el servidor TCP y permanecerá en espera hasta que un cliente (dispositivo móvil) se conecte para enviar las señales de detección de somnolencia.
Actores	Sistema
Acciones	<p>El usuario deberá loguearse en el sistema</p> <p>El sistema iniciara el servido TCP</p> <p>El sistema permanecerá en espera hasta que un cliente se conecte</p> <p>Le sistema podrá enviarle información al cliente</p>

Tabla 17 Documentación caso de uso iniciar servidor

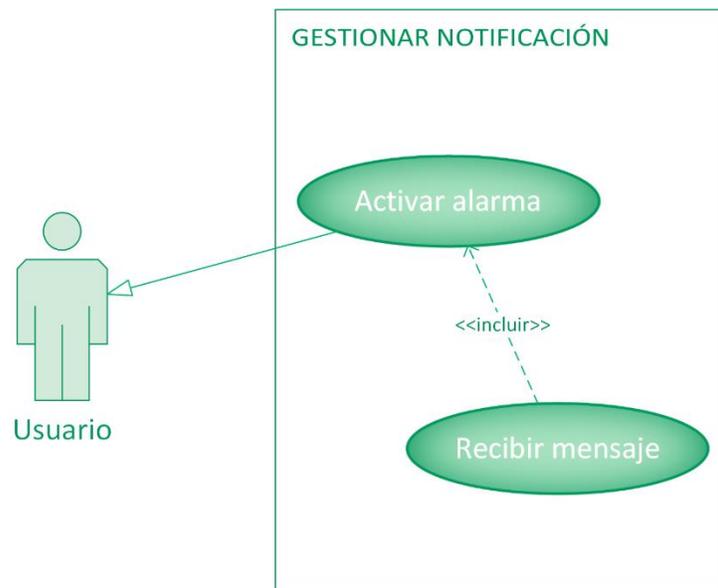


Ilustración 18 Caso de uso gestionar notificación

DCU – 11 RECIBIR MENSAJE	
Objetivos vinculados	OBJ – 03
Requisitos de información vinculados	Ninguno

Descripción del caso de uso	La aplicación móvil Android recibirá todos los mensajes enviados desde el servidor.
Actores	App
Acciones	El usuario deberá tener instalado Vigilant.apk El usuario deberá iniciar Vigilant.apk sincronamente con el inicio de sesión

Tabla 18 Documentación caso de uso recibir mensaje

DCU – 12 ACTIVAR ALARMA	
Objetivos vinculados	OBJ – 03
Requisitos de información vinculados	Ninguno
Descripción del caso de uso	El aplicación móvil Android activará una alerta auditiva cada vez que ésta reciba mensajes desde el servidor
Actores	App
Acciones	El usuario deberá tener instalado Vigilant.apk El usuario deberá iniciar Vigilant.apk sincronamente con el inicio de sesión.

Tabla 19 Documentación caso de uso activar alarma

5.2.6. Diagramas de secuencia

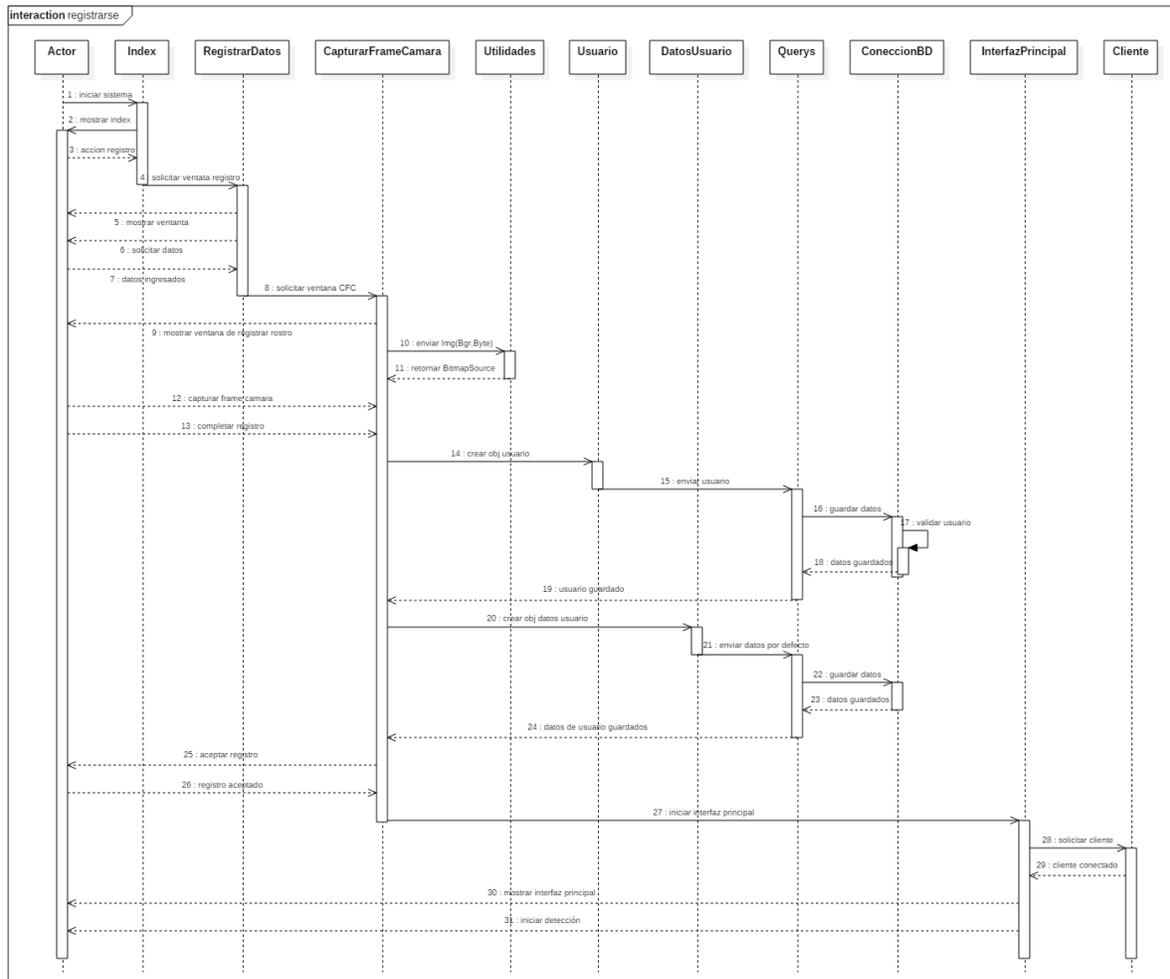


Ilustración 19 Diagrama de secuencia registrarse

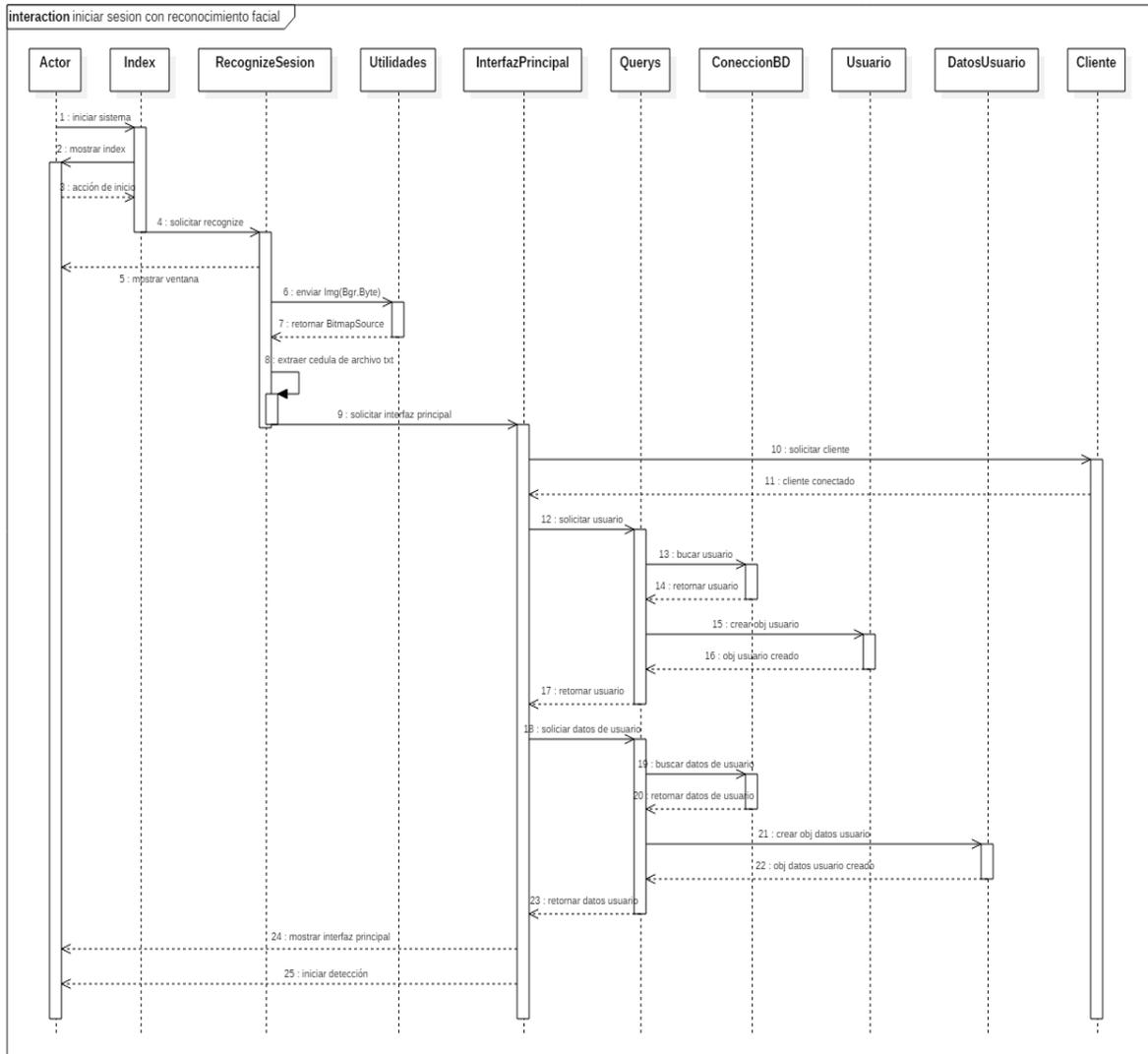


Ilustración 20 Diagrama de secuencia iniciar sesión con reconocimiento facial

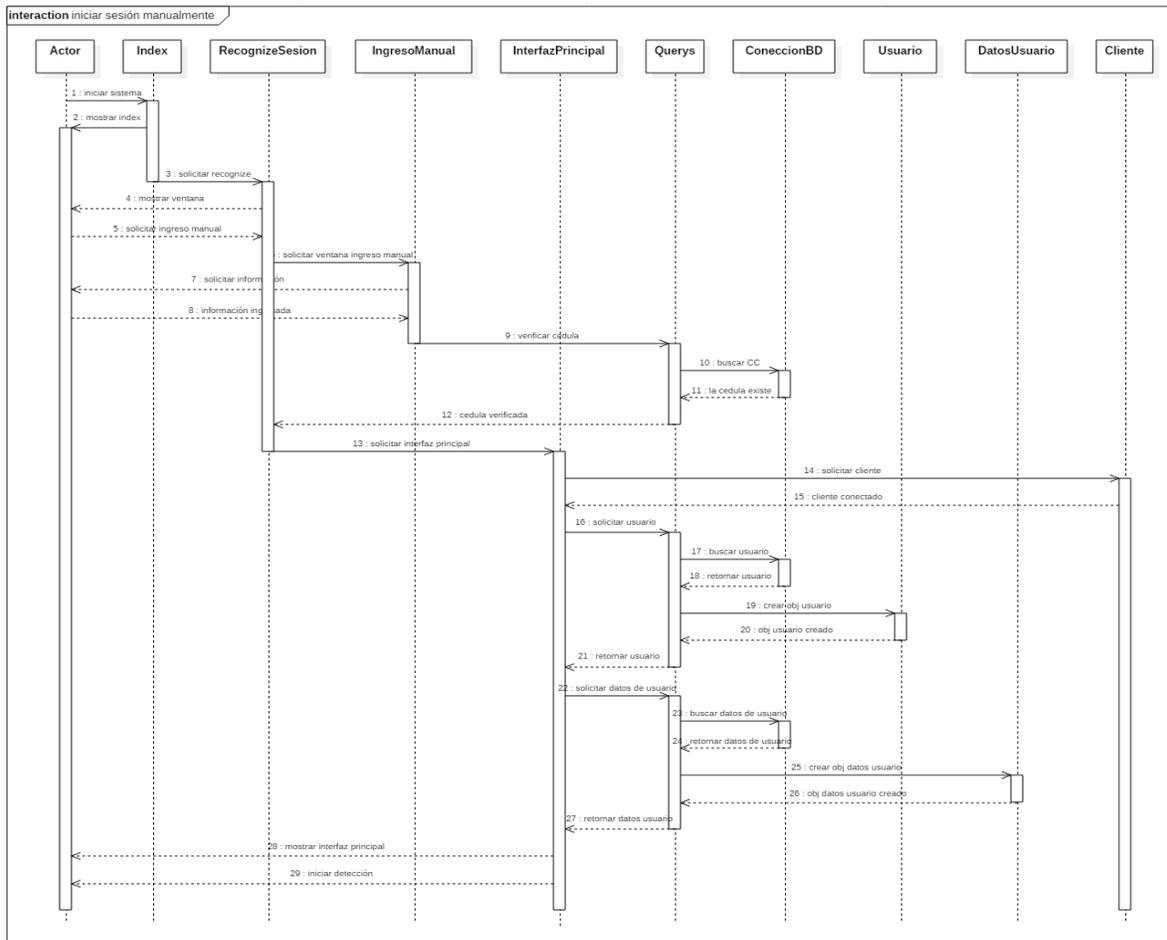


Ilustración 21 Diagrama de secuencia iniciar sesión manualmente

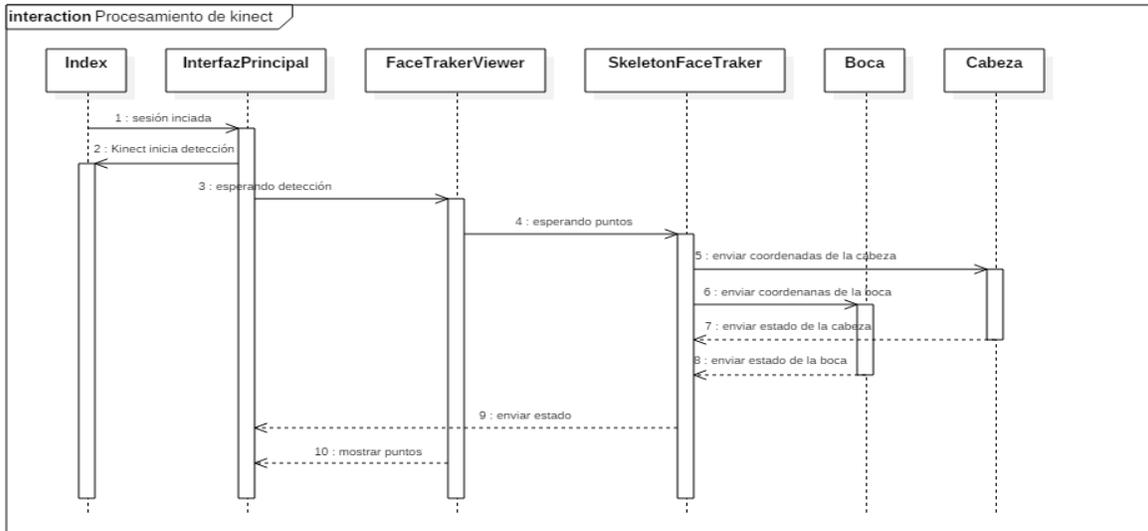


Ilustración 22 Diagrama de secuencia procesamiento de Kinect

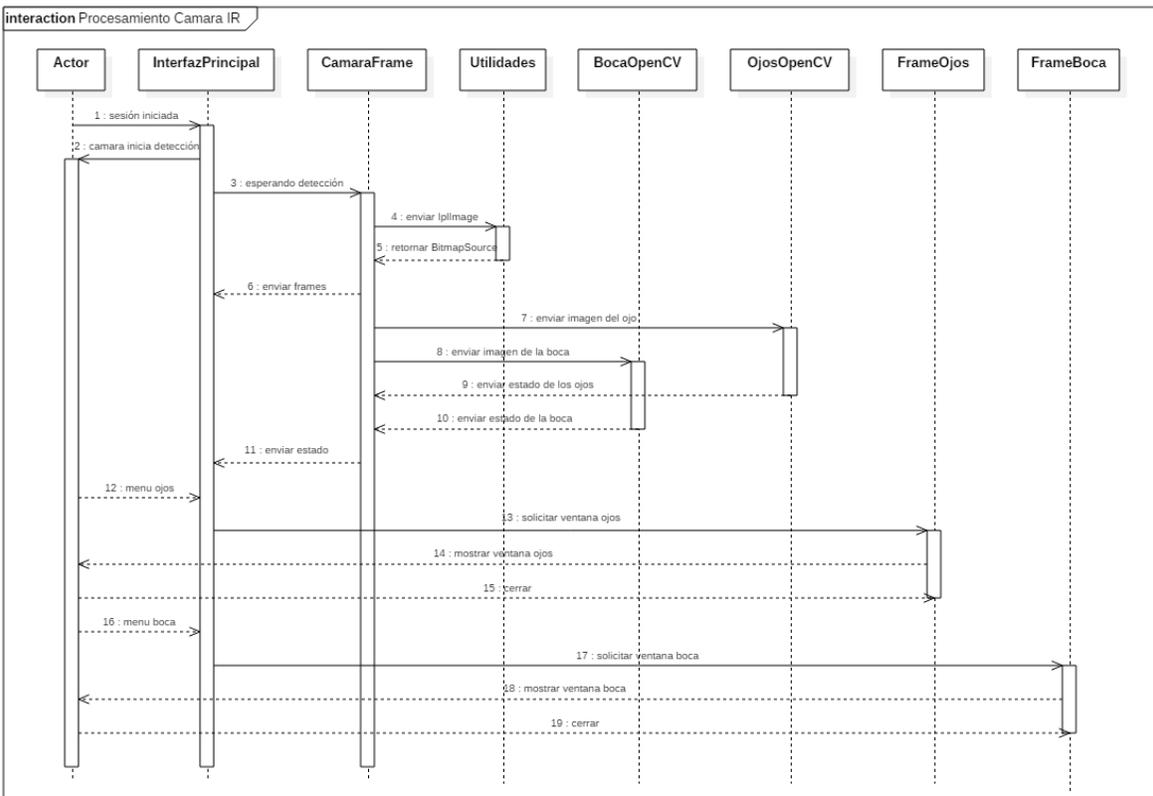


Ilustración 23 Diagrama de secuencia procesamiento cámara IR

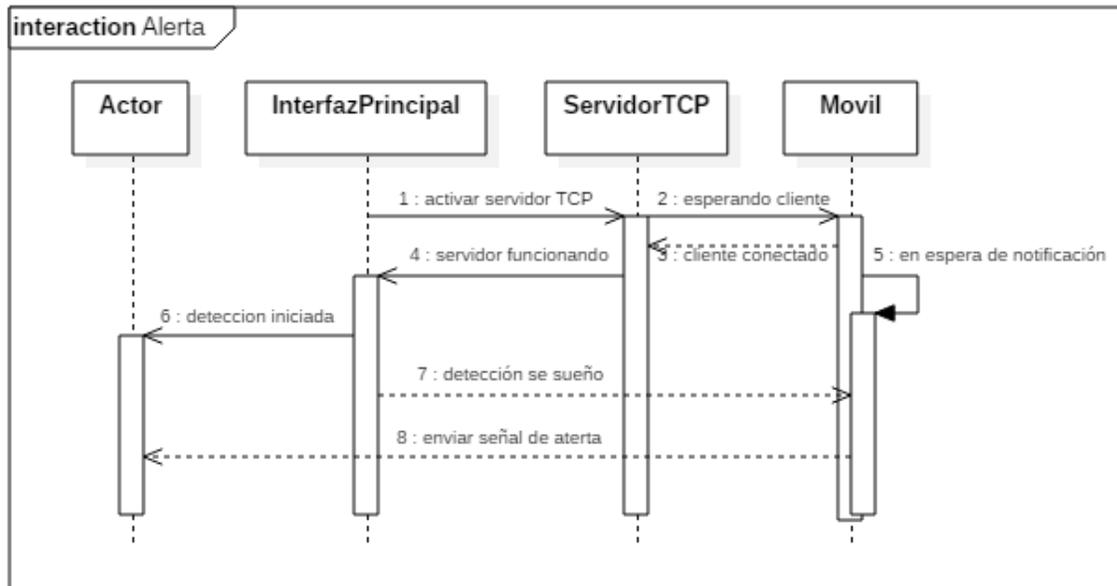


Ilustración 24 Diagrama de secuencia alerta

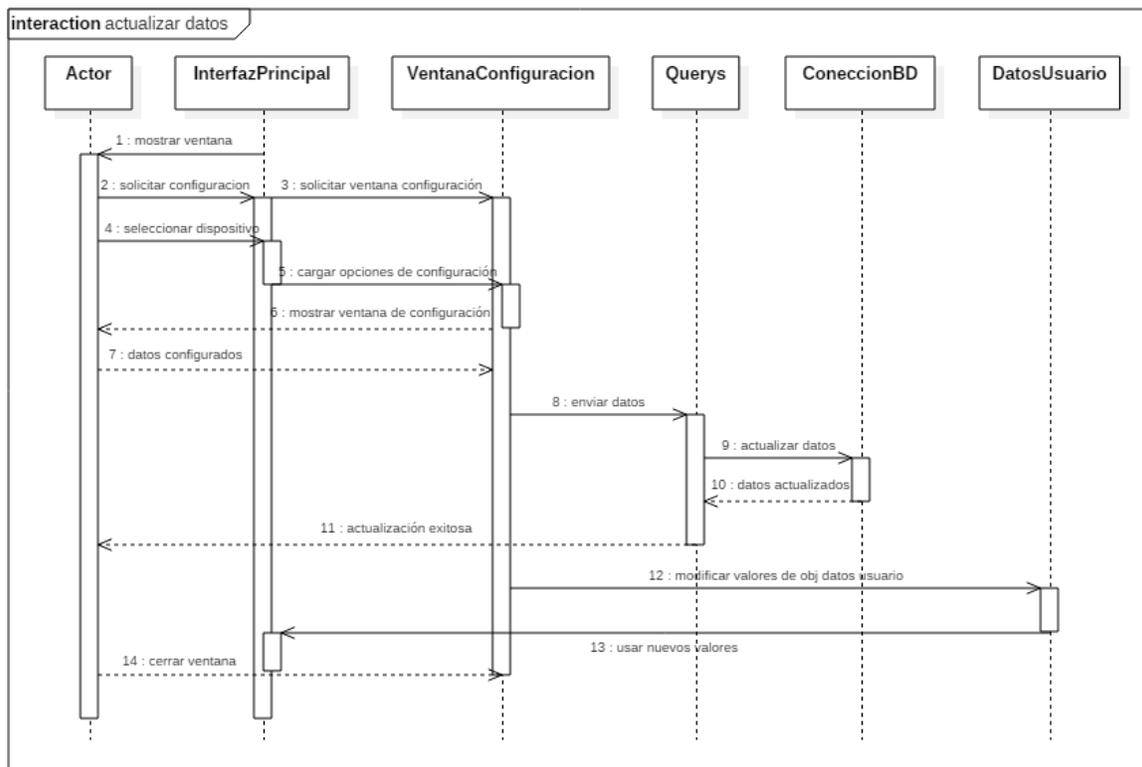


Ilustración 25 Diagrama de secuencia actualizar datos

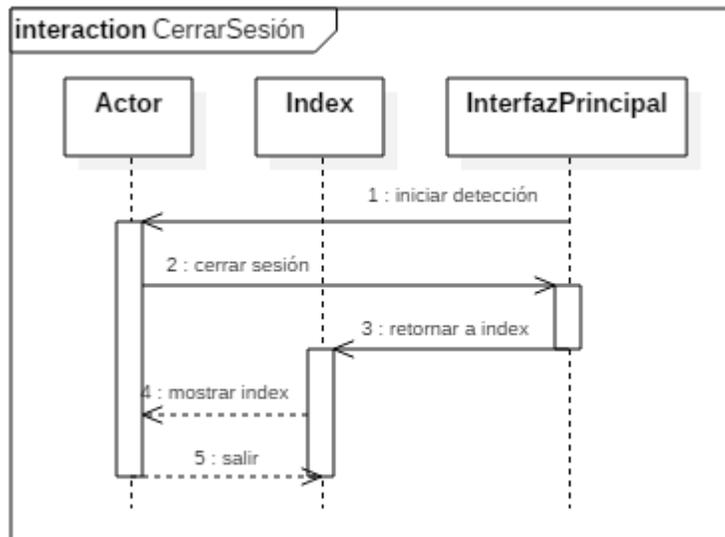


Ilustración 26 Diagrama de secuencia cerrar sesión

5.2.7. Diagramas de actividad

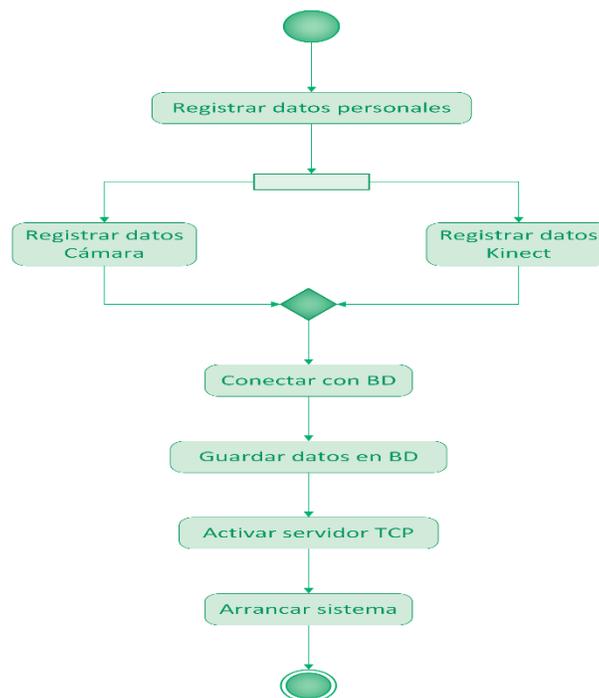


Ilustración 27 Diagrama de actividad registrarse

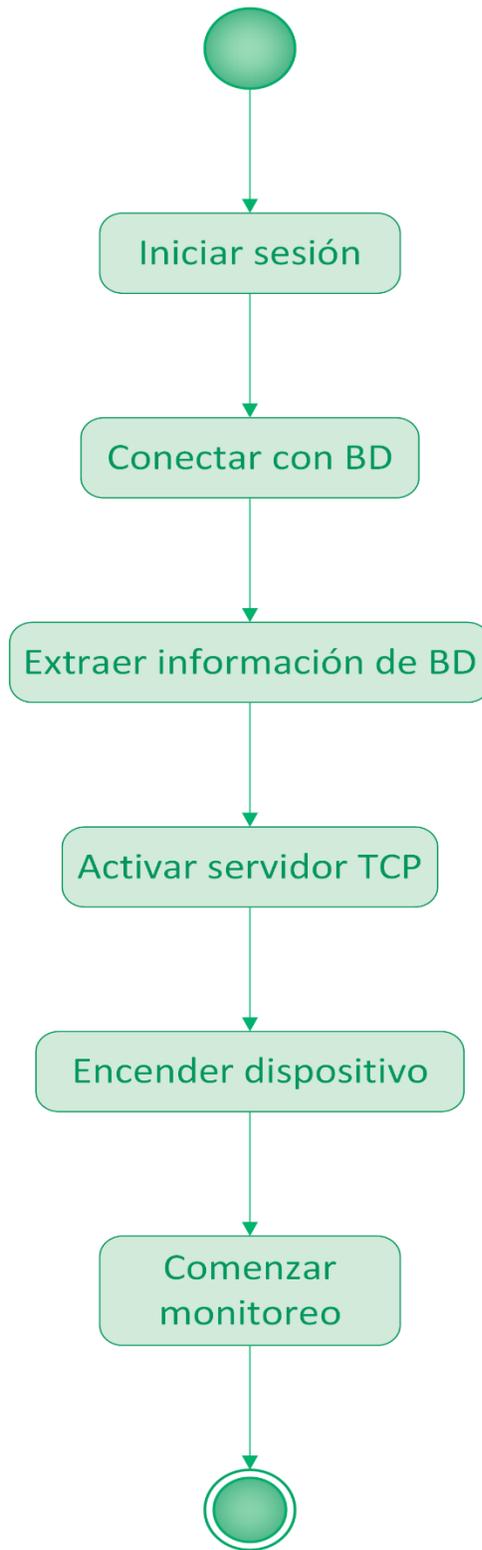


Ilustración 28 Diagrama de actividad iniciar sesión

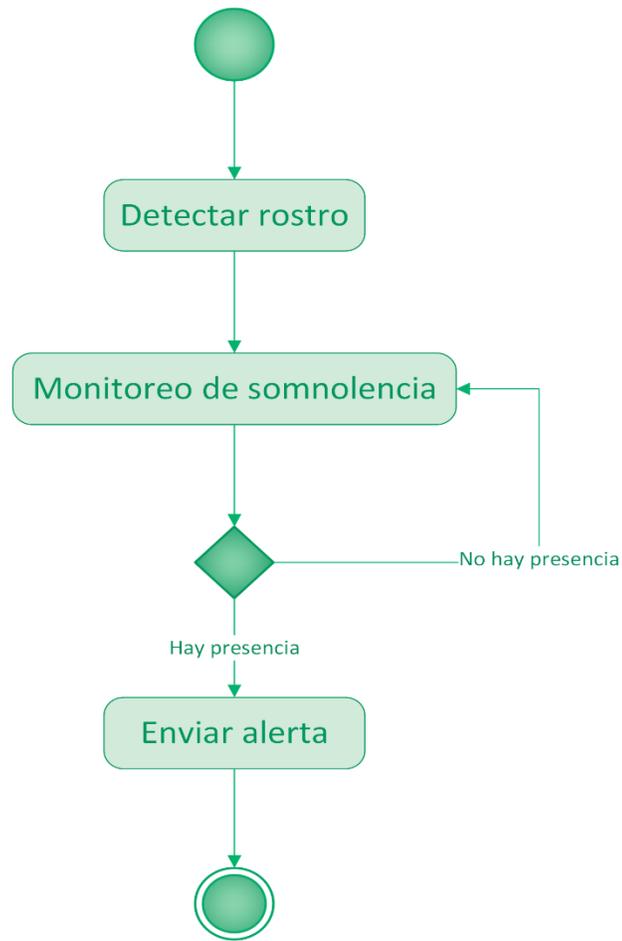


Ilustración 29 Diagrama de actividad análisis facial

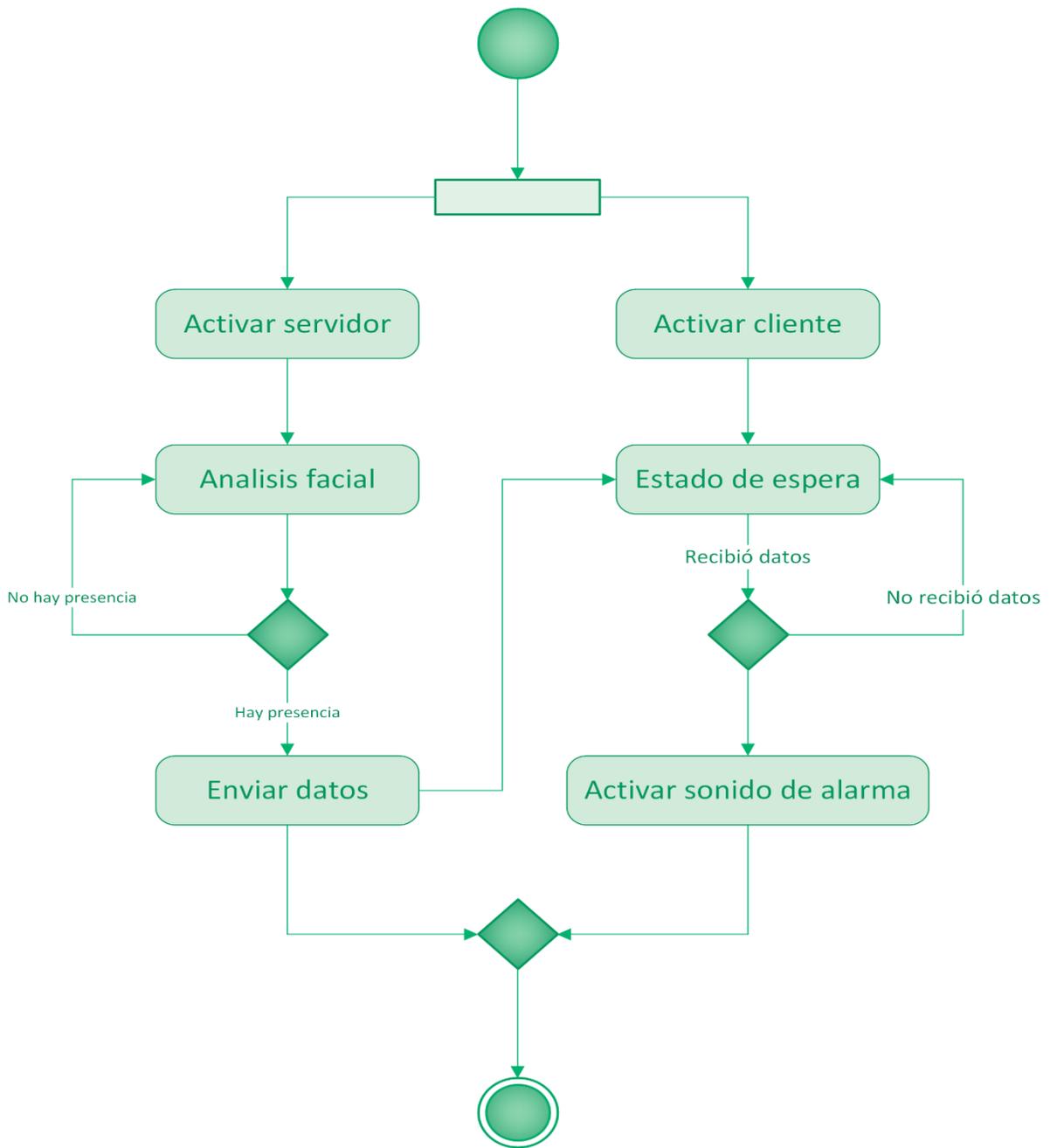


Ilustración 30 Diagrama de actividad Conexión TCP

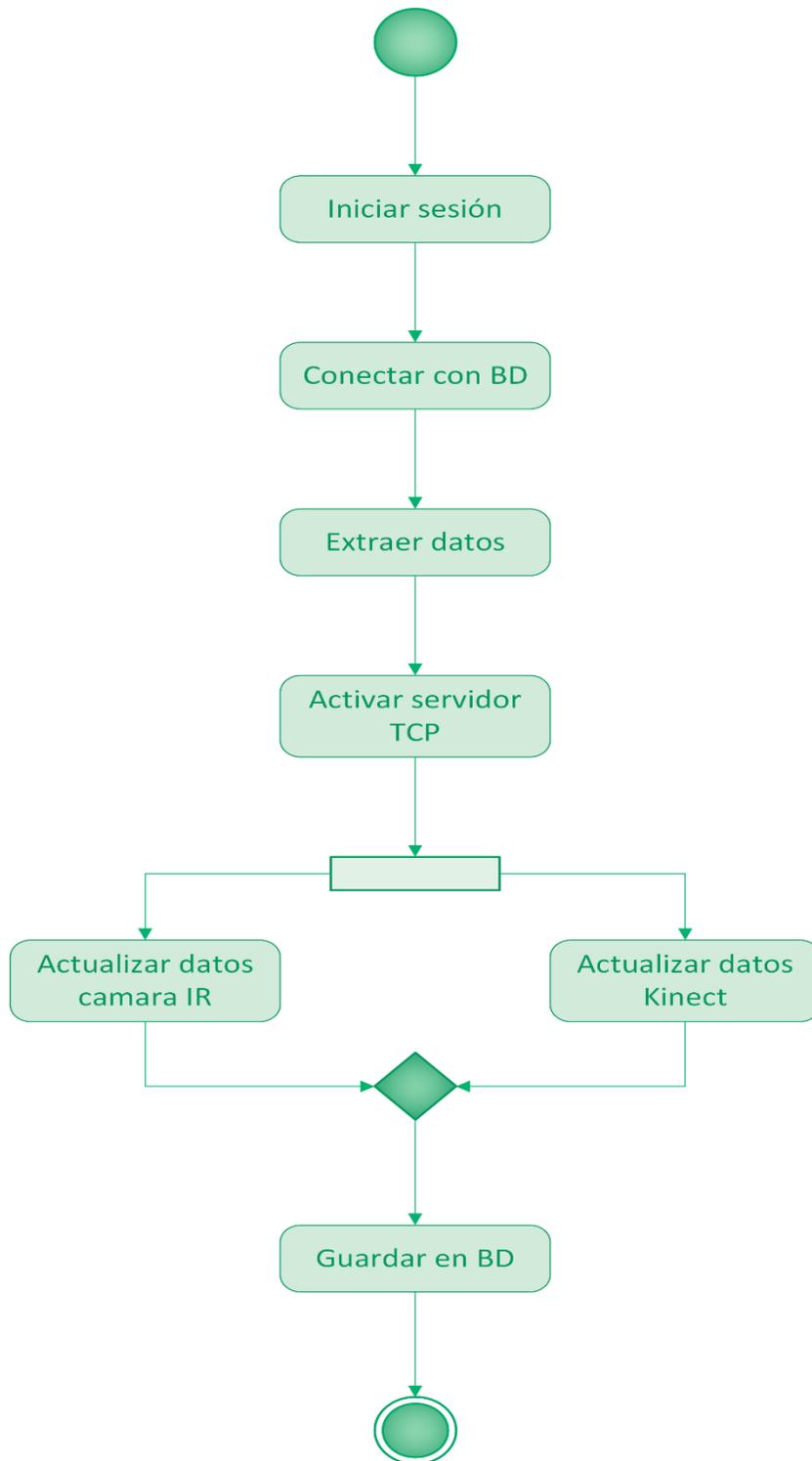


Ilustración 31 Diagrama de actividad actualizar datos

5.2.8. Diagrama de flujo

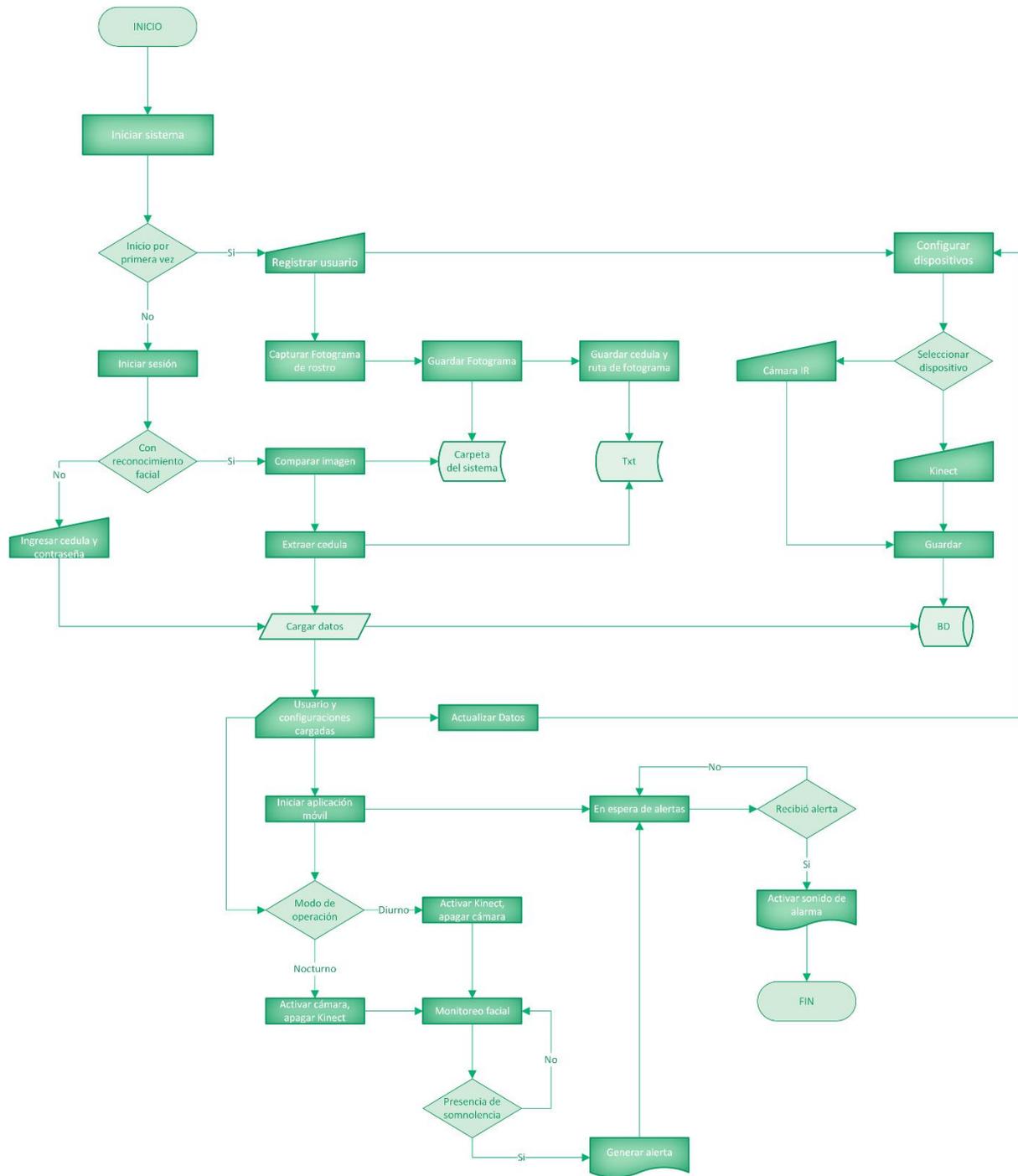


Ilustración 32 Diagrama de flujo Vigilant

5.3. Desarrollo del proyecto

Inicialmente se optó por utilizar el dispositivo Kinect por 2 motivos, el primero fue debido a su cámara infrarroja permitiendo capturar la imagen en un ambiente sin luz, la segunda razón era por su librería FaceTracking que deduce la posición de la cabeza y las expresiones faciales de la cara. Mientras se desarrollaba el proyecto se descubrió que era difícil trabajar con el dispositivo en las noches, ya que pese a que éste cuenta con sensor infrarrojo para poder capturar la imagen en ausencia de luz, tuvo los siguientes problemas:

- La librería FaceTracking de Kinect utilizada para capturar los rostros de las personas necesita que el stream de datos pasados sea a color, y aunque el stream IR es capturado como parte de una imagen a color (ver ilustraciones 32, 33, 34), esta se comporta de una manera diferente, ya que su salida está relacionada con el proyector IR, motivo por el cual su comportamiento es diferente al flujo de datos de una imagen a color, si se intenta pasar el flujo de tipo IR se obtiene un error (ver ilustración 35).

```
ColorImageFormat.YuvResolution640x480Fps15
```

Ilustración 33 Yuv Fps 15

```
ColorImageFormat.RgbResolution640x480Fps30
```

Ilustración 34 Rgb Fps 30

```
ColorImageFormat.InfraredResolution640x480Fps30
```

Ilustración 35 IR Fps 30



Ilustración 36 Error

- La otra razón por la cual no se utilizó el sensor IR de Kinect fue debido a que para la integración de este tipo de imagen con las librerías de procesamiento (Emgu CV, OpenCvSharp) era necesario convertir la imagen en un objeto Bitmap, para hacer esto se utilizaron 2 diferentes funciones.

```
public static System.Drawing.Bitmap Funcion1(BitmapSource bitmapsources)
{
    System.Drawing.Bitmap bitmap;
    using (var outputStream = new MemoryStream())
    {
        BitmapEncoder enc = new BmpBitmapEncoder();
        enc.Frames.Add(BitmapFrame.Create(bitmapsources));
        enc.Save(outputStream);
        bitmap = new System.Drawing.Bitmap(outputStream);
        return bitmap;
    }
}
```

Ilustración 37 Función 1

```
System.Drawing.Bitmap Funcion2(ColorImageFrame Image)
{
    byte[] pixeldata = new byte[Image.PixelDataLength];

    Image.CopyPixelDataTo(pixeldata);

    System.Drawing.Bitmap bmap = new System.Drawing.Bitmap(Image.Width, Image.Height,
        System.Drawing.Imaging.PixelFormat.Format16bppRgb555);

    System.Drawing.Imaging.BitmapData bmapdata = bmap.LockBits(
        new System.Drawing.Rectangle(0, 0, bmap.Width, bmap.Height),
        System.Drawing.Imaging.ImageLockMode.WriteOnly,
        bmap.PixelFormat);

    IntPtr ptr = bmapdata.Scan0;

    Marshal.Copy(pixeldata, 0, ptr, Image.PixelDataLength);
    bmap.UnlockBits(bmapdata);

    return bmap;
}
```

Ilustración 38 Función 2

Al utilizar la primera (utilizado en la cámara web), como los objetos Bitmap tienen inconvenientes al utilizar el formato Format16bppGrayScale, este al hacer la conversión se bloquea, es diferente a lo que sucede con la cámara web porque esta trabaja diferente ya que no cuenta con un proyector IR sino con luz infrarroja y la ausencia de un filtro que permite capturar la imagen en un solo canal (gris) cuando hay ausencia de luz, esto es decir que la imagen obtenida se considera como imagen a color lo cual permite hacer la conversión sin problema (ver diferencia de imágenes)

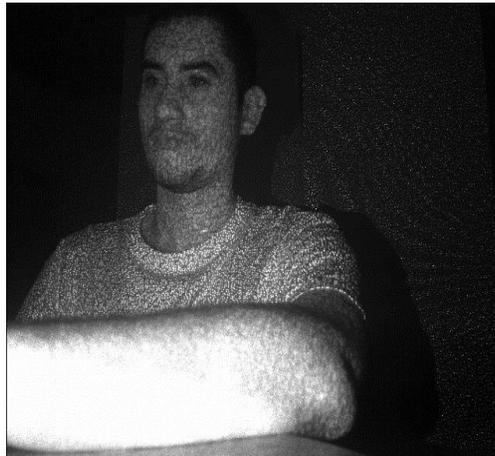


Ilustración 39 Imagen IR Kinect



Ilustración 40 Imagen IR Cámara

la segunda función si logra el objetivo de convertir la imagen a bitmap que permite no utilizar el formato Format16bppGrayScale sino otro tipo de formato, el inconveniente se presenta en la imagen que devuelve, esto es una imagen con mucho ruido y con

una sombra al lado, el ruido es producto del haces de puntos que el proyector emite, y la sombra se debe a la separación entre el emisor y el receptor, la cámara puede ver algunas partes de la escena que el emisor no llega y estas aparecen como sombras, si se tapa con algún objeto el proyector la imagen mejora (ver Ilustración 40 y 41), por otra parte el procesamiento de esta función es fuerte, lo que provoca que a cierto tiempo el programa deje de funcionar por falta de memoria, esta parte es un tema que se sale de nuestros conocimientos.



Ilustración 41 Imagen obtenida de la función 2



Ilustración 42 Imagen obtenida de la función con el proyector medianamente obstruido

Para la solución de este problema se decidió adaptar una cámara web para poder capturar imágenes de noche, es decir que se tendrían dos sistemas, uno que funcionase de día utilizando Kinect y FaceTracking y otro que funcionase de noche con la cámara web y la librería de visión artificial OpenCV, en cuanto a esta última, se utilizaron los Frameworks EmguCV y OpenCvSharp para el lenguaje de programación C#.

5.3.1. Desarrollo de algoritmos

Para detectar somnolencia nos basamos en rasgos faciales como bostezo, cabeceo y parpadeo, cuando una persona bosteza este permanece con la boca abierta o bien se tapa la boca con una de sus manos, la persona puede pestañear, llevarse las manos hacia los ojos o bien permanecer con los ojos cerrados y cuando cabecea, la persona lleva su cabeza hacia adelante, el alcance de este proyecto no se estudia técnicas de aprendizaje como redes neuronales, por ende no será posible saber si la persona se está quedando dormida si pestañea bajo ciertos intervalos, tampoco se estudiaran los casos para cuando la persona se lleva las manos hacia la cara. Para cada algoritmo se especifica cómo trabaja cada uno de estos.

5.3.1.1. Sistema nocturno

En este apartado se hará una comparación entre dos algoritmos que se diseñaron para la detección de somnolencia que serán utilizados en el sistema de noche. Uno de los algoritmos utiliza el SDK Luxand y el otro solo utiliza el Framework de OpenCV para C# OpenCvSharp.

5.3.1.1.1. OpenCvSharp

Ojos cerrados

El primer paso que se hace en este algoritmo es detectar la región de los ojos de la persona, este proceso se hace a través de un método contenido en este framework basado en el algoritmo de viola-jones utilizando un clasificador en cascada. En este caso el clasificador necesario es el de los ojos, lamentablemente cuando se aplicó este método se observó que los resultados obtenidos era inestables, la detección de los ojos era bastante intermitente lo que dificultaba la continuación del desarrollo de este algoritmo, se probó con otros clasificadores pero los resultados eran iguales, el único clasificador que era estable en gran proporción era el que detectaba cara.



Ilustración 43 Detección de cara OpenCvSharp

Se decidió utilizar este clasificador y obtener la región de los ojos a través de las dimensiones del rectángulo de la cara.

El punto X del rectángulo del ojo izquierdo se encuentra ubicado a un 37% de distancia del eje X del rectángulo del rostro, el punto Y del mismo ojo se encuentra ubicado a un 24% de distancia del eje Y del eje Y del rectángulo del rostro. El ancho es un 17% del ancho del rectángulo del rostro y el alto es un 8% del alto del mismo. Con estos valores se obtiene la región de interés del ojo izquierdo.

Para el rectángulo de la boca, el punto X se encuentra ubicado a un 70% de distancia del eje X del rectángulo del rostro, el punto Y se establece a un 40% de distancia del eje Y. El ancho es un 17% del ancho del rectángulo del rostro y el alto es un 16% del alto del mismo. Con estos valores se obtiene la región de interés de la boca (ver Ilustración 43 y 44).

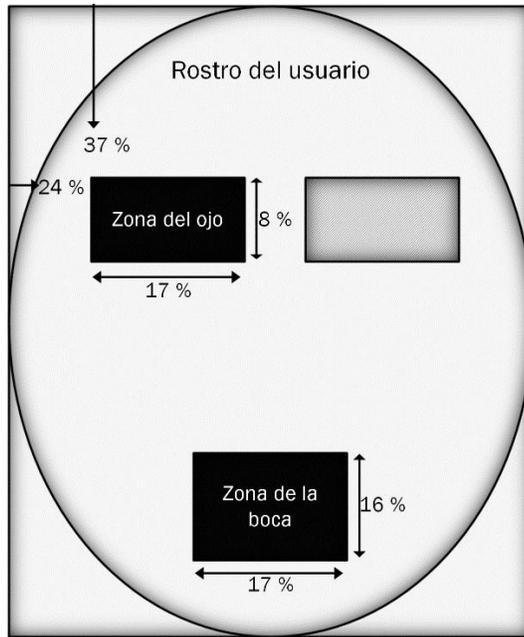


Ilustración 44 Dimensiones de los ojos y la boca

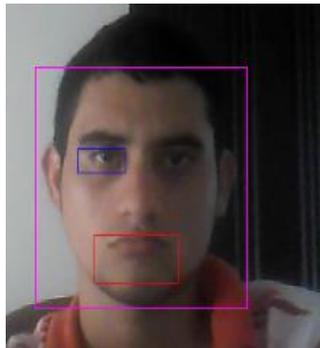


Ilustración 45 Ojos y boca en un rostro real

Luego de haber obtenido esta área se decidió aplicar el algoritmo para saber si el ojo está cerrado utilizado en la mayoría de los antecedentes investigados en la fase de análisis, este algoritmo trata de localizar la pupila del ojo por medio del método de reconocimiento de óvalos aplicado por el framework, al probar este algoritmo los resultados obtenidos no fueron los esperados, tardaba mucho tiempo en reconocer el círculo del iris, además la persona tenía que abrir demasiado el ojo para reconocerlo, posiblemente sea por la resolución de la cámara, para seguir con este proceso se

tuvo que desarrollar un algoritmo que cumpliera con este objetivo. Se convirtió la imagen a escala de gris y luego a binario.

Una imagen binaria es una imagen que solo presenta dos colores: blanco y negro (ver Ilustración 45)



Ilustración 46 Imagen binaria (derecha) con Thresload 50

Este tipo de imágenes se ven afectada por un valor numérico entre 0-255, este valor comúnmente llamado Threshold indica que cantidad de negro se quiere, en la imagen anterior el valor del Threshold era 50



Ilustración 47 Imagen binaria (derecha) con Thresload 20

Esta es una imagen binaria con un valor de 20. Hay que tener presente que la luz también afecta la salida de estas imágenes, por ejemplo una imagen binaria con un Threshold de 50 capturada en el día, va a ser diferente a una imagen binaria con un Threshold de 50 tomada en la noche.

Continuando con este paso, se observó que cuando el ojo está abierto el porcentaje de negro es mayor que cuando se encuentra cerrado, basados en este hecho se creó un ciclo que recorriera los píxeles de la imagen en cada fotograma, en este proceso se obtendrá el total de píxeles negros, si el índice de píxeles negros se mantiene en cierto nivel, entonces la persona se encuentra con los ojos abiertos, en caso de que el índice se reduzca en cierta cantidad de píxeles, entonces la persona tiene los ojos cerrados.



Ilustración 48 Imagen binaria ojos abiertos

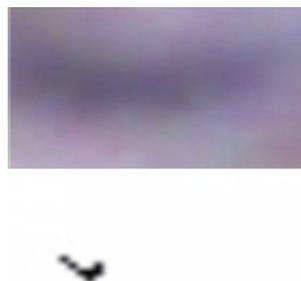


Ilustración 49 imagen binaria ojos cerrados

Cuando una persona se duerme cierra los dos ojos, así que por motivos de reducir procesos solo se utilizó este algoritmo para un solo ojo.

Bostezo

Para el desarrollo de este algoritmo se intentó utilizar un clasificador para la detección de la boca, pero al igual que en el caso de los ojos, la detección de este clasificador era demasiado inestable, así que se procedió de la misma manera que el algoritmo anterior, esto es se obtuvo el rectángulo de la boca a través del rectángulo del rostro por medio de cálculos matemáticos básicos, luego de obtener éste rectángulo, se procedió a pasar la imagen a binario, aquí se observó que la boca cerrada presenta menos cantidad de negro que cuando se encuentra abierta, se hizo lo mismo que en el caso de los ojos, se recorrió la imagen para saber el índice de negros, si el índice aumenta en cierta cantidad en un número determinado de fotogramas, entonces la boca se encuentra abierta.

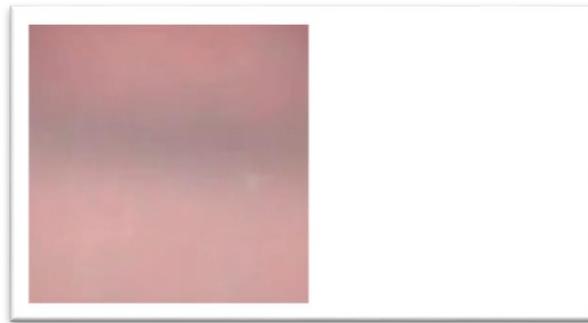


Ilustración 50 Imagen binaria (derecha) boca cerrada

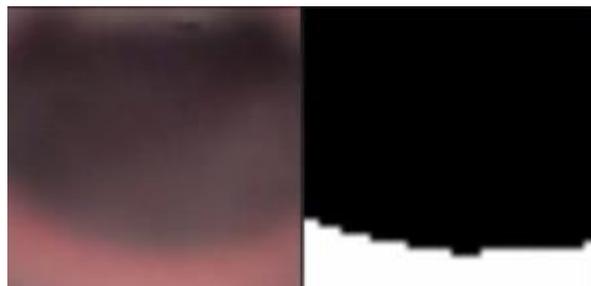


Ilustración 51 Imagen binaria (derecha) boca abierta

Cabeceo

No se pudo calcular cuando la persona está cabeceando, debido a que no se encontró ningún patrón con el rectángulo de la cara, cuando una persona baja su cabeza, este rectángulo queda casi con sus mismas proporciones, por ende es muy difícil tomar decisiones basados en esos hechos.

5.3.1.1.2. Luxand con EmguCV

Luxand es una librería que permite trabajar con 66 características faciales representadas en punto. Los puntos de interés para el desarrollo de este algoritmo son los correspondientes a los ojos, los puntos 54 y 55 que representen la mitad del labio superior y la mitad del labio inferior respectivamente.

Bostezo

Como se mencionó anteriormente se utilizaron los puntos 54 y 55, cuando la persona permanece con la boca cerrada estos puntos permanecen casi unidos, si la persona abre la boca estos puntos se alejan, para calcular la distancia de éstos se utilizó la fórmula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Donde x_1 y y_1 son las coordenadas del punto 54 y x_2 , y_2 son las coordenadas del punto 55. Si los puntos superan cierta distancia en una cantidad de frames, entonces el sujeto está bostezando.

Cabeceo

Para el algoritmo de detección de cabeceo se utilizó el punto 22 que se localiza en la mitad de los ojos, cuando la persona lleva su cabeza hacia adelante, este punto aumenta su valor en el eje Y, así que lo que se hizo fue fijar una zona en la cual si el punto 22 entra significa que está cabeceando.

Ojos Cerrados

Para la detección de los ojos cerrados se utilizaron los puntos asociados a éste. Lamentablemente estos puntos están tan cerrados que si la persona tiene los ojos cerrados o abiertos es muy difícil tomar decisiones, ya que permanecen en la misma posición, para seguir con el desarrollo de este algoritmo se utilizó el Framework de OpenCV, EmguCV. Este Framework es parecido a OpenCvSharp con la diferencia de que sus métodos son más fáciles de usar, no se utilizó en el otro sistema, debido a que la implementación de los clasificadores de este Framework utilizan más recursos haciendo que el programa Funcione más lento, en este caso no se necesita usar clasificadores para deducir el rectángulo de los ojos, ya que es posible hacerlo utilizando las coordenadas de los puntos 23, 24, 27 y 28. Luego de obtener esta zona, se transforma la imagen a binario y se realiza el mismo procedimiento del sistema que trabaja con OpenCvSharp. Cuando se depuro este algoritmo se observó que la imagen binaria era demasiado inestable ya que todos los puntos se mueven mucho haciendo que las dimensiones del rectángulo cambien mucho en cada frame y por ende el índice de negros tuviera cambios bruscos en cada frame. Para solucionar este problema solo se tomó las coordenadas del punto 12 y se estableció el ancho y el alto con un valor estático, así en vez de trabajar con 4 puntos inestables, trabajaríamos solo con un punto inestable, el resultado fue que la imagen binaria continuo cambiando entre cada frame pero no eran cambios muy exagerados.

5.3.1.1. Sistema diurno

5.3.1.1.1. Kinect vs Luxand

Para el sistema diurno también se compararon dos tecnologías, en este caso la librería FaceTracking del dispositivo Kinect y el SDK de Luxand, FaceTracking al igual que Luxand devuelve algunas características faciales representadas en puntos con la diferencia de que éste permite trabajar con 87 puntos en 2D. Los algoritmos utilizados para la detección de características de somnolencia con Kinect son parecidos a los utilizados con Luxand pero utilizando los puntos correspondientes de FaceTracking, 82 y 83 de los labios superior e inferior respectivamente para la detección de bostezo y el punto 0 para la detección del cabeceo. Para estas dos tecnologías se omitieron los algoritmos de detección de ojos cerrados, debido a que se tenía que utilizar algunos de los frameworks de OpenCV mencionados anteriormente y las imágenes binarias sufrían cambios debido a la variación de la iluminación, los resultados obtenidos bajo este efecto eran bastante inexactos, por lo que se tomó la decisión de quitarlos para el sistema diurno.

5.3.1.3. Notificación

Para notificar el momento en que se detecta somnolencia, el sistema envía mensajes a través de un socket TCP al dispositivo móvil, el móvil siempre está revisando su buffer en busca de contenido, cuando encuentra un mensaje este activará una alarma sonora que alerta a la persona de la presencia de síntomas de somnolencia, éste funciona en background por lo que el usuario puede recibir notificaciones de otras aplicaciones, sin embargo el ciclo de vida finalizará cuando el usuario cierre el programa principal, ya que este actúa como servidor o cuando se cierre el proceso del App.



Ilustración 52 Vigilant.Apk en ejecución

5.3.2. Ubicación de dispositivos

El vehículo que se utilizó para el montaje y pruebas del sistema fue un Chevrolet Sail Sedan (ver ilustración 51). A continuación se mostrara como fueron ubicados los dispositivos.



Ilustración 53 Vehículo de prueba

Laptop

La laptop donde se encuentra el software se ubicó en el asiento del copiloto para que la conexión de los otros dispositivos fuera más sencilla (ver ilustración 52).



Ilustración 54 ubicación laptop

Kinect

Inicialmente se ubicó el dispositivo en la parte central del vehículo mirando al conductor con la intención de abarcar completamente la cara del piloto, la distancia que había en estos era de aproximadamente 80cm que es la distancia mínima que necesita Facetracking para la detección del rostro, lamentablemente la detección del rostro era un poco intermitente, por lo cual se movió el dispositivo hacia la esquina del lado del copiloto, ya que la distancia en esa parte aumento aproximadamente 1m de distancia, en esta posición la detección es un poco más eficiente, sin embargo el lado izquierdo del rostro de la persona es despreciado, además si el conductor hace un movimiento hacia la izquierda la detección se pierde en ese instante, pero cuando se devuelve la mirada hacia el frente la detección se vuelve a retomar (ver ilustración 53).



Ilustración 55 ubicación Kinect



Ilustración 56 Dimensiones Vehiculo

Kinect necesita que se le suministre un voltaje de 12V, para este existe un cable con dos entradas, la primera es la que se conecta al computador y el otro se dirige a un enchufe. Para el montaje dentro del vehículo se modificó el cable para adaptarlo a la cigarrera del carro que tiene una salida de 12 voltios, por lo que se conecta el dispositivo por corriente directa.

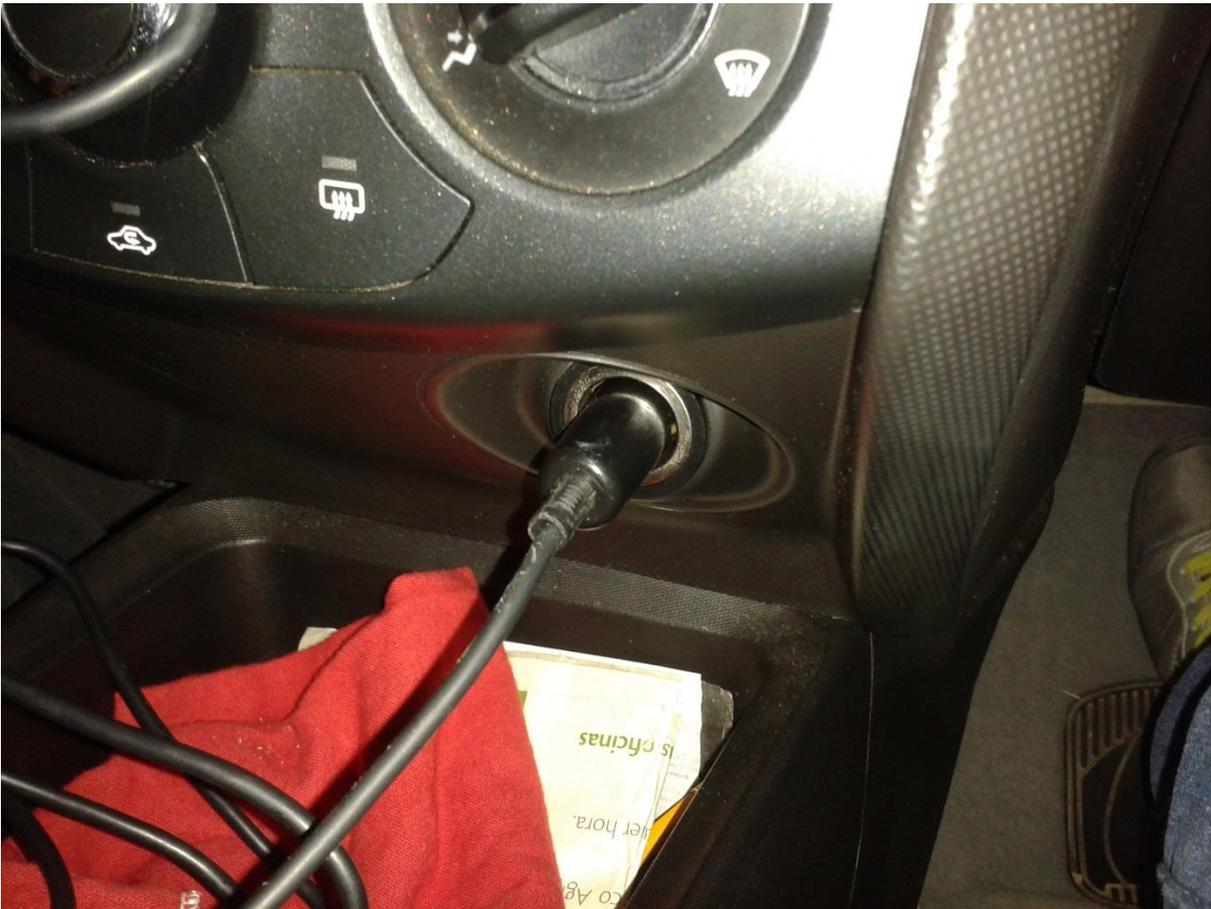


Ilustración 57 Conexión de Kinect al Vehículo

Cámara

Debido a que el software realizado con OpenCvSharp requiere que el usuario este frente la cámara, se ubicó está en una posición tal que permitiera cumplir con este objetivo, esto fue en la zona detrás del volante (ver ilustración 54).



Ilustración 58 Ubicación cámara para OpenCvSharp

Para probar el software que utiliza el SDK de Luxand la cámara fue ubicada a una distancia aproximada de 50cm respecto al costado izquierdo del vehículo, ya que en esta zona el dispositivo está en una distancia considerable y se logró detectar los puntos de las características de la cara de buena manera (ver siguiente figura).



Ilustración 59 Ubicación cámara para Luxand

5.3.3. Pruebas y análisis

5.3.3.1. Pruebas

Las pruebas del sistema se realizar dentro de 3 ambientes distintos, como se describe a continuación con sus respectivos resultados:

- simulación en un ambiente común (usuario frente a los dispositivos en cualquier lugar), de día con:
 - Kinect



Ilustración 60 Mensaje de bostezo Kinect SAC – Día

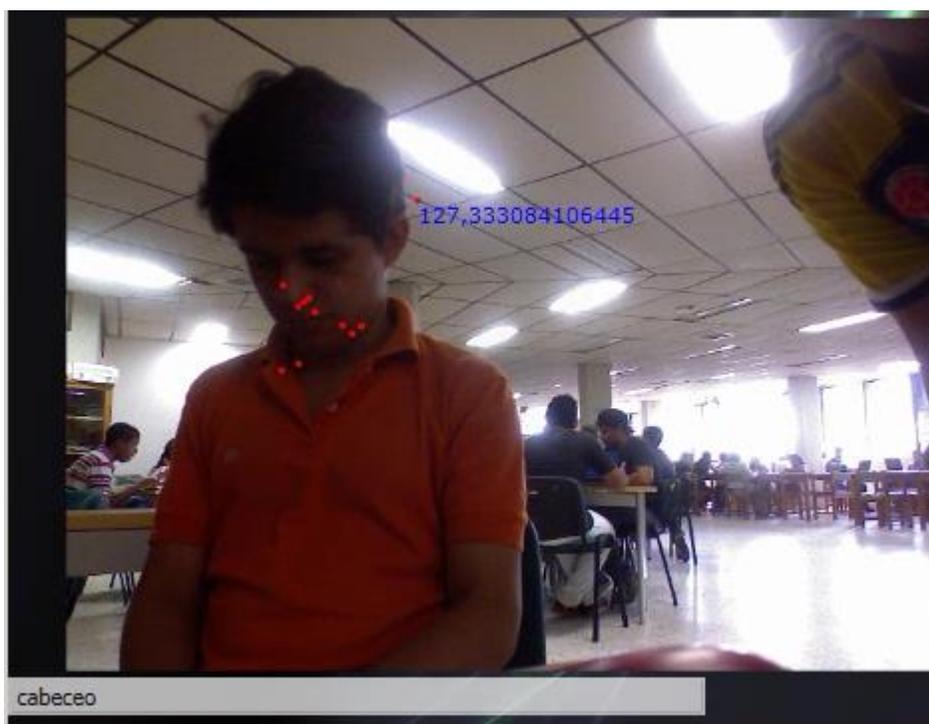


Ilustración 61 Mensaje Cabeceo Kinect SAC – Día

Nombre	Bostezo					Cabeceo				
	1	2	3	4	5	1	2	3	4	5
Eder	1	1	1	1	1	1	1	1	1	1
Eduardo	1	1	1	1	1	1	1	1	1	1
Harold	1	0	1	1	1	1	1	1	1	1
Mauricio	1	1	1	1	1	1	1	1	1	1
Omar	1	1	1	1	1	1	1	1	1	1

Tabla 20 Resultados SAC Kinect

Cabeceo: Fue eficiente en un 100%

Bostezo: Fue eficiente en un 96%

- Luxand Face SDK

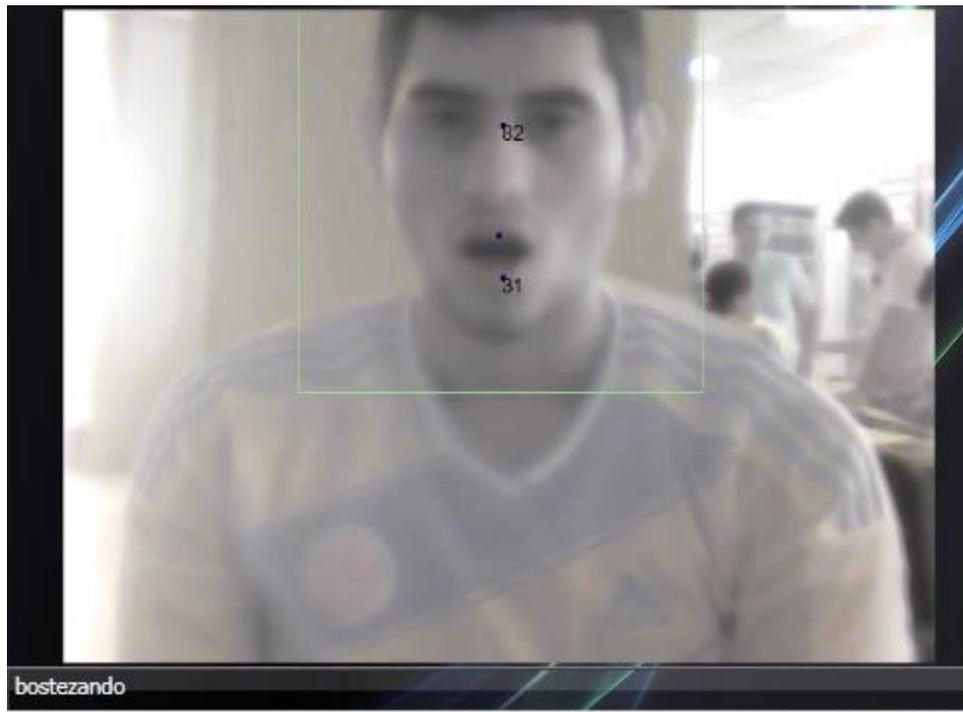


Ilustración 62 Bostezo Luxand SAC – Día

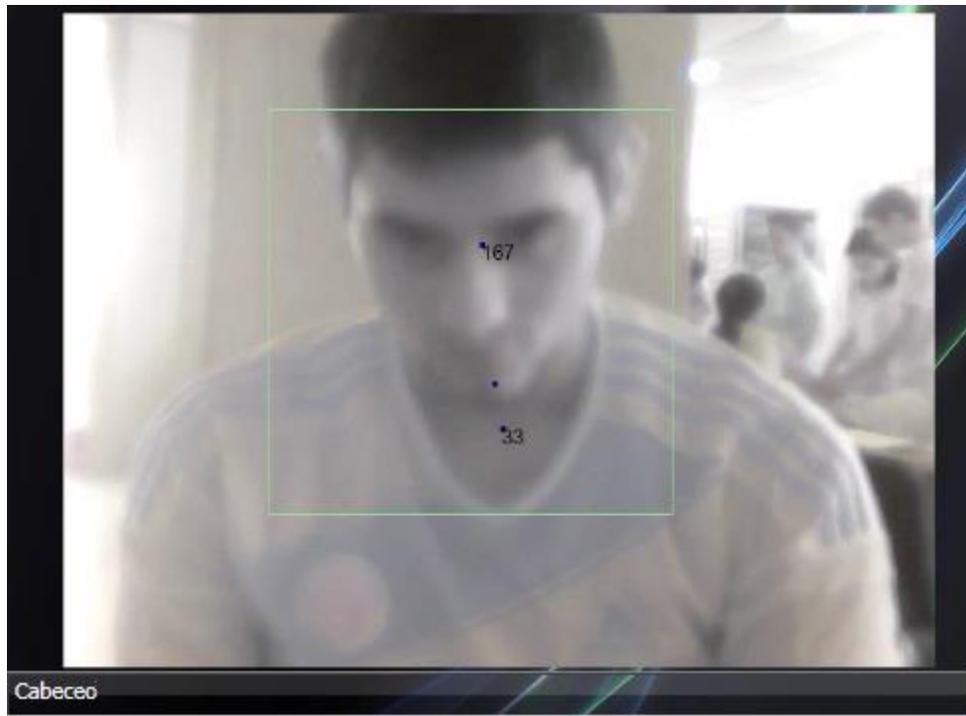


Ilustración 63 Cabeceo Luxand SAC – Día

	Bostezo					Cabeceo				
Nombre	1	2	3	4	5	1	2	3	4	5
Eder	0	1	1	0	1	1	1	1	1	1
Eduardo	1	1	0	1	1	1	1	1	1	1
Harold	1	1	0	0	1	1	1	1	1	1
Mauricio	1	1	1	0	1	1	1	1	1	1
Omar	0	0	0	1	0	1	1	1	1	1

Tabla 21 Resultados SAC Luxand - Día

Cabeceo: Fue eficiente en un 100%

Bostezo: Fue eficiente en un 60%

- simulación en un ambiente común, de noche con:
 - OpenCvSharp



Ilustración 64 Bostezo OpenCvSharp SAC – Noche

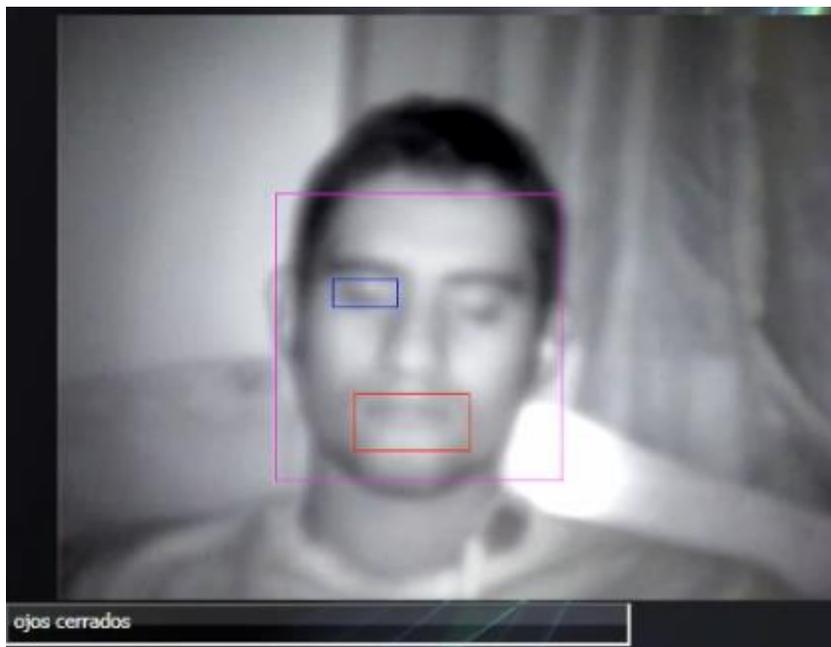


Ilustración 65 Ojos cerrados OpenCvSharp SAC – Noche

Nombre	Bostezo					Ojos Cerrados				
	1	2	3	4	5	1	2	3	4	5
Eduardo	0	0	1	0	1	1	1	1	1	1
Harold	1	1	1	0	0	1	1	0	1	1
Mauricio	1	1	1	1	1	1	1	1	1	1
Omar	0	1	1	1	1	0	1	1	0	1

Tabla 22 Resultados SAC OpenCvSharp Noche

Bostezo: Efectivo 60%

Ojos cerrados: Efectivo 85%

- Luxand Face SDK

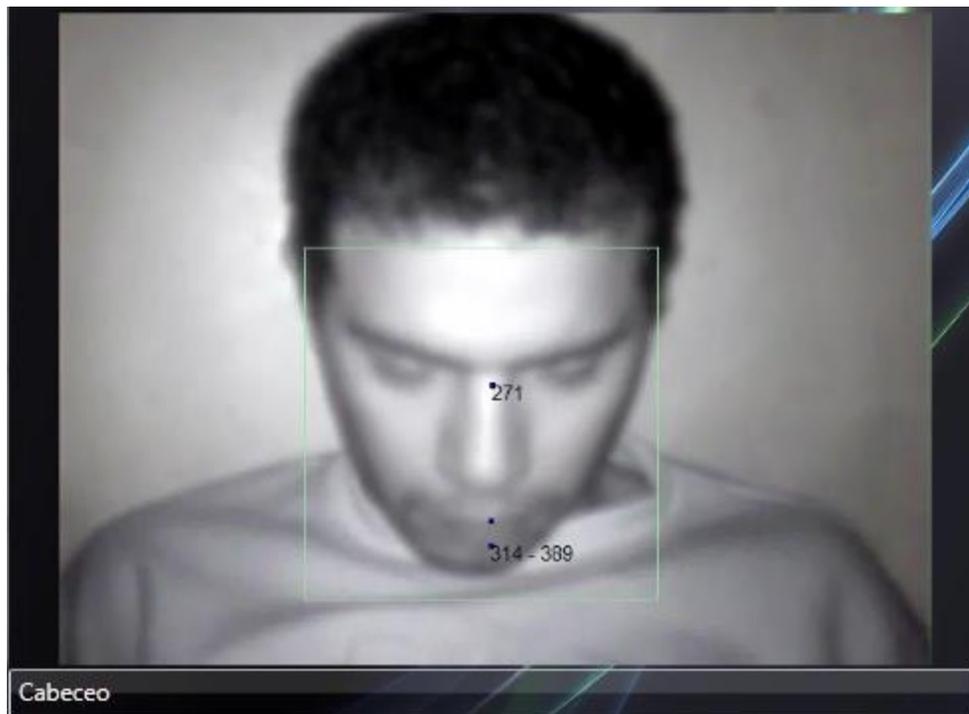


Ilustración 66 Cabeceo Luxand SAC – Noche



Ilustración 67 Bostezo Luxand SAC – Noche



Ilustración 68 Ojos cerrados Luxand SAC – Noche

Nombre	Bostezo					Cabeceo					Ojos cerrados				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Eduardo	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
Harold	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
Mauricio	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
Omar	0	0	1	1	1	1	1	1	1	1	0	0	1	1	0

Tabla 23 Resultados SAC Luxand Noche

Cabeceo: Efectivo 100%

Bostezo: Efectivo 90%

Ojos cerrados: Efectivo 50%

- Simulación con el vehículo dentro del garaje, de día con:
 - Kinect



Ilustración 69 Bostezo Kinect SVDG – Día



Ilustración 70 Cabeceo Kinect SVDG - Día

Nombre	Bostezo					Cabeceo				
	1	2	3	4	5	1	2	3	4	5
Eduardo	1	1	1	1	1	1	1	1	1	1
Mauricio	1	1	1	0	1	1	1	1	0	1

Tabla 24 Resultados SVDG Kinect

Cabeceo: Efectivo 90%

Bostezo: Efectivo 90%

- Luxand Face SDK

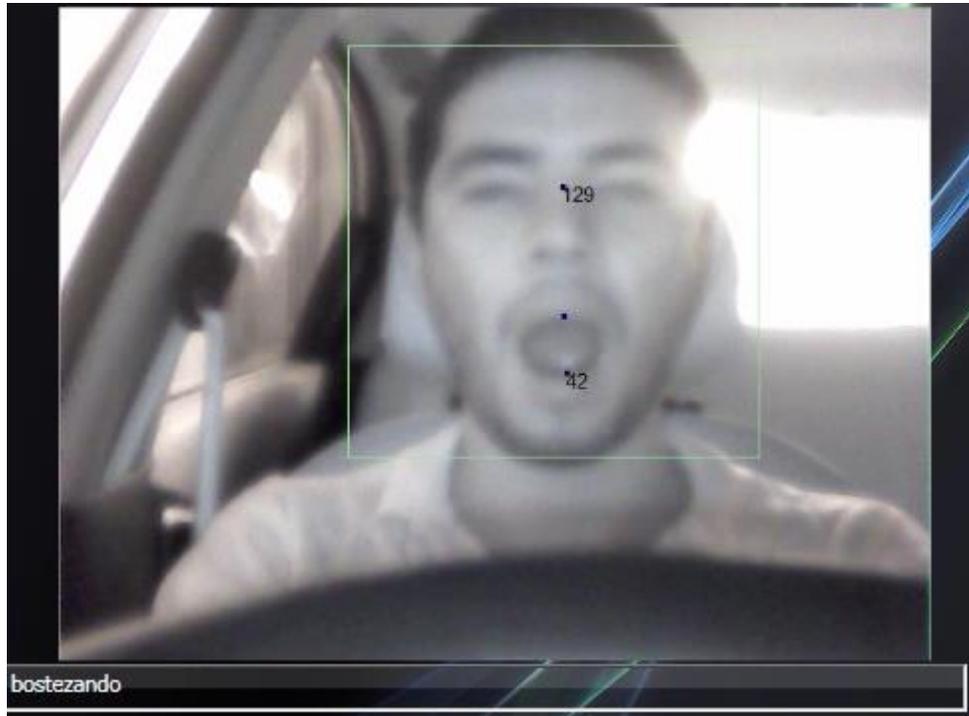


Ilustración 71 Bostezo Luxand SVDG – Día

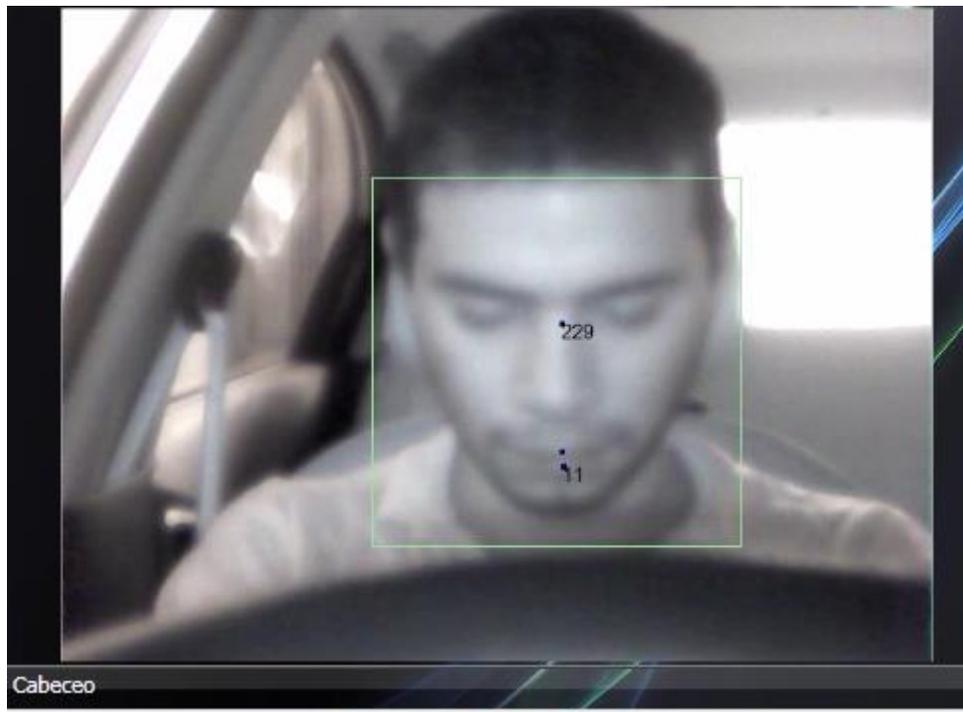


Ilustración 72 Cabeceo Luxand SVDG - Día

Nombre	Bostezo					Cabeceo				
	1	2	3	4	5	1	2	3	4	5
Eduardo	1	1	1	0	1	1	1	1	1	1
Mauricio	1	0	1	1	0	1	1	1	1	0

Tabla 25 Resultados SVDG Luxand Dia

Cabeceo: 90%

Bostezo: 70%

Simulación con el vehículo dentro del garaje, de noche con:

- OpenCvSharp

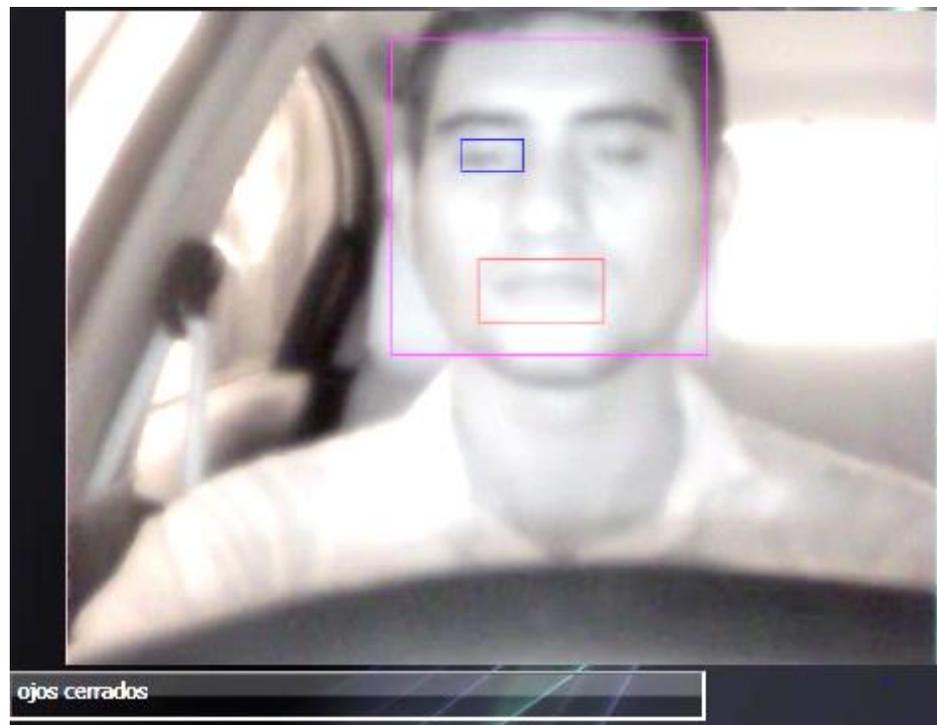


Ilustración 73 Ojos cerrados OpenCvSharp SVDG – Noche

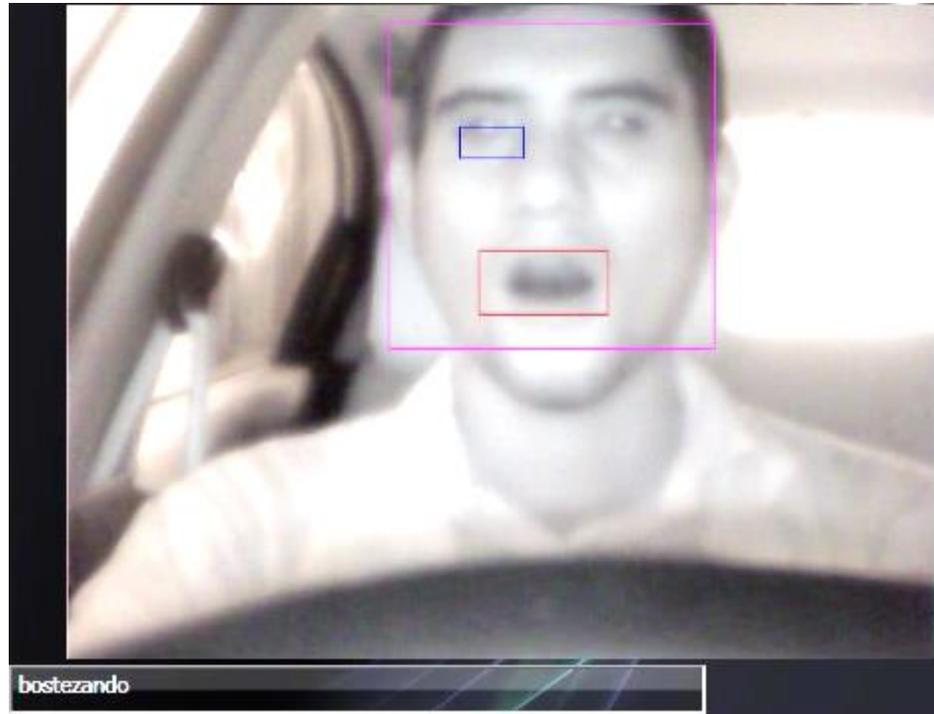


Ilustración 74 Bostezo OpenCvSharp SVDG - Noche

Nombre	Bostezo					Ojos Cerrados				
	1	2	3	4	5	1	2	3	4	5
Eduardo	0	0	1	1	1	0	1	1	1	1
Mauricio	1	1	1	1	1	1	1	1	1	1

Tabla 26 Resultados SVDG OpenCvSharp Noche

Bostezo: Efectivo 80%

Ojos cerrados: Efectivo 90%

- Luxand Face SDK

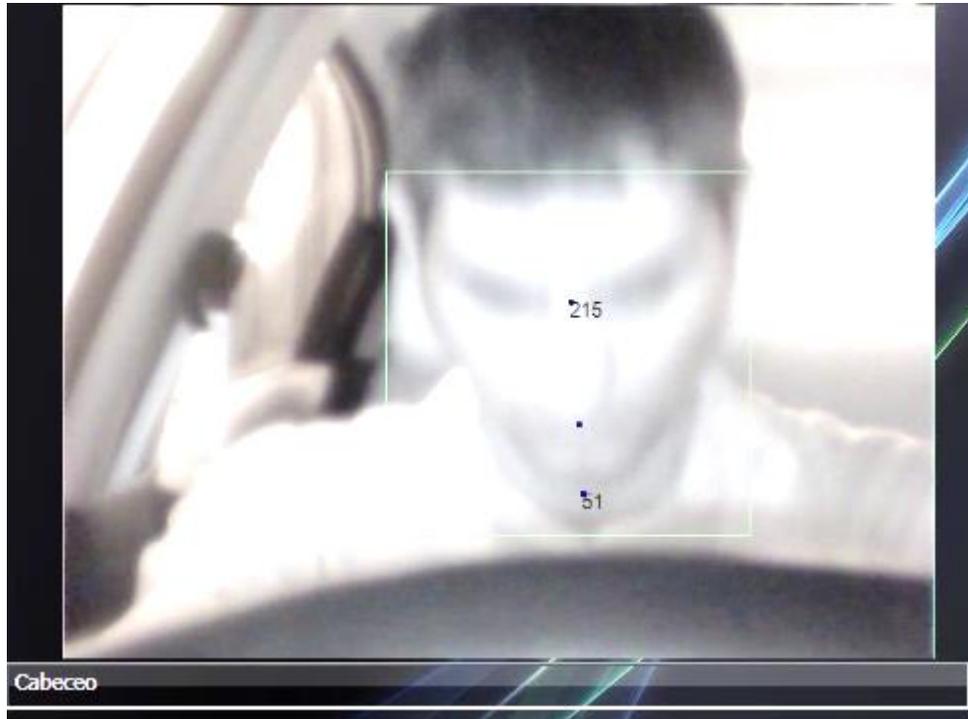


Ilustración 75 Cabeceo Luxand SVDG – Noche

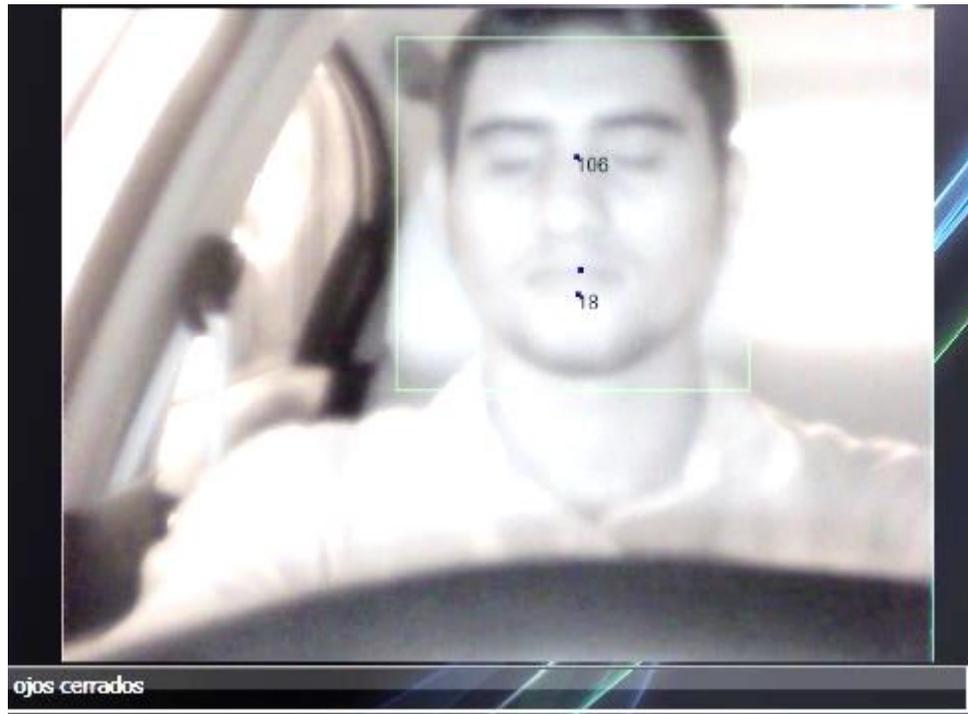


Ilustración 76 Ojos Cerrados Luxand SVDG – Noche

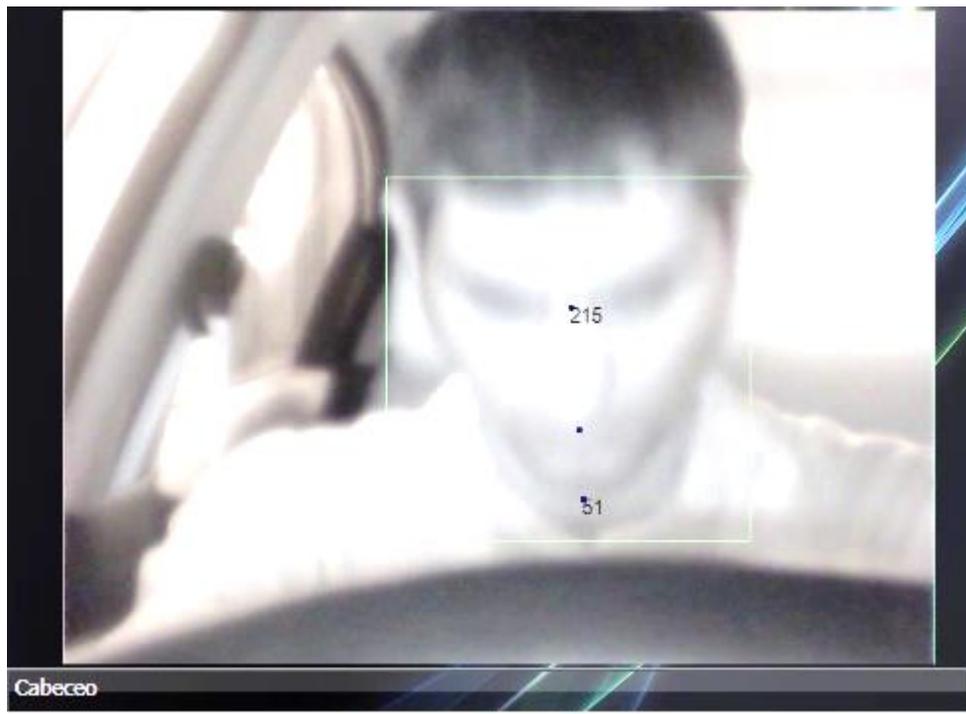


Ilustración 77 Cabeceo Luxand SVDG - Noche

	Bostezo					Cabeceo					Ojos cerrados				
Nombre	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Eduardo	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
Mauricio	1	1	0	0	0	1	1	1	1	1	1	1	1	0	1

Tabla 27 Resultados SVDG Luxand Noche

Cabeceo: Efectivo 100%

Bostezo: Efectivo 70%

Ojos cerrados: Efectivo 50%

- Simulación con el vehículo expuesto a condiciones reales, de día con:
 - Kinect

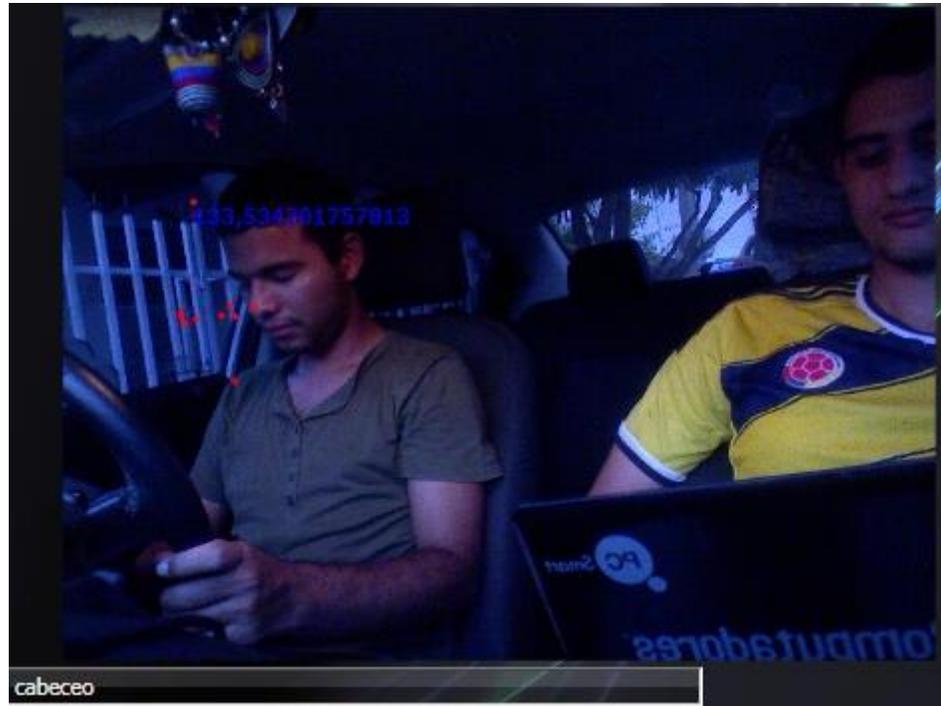


Ilustración 78 Cabeceo Kinect VCR – Día

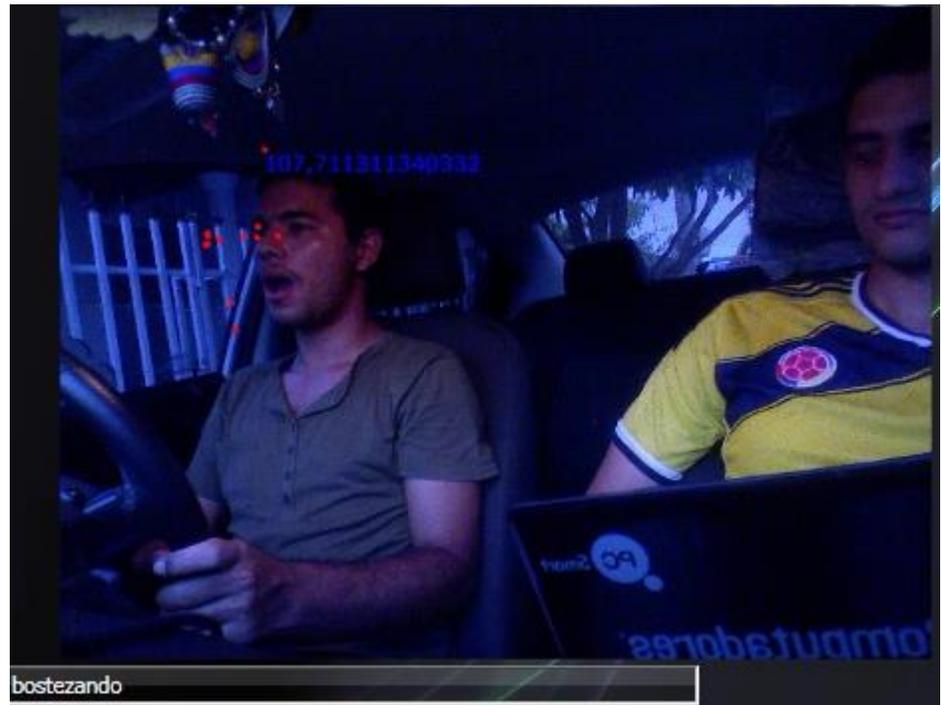


Ilustración 79 Bostezo Kinect VCR - Día

Nombre	Bostezo					Cabeceo				
	1	2	3	4	5	1	2	3	4	5
Eduardo	1	1	1	1	1	1	1	1	1	1
Mauricio	1	1	1	1	1	1	1	1	1	1

Tabla 28 Resultados VCR Kinect

Cabeceo: Efectivo 100%

Bostezo: Efectivo 100%

- Luxand Face SDK

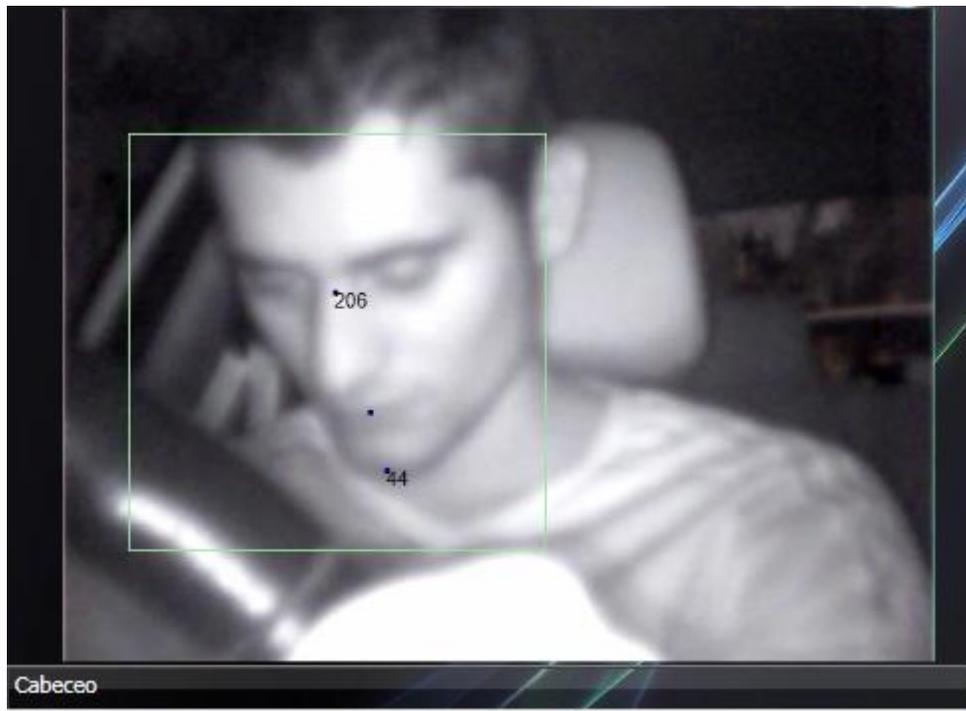


Ilustración 80 Cabeceo Luxand VCR - Día

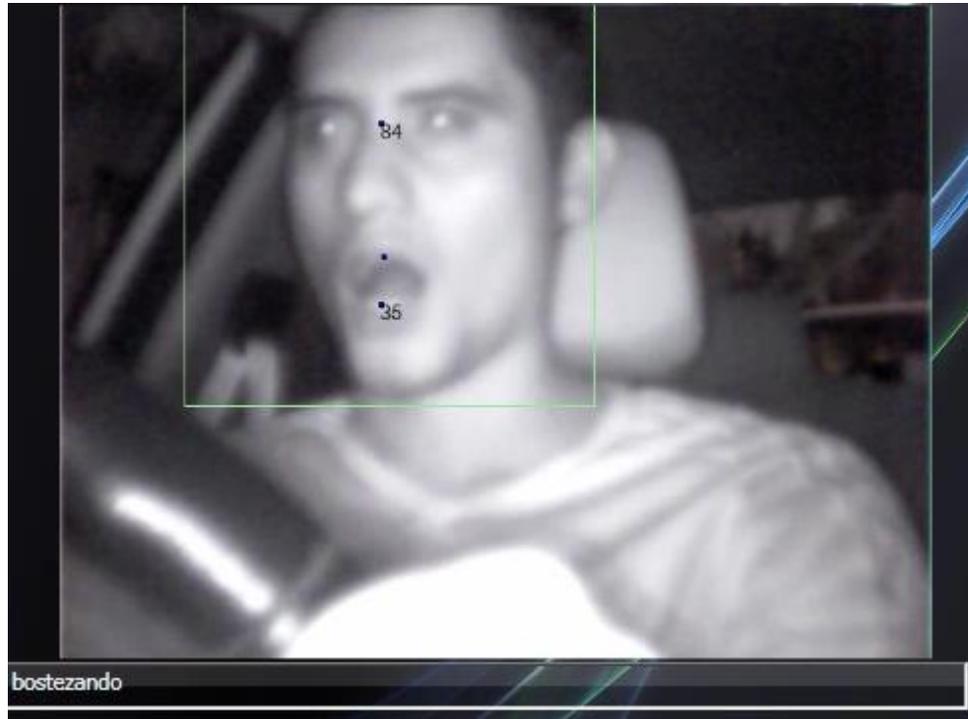


Ilustración 81 Bostezo Luxand VCR - Día

Nombre	Bostezo					Cabeceo				
	1	2	3	4	5	1	2	3	4	5
Eduardo	1	0	1	0	0	1	1	1	1	1
Mauricio	1	1	1	0	1	1	1	1	1	1

Tabla 29 Resultados VCR Luxand Día

Cabeceo: Efectivo 100%

Bostezo: Efectivo 60 %

- Simulación con el vehículo expuesto a condiciones reales, de noche con:
 - OpenCvSharp

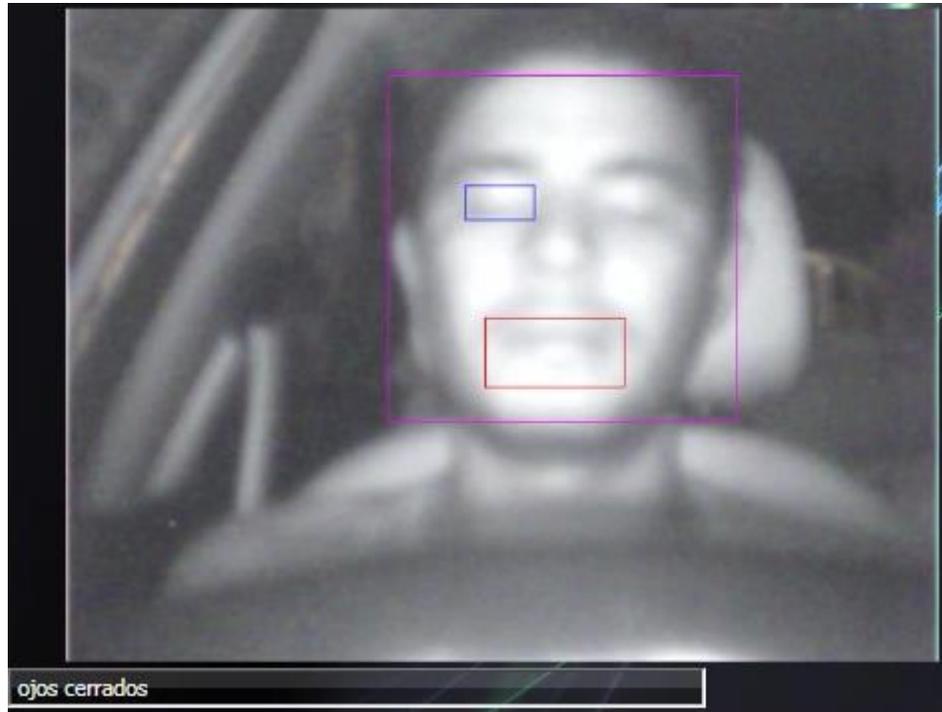


Ilustración 82 Ojos cerrados OpenCvSharp VCR – Noche



Ilustración 83 Bostezo OpenCvSharp VCR – Noche

Nombre	Bostezo					Ojos cerrados				
	1	2	3	4	5	1	2	3	4	5
Eduardo	0	0	1	1	1	1	1	1	1	0
Mauricio	1	1	1	1	1	1	0	1	1	1

Tabla 30 Resultados VCR OpenCvSharp Noche

Ojos cerrados: Efectivo 80%

Bostezo: Efectivo 80 %

- Luxand Face SDK



Ilustración 84 Bostezo Luxand VCR – Noche



Ilustración 85 Cabeceo Luxand VCR - Noche



Ilustración 86 Ojos cerrados ERROR Luxand VCR - Noche

Nombre	Bostezo					Cabeceo					Ojos cerrados				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Eduardo	0	0	1	1	0	1	1	1	1	1	0	0	0	0	1
Mauricio	0	0	1	1	1	1	1	1	1	1	1	0	0	1	0

Tabla 31 Resultados VCR Luxand Noche

Cabeceo: Efectivo 100%

Bostezo: Efectivo 50%

Ojos cerrados: Efectivo 30%

Se hicieron algunas pruebas en unos conductores de taxis, pero por problemas relacionados con la luz como se menciono anteriormente y por la poca distancia entre la cabina muy angosta del vehículo los resultados obtenidos no fueron satisfactorios.



Ilustración 87 FaceTracking no puede detectar el rostro



Ilustración 88 Vista trasera de la ubicación del dispositivo en el taxi

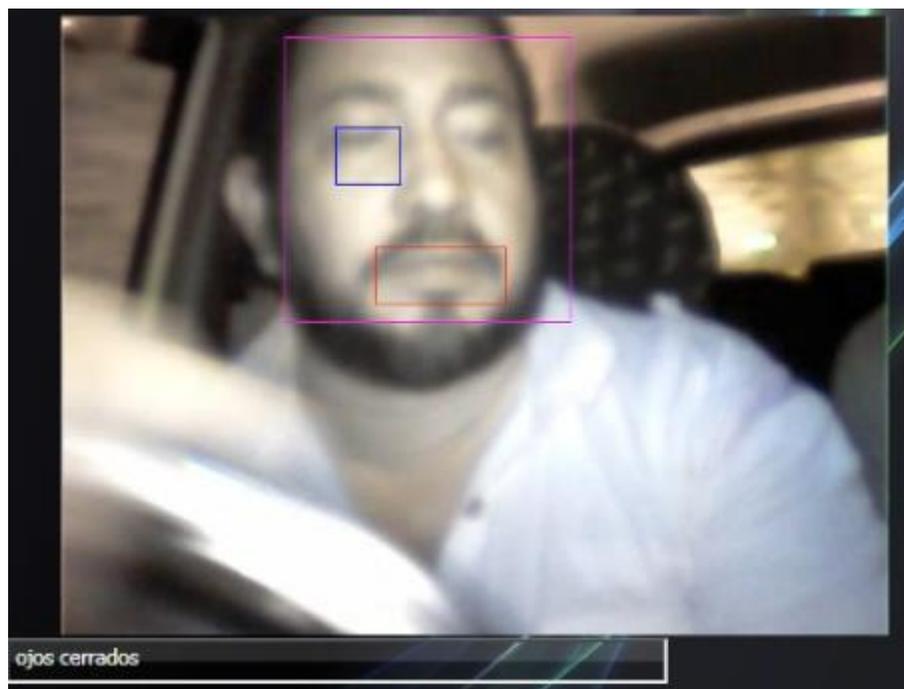


Ilustración 89 Falso positivo ojos cerrados

En la ilustración anterior, el usuario se encuentra mirando lateralmente hacia abajo, entonces el sistema encuentra la cantidad de píxeles negros aproximadamente igual a la establecida para los ojos cerrados.

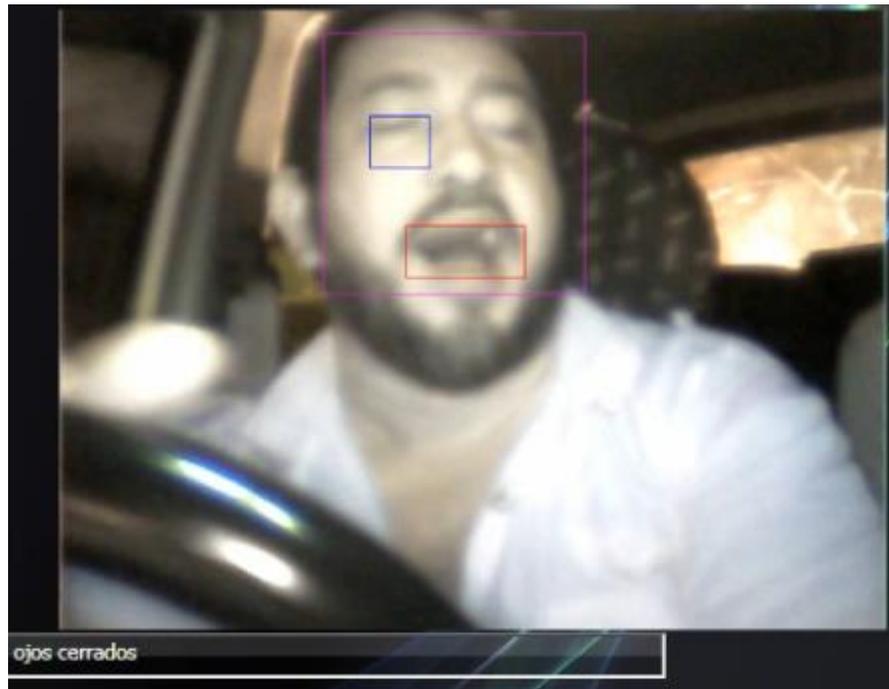


Ilustración 90 Falso positivo bostezo

En la (Ilustración 86) el usuario se encuentra bostezando pero el sistema no encuentra demasiada sombra en el interior de la boca por lo cual no puede determinar y sigue enviando falsas alarmas de ojos cerrados.

5.3.3.2. Análisis.

Día Kinect

- 1) Para la detección se hizo necesario que el conductor hiciera varios movimientos mirando hacia el dispositivo para que facetracking entrara en funcionamiento.

- 2) Se presentan falsos positivos en presencia de demasiada luz solar o en poca ausencia de esta
- 3) Si la persona es demasiado alta o demasiado baja tal que la posición de la cabeza se encuentre demasiado desviada del enfoque del sensor, entonces pueden suceder 2 cosas: la primera es que la detección sea muy intermitente, la segunda es que no reconozca al sujeto.
- 4) A 80 cm de distancia que es la mínima requerida para el funcionamiento de Kinect para Xbox, la detección solo se produjo en 2 intentos de 10, con una duración de los puntos faciales muy corta.
- 5) La distancia ideal es de 1 metro ubicando el dispositivo enfrente del copiloto y enfocando el sensor hacia el conductor.
- 6) Posteriormente al reconocimiento facial, registro del punto de la cabeza e inicio de sesión, el sistema diurno no presenta falsos positivos en lo referente a cabeceo y bostezo mientras el ambiente sea el adecuado.

Noche cámara IR

Ojos

Lanza falsos positivos en presencia de cambios de luminosidad.

Boca

Lanza falsos positivos mientras el Thresload es nivelado y el porcentaje de pixeles para el bostezo varía según la luz que exista en el momento, esto es, a más luz menos porcentaje de pixeles y viceversa.

Funcionan bajo ciertos parámetros establecidos para todos los usuarios, como son:

- Porcentaje de pixeles negros para un ojo que se encuentre cerrado
- Porcentaje de pixeles negros para una boca que se encuentre abierta.

Ambos parámetros funcionan mientras no se altere de forma violenta la luz del ambiente.

Los rectángulos del ojo y boca fueron establecidos predeterminadamente (ver Ilustración 43) debido a que la poca resolución ocasionada por la modificación de ésta para que funcionara de noche en ausencia de luz con los 4 leds infrarrojos que se le soldaron hizo que nitidez de disminuyera en un 30 % aproximadamente, lo que provoco que el clasificador mostrara detecciones falsas.

Los rectángulos pequeños dependen de la posición del rectángulo mayor que es el de la cara, esta detección es bastante eficiente pero el cuadro de detección desafortunadamente siempre está variando su posición lo que afecta directamente las luces o sombras que se encuentren en el rostro.

El método de conteo de pixeles es un 80% eficiente en condiciones propicias con pocas alteraciones de luz o en ambientes sin luz.

6. CONCLUSIONES

Una persona que conduzca con somnolencia es un peligro para la vida de cualquier ser humano, por eso se hace necesario incursionar en nuevas tecnologías que hagan de la vida en las carreteras algo más segura. Tecnologías como la visión artificial son una buena opción para propósitos de análisis de características faciales, debido que no implican un sistema invasivo al usuario, una pequeña cámara u otro tipo de sensor infrarrojos instalado en la cabina del vehículo no representa ninguna molestia o incomodidad al conductor debido a que por ser un dispositivo pequeño, su ubicación dentro de éste se hace de forma sencilla y sin estorbar la visión del conductor.

El proyecto Vigilant presenta unas herramientas y unos métodos para el análisis de características faciales y consta de dos sistemas:

El sistema diurno enlazado con Kinect funciona bajo ciertas características y parámetros establecidos, o dicho de otra forma, es eficiente según las condiciones de luminosidad a las que se encuentre expuesto el vehículo, como cualquier cámara se ve afectado ante la luz solar directa o en presencia de oscuridad, además es necesario que exista una distancia considerable entre el conductor y la cabina del vehículo para que Kinect funcione de forma óptima.

El sistema nocturno el cual trabaja con una pequeña cámara web de baja resolución y que fue modificada por la necesidad de capturar datos de noche, también se ve afectada ante las luces de neón u otros tipos de luces utilizadas cuando se oculta el sol, no obstante el método de conteo de porcentaje de píxeles funciona cuando la luz es estable tanto para determinar si los ojos están cerrados o si la persona está bostezando aunque la cámara ha de ser ubicada en frente del conductor y puede implicar depende la estatura del sujeto un pequeño estorbo para una visión completa de la carretera. No obstante al utilizar la tecnología de LuxandFace SDK la cámara se pudo ubicar al costado derecho del volante evitando así obstrucción visual y

detectando prácticamente de forma eficiente la presencia de bostezo y cabeceo pero fallando en un 90% aproximadamente en la detección de ojos cerrados.

No se pudo obtener un sistema diurno/nocturno completo realmente, lo que le falta a una tecnología u herramienta lo tiene otra y viceversa. Lamentablemente el hecho de que la luz no se puede contralar hacen al sistema vulnerable, ésta es una variable natural no manipulable y significa un problema para este tipo de proyectos.

Como apartado final se puede hacer la siguiente observación: posiblemente utilizando otro tipo de herramientas como Kinect 2 u otras cámaras de alta resolución e implementado un sistema más complejo de visión diurna/nocturna se podría aumentar considerablemente la efectividad del software teniendo como presente que el objetivo principal sería evitar el daño de la luz al sistema de igual forma se puede probar otras librerías y comparar si es más eficiente en cada uno de las características mencionadas o quizás extraer nuevas características.

BIBLIOGRAFIA

ASUS. (s.f.). Obtenido de <http://www.asus.com/>

Burbano, L. C. (11 de Abril de 2013). *HSB NOTICIAS*. Obtenido de <http://hsbnoticias.com/conductor-que-se-queda-dormido-causa-accidente-en-nari%c3%b1o-36652>

Caracol Noticias. (5 de Marzo de 2015). Obtenido de <http://st.caracolnoticias.co/mundo/video-conductor-se-queda-dormido-y-causa-tremendo-accidente>

Clinica del ronquido y trastornos de sueño. (s.f.). Obtenido de Accidentes de tránsito y sueño: <http://www.clinicadelronquidoysueno.com/consulta/accidentes-de-transito-y-sueno/>

Colombia, U. N. (s.f.). Obtenido de Trastornos de Sueño: de http://www.virtual.unal.edu.co/cursos/medicina/2007860/lecciones/cap5/05_06.htm

Corporación Fondo de Prevención vial. (s.f.). Obtenido de http://www.fpv.org.co/images/repositorioftp/CFPV_ARojas.pdf

Corporación Fondo de Prevención vial. (s.f.). Obtenido de http://www.fpv.org.co/investigacion/balance_cifras_accidentalidad/gen/12

de Castro, J. R., Gallo, J., & Loureiro, H. (2004). Cansancio y somnolencia en conductores de ómnibus y accidentes de carretera en el Perú: estudio cuantitativo. *Rev Panam Salud Publica*, 16(1), 11.

Dirección General de Tráfico (DTG), Gobierno de España. (s.f.). Obtenido de http://www.dgt.es/PEVI//documentos/catalogo_recursos/didacticos/did_adultas/suenio.pdf

- EmguCV. (s.f.). *Main Page*. Obtenido de http://www.emgu.com/wiki/index.php/Main_Page
- Espinola Gonzales, J. E., Lopez, A., & Maximiliano E y Rodriguez Sabino, V. G. (2011). Sistema de visión artificial para la detección de somnolencia de conductores, basado en el comportamiento ocular. *Aporte Santiaguino*, 4(2), 145-151.
- Ficosa. (s.f.). Obtenido de http://www.ficosa.com/wps/wcm/connect/a1a98c7a-c9c9-4278-ac86-25db5d6d0c7d/DATASHEET+P8_Set_2014.pdf
- Flores, M. J. (2009). *Sistema avanzado de asistencia a la conducción mediante visión por computador para la detección de la somnolencia* (Doctoral dissertation, Universidad Carlos III de Madrid).
- García, J. L., Rogado, E., Barea, R., Bergasa, L. M., López, E., Ocaña, M., & Schleicher, D. (2008). Sistema detector de fatiga en la conducción.
- Intel. (s.f.). *RealSense Camera F200*. Obtenido de <https://software.intel.com/es-es/realsense/f200camera>
- Ismael, C. (19 de Junio de 2013). *List of 10+ Face Detection / Recognition APIs, libraries, and software*. Obtenido de <http://blog.mashape.com/list-of-10-face-detection-recognition-apis/>
- Jimenez Moreno, R. (2011). *Sistema de detección de nivel de cansancio en conductores mediante técnicas de visión por computador* (Doctoral dissertation, Universidad Nacional de Colombia).
- Liendo, G. R., Castro, C. L., & Rey de Castro, J. (2010). Cansancio y somnolencia en conductores de ómnibus interprovinciales: estudio comparativo entre formalidad e informalidad. *Revista Peruana de Medicina Experimental y Salud Publica*, 27(2), 187-194
- Luxand. (s.f.). *LuxandFace SDK*. Obtenido de <https://www.luxand.com/facesdk/>
- Marín, C. C. (18 de Abril de 2009). *El Tiempo*. Obtenido de <http://www.eltiempo.com/archivo/documento/mam-3404798>

- Microsoft. (s.f.). Obtenido de <http://www.microsoft.com/>
- OPTALERT. (s.f.). Obtenido de <http://www.optalert.com>
- Play Station. (s.f.). Obtenido de <http://latam.playstation.com/gg>
- Real Academia Española. (s.f.). Obtenido de Diccionario de la lengua española, vigésima segunda edición: <http://lema.rae.es/drae/>
- Rosales Mayor, E., & Rey de Castro Mujica, J. (2010). Somnolencia: Qué es, qué la causa y cómo se mide. *Acta Médica Peruana*, 27(2), 137-143.
- Seguimiento al sector salud en Colombia. (s.f.). Obtenido de <http://www.asivamosensalud.org/inidicadores/estado-de-salud/grafica.ver/58>
- Somnolencia. (s.f.). Obtenido de <http://somnolencia.org/sintomas/>
- Suarez, J. F. (19 de Enero de 2015). *El Colombiano*. Obtenido de <http://www.elcolombiano.com/en-colombia-se-vendieron-328-526-carros-nuevos-en-2014-KE1125815>
- Sura. (s.f.). *Accidentes de transito en Colombia*. Obtenido de <http://www.sura.com/blogs/autos/accidentes-transito-colombia.aspx>
- Técnicas Tecnogías de Diseño Electrónico. (23 de Junio de 2012). Obtenido de http://157.253.195.181/ttdewikim1/index.php?title=Android_Sistema_para_identificar_condici%C3%B3n_de_sue%C3%B1o&oldid=1582
- Villamizar, M., Suarez, J., Villanueva, J., Borja, G., & De Los Rios, E. (2014, April). Design and implementation of sleep monitoring system using electrooculographs signals. In *Health Care Exchanges (PAHCE), 2014 Pan American* (pp. 1-1). IEEE. (Universidad Autónoma del Caribe, Barranquilla-Colombia)
- Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2), 137-154.

Wikipedia. (10 de Agosto de 2015). *OpenCV*. Obtenido de Wikipedia, la enciclopedia libre: <https://es.wikipedia.org/wiki/OpenCV>

Wikipedia. (15 de Julio de 2015). *Kinect*. Obtenido de Wikipedia, la enciclopedia libre: <https://es.wikipedia.org/wiki/Kinect>

Wikipedia. (24 de Octubre de 2015). *Detección de caras*. Obtenido de Wikipedia, La enciclopedia libre: https://es.wikipedia.org/w/index.php?title=Detecci%C3%B3n_de_caras&oldid=86069296.

Wikipedia. (28 de Julio de 2015). *Sistema de reconocimiento facial*. Obtenido de Wikipedia, La enciclopedia libre: https://es.wikipedia.org/w/index.php?title=Sistema_de_reconocimiento_facial&oldid=84061043.

Wikipedia. (31 de Octubre de 2015). *Visión artificial*. Obtenido de Wikipedia, La enciclopedia libre: https://es.wikipedia.org/w/index.php?title=Visi%C3%B3n_artificial&oldid=86150239

ANEXOS

1. Mortalidad por siniestros de tránsito

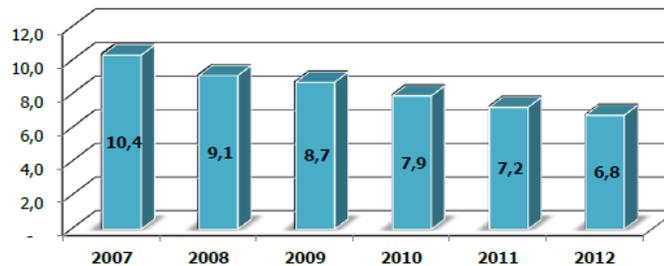


Fuente: Instituto Nacional de Medicina Legal, suministrada el 24 de enero de 2013; procesado por la Dirección de Investigación de la CFPV bajo la metodología de la OMS.
Parque Automotor, Registro Único Nacional de Tránsito RUNT, con corte al 5 de junio de 2012.



Ilustración 91 Mortalidad por siniestros de tránsito. Colombia 1995-2012 Py

2. Tasas de mortalidad



Fuente: Instituto Nacional de Medicina Legal, suministrada el 24 de enero de 2013; procesado por la Dirección de Investigación de la CFPV bajo la metodología de la OMS.
Parque Automotor, Registro Único Nacional de Tránsito RUNT, con corte al 5 de junio de 2012.



Ilustración 92 Tasas de Mortalidad Colombia

3. Manual de instalación

A continuación se describen los pasos necesarios para la correcta instalación del software y de los dispositivos.

Para que la aplicación funcione es indispensable una computadora que cuente con las siguientes especificaciones:

- Sistema operativo Windows Vista/7/8/10
- Procesador dual core de 1.8 ghz
- Memoria RAM de 4Gb DDR3
- Disco duro de 320 Gb
- .Net Framework 4.5
- Kinect SDK 1.8

Es necesario contar con un servidor de base de datos, para ello se optó por el uso del software XAMPP que cuenta con un gestor de descarga además de otras herramientas. Para la instalación de XAMPP se procede a su descarga desde la página oficial <https://www.apachefriends.org/es/index.html>, como ésta aplicación está bajo licencia GPL no hay problema alguno en su utilización.



Ilustración 93 Pagina de descarga de XAMPP

Posterior a su descarga e instalación de forma correcta, se ejecuta y desplegará una ventana con el panel de XAMPP en éste se precede a iniciar Apache y MySQL.

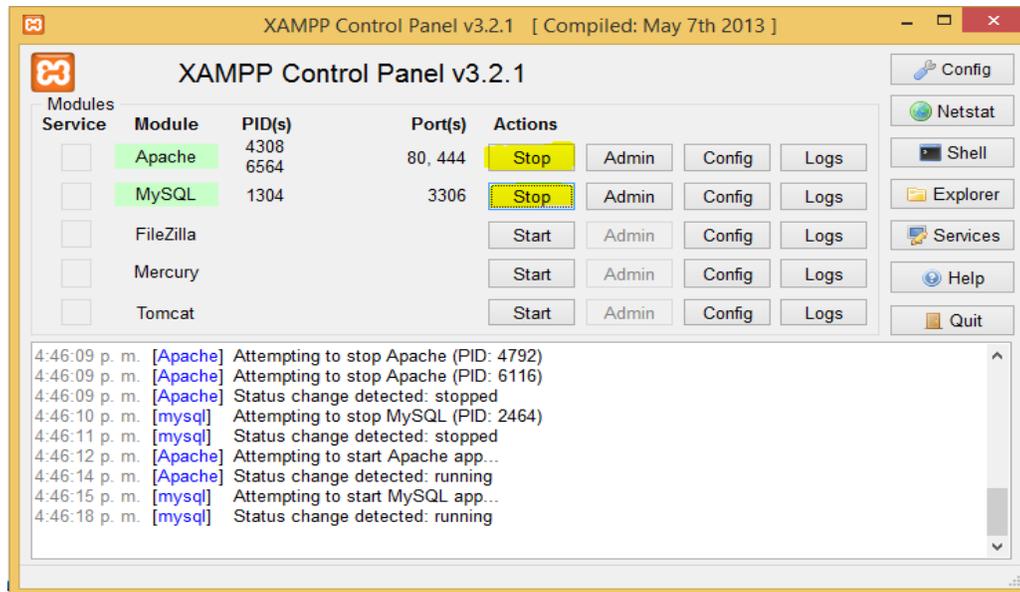


Ilustración 94 Panel de control XAMPP

Lo siguiente es montar la base de datos, para ello se abre la página de inicio de XAMPP, desde nuestro navegador preferido a través de la dirección <http://localhost>, a continuación se selecciona el link de phpMyAdmin.



Ilustración 95 Página de inicio XAMPP

El navegador enviará a la página de phpMyAdmin y de esta forma conseguir trabajar con las opciones del gestor de la base de datos MySQL. Para crear una nueva base de datos, se selecciona “Nueva” ubicado en el panel izquierdo, se le da un nombre y se crea.

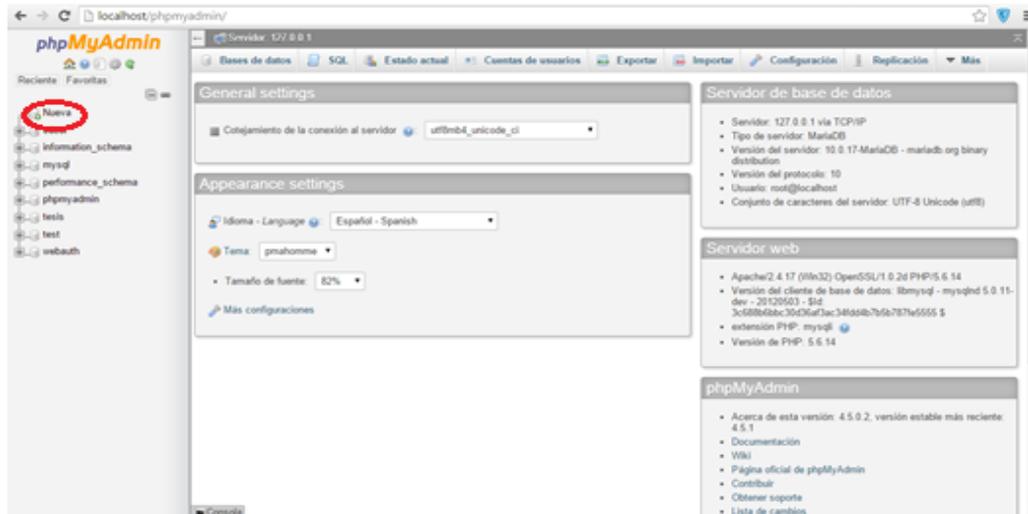


Ilustración 96 Página de phpMyAdmin

Luego de haber creado la base de datos se importa el archivo `vigilant.sql` ubicado dentro del proyecto “`vigilant_kinent_luxand`” o “`vigilant_kinect_opencv`”, es necesario instalar las 2 bases de datos si se tiene la intención de utilizar ambas aplicaciones.

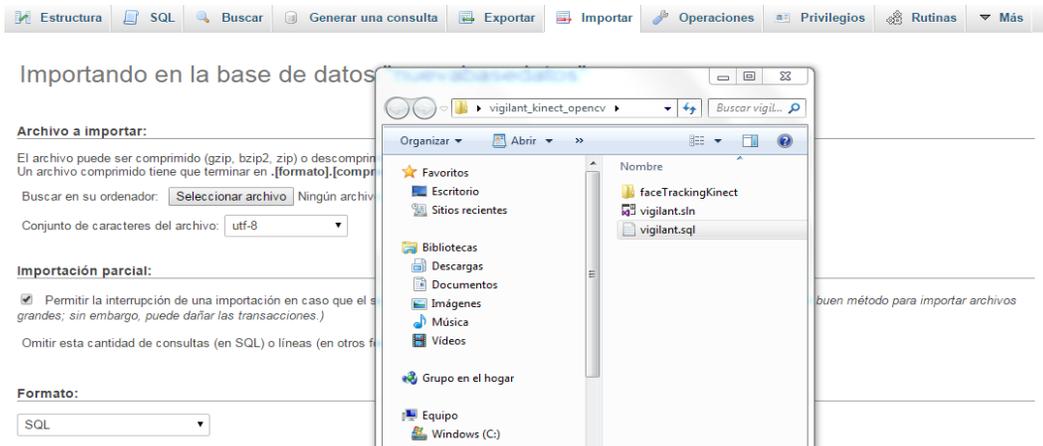


Ilustración 97 Importación de la base de datos

Por último solo queda ejecutar la aplicación “`Vigilant.exe`”.

4. Manual de usuario

Luego de haber instalado los dispositivos en el vehículo se procede a iniciar el sistema, la interfaz principal del sistema se verá como se muestra a continuación.



Ilustración 98 Ventana Índice

Para registrar un usuario, se presiona el botón “Registrar”, se abrirá una ventana donde se pedirá los siguientes datos:

- Nombre del conductor
- Apellido del conductor
- Cedula, debe ser única y solo debe contener caracteres alfanuméricos.
- Contraseña
- Verificación de contraseña



Ilustración 99 Ventana Registrar Datos

Si la cedula ya se encuentra registrada en la base de datos, entonces se lanzara un mensaje de error indicando el problema. El botón “Atrás” devolverá al usuario a la ventana principal.

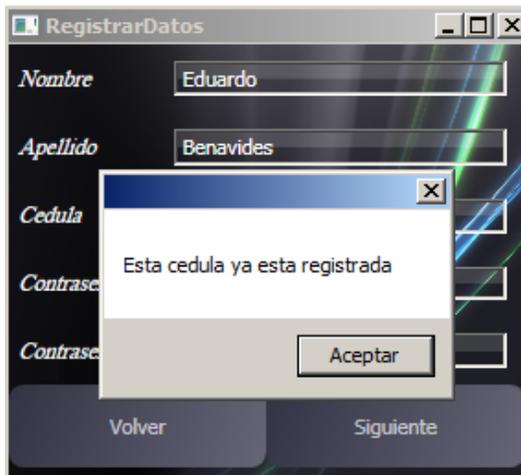


Ilustración 100 Erros en los Datos

Cuando se presiona el botón “Siguiete” la próxima ventana será la de registrar el fotograma de la cara del conductor.



Ilustración 101 Ventana Capturar fotograma del rostro

Solo se puede capturar un fotograma, mientras que este fotograma no es capturado, no se podrá continuar con el sistema.



Ilustración 102 Frame Capturado

Una vez se complete el registro se notificara al conductor con el siguiente mensaje.

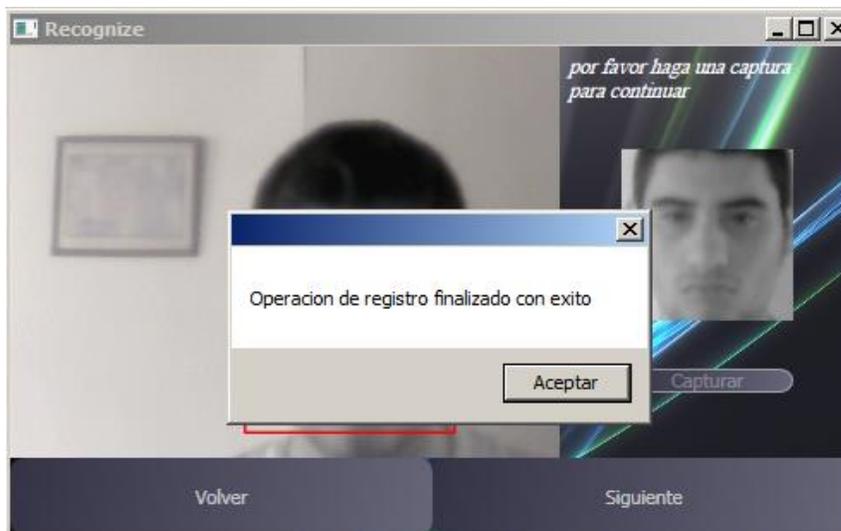


Ilustración 103 Ventana de Confirmación de Datos

El programa permanecerá en espera hasta que el cliente (sistema móvil) se conecte a éste, luego de este paso saldrá la ventana que maneja el flujo principal del sistema.



Ilustración 104 Ventana principal

Presionar el botón “Iniciar Sesión” en la ventana principal para identificar el usuario. En la próxima ventana, el sistema tratara identificar al conductor, en caso de que lo logre, el sistema estará en estado de espera mientras se conecta el cliente. Luego de esto se enviara al usuario a la ventana que maneja el flujo principal del sistema.

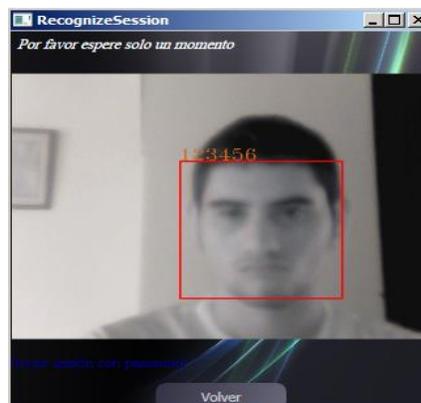


Ilustración 105 Ventana de iniciar sesión con reconocimiento facial

En caso de que no se pueda identificar al usuario a través del rostro del conductor, en esa misma ventana se encuentra un link para iniciar sesión a través de la cedula y contraseña con la que el conductor se registró.



Ilustración 106 Ventana de inicio de sesión manual

En la ventana que maneja el flujo principal del sistema se encuentran dos radio para activar o desactivar el Kinect o la cámara IR

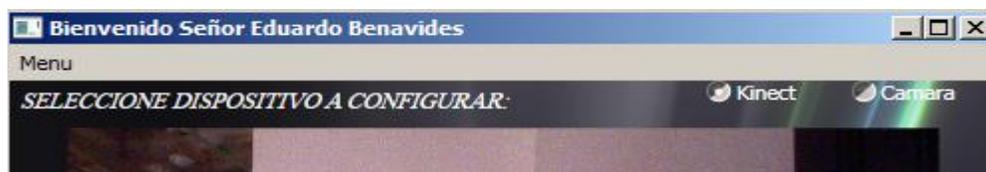


Ilustración 107 Opciones para configurar dispositivos

En esta ventana se encuentran algunos ítems dentro de la barra de menú.



Ilustración 108 Menú de configuraciones

El menú de configuración abre una ventana que muestra los datos de configuración de un conductor, en esta ventana el conductor también puede actualizar estos datos.



Ilustración 109 Ventana Configuraciones

El menú “Mostrar Ojos” abre una ventana con la imagen del ojo en su forma original y binaria ver ilustración 109. El menú “Mostrar Boca” abre una ventana con la imagen de la boca en su forma original y binaria ver ilustración 110.



Ilustración 110 Ventana Ojos



Ilustración 111 Ventana Boca

El menú “Cerrar Sesión” devuelve al usuario a la ventana principal.

El menú “Salir” cierra el sistema.

5. De cámara RGB a cámara IR

Kinect cuenta con una cámara IR y un proyector de infrarrojos pero el flujo de datos arrojados por éste sensor (no es algo con lo que se pudo obtener una imagen para aplicarle un clasificador en cascada u otro método de detección de características faciales). Por lo tanto se desarrolló un sencillo sistema de visión nocturna compuesto en su totalidad con una pequeña cámara web convencional y 4 leds infrarrojos.



Ilustración 112 Cámara Normal



Ilustración 113 LEDs Infrarrojos

Toda cámara es capaz de captar la luz infrarroja una prueba de ello es encender la cámara de un celular y enfrente de ésta presionar cualquier tecla de un control remoto convencional y se podrá observar el destello que es totalmente invisible para

los ojos humanos, solo que éstas cámaras poseen un filtro infrarrojo que permite capturar todos los colores del “espectro de luz” evitando prácticamente en su totalidad la vista de luz infrarroja.

1.1. Fabricación

Para transformar la cámara en un dispositivo capaz de ver toda luz infrarroja se llevaron a cabo los siguientes pasos:

1. Desarmar la cámara.



Ilustración 114 Cámara desarmada

2. Retirar el filtro y los leds normales con ayuda de una pistola de estaño y soldar los leds infrarrojos. Se decidió que 4 era el número correcto debido a que podían proporcionar la luz suficiente para poder capturar la imagen, si se excedía de este número el reflejo de la luz infrarroja en el lente de la cámara sería demasiado y no se podría capturar una buena imagen por la cercanía que para este proyecto el rostro se encontraba de la cámara



Ilustración 115 LEDs Infrarrojos instalados

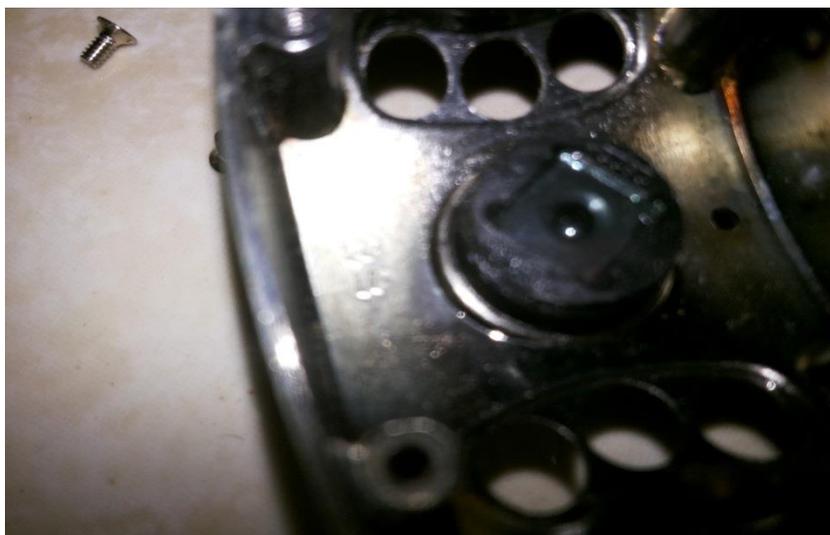


Ilustración 116 Filtro que se debe remover

3. Armar el dispositivo.



Ilustración 117 Cámara IR terminada

De esta forma concluye la fabricación del sistema de visión nocturna.