UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

"SISTEMA ELECTRÓNICO CON APLICACIÓN IoT PARA MONITOREO FACIAL QUE BRINDE ESTIMADORES DE DESCONCENTRACIÓN DEL ESTUDIANTE UNIVERSITARIO EN EL AULA A ESCALA DE LABORATORIO"

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

AUTOR: VANESSA ESTEFANÍA ALVEAR PUERTAS

DIRECTOR: ING. JAIME MICHILENA

Ibarra – Ecuador

2016

UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

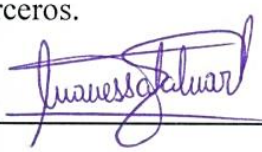
DATOS DEL AUTOR	
Cédula de identidad	100333530-2
Apellidos y Nombres	Alvear Puertas Vanessa Estefanía
Dirección	Juan de Velasco 08-97 y Dos de Marzo
E-mail	vealvearp@utn.edu.ec
Teléfono	0988616042
DATOS DE LA OBRA	
Título	SISTEMA ELECTRÓNICO CON APLICACIÓN IoT PARA MONITOREO FACIAL QUE BRINDE ESTIMADORES DE DESCONCENTRACIÓN DEL ESTUDIANTE UNIVERSITARIO EN EL AULA A ESCALA DE LABORATORIO
Autor	Alvear Puertas Vanessa Estefanía
Fecha	28 de noviembre de 2016
Programa	Pregrado
Título por el que opta	Ingeniera en Electrónica y Redes de Comunicación
Director	Ing. Jaime Michilena

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Alvear Puertas Vanessa Estefanía, con cédula de identidad Nro. 100333530-2, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad de material y como apoyo a la educación, investigación y extensión , en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIAS

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



Vanessa Estefanía Alvear Puertas

Cédula: 100333530-2

Ibarra, noviembre de 2016

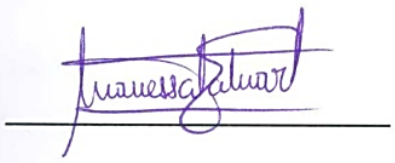
UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

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

Yo, VANESSA ESTEFANÍA ALVEAR PUERTAS, con cédula de identidad Nro. 100333530-2, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador artículos 4,5 y 6, en calidad de autor del trabajo de grado con el tema: "SISTEMA ELECTRÓNICO CON APLICACIÓN IoT PARA MONITOREO FACIAL QUE BRINDE ESTIMADORES DE DESCONCENTRACIÓN DEL ESTUDIANTE UNIVERSITARIO EN EL AULA A ESCALA DE LABORATORIO". Que ha sido desarrollado para optar por el título de Ingeniero en Electrónica Redes de Comunicación de la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.



Vanessa Estefanía Alvear Puertas

Cédula: 100333530-2

Ibarra, a los 28 días del mes de noviembre de 2016

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL ASESOR

INGENIERO JAIME MICHILENA, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN,

CERTIFICA:

Que, el presente trabajo de titulación: **"SISTEMA ELECTRÓNICO CON APLICACIÓN IoT PARA MONITOREO FACIAL QUE BRINDE ESTIMADORES DE DESCONCENTRACIÓN DEL ESTUDIANTE UNIVERSITARIO EN EL AULA A ESCALA DE LABORATORIO".** Ha sido desarrollado por la Señorita Vanessa Estefanía Alvear Puertas, portadora de la cédula de identidad número 100333530-2, bajo mi supervisión.

Es todo cuanto puedo certificar en honor a la verdad.

Ing. Jaime Michilena

DIRECTOR DE TESIS

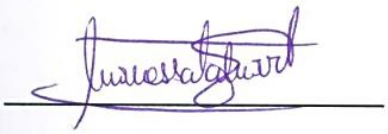
UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DECLARACIÓN

Yo, Vanessa Estefanía Alvear Puertas, con cédula de identidad 100333530-2, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; y que éste ha sido previamente presentado para ningún grado o calificación profesional.

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



Vanessa Estefanía Alvear Puertas

Cédula: 100333530-2

Ibarra, noviembre del 2016.

DEDICATORIA

Dedicado a Alex, por haber sido el mejor padre, hermano y amigo que pude tener. Tu recuerdo perdurará para siempre en mi vida. A Erick, Orly y Eimmy con inmenso amor.

AGRADECIMIENTO

Agradezco a todas las personas que confiaron y creyeron en que podía cumplir mis metas. A mi familia por ser un apoyo incondicional, de manera especial a mi madre por ser un faro en el camino, y jamás dejarme sola. A Isaac, Alex, Nathalie, Sayeli, Fernando, Piter y Lorena por siempre estar a mi lado y ser una parte tan importante de mi vida.

Mi profundo agradecimiento al Ingeniero Paul Rosero quien además de idear temas tan innovadores y creativos como este Trabajo de Grado, ha sido un excelente profesional, un guía y un gran mentor para muchos estudiantes que buscamos crear nuevos proyectos y desarrollar nuevas aplicaciones en el campo de la Electrónica. Durante el desarrollo de este proyecto nos encaminó a una forma diferente de investigar y de hacer que el trabajo de los estudiantes universitarios sea valorado más allá de las aulas de clase, permitiendo que un proyecto se convierta en una realidad, gracias por impulsarnos a superarnos e invitarnos a seguir preparándonos, por demostrarnos que todo sacrificio tiene su recompensa y por ser un gran ejemplo a seguir.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO DE TÉRMINOS

- **Adaboost**: Adaptación del algoritmo de Boosting (Adaptative Boosting)
- **Boosting:** Algoritmo de aprendizaje que comprende una mejora del algoritmo Base, se aplica repetidamente el algoritmo hasta conseguir el mejor resultado
- **Falsos positivos:** objetos que se detectan en una imagen a pesar de que no representan una característica para la detección
- **LTE:** por sus siglas Long Term Evolution, comprende un estándar para comunicaciones inalámbricas con altas tasas de transmisión
- **M2M:** Componente de internet de las cosas que comprende la comunicación entre dos máquinas (Machine to Machine = Máquina a Máquina)
- **M2P:** Componente de internet de las cosas que comprende la comunicación entre una máquina y una persona (Machine to People = Máquina a Persona)
- **Script:** es un archivo en texto plano que contiene instrucciones en un orden establecido que pueden ser ejecutadas para el desarrollo de programas de distintos lenguajes de programación
- **Wimax:** tecnología inalámbrica de largo alcance, puede llegar a cubrir distancias de hasta 50 km, utiliza las ondas de radio en las frecuencias de 2,5 a 5,8 GHz
- **WLAN:** de las siglas Wirelles Local Area Network, representa las comunicaciones inalámbricas en redes de área local
- **xDSL:** grupo de tecnologías de acceso a internet de banda ancha basadas línea de abonados digitales
- **XML:** Extensible Markup Language, estándar que permite el intercambio de información estructurada entre distintas plataformas

RESUMEN

El sistema de monitoreo facial para determinar la desconcentración en estudiantes universitarios, es un sistema que tiene como objetivo brindar estimadores de desconcentración de los estudiantes de la carrera de Ingeniería en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte. Para el desarrollo de este sistema se emplearon herramientas de visión artificial mediante el uso de librerías de OpenCv y scripts de Phyton. El hardware utilizado fue una placa Raspberry Pi 2 Modelo B y el módulo Pi Camera para la detección del rostro. Se realizó una fase de levantamiento de información de la situación actual con lo cual se determinó que la principal causa de desconcentración de los estudiantes se debe a la fatiga y a la falta de horas de descanso. Se tomaron en cuenta las variables de número de pestañeos y número de bostezos como factores fisiológicos que determinan si una persona está cansada o no.

Se realizaron pruebas preliminares y pruebas finales que permitieron verificar el correcto funcionamiento del sistema de monitoreo facial, éste permite identificar la fatiga en los estudiantes y por lo tanto la relaciona con la falta de concentración, los datos obtenidos son subidos en tiempo real a una plataforma en Internet donde los docentes pueden visualizar los estimadores de desconcentración en sus periodos académicos.

ABSTRACT

The system of facial monitoring to determine the deconcentration in university students, is a system that aims to provide estimators of deconcentration of the students of the Engineering Degree in Electronics and Communication Networks of the Universidad Técnica del Norte. For the development of this system artificial vision tools were used through the use of OpenCv libraries and Phyton scripts. The hardware used was a Raspberry Pi 2 Model B and the Pi Camera module for face detection. A phase of information gathering of the current situation was carried out, which determined that the main cause of deconcentration of the students is due to the fatigue and the lack of rest hours. The variables of number of eyelashes and number of yawns were taken into account as physiological factors that determine if a person is tired or not.

Preliminary tests and final tests were carried out to check the correct functioning of the facial monitoring system, to identify the fatigue in the students and thus to relate it to the lack of concentration, the data obtained are uploaded in real time to a platform On the Internet where teachers can visualize the deconcentration estimators in their academic periods.

CAPÍTULO I

Antecedentes

El capítulo de Antecedentes busca dar una breve introducción al lector debido a que le permite conocer las razones que impulsaron al desarrollo de este proyecto. A continuación, se presenta la problemática, los objetivos tanto general como específicos planteados, el alcance y la justificación que fundamenta este proyecto.

1.1 Problema

Internet de las Cosas (o también conocido como IoT, por sus siglas en inglés Internet of Things) es el instante del tiempo en el que se conectaron a internet más “cosas u objetos” que personas, así lo explica el IBSG (Internet Business Solution Group) de Cisco. Para el año 2003 se tenía 6,3 mil millones de personas y unos 500 millones de dispositivos conectados a Internet, es decir existía 0,08 dispositivos por persona; para finales de 2010 la cantidad de dispositivos por persona fue de 1,84. Con el nacimiento de IoT, el número de dispositivos conectados en el 2015 es de alrededor de 25 mil millones y se prevé que se elevará a 50 mil millones para 2020, con esta última cifra se estima que el número de dispositivos por persona será de 6,58.

La situación del Ecuador respecto al acceso a las Tecnologías de la Información y Comunicación ha tenido de igual manera un crecimiento muy acelerado, según datos del Instituto Nacional de Estadísticas y Censos, el porcentaje de personas que tenían acceso a internet en el 2013 ha subido al 28,3% y para el 2015 se estima que ha llegado al 45%. En los

estudios realizados sobre el acceso a las TICs[1] las provincias con mayores índices son Pichincha, Azuay e Imbabura, tanto en uso de medios informáticos como en conocimiento general sobre las tecnologías actuales. Este aumento que se registra cada año se debe en gran parte a la implementación del Plan Nacional del Buen Vivir 2013-2017 que contempla entre sus objetivos estratégicos el acceso universal a las TICs, respaldando de esta manera al desarrollo de nuevas aplicaciones y servicios tecnológicos en los campos educativos, empresariales e informativos para lograr un cambio en la matriz productiva del país. Sin embargo, a pesar del PNVB en la actualidad en el sector educativo del país no se fomentan proyectos tecnológicos que mejoren las capacidades de los estudiantes. (INEC, 2013)

Según el Instituto Nacional de Estadísticas y Censos, la tasa neta de escolarización superior es de 59,1% y la edad promedio en la que un ecuatoriano/a obtiene su título universitario es de 24 a 25 años, lo que indica que se tarda alrededor de 7 años para finalizar una carrera universitaria cifra que es alarmante para los expertos en educación. (INEC, 2013)

Los factores que intervienen para que un estudiante universitario se tarde este tiempo en cursar sus estudios son principalmente ligados a su desempeño académico, debido a que la mayoría de universidades ecuatorianas tienen altos índices de estudiantes repitiendo una o varias materias, en el caso específico de la UTN en la Facultad de Ingeniería en Ciencias Aplicadas en la Carrera de Electrónica y Redes de Comunicación para el semestre de Octubre 2015 a Febrero 2016 se tiene un 44% de estudiantes con segunda matrícula. Este tipo de casos serían ocasionados por la falta de concentración de los estudiantes en sus clases, afectando su aprendizaje y por ende su desempeño dentro del aula.

[1] TICs: Tecnologías de la Información y Comunicación.

Los estudiantes universitarios deben cumplir con numerosas tareas fuera del aula lo cual conlleva a que dediquen parte de sus horas de sueño para cumplir con estas responsabilidades, lo cual afecta a su estado físico dejándolos cansados y con falta de horas de descanso, si a esto le agregamos que deben cumplir con un horario de clases muy extenso encontramos que son factores que sin duda contribuyen a la desconcentración. El bajo rendimiento de un estudiante afectará no solo a su aprobación de la materia, sino que conlleva a que su conducta en ámbitos familiares y sociales no sea la más adecuada, afectando además a su futuro como profesional.

En base a estos antecedentes de acceso a las TICs y el desarrollo de IoT se plantea la creación de un sistema de monitoreo a los estudiantes dentro de sus aulas de clase que permita indicar al profesor en que momento sus alumnos están perdiendo la concentración realizando un monitoreo facial. Brindándole al docente la posibilidad de analizar su método de aprendizaje y cómo este es receptado por los alumnos y a partir de esto poder mejorar los problemas antes indicados.

1.2 Objetivos

1.2.1 Objetivo General.

Desarrollar un sistema electrónico con aplicación IoT para el monitoreo facial que brinde estimadores de desconcentración de un estudiante universitario dentro del aula, durante ocho meses en la Universidad Técnica del Norte.

1.2.2 Objetivos Específicos.

- Realizar una revisión sistemática de literatura para determinar las bases teóricas comprendidas en la revisión bibliográfica correspondiente al sistema.
- Definir los requerimientos de los usuarios tanto de software como de hardware bajo el estándar IEEE 29148 para determinar los materiales a usarse en el sistema.
- Analizar y determinar el hardware y el software idóneo en relación a los requerimientos establecidos para el diseño y desarrollo del sistema.
- Realizar las pruebas de funcionamiento tanto del sistema electrónico como de la aplicación para la depuración de errores.
- Analizar los datos que se obtengan como resultado del monitoreo facial a los estudiantes universitarios para establecer estimadores de desconcentración dentro del aula.

1.3 Alcance

En el transcurso de ocho meses se desarrollará un sistema de monitoreo facial que brindará estimadores de desconcentración de estudiantes universitarios dentro de un aula de clases, está orientado específicamente a los alumnos de la Facultad de Ingeniería en Ciencias Aplicadas, Carrera de Electrónica y Redes de Comunicación de la Universidad Técnica del Norte y se trata de un proyecto a escala de laboratorio.

Con la finalidad de obtener las bases teóricas requeridas en la revisión bibliográfica, se realizará una revisión sistemática de literatura especialmente en artículos científicos de revistas tecnológicas indexadas, entre otros textos referentes científicos.

Para realizar el levantamiento de requerimientos de los usuarios se seguirá la norma IEEE 29148, la cual facilitará el análisis y selección del software y hardware idóneos para el diseño y desarrollo del sistema. Una vez finalizado el trabajo se procederá a comprobar el correcto funcionamiento tanto de la placa como de la aplicación con IoT.

La selección del sistema embebido, cámaras para el reconocimiento facial y lenguaje para desarrollar la aplicación con IoT, serán seleccionados conforme se avance con el proyecto debido a que se debe realizar pruebas para determinar cuáles son los que mejor se adaptan a la solución planteada.

Finalmente se tendrá una etapa de análisis de datos del sistema de monitoreo facial. Es importante indicar que dentro del alcance del proyecto se brindará los estimadores de desconcentración sin embargo las actividades o decisiones que tome cada docente para mejorar dichos indicadores dentro de sus horas clase, serán independientes y no están consideradas dentro de este proyecto.

1.4 Justificación

El desarrollo de IoT abre un mundo de posibilidades y mejoras en la calidad de vida de las personas debido a que nos permite hacer de Internet una red sensorial a través de la cual podamos conocer lo que está sucediendo a nuestro alrededor, desde conocer la temperatura ambiente hasta asegurarnos de la seguridad de nuestro hogar. En la opinión de IBSG, IoT nos permitirá ser más proactivos y menos reactivos, llevando la tecnología a sus niveles más altos.

Dentro del Ecuador los proyectos tecnológicos han adquirido un gran respaldo por parte del Gobierno Nacional ya que con el desarrollo de éstos se contribuye directamente al cumplimiento del Plan Nacional del Buen Vivir, específicamente se hace un aporte al objetivo 11, en su sección 11.3 cuyo principio es democratizar la prestación de servicios de telecomunicaciones y el acceso universal a las TICs. El cumplimiento de este objetivo del PNBV no solo contribuye a una sociedad con mayor conocimiento tecnológico, sino que también asegura el correcto desarrollo de este proyecto al disponer de los recursos tecnológicos en las instituciones educativas donde podría ser implementado.

Según la página web oficial del Banco Mundial y sus estudios realizados sobre indicadores de desarrollo a nivel mundial, revelan que en universidades de países desarrollados se tiene como edad promedio para obtener un título de pregrado los 23 años y la edad promedio para estudiantes de posgrado es de 27 años, convirtiéndose a tempranas edades en entes productivos para la sociedad y con una mejor calidad de vida no solo en lo económico sino también en el aspecto social. Por su parte los estudiantes de universidades de países subdesarrollados han retrasado el proceso de obtención de su título de pregrado reduciendo su productividad y aporte a la sociedad. Por esta razón es de vital importancia buscar alternativas que trabajen en forma conjunta con la tecnología y que permitan mejorar los procesos educativos dentro del país.

En la actualidad una de las herramientas tecnológicas más aplicadas es la visión artificial, esta tecnología consiste en captar con cámaras (de distintos tipos) eventos de la vida diaria para ser interpretados por un computador o sistema embebido, ofrece diversas ventajas, pero principalmente permite automatizar sistemas mejorando su rendimiento. Al aplicar este

concepto junto con un sistema embebido se obtiene un sistema inteligente capaz de tomar decisiones por su propia cuenta el cual nos permite un monitoreo a tiempo real.

Con la implementación del sistema de monitoreo facial y un adecuado manejo y análisis de los datos obtenidos se podría mejorar las estrategias de enseñanza dentro del aula, donde cada docente tomaría las medidas adecuadas para dictar sus clases lo cual mejoraría el proceso de aprendizaje y por ende el rendimiento académico de los estudiantes.

CAPÍTULO II

Revisión Bibliográfica

En el presente capítulo se dará a conocer aspectos generales sobre Internet de las Cosas, sus limitaciones, aplicaciones, arquitectura, protocolos y componentes. También se abordarán los principios de la Visión Artificial y dentro de ella la Detección Facial y el Tratamiento de Imágenes, indicando los principales algoritmos de detección facial dando un especial enfoque en el algoritmo Adaboost. Más adelante se trata sobre los conceptos de Open Source refiriéndose tanto a Open Hardware y Open Software, los beneficios que estos proporcionan al usuario y sus condiciones de uso.

Respecto al software y hardware empleados para este proyecto, se explica en este capítulo de manera breve sus principales características que fundamentan su utilización en este proyecto. Finalmente se abordan los conceptos básicos de estadística que constituye una parte fundamental para el desarrollo del proyecto.

2.1 Internet of Things

Internet of Things (conocida también como IoT) es una plataforma capaz de conectar dispositivos y sensores que permitan monitorear aspectos de la vida cotidiana, representa de cierta manera la próxima evolución de Internet. En la actualidad se aplica IoT en áreas de salud, construcciones, tráfico vehicular, agricultura, etc. Se prevé que para el 2020 existirán alrededor de 50 mil millones de dispositivos conectados. (Ansari, A. N., Sedky, M., Sedky, M., & Tyagi, A., 2015)

Los inicios de Internet of Things se remontan al año de 1999 en el Instituto Tecnológico de Massachusetts, donde se trabajó con la implementación de un sistema de identificación basado en RFID el cual sería el punto de partida para futuras investigaciones de IoT (Cisco IBSG, 2011) . Para el año 2005 la ITU[2] publica su primer trabajo en el que define IoT como una nueva dimensión al mundo de las TICs, en donde se multiplicarán las conexiones y crearán una red dinámica de redes con redes.

En varios textos científicos se definen tres componentes fundamentales de IoT que son: comunicaciones Machine to Machine (M2M), comunicaciones Machine to People (M2P) y por último People to People (P2P), todos estos factores deben mantenerse en constante interacción permitiendo el intercambio de información, más adelante en este trabajo se indicará el principio que rige a cada uno de estos componentes.

2.1.1 Limitaciones de Internet of Things.

Las principales limitaciones de IoT se han evidenciado a medida que los dispositivos han excedido la capacidad de procesamiento y almacenamiento disponible en los sistemas actuales, a continuación se indican cuáles son los principales retos que deben ser superados por IoT:

- Limitaciones a nivel de seguridad: Ocasionadas por la falta de un protocolo que permita que los datos puedan ser encriptados y desencriptados. (Ansari, A. N., Sedky, M., Sedky, M., & Tyagi, A., 2015)

[2] International Telecommunications Union

- Limitaciones a nivel del número de direcciones IPv4 disponibles: se requiere migrar al protocolo IPv6 donde el número de direcciones IP asignables se adapta al acelerado aumento de dispositivos inteligentes en el mundo, se prevé que para el año 2020 existan 50 billones de dispositivos conectados a la Internet. (Cisco IBSG, 2011)
- Limitaciones a nivel de almacenamiento de información: debido al crecimiento exponencial de los datos que se envían y se reciben cada segundo en la Internet, se requiere implementar un storage con alta disponibilidad y capacidad de almacenamiento. (Ansari, A. N., Sedky, M., Sedky, M., & Tyagi, A., 2015)

2.1.2 Aplicaciones de Internet of Things

IoT ha cambiado el paradigma referente al uso de la tecnología y es así que ahora podemos conectar a Internet un sinnúmero de objetos que van desde termostatos, electrodomésticos, dispositivos móviles, sistemas de vigilancia, etc., lo único que se requiere es poder asociarles una dirección IP. A partir de este concepto se han desarrollado aplicaciones en distintos ámbitos de la vida cotidiana que buscan mejorar la calidad de vida de los habitantes tal como es el objetivo de las Smart Cities. A continuación, se citan algunos ejemplos de aplicaciones con IoT que se están desarrollando en la actualidad:

- IoT en la medicina: La reciente investigación realizada por (Alwakeel, Alhalabi, Aggoune, & Alwakeel, 2015) describen un Sistema Electrónico y Sistema de Asistencia para niños con autismo, el cual se basa en algoritmos de Machine Learning usando una red de sensores inalámbricos (WSN) para identificar los movimientos de los niños y alertar a sus padres de una posible conducta peligrosa, con el desarrollo de este sistema se busca dar una mayor protección a los niños autistas en situaciones que

podrían comprometer su integridad física. En el campo de los textiles inteligentes también se han desarrollado aplicaciones que vinculan IoT y la medicina como lo son los dispositivos encargados de monitorear la presión arterial, problemas cardiacos, niveles de azúcar, entre otros, donde se permite realizar un seguimiento más adecuado de las condiciones físicas de los pacientes.

- IoT en la agricultura: (Dlodlo & Kalezhi, 2015) en su reciente investigación indican ciertos mecanismos que vinculan IoT con la agricultura, el estudio realizado en Sudáfrica muestra como las tecnologías de IoT pueden maximizar el retorno de la inversión en agricultura. La detección de la cantidad de agua de riego y detección de suelos a través de sensores ayudan a proteger los cultivos de los agricultores, además de forma inalámbrica se indica a la reserva de agua cuándo y que cantidad se debe irrigar. Igualmente han estudiado otra alternativa para los agricultores quienes pueden adoptar sistemas de goteo automatizado en las zonas donde el agua es escasa. Esto se puede lograr mediante la vinculación de datos de varios sensores que controlan los cultivos mejorando la producción y la eficiencia del trabajo de los agricultores. Estas aplicaciones se han adoptado en otros países donde los resultados han mostrado una mejora en la productividad.
- IoT en la educación: (Tianbo, 2012) en su trabajo de investigación indica como IoT está promoviendo una revolución en la educación superior, debido a que con el uso de estas tecnologías puede mejorar el proceso de aprendizaje y la relación entre docentes y estudiantes. Los cambios que deben ser adoptados en la educación superior van desde cambios en el campus hasta cambios en las metodologías de enseñanza, todos estos en busca de dinamizar el aprendizaje.

2.1.3 Arquitectura de Internet of Things.

En la actualidad no existe un estándar que especifique una arquitectura para IoT, debido a que se encuentra en proceso de desarrollo, sin embargo la mayoría de autores coincide en una estructura dividida en capas, es así que (Plauska & Damaševičius, 2014) indican que la estructura de IoT podría representarse como un modelo de tres capas como se muestra a continuación:

- HARDWARE (sensores, actuadores y dispositivos de comunicación), Los sensores permiten a los usuarios obtener información sobre su entorno, permitir nuevas formas de interacción con el usuario, y conectar el mundo real con información.
- MIDDLEWARE (herramientas informáticas), utilizadas para la captura y análisis de los datos. La información secundaria que se obtiene del sensor también se puede utilizar para sincronizar actividades de aprendizaje con el entorno físico.
- CAPA PRESENTACIÓN (o servicio web), permite que los datos aprendidos y analizados puedan ser visibles para el usuario.

2.1.4 Protocolos de Internet of Things

Un gran número de protocolos de comunicación, también denominado como "middleware" son utilizados para la comunicación extremo a extremo en IoT. En la actualidad los más usados son MQTT y CoAP, cabe recalcar que todos estos protocolos dependen directamente de la existencia del stack de protocolos de TCP/IP.

2.1.4.1 Protocolo MQTT

Las siglas MQTT provienen de MQ Telemetry Transport, este protocolo es una implementación realmente ligera y sencilla de un protocolo de mensajería, su diseño se orienta a redes con bajo ancho de banda y altos niveles de latencia. Las principales características de diseño se fundamentan en utilizar un mínimo ancho de banda y de energía, por lo cual se adapta a las tecnologías emergentes de M2M e IoT. Fue desarrollado en el año de 1999 por el Dr. Andy Stanford-Clark de IBM y Arlen Nipper de Arcom. Los puertos de TCP/IP reservados para MQTT son el 1883 y el 8883, este último para MQTT sobre SSL. (Luzuriaga, y otros, 2015)

El protocolo MQTT asocia un nivel de calidad de servicio (QoS) a su modo de operación. Se definen tres niveles de QoS: nivel cero (QoS = 0) en el cual se envía el mensaje solo una vez, nivel uno y nivel dos (QoS> 0) donde se garantiza que ningún mensaje enviado se perderá, esto implica un proceso de reconocimiento de una o dos etapas, respectivamente.

2.1.4.2 Protocolo CoAP.

El protocolo de aplicación restringida (COAP) es un protocolo de transferencia especializado para el uso con nodos y redes restringidas, entiéndase por restringidas aquellas que presentan baja potencia y altos índices de pérdidas. CoAP está diseñado para interactuar fácilmente con HTTP para la integración con la Web y su operación se basa en el protocolo UDP de capa transporte del modelo TCP/IP. (Collina, Bartolucci, Vanelli, & Corazza, 2014)

El modelo de interacción de CoAP es similar al modelo cliente / servidor de HTTP. Una solicitud CoAP es equivalente a la de HTTP y se envía por un cliente para solicitar una acción en un recurso en el servidor. El servidor envía una respuesta que puede incluir una representación de recursos. Este intercambio de información se realiza de forma asíncrona orientado al uso de datagramas UDP. El protocolo COAP define cuatro tipos de mensajes: Confirmable, No confirmable, Reconocimiento, Reset.

2.1.4.3 Diferencias entre CoAP y MQTT.

Luego de haber analizado cada protocolo de IoT se presentan las principales diferencias entre MQTT y CoAP:

a) El protocolo CoAP ejecuta en su capa de transporte el User Datagram Protocol (UDP) mientras que MQTT ejecuta TCP. (Collina, Bartolucci, Vanelli, & Corazza, 2014)
b) Por la falta de confiabilidad del protocolo UDP, se desarrolló para CoAP su propio mecanismo de fiabilidad mediante el uso de los mensajes propios de este protocolo que son: “Confirmable” y “No Confirmable”, solo los primeros requieren un acuse de recibo. Por su parte TCP no requiere un mecanismo de fiabilidad como el desarrollado para CoAP. (Collina, Bartolucci, Vanelli, & Corazza, 2014)
c) El protocolo CoAP no dispone de niveles de calidad de servicio diferenciados mientras que MQTT define 3 niveles distintos de calidad de servicio garantizando entrega de paquetes sin pérdidas de información. (Collina, Bartolucci, Vanelli, & Corazza, 2014)

2.1.5 Componentes de Internet Of Things

2.1.5.1 Machine to Machine (M2M)

(Fan, Chen, Kalogridis, Tan, & Kaleshi, 2012) *"En las comunicaciones máquina a máquina (M2M), un gran número de máquinas inteligentes / dispositivos se comunican directamente uno con el otro con poca o ninguna intervención humana para compartir información y tomar decisiones colaborativas de forma automática"*

En la figura 1 se muestra la arquitectura de M2M la cual está formada por los siguientes elementos: dispositivos M2M, red de área M2M (dominio de dispositivos), Gateway M2M, red de comunicaciones M2M (dominio de red) y las aplicaciones M2M (dominio de aplicación).

Los dispositivos M2M una vez conectados forman a su vez una red de área M2M, esta red puede estar basada en tecnologías Wireless como Zigbee[3], Bluetooth y RFID. Por otra parte, las redes de acceso encargadas de conectar el Gateway M2M con el núcleo M2M pueden ser xDSL, WLAN, LTE, Wimax, entre las principales. En M2M cada dispositivo inteligente conectado a la red analiza el entorno y aprende a tomar decisiones propias para mejorar el rendimiento de la red.

[3] Zigbee: tecnología inalámbrica de corto alcance y bajo consumo, trabaja sobre IEEE 802.15.4.

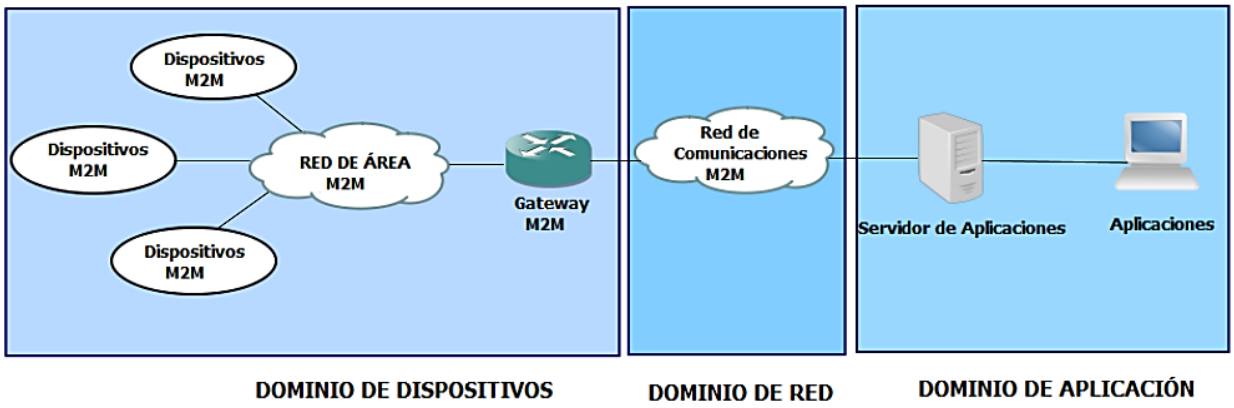


Figura 1. Arquitectura de comunicaciones M2M

Fuente: Autoría

Las aplicaciones M2M son diversas y requieren dispositivos con diferentes características y funciones para que puedan adaptarse a las necesidades de los usuarios, pueden ser aplicaciones de carácter público orientadas a seguridad y vigilancia de carreteras, cuidado del medio ambiente, mediciones de temperatura, automatización en servicios de transporte urbano o de carácter privado por ejemplo en Smart Homes, gestión de activos en empresas, sistemas de localización. Dependiendo de la aplicación se requiere o no características como movilidad, eficiencia energética y procesamiento, parámetros que deben ser considerados el momento de diseñar la red de comunicaciones M2M. (Kim, Lee, Kim, & Yun, 2014)

2.1.5.2 Machine Learning.

Machine Learning (ML) o Aprendizaje Automático es una herramienta creada para desarrollar tareas de inteligencia artificial a través de la implementación de robustos algoritmos computacionales, su principal objetivo es permitir que los sistemas puedan aprender por sí mismos sin la necesidad de que exista una reprogramación en el software,

ahorrando tiempo y mejorando el funcionamiento de las aplicaciones y sistemas electrónicos. (Alwakeel, Alhalabi, Aggoune, & Alwakeel, 2015).

Por otra parte, Arthur Samuel describe a Machine Learning como “el campo de estudio que proporciona a las computadoras la habilidad para aprender sin ser programadas de forma explícita”. Tom Mitchell nos proporciona una definición más formal: “Un programa de computadora se dice que aprende de la experiencia E con respecto a algún tipo de tareas T y con una medida de rendimiento P, si su desempeño en las tareas T, medido por P mejora con la experiencia E”.

El campo de desarrollo de ML se divide en dos áreas que son: Aprendizaje Supervisado y Aprendizaje no Supervisado a continuación se indican las principales características de cada uno:

- APRENDIZAJE SUPERVISADO: para este tipo de aprendizaje el objetivo principal es hacer predicciones a futuro basados en el conjunto de datos almacenados en el sistema, para ello se buscan patrones tomando en cuenta las decisiones anteriores del usuario. Los problemas de Aprendizaje Supervisado se dividen en dos categorías: Regresión y Clasificación. En un problema de Regresión se trata de predecir el resultado dentro de una salida continua mientras que para los problemas de Clasificación se predice los resultados para una salida discreta. (Deng & Li, 2013)
- APRENDIZAJE NO SUPERVISADO: en este tipo de aprendizaje los datos proporcionados no guardan relación con anteriores datos almacenados por lo que no se puede encontrar un patrón definido, en este caso lo que hace es analizar la estructura de los datos y agruparlos en función de características similares. Un

ejemplo de aplicación son los estudios de marketing donde se segmenta el mercado agrupando a los clientes con comportamientos similares. (Deng & Li, 2013)

El objetivo que rige al proceso de aprendizaje automático es encontrar la Función de Hipótesis $\boldsymbol{h_\theta(x)}$**,** que permita asignar los datos de entrada **x** los datos de salida **y.** Un ejemplo citado en la mayoría de textos nos dice que, para el caso de la compra de una casa, sea x= (metros de construcción, número de dormitorios, número de baños, años de construcción), el valor de la función $\boldsymbol{h_\theta(x)}$ debe predecir el precio de venta de la casa basado en los parámetros conocidos. En la ecuación 1 se representa la Función de Hipótesis con su forma general. (Standford University, Cousera, s.f.)

$$\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x}$$

Ecuación 1. Forma General de la Función de Hipótesis en Machine Learning

Fuente: (Standford University, Cousera, s.f.)

Para medir la exactitud de la Función de Hipótesis se utiliza la Función de Costo, esta última toma un promedio del resultado de la hipótesis y compara con valores reales obtenidos para cada *y*. En la ecuación 2 se presenta la forma general de la Función de Costo. (Standford University, Cousera, s.f.)

$$\boldsymbol{J(\theta_0\,,\theta_1) = \frac{1}{2m} + \sum_{i=1}^{m} \left(h_\theta(x^{(i)} + y^{(i)}\right)^2}$$

Ecuación 2. Forma General de la Función de Costo en Machine Learning

Fuente: (Standford University, Cousera, s.f.)

donde m= cantidad de elementos del conjunto de datos

2.2 Visión Artificial

El campo de estudio de la visión artificial o visión por computador es uno de los componentes de la Inteligencia Artificial, comprende un conjunto de técnicas que permiten que un computador o dispositivo electrónico sea capaz de entender e interpretar el significado de distintas imágenes o escenarios con distintas características. A partir del reconocimiento de patrones y procesamiento de imágenes los sistemas que usan visión artificial pretenden dotar a los computadores de la habilidad de simular la visión humana, siendo capaces de tomar decisiones autónomas basados en conjuntos de entrenamiento previos y ejecutando procesos que mejoran la automatización.

2.2.1 Objetivos de la Visión Artificial

En el trabajo de Cazorla sobre Robótica y Visión Artificial, se detallan los principales objetivos de la visión artificial:

- Detección, segmentación y localización de patrones en imágenes con distintas características.
- Captura y registro de características imágenes en una solo captura o escena.
- Seguimiento de objetos en secuencias de imágenes, ya sea determinado por un color, forma, o figura especifica.
- Estimación de posturas tridimensionales.

2.2.2 Limitaciones de la Visión Artificial

El trabajo de Fernández García indica que al igual que capacidades, la visión artificial posee limitaciones al momento de simular la visión humana en un ordenador, las principales dificultades se presentan a continuación:

- *Cambios de iluminación:* con los distintos tonos de luz y los cambios bruscos de iluminación se crean sombras o reflejos que impiden tener una imagen clara para ser procesada por el computador.
- *Cambios de escala*: los cambios de tamaños en las imágenes impiden que se realice el seguimiento adecuado de los objetos, con lo que se requiere que se reinicie la búsqueda y procesamiento del mismo, incrementando el tiempo de procesamiento y empleando más recursos del hardware.
- *Deformación de los objetos*: una imagen deformada impide que se realice el seguimiento y extracción de las características de los objetos de interés
- *Oclusión*: ocurre cuando la imagen que se desea procesar se encuentra detrás de otra, con lo que la única información que obtenemos es la que se encuentra visible, generando pérdida de datos sobre la imagen.

2.2.3 Principales aplicaciones de la Visión Artificial

Las aplicaciones potenciales relacionadas a la visión artificial comprenden una amplia variedad de sistemas tanto en el ámbito industrial como en el ámbito académico, a continuación, se presenta un breve listado de estas aplicaciones (Morante Cendrero, 2012)

- *Vigilancia y seguridad*, aplicado en edificios inteligentes para conteo de personas, sistemas de control de acceso dentro de empresas, detección de acceso no autorizado en espacios controlados, sistemas de rastreo de personas y vehículos, entre los principales.
- *Control de calidad en industrias*, incluyen sistemas que permiten distinguir entre los productos alimenticios en buen o mal estado, hasta aplicaciones que requieren mayor precisión como la clasificación de materiales, por ejemplo, la diferenciación entre tuercas y tornillos
- *Interacción humano-robot*, este campo de estudio es imprescindible para el desarrollo de futuros robots con capacidades avanzadas aplicadas al desarrollo de actividades cotidianas
- *Seguimiento de trayectorias,* aplicaciones como el famoso ojo de halcón implementado en los partidos de tenis permiten reconstruir el movimiento de objetos en tiempo real.

2.2.4 Detección facial.

Los sistemas de detección facial han tomado fuerza en los últimos años debido a que son ampliamente utilizados en sistemas biométricos, el objetivo de estos sistemas es identificar a un individuo en una imagen digital de forma automática.

La detección facial automática tuvo sus inicios en los años 60, el primer sistema semiautomático necesitaba que el administrador localice rasgos como ojos, cejas, nariz y boca en las imágenes para calcular distancias a determinados puntos de referencia que después serían analizados con datos almacenados en el sistema. Los próximos sistemas de

detección usaron distintos marcadores y técnicas para mejorar el proceso de detección. En la actualidad existen sofisticadas herramientas que son aplicadas en distintas áreas sociales como entretenimiento, tarjetas inteligentes (pasaportes, licencias de conducir), seguridad de la información, aplicación de la ley y vigilancia, entre otros.

A pesar de que la detección facial es una tarea cotidiana para el ser humano, la implementación de un software no ha sido tarea fácil y ha requerido amplias investigaciones que mejoren el proceso. El desarrollo e implementación de los algoritmos requeridos para los sistemas de detección facial se han subdividido en áreas específicas: detección de rostro, seguimiento, alineamiento, extracción de rasgos, entrenamiento, identificación, clasificación, análisis de expresión facial, análisis en 2 y 3 dimensiones (García, 2009).

2.2.5 Clasificación de los Métodos de Reconocimiento Facial.

Dentro de esta clasificación se abarca tres métodos que son los holísticos, los que se basan en características y los híbridos. Para el caso de los Métodos Holísticos son aquellos que utilizan toda la región de la cara; los Métodos Basados en Características buscan marcadores específicos como ojos, nariz, boca para luego usar un marcador estructural, requieren mayor precisión el momento de extraer los marcadores; los Métodos Híbridos son los más similares a la percepción humana, estos combinan los métodos holísticos y basados en características para reconocer un rostro de manera más precisa. (Jiménez Encalada, 2015)

Para cada uno de estos métodos se ha desarrollado varios trabajos que permitan mejorar el procedimiento de detección, a continuación en la figura 2, se muestran los más conocidos:

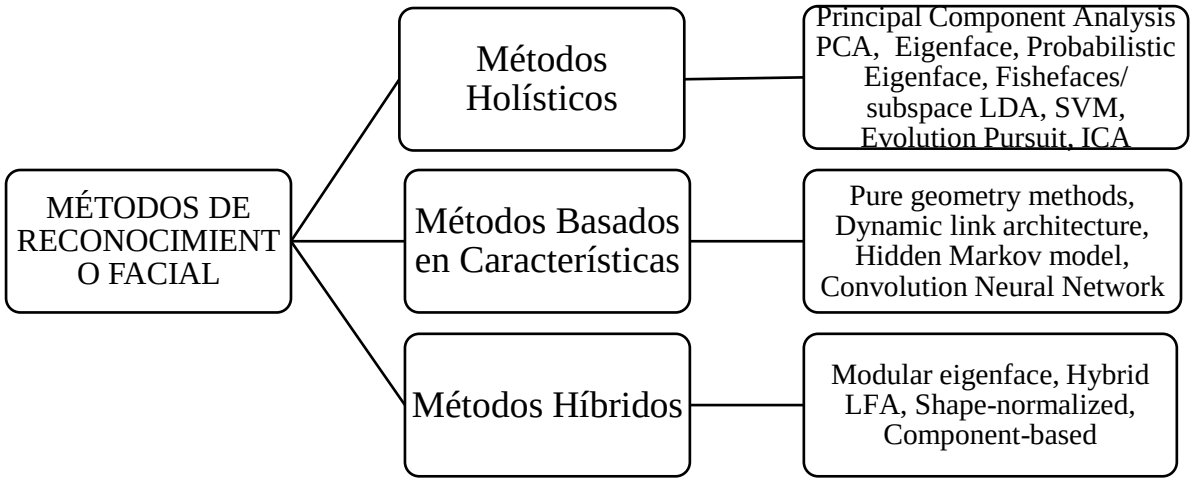


Figura 2. Clasificación de Métodos de Reconocimiento Facial

Fuente: Autoría

2.2.6 Algoritmos de Detección Facial.

En varios trabajos científicos se han desarrollado algoritmos de detección facial y cada uno ha tomado las mejores funciones de algoritmos anteriores, siempre buscando mejorar y reducir el tiempo que un sistema electrónico se tarda en detectar el rostro, sin embargo, ha llamado la atención de manera especial a los investigadores el denominado Boosting y posteriormente la adaptación a este algoritmo que se denomina Adaboost, a continuación, se indican las características de éstos.

2.2.3.1 Boosting y Adaboost.

El Boosting es un algoritmo basado en el aprendizaje de clasificadores débiles con respecto a una distribución para formar un clasificador final más fuerte. Su funcionamiento se basa en la pregunta: ¿Puede un conjunto de clasificadores débiles unificarse para crear un clasificador fuerte?

Se define a un clasificador débil como un clasificador que guarda una mínima relación con la clasificación final, mientras que un clasificador fuerte es aquel que guarda una fuerte relación con la clasificación final. (García, 2009)

El algoritmo Adaboost es un tipo de Boosting Adaptativo, fue propuesto por Freund y Schapire en el año de 1996, su objetivo es conseguir el aprendizaje a partir de modificar la distribución de muestras al finalizar cada iteración, tomando ventaja de los clasificadores débiles para formar un clasificador fuerte. Este algoritmo es comúnmente implementado usando árboles de decisión. A partir de Adaboost se han desarrollado diversos trabajos de detección facial, entre los más conocidos tenemos el realizado por Viola y Jones que crea un clasificador no lineal basado en el funcionamiento de Adaboost (Vivas).

2.3 Open Source

Open Source es una marca de certificación propiedad de la Open Source Initiative. Los desarrolladores que diseñan este tipo de software para ser compartido, mejorado y distribuido libremente, pueden usar la marca registrada Open Source siempre y cuando se adapten a las condiciones de Open Source Initiative.

Entre los términos que deben ser considerados tenemos: *Libre distribución*, no deben existir restricciones para vender o distribuir el software; *Código fuente*, el software debe incluir el código fuente y debe permitir crear distribuciones compiladas siempre y cuando la forma de obtener el código fuente esté expuesta claramente; *Trabajos derivados*, se debe permitir crear trabajos derivados que deben ser distribuidos bajo los mismos términos que la licencia original del software; Integridad del código fuente del autor, las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches; *No discriminar personas o grupos*, la licencia no debe discriminar a ninguna persona o grupo; *No discriminar áreas de iniciativa*, los usuarios comerciales no pueden ser excluidos (Open Source Initiative, s.f.).

2.3.1 Open Hardware.

Open Source Hardware Association afirma:

> Open Hardware es aquel hardware cuyo diseño se hace disponible públicamente para que cualquier persona lo pueda estudiar, modificar, distribuir, materializar y vender, tanto el original como otros objetos basados en ese diseño. Las fuentes del hardware (entendidas como los ficheros fuente) habrán de estar disponibles en un formato apropiado para poder realizar modificaciones sobre ellas. Idealmente, el hardware de fuentes abiertas utiliza componentes y materiales de alta disponibilidad, procesos estandarizados, infraestructuras abiertas, contenidos sin restricciones, y herramientas de fuentes abiertas de cara a maximizar la habilidad de los individuos para materializar y usar el hardware. (Open Source Hardware Association, s.f.)

2.3.1.1 Raspberry Pi 2 Modelo B.

Ordenador de placa reducida o SBC[4] (Single Board Computer) desarrollado por la Fundación Raspberry Pi, fue creado con el objetivo de mejorar la metodología de enseñanza de informática en las escuelas de Reino Unido, actualmente es utilizado a nivel mundial y contempla aplicaciones en otras áreas además de la educación.

La placa incluye: chip Broadcom BCM2835 que contiene un CPU a 700 MHz, procesador gráfico Broadcom Video Core IV, memoria RAM 2GB (dependiendo del modelo de la placa), el sistema operativo por defecto es RaspBian aunque soporta la mayoría de versiones de Linux e incluso se podría instalar Windows 10.

El sistema requiere una SD card para el almacenamiento debido a que no dispone de un disco duro incorporado. Entre las principales características de hardware cuenta con un puerto RJ-45, puertos USB, entradas y salidas de video con un consumo energético que va desde los 500mA hasta los 800mA en los modelos más recientes. En la figura 3 se puede apreciar una vista frontal de la placa Raspberry Pi 2 Modelo B (Raspberry Pi Foundation, s.f.)

[4] Single Board Computer: computador completo en un solo circuito, contiene microprocesador, RAM, periféricos E/S.

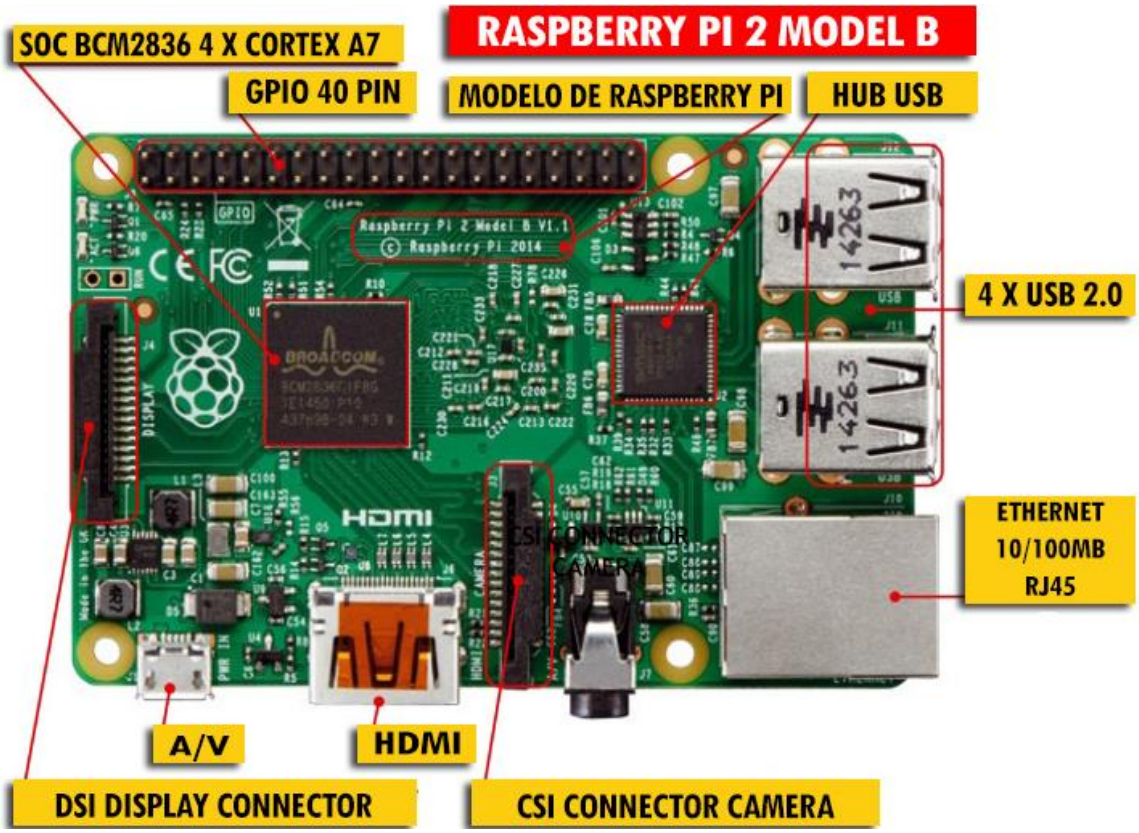


Figura 3. Vista Superior de Raspberry Pi 2 Modelo B y sus componentes

Fuente: (Raspberry Pi Foundation, s.f.)

2.3.1.2 Intel Galileo Gen2

La placa Intel Galileo Gen 2 es un sistema embebido que posee un microcontrolador con procesador de aplicaciones Intel Quark SoC X1000 y un chip (SoC) Intel Pentium de 32-bit. Esta placa fue diseñada para ser compatible con el software y hardware de un Arduino Uno, esta compatibilidad permite el desarrollo de aplicaciones con la arquitectura Intel creadas desde el sencillo entorno de desarrollo de Arduino. (Intel Corporation, s.f.)

Las principales características de hardware y software que incluye la placa Intel Galileo Gen 2 se indican a continuación:

- Ranura mini PCI Express de tamaño completo,
- Puerto Ethernet de 100 Mb
- Cabezal 3.3V USB TTL UART de 6 clavijas
- Puerto de host USB
- Puerto de cliente USB
- NOR Flash* de 8 MB.
- SRAM integrada de 512 KB
- DRAM de 256 MB
- Tarjeta microSD opcional que ofrece hasta 32 GB de almacenamiento

2.3.1.3 pcDUINO 3B.

La placa de desarrollo pcDUINO 3B es un potente sistema embebido con las características de una mini PC, soporta sistemas operativos Linux y Android y es compatible con lenguajes de programación como Python, Java, C y C++, entre los principales. Posee un procesador dual core A20, resultando ser una de las más mejores opciones en cuanto a sistemas embebidos existentes en el mercado. (SparkFun Electronics, s.f.)

Las principales especificaciones de hardware y software se presentan a continuación:

- CPU: AllWinner A20 SoC 1GHz ARM Cortex A7 Dual Core
- GPU: OpenGL ES2.0, OpenVG 1.1 Mali 400 Dual Core
- 1GB DRAM
- Almacenamiento propio: 4GB Flash, microSD card (TF) slot hasta 32GB
- Salida video HDMI con soporte HDCP

- Receptor IR
- Conector para batería Li-Poly
- Interfaz cámara MIPI
- Ubuntu 12.04 y Android ICS 4.2
- Conexión Ethernet RJ45 y módulo Wifi integrado
- Requisitos para la alimentación: 2A y 5VDC

2.3.1.4 Comparativa entre Raspberry Pi, Intel Galileo y pcDuino

Una vez analizadas las especificaciones técnicas de los tres sistemas embebidos antes mencionados, se presenta a continuación en la tabla 1 una comparativa teniendo en cuenta las principales características de cada placa.

Tabla 1. Comparativa entre Raspberry Pi, Intel Galileo y pcDUINO

Especificación	**Raspberry Pi 2 Modelo B**	**Intel Galileo Gen2**	**pcDUINO 3B**
CPU	Quad-core ARM Cortex-A7 a 900 MHz	Intel Quark X1000 32 bits 400 MHz	AllWinner A20 SoC 1GHz ARM Cortex A7 Dual Core
GPU	Open GL ES 2.0, hardware-accelerated OpenVG	No tiene	OpenGL ES2.0, OpenVG 1.1 Mali 400 Dual Core
RAM	1GB	256 MB	1GB
Almacenamiento interno	No tiene	8MB	4GB Flash

Almacenamiento externo	Slot para microSD card hasta 16GB	Slot para microSD card hasta 32 GB	Slot para microSD hasta 32GB
Salida de Video	HDMI RCA	No tiene	HDMI
Sistemas operativos soportados	Debian GNU Linux Raspbian OS Fedora Android	Linux	Linux Android
Puertos USB	4 puertos USB 2.0	3 puertos USB 2.0	USB Host USB OTG
Interfaz de red	Puerto Ethernet RJ-45	Puerto Ethernet RJ-45	Puerto Ethernet RJ-45 Módulo WiFi integrado
Alimentación	5V	De 7 a 12 V	5V

Fuente: Autoría

2.3.2 Open Software

Open Source Software es software cuyo código está disponible para modificación o mejora por parte de cualquier usuario. El concepto de software libre respeta la libertad de los usuarios y la comunidad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software (GNU & Free Software Foundation, s.f.).

2.3.2.1 Raspbian.

Sistema operativo libre desarrollado específicamente para las placas Raspberry Pi, es una versión modificada de Debian. Raspbian se caracteriza por ser más que un sistema operativo puro debido a que incluye 35000 paquetes pre compilados para ser instalados en las placas Raspberry. La construcción inicial de 35000 paquetes se completó en junio de 2012, hasta la actualidad se siguen desarrollando nuevos paquetes para mejorar la estabilidad y el manejo de las placas (Raspbian.org, s.f.).

2.3.2.2 Python.

Python es un lenguaje de programación orientado a objetos, interpretado o de script, es decir que no requiere compilar el código fuente para poder ejecutarlo proporcionando rapidez el momento del desarrollo, además es un lenguaje independiente de las plataformas teniendo versiones libres para Linux, Windows y Mac OS. Cuenta con numerosas librerías y funciones para el tratamiento de archivos, strings, números, etc, las cuales vienen incorporadas en el propio lenguaje (Python Software Foundation, s.f.).

2.3.2.3 Java.

Java es un lenguaje de programación orientado a objetos, inicialmente fue desarrollado por la compañía Sun Microsystems en el año de 1995. El principal objetivo de Java fue que los desarrolladores de software puedan crear programas capaces de ser ejecutados desde cualquier dispositivo sin necesidad de ser recompilados en otra plataforma. Actualmente los principales usos de Java son las aplicaciones web de cliente-servidor. La sintaxis empleada en

Java está basada principalmente en el lenguaje C++ combinada con programación estructurada y orientada a objetos. Posee numerosas librerías siendo compatible con otras herramientas de software para aplicaciones de bases de datos, electrónica, visión artificial, IoT, entre otras. (Oracle Corporation, s.f.)

2.3.2.4 OpenCV.

OpenCV (Open Source Computer Vision) es una biblioteca de distribución libre creada por Intel para el desarrollo de sistemas de visión artificial con un enfoque especial en aplicaciones en tiempo real, cuenta con interfaces de C++, C, Python y Java, es multiplataforma siendo compatible con Windows, Linux, Mac OS y Android. Integra alrededor de 500 funciones para el reconocimiento y posición de un objeto, reconocimiento de gestos, calibración de cámaras y distintos tipos de visión. Proporciona manejo de datos estáticos y dinámicos (Deepthi & Sankaraiah, 2011).

La principal ventaja de OpenCV y la razón por la que es utilizado en la mayoría de aplicaciones de visión artificial es por su fuerte enfoque en tiempo real, además de su optimización en lenguaje C que brinda mejoras a los procesadores de múltiples núcleos (Marengoni & Stringhini, 2011).

2.3.2.5 SimpleCV.

SimpleCV es un entorno de desarrollo de código abierto diseñado para la creación de aplicaciones de visión artificial. Permite el acceso a librerías de visión por computador de alta potencia como lo son las librerías de OpenCV. Su objetivo es brindar un entorno que

permita a nuevos desarrolladores hacer uso de estas herramientas sin la necesidad de conocer a fondo formatos de archivo, espacios de color, almacenamiento de valores, bitmaps y otros conceptos empleados en el tratamiento y procesamiento de imágenes. Ha sido desarrollado por la compañía Sight Machine y al compartir el concepto de Open Source se puede encontrar en la web numerosos ejemplos y aplicaciones para iniciar en el desarrollo de la visión artificial. (Sight Machine, Inc, s.f.)

2.4 Conceptos Estadísticos: Estimadores

2.4.1 Estimadores.

Para comprender el concepto de un estimador primero se debe conocer el principio de los estadísticos muestrales, que son valores que actúan como referencia empírica de los valores desconocidos de toda una población. Debido a la función que desempeñan los estadísticos muestrales se han denominado como "estimadores". Se debe considerar que para cada parámetro poblacional existirá un estadístico muestral que es su mejor estimador. (Vivanco, 2005)

> Vivanco (2005, pág. 31) afirma que *"Un estimador es un suceso aleatorio que asume diversos valores con probabilidades distintas, un estimador puede variar en torno al parámetro poblacional"*

Un ejemplo breve para entender el concepto de estimador puede presentarse cuando se desea conocer el costo promedio de determinado artículo del mercado, entendemos que este valor es un parámetro desconocido, por lo tanto, para encontrar el valor se deben recoger

muestras de los costos del artículo en distintos puntos de venta, correspondientes a los dispuestos para la muestra. Una vez que se obtienen los valores, el cálculo de la media aritmética obtenida de los puede emplearse como un estimador del costo promedio requerido. El parámetro que en un inicio fue desconocido ahora ha tomado un valor estimado.

2.4.1.1 Propiedades de los estimadores.

Para determinar si un estimador es pertinente con respecto a un parámetro poblacional debe cumplir con ciertas propiedades, es decir para que se considere como un buen estimador debe ser: insesgado, consistente, eficiente, suficiente y resistente. A continuación, en la tabla 2 se indica las propiedades que deben cumplir los estimadores:

Tabla 2. *Propiedades de los Estimadores*

PROPIEDAD	DESCRIPCIÓN
Insesgado	Un estimador será insesgado cuando el promedio de la distribución formada por los estadísticos muestrales tienen el mismo valor que el valor de la población
Consistente	Se considera estimador consistente siempre y cuando el valor de la estimación se asemeje con el valor del parámetro, cuando el tamaño de la muestra se aproxima al tamaño de la población.
Eficiente	Será un estimador eficiente cuando sus estimaciones se aproximen más al parámetro poblacional que las de cualquier otro estimador
Suficiente	Se obtiene un estimador suficiente cuando utiliza la información muestral ampliamente, con lo que ningún otro estimador

	ofrecerá una mejor estimación del parámetro poblacional.
Resistente	Se trata de estimadores que no se afectan por valores anómalos, es decir no sufre cambios por valores distintos a los valores típicos de la muestra.

Fuente: Autoría

CAPÍTULO III

Desarrollo Experimental

El capítulo de Desarrollo Experimental presenta el Análisis de la Situación Actual incluyendo la tabulación de resultados de las encuestas aplicadas a estudiantes de CIERCOM de cuarto a noveno semestre de la Universidad Técnica del Norte y explica cómo estas encuestas ayudaron a resolver las preguntas planteadas para diseñar el sistema de monitoreo facial. A continuación, se realiza una descripción del sistema señalando sus beneficiarios y posibles limitaciones, también se presenta el análisis de requerimientos de arquitectura, de funciones y de stakeholders.

En el apartado sobre Diseño del Sistema se muestra el diagrama de bloques y diagrama de flujo diseñados para el desarrollo del proyecto. Se explicará el funcionamiento de los algoritmos de detección de rostro y detección de ojos y los scripts de Python y OpenCV implementados para el sistema.

3.1 Análisis de la Situación Actual

Para el análisis de la situación actual se buscó recopilar información que permita responder las preguntas referentes al diseño del sistema y de esta manera sustentar el desarrollo del proyecto, esta información se obtuvo de los beneficiarios directos que serían los estudiantes de la Carrera de Ingeniería en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte. A continuación, en la tabla 3 se explica más detalladamente esta información.

Tabla 3 *Método y formato para levantamiento de información de la situación actual*

SITUACIÓN ACTUAL	
Método:	Para el desarrollo de esta tesis se plantea un método de la investigación descriptiva para recolectar información, este método corresponde a la encuesta. Se ha seleccionado la encuesta debido a que se requiere conocer las ideas, opiniones y datos reales acerca de las posibles causas de desconcentración en los estudiantes antes mencionados. La encuesta se aplicará a una muestra de estudiantes de CIERCOM tanto hombres como mujeres de indistintas edades de la Universidad Técnica del Norte, que se encuentren matriculados de cuarto semestre en adelante, lo que se busca conocer principalmente es el número de horas clase que recibe cada estudiante y el número de horas de descanso diarias, y cual consideran ellos es la principal causa de desconcentración. El cálculo de la muestra seleccionada para aplicar la encuesta se muestra más adelante en el apartado de Identificación de la Población.
Formato:	La encuesta según su objetivo es de tipo analítica, ya que busca llegar a conocer las verdaderas razones por la que los estudiantes pierden la concentración durante sus horas clase. El tipo de preguntas planteadas fueron tanto de respuestas cerradas como de respuestas abiertas. Las preguntas de respuestas cerradas fueron diseñadas para facilitar la cuantificación y tabulación de resultados, estas fueron referidas al semestre, número de créditos y razones de la desconcentración dando al encuestado un número limitado de respuestas entre las cuales puede escoger dependiendo de su caso personal. Las

preguntas de respuestas abiertas fueron planteadas en cuanto a las horas de estudio y horas de descanso, la razón de escoger este tipo de preguntas es para que el encuestado tenga la libertad de responder de acuerdo a sus hábitos diarios permitiendo obtener respuestas más profundas que en un inicio no estaban consideradas dentro de los formularios de tabulación.

El formato de la encuesta aplicada se muestra en el ANEXO 1 de este trabajo.

Fuente: Autoría

3.1.1 Identificación de la población.

Para determinar la población a la cual se va a aplicar la encuesta se utiliza la ecuación 3 la cual corresponde a la fórmula propuesta por Fisher y Navarro (1994)

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

Ecuación 3. Cálculo de la muestra, propuesto por Fisher y Navarro

Fuente: (Fisher & Navarro, 1994)

Donde:

n= tamaño de la muestra

N= tamaño de la población

K^2= nivel de confianza

e = error porcentual

p= probabilidad a favor

q= probabilidad en contra.

Para este caso de estudio tenemos un tamaño de la población de 256 estudiantes matriculados en el periodo académico Abril – Agosto 2016 entre los niveles cuarto a noveno, con este tamaño de la población se aplica la ecuación 3. Los parámetros considerados para la fórmula de Fisher y Navarro son los siguientes: el valor de N corresponde a los 256 estudiantes, para el nivel de confianza *k* se ha tomado un valor del 95,5% (k=2) esperando que únicamente el 4,5% de los resultados obtenidos no correspondan a información verídica; tanto para el valor de la probabilidad a favor *p* como el valor de la probabilidad en contra *q* se tomó el valor de 0,5 considerando obtener datos reales de los encuestrados, el error porcentual *e* con un valor de e=0,10 teniendo un porcentaje de respuestas correctas del ±10% de los resultados totales obtenidos. El cálculo de la muestra aplicando la ecuación 3 se presenta a continuación:

$$n = \frac{256 * 2^2 * 0.5 * 0.5}{(0{,}1^2 * (256-1)) + 2^2 * 0.5 * 0.5}$$

$$n = 72$$

Con el resultado de la muestra que se obtuvo se procede a aplicar la encuesta a un número de 72 estudiantes para posteriormente analizar los resultados.

3.1.2 Análisis de Resultados.

Una vez aplicada la encuesta a 72 estudiantes de la Carrera de Ingeniería en Electrónica y Redes de Comunicación, se obtuvo resultados que fundamentan el desarrollo de este proyecto, a continuación se indican las conclusiones derivadas de la encuesta.

Las preguntas realizadas permitieron comprobar que el número de créditos de un estudiante es directamente proporcional al número de horas dedicadas a estudiar, mientras

más créditos mayor número de horas dedicadas al estudio, y a su vez es inversamente proporcional al número de horas dedicadas a descansar, teniendo un elevado número de estudiantes que descansan un tiempo muy reducido respecto a las horas recomendadas para tener una buena salud. En la figura 4 se puede ver la relación directamente proporcional e inversamente proporcional entre las variables antes mencionadas. Las gráficas correspondientes a la tabulación de resultados de las encuestas aplicadas se encuentran disponibles en el ANEXO 2 de este trabajo.

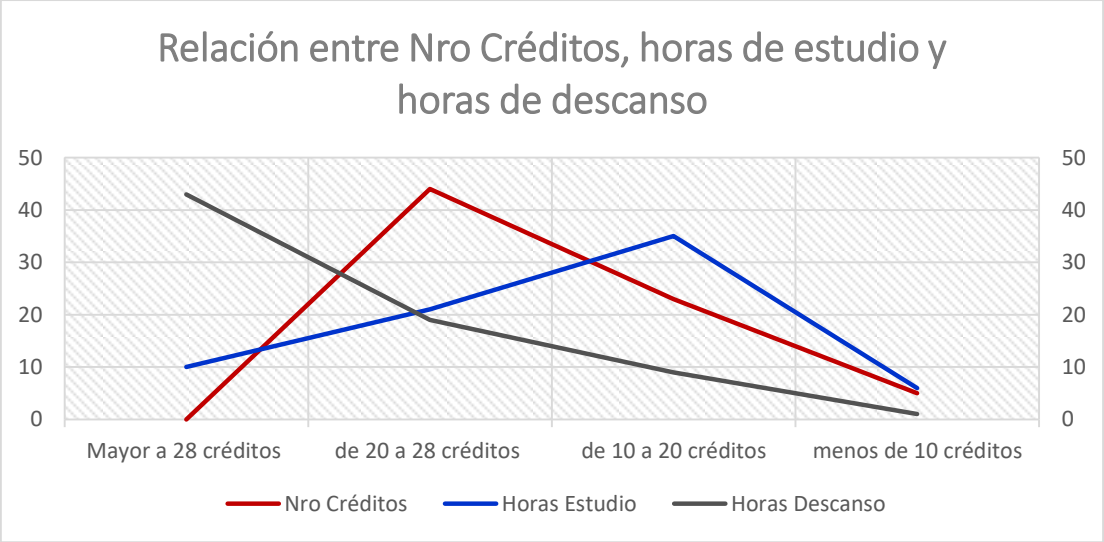


Figura 4. Relación entre las preguntas aplicadas en la encuesta

Fuente: Autoría

Para conocer más a fondo la opinión de los estudiantes se solicitó que señalen la razón que consideran que más influencia tiene en la falta de concentración durante sus periodos académicos, se obtuvo que más del 50% de encuestados consideran que la principal causa es la falta de horas de descanso. Este resultado está totalmente respaldado por varios estudios e investigaciones científicas que muestran la relación entre la calidad y cantidad de horas de sueño y el desempeño de una persona en el ámbito laboral y social.

El estudio de Miró *et al.* (2002) cita varios trabajos realizados respecto a los patrones de sueño e indican que respecto a la inteligencia las personas con patrones de sueño largos obtienen mayores puntuaciones en los test de inteligencia fluida y pensamiento divergente, además existen mejores resultados en cuanto a la consolidación de la memoria y aprendizaje.

Con la información recolectada y las bases teóricas ya mencionadas se puede definir que la determinación de estimadores de desconcentración, que es el objetivo de este trabajo de titulación, puede realizarse basándose en características fisiológicas relacionadas a la falta de horas de descanso, las principales señales de cansancio que pueden presentar los estudiantes son bostezos repetidos y reducido número de pestañeos, estos factores pueden ser analizados mediante el uso de software de visión artificial y sus diversas librerías destinadas a la detección de objetos e identificación de patrones.

Vandna Saini & Rekha Siani (2014) presentan un sistema de detección de somnolencia de conductores al manejar, basado en la medición de ciertos comportamientos, estas conductas incluyen bostezos, parpadeos e incluso posición de la cabeza. Una vez analizados los trabajos de varios autores, las bases teóricas y los resultados de las encuestas aplicadas a los estudiantes se determina que los parámetros a considerar para este trabajo serán el pestañeo y el bostezo como muestras del cansancio o fatiga en los estudiantes.

A pesar de que existen varios comportamientos y características fisiológicas propias de un estado de somnolencia, se determinó que los parámetros de número de bostezos y número de parpadeos son los más adecuados para ser monitoreados debido a que no requieren de hardware adicional, únicamente con una cámara se pueden adquirir los datos relacionados a

estas variables. Existen sistemas que emplean el pulso, el movimiento de la cabeza e incluso la temperatura corporal para detectar somnolencia (Vandna Saini & Rekha Siani, 2014) (Häkkänen, Summala, Partinen, Tiihonen, & Silvo, 1999), sin embargo no es factible para el monitoreo a los estudiantes debido a que estas mediciones requieren de sensores adicionales que impedirían el desarrollo normal de las actividades del estudiante dentro del aula de clases.

Finalmente, se concluye que en el grupo de estudiantes encuestados se observó una alta tendencia que coinciden que los factores de desconcentración están relacionados a la fatiga y cansancio lo cual respalda el desarrollo del sistema de monitoreo facial propuesto para este trabajo.

3.2 Introducción al desarrollo del Proyecto

A continuación, se presentan los apartados necesarios para aclarar el propósito, ámbito, beneficiarios y objetivos del sistema, para poder mejorar el desarrollo del diseño experimental del sistema de monitoreo facial.

3.2.1 Propósito del Sistema.

El objetivo fundamental de este proyecto es brindar estimadores de desconcentración de los estudiantes dentro del aula de clases, con la ayuda de la visión artificial y el procesamiento de imágenes, para esta finalidad se pretende monitorear a los alumnos de la Carrera de Electrónica y Redes de Comunicación de la UTN, para conocer cuando han perdido la concentración en sus horas clase, estos resultados estarán disponibles en la Internet

para que los docentes puedan acceder a ellos y conocer si durante sus periodos académicos sus alumnos estuvieron desconcentrados.

3.2.2 Ámbito del Sistema.

El sistema de monitoreo facial que brinda estimadores de desconcentración de los estudiantes dentro del aula de clases con aplicación IoT, es un sistema a escala de laboratorio, lo que indica que sus pruebas servirán para futuras implementaciones dentro de los planteles educativos del país.

Para comprender de mejor manera el concepto de escala de laboratorio, se cita el trabajo "Principios Básicos de Escalado" presentado por Gónzalez Castellanos (2000) en donde se encontró la siguiente información: En general, un laboratorio constituye un espacio primario de investigación en donde se plantean las metodologías propicias para confirmar o rechazar una hipótesis y formular o comprobar un modelo matemático. Al indicar que un proyecto será desarrollado a escala de laboratorio se habla de que se desarrollará a una escala menor que la industrial, donde no se requiere que se siga todas las etapas convencionales de un proceso de I+D. *"No existe duda alguna que es técnicamente posible transferir casi cualquier proceso desarrollado a nivel de laboratorio, directamente a la producción industrial a gran escala (...)"* (González Castellanos, 2000),

Por las razones ya citadas se confirma que el sistema de monitoreo facial, se desarrollará a una escala de laboratorio dejando los mejores precedentes de diseño y resultados para que en un futuro puedan ser implementados a nivel industrial.

3.2.3 Beneficiarios.

Se consideran como beneficiarios iniciales a los alumnos y docentes de la carrera de Electrónica y Redes de Comunicación de la UTN. Como beneficiarios directos tenemos a todos los estudiantes de las distintas carreras de la Facultad de Ingeniería en Ciencias Aplicadas, debido a que el sistema será desarrollado para cumplir su funcionamiento dentro de las aulas del edificio de la FICA las mismas que son compartidas por todos los estudiantes ya mencionados, permitiendo la escalabilidad del proyecto. Los resultados obtenidos permitirán analizar el desempeño tanto de estudiantes como de docentes dentro del aula de clases y su participación activa en el proceso de aprendizaje.

3.2.4 Objetivos del Sistema.

Los objetivos que se detallan a continuación son el resultado de un amplio análisis de las bases teóricas y técnicas, levantamiento de información e investigación dedicada a la visión artificial e IoT, todo este proceso ha permitido definir criterios de diseño para el sistema y a su vez se han permitido determinar los propósitos principales con los que debe cumplir el sistema de monitoreo:

- Monitorear a los estudiantes dentro del aula de clases en distintos horarios para determinar si existe desconcentración.
- Emplear los algoritmos de visión artificial existentes de manera que con el uso de sus librerías y el procesamiento de imágenes permitan determinar si un estudiante está desconcentrado.
- Analizar los datos obtenidos del monitoreo para brindar los estimadores de desconcentración.

- Implementar la comunicación entre el sistema desarrollado y una plataforma en la Internet donde se puedan visualizar los estimadores de desconcentración.

3.3 Descripción General del Proyecto

Para describir de forma más clara el proyecto se presenta a continuación la perspectiva, funciones e interfaces a ser empleadas en el proyecto. En la sección de funciones del proyecto se propone analizar el número de pestañeos y bostezos que un estudiante realiza por minuto, considerando estos aspectos fisiológicos como indicadores de cansancio lo cual a su vez nos lleva a determinar la desconcentración del estudiante.

Para el monitoreo del número de pestañeo se propone desarrollar un contador para conocer cuántas veces por minuto ha pestañeado un estudiante, para ello se ha revisadon varios trabajos que permitan conocer los valores normales de este movimiento, de manera especial se analizó el trabajo de Häkkänen *et al*. (1999) el cual brinda datos específicos sobre tiempos y frecuencia de pestañeo.

De igual manera para el monitoreo de bostezos se propone desarrollar un contador que tome datos del estudiante minuto a minuto, para ello se toma como referencia un estudio realizado en la Facultad de Ciencias Médicas en la Universidad de Buenos Aires que explica que el bostezo es un reflejo normal desencadenado por la fatiga y somnolencia (entre otras causas) los valores presentados en este trabajo se tomarán como referencia para el desarrollo del contador. (Muchnik, Finkielman, Seemeniuk, & de Aguirre, s.f.)

A continuación, se presenta la tabla 4 correspondiente a la descripción del proyecto y los subtemas antes indicados:

Tabla 4. Descripción General del proyecto

DESCRIPCIÓN GENERAL	
Perspectiva del sistema:	El sistema busca emplear la visión artificial destinada a la extracción de características visuales que típicamente muestran señales de fatiga, con lo cual se puede determinar estimadores de desconcentración mediante el uso de un sistema embebido, una cámara y las herramientas de software libre para procesamiento de imágenes y programación.
Funciones del proyecto	- Activar la cámara mediante un script. - Detectar el rostro y enmarcarlo para determinar que se inicia el monitoreo. - Detectar los ojos y enmarcarlos. - Detectar el pestañeo e iniciar un contador para conocer cuántas veces por minuto se ha realizado este movimiento. - Detectar la boca y detectar cuando el estudiante bostece. - Comparar los datos del contador y de la detección de bostezo y determinar si se encuentra dentro de los valores normales. (Häkkänen, Summala, Partinen, Tiihonen, & Silvo, 1999), (Muchnik, Finkielman, Seemeniuk, & de Aguirre, s.f.) - Determinar estadísticamente los estimadores de desconcentración. - Subir los valores de los estimadores a una plataforma en la Internet para que pueda ser visualizada por los usuarios, inicialmente estará disponible únicamente para los docentes.

Interfaces:	
Estará disponible para el usuario la interfaz proporcionada por la plataforma donde se ejecute la aplicación IoT, es decir donde se almacenan los datos de los estimadores de desconcentración para que puedan ser analizados por los docentes.	
Software:	**Hardware:**
Plataforma Libre Ubidots	Puerto Ethernet Cable de Red
Tipos de información:	
La información será almacenada en un archivo .csv, para su análisis estadístico y posteriormente su almacenamiento en la aplicación IoT	
Software:	**Hardware:**
Plataforma Libre Ubidots	No requiere hardware adicional
Tipos de tecnología:	
Se emplea tecnología de tipo Open Source.	

Fuente: Autoría

3.4 Análisis de las Características del Usuario

Los usuarios del sistema podrían dividirse en dos grupos, por una parte, son usuarios los estudiantes a ser monitoreados y por otra parte son usuarios las personas que van a poder acceder a la información sobre los estimadores de desconcentración en la plataforma IoT. A continuación, se explica cada tipo de usuario del sistema:

- Los estudiantes como usuarios, resultan ser la parte fundamental del proyecto debido a que serán los sujetos de prueba de donde se recolectará la información.

- Los docentes como usuarios, podrán acceder a la aplicación IoT creada en la plataforma en Internet, dentro de esta plataforma podrán visualizar de forma clara los estimadores obtenidos del monitoreo.

Además de las características de los usuarios es de interés las limitaciones del sistema relacionadas a los usuarios, previo el diseño del sistema se analizaron las siguientes limitaciones:

- El sistema de monitoreo presenta una limitación en cuanto a la ubicación correcta de la cámara, ésta debe estar ubicada frontalmente y a una altura que pueda adaptarse a las diferentes estaturas de los estudiantes y las diversas formas de sentarse. Esta limitación debe ser superada con los criterios adecuados de diseño para que el usuario no interfiera con la cámara del sistema impidiendo que el monitoreo se realice con éxito.
- El sistema estará expuesto a la vista de los estudiantes, por lo que se requiere un manejo adecuado del hardware asegurando que no se dañe ninguno de los elementos de hardware.
- El manejo de plataformas de IoT no resulta muy sencillo para los usuarios, para superar esta limitación se debe crear una plataforma amigable y dejar los respectivos manuales de uso para evitar posibles inconvenientes.

Las características y limitaciones de los usuarios antes mencionadas serán tomadas en cuenta el momento del diseño e implementación del sistema para mejorar el desarrollo del proyecto y evitar posibles errores en la fase de pruebas.

3.5 Requerimientos del Sistema

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

El estándar define la construcción de un buen requisito que proporcione atributos y características teniendo en cuenta la aplicación reiterativa a lo largo del ciclo de vida del sistema. El ISO / IEC / IEEE 29148: 2011 guarda una estrecha relación con anteriores normas destinadas al proceso de aplicación de requerimientos, como son la norma ISO / IEC 12207: 2008 e ISO / IEC 15288: 2008.

Las tablas que se muestran a continuación se han diseñado teniendo en cuenta las consideraciones del estándar antes mencionado, éstas contienen los requerimientos iniciales del sistema, requerimientos de arquitectura y los requerimientos de stakeholders, el objetivo es presentar de una manera más clara dicha información la cual permitirá realizar la selección de software, hardware y algunos aspectos específicos para el diseño del script de monitoreo.

El diseño propuesto para cada tabla incluye una columna donde se identifica el número de requerimiento, una columna destinada a la descripción detallada del requerimiento, la siguiente columna está destinada a indicar la prioridad del requerimiento y a su vez se subdivide en Alta, Media y Baja, esta valoración es de suma importancia para la selección del

software y hardware, se incluye también una columna de relación y de verificación que se utilizarán en el caso de que un requerimiento sea totalmente dependiente de otro.

La tabla 5 presentada a continuación contiene los requerimientos iniciales del sistema, aquí se definen los límites funcionales del sistema, se describen los requerimientos de uso, de interfaces, de modos y estados y requerimientos físicos. La tabla 5 emplea la abreviatura SySR la cual nos permite identificar a qué número de requerimiento estamos haciendo referencia y además lo identifica como requerimiento propio de la tabla de *Requerimientos Funcionales del Sistema.*

Tabla 5. *Requerimientos de funciones del sistema*

SySR						
REQUERIMIENTO DE FUNCIONES						
#	**REQUERIMIENTO**	**PRIORIDAD**			**RELACIÓN**	**VERIFICACIÓN**
		Alta	**Media**	**Baja**		
SySR 1	*El sistema deberá permanecer en un lugar donde no esté expuesto a altas temperaturas ni humedad*	☒	☐	☐		
REQUERIMIENTO DE USO						
SySR 2	*El sistema deberá estar conectado a la corriente eléctrica todo el tiempo para que pueda funcionar, también debe tener conexión a internet para que pueda subir los datos a la Internet*	☒	☐	☐		
REQUERIMIENTOS DE INTERFACES						
SySR 3	*Se requiere ingresar a la plataforma en la web para visualizar los datos de los estimadores*	☒	☐	☐		
SySR 4	*La visualización debe ser clara con gráficas que permitan la correcta interpretación de los datos*	☒	☐	☐		
REQUERIMIENTO DE MODOS/ESTADOS						
SySR 5	*El sistema debe permanecer activo durante las horas clase*	☒	☐	☐		
REQUERIMIENTOS FÍSICOS						
SySR 6	*El sistema debe estar correctamente ubicado en un lugar que no interfiera con las actividades de los estudiantes en sus horas clase*	☒	☐	☐		

Continuando con el análisis de requerimientos, se presenta la tabla 6 o tabla de *Requerimientos de Arquitectura* aquí se describen los requerimientos de hardware, software y requerimientos eléctricos del sistema. La abreviatura que identifica a los requerimientos de esta tabla es SRSH y tienen una importancia adicional debido a que serán los empleados el momento de la selección de software y hardware que se presenta más adelante en este trabajo por lo que es necesario ser analizados detenidamente en el proceso de diseño.

Tabla 6. Requerimientos de Arquitectura

SRSH						
REQUERIMIENTO DE ARQUITECTURA						
#	**REQUERIMIENTO**	**PRIORIDAD**			**RELACIÓN**	**VERIFICACIÓN**
		Alta	**Media**	**Baja**		
REQUERIMIENTOS DE SOFTWARE						
SRSH 1	*Se requiere un sistema operativo que sea de distribución libre y compatible con el sistema embebido*	☒	☐	☐		
SRSH 2	*Se requiere software para el tratamiento de imágenes que permita procesamiento en tiempo real y disponga clasificadores que permitan la detección de ojos y boca*	☒	☐	☐		
SRSH 3	*Se requiere software de programación que sea compatible con el software de tratamiento de imágenes*	☒	☐	☐		
SRSH 4	*Se requiere software que permita ejecutar un script desde un sistema embebido en tiempo real*					
REQUERIMIENTOS DE HARDWARE						
SRSH 5	*Es necesario un sistema embebido que tenga entradas para conectar una cámara de alta resolución*	☒	☐	☐		
SRSH 6	*El sistema embebido requiere un procesador que soporte el tratamiento de imágenes en tiempo real*	☒	☐	☐		
SRSH 7	*Se requiere un sistema embebido que tenga un puerto Ethernet para poder transmitir los datos al servidor*	☒	☐	☐		
SRSH 8	*Se debe considerar el tamaño de la cámara antes de su selección*	☐	☒	☐		
SRSH 9	*Se requiere una cámara que posea una entrada USB o sea compatible con la placa del sistema embebido y el software de tratamiento de imágenes*	☒	☐	☐		
REQUERIMIENTOS ELÉCTRICOS						
SRSH 10	*El sistema no debe usar una batería debido a su uso prolongado deberá estar conectado a la red eléctrica*	☒	☐	☐		

Fuente: Autoría

Para finalizar la descripción de requerimientos se presenta tabla 7, esta tabla corresponde a los requerimientos de Stakeholders, recuérdese que un Stakeholder es todo grupo o individuo que tiene un interés directo en el resultado obtenido por el desarrollo del proyecto. Se especifican también los requerimientos funcionales y requerimientos de usuarios. La abreviatura empleada en esta tabla corresponde a StSR la que permitirá identificar de manera específica a cada requerimiento.

Tabla 7. Requerimientos de Stakeholders

StSR						
REQUERIMIENTOS DE STAKEHOLDERS						
#	**REQUERIMIENTO**	**PRIORIDAD**			**RELACIÓN**	**VERIFICACIÓN**
		Alta	**Media**	**Baja**		
StRS 1	*La posición del estudiante debe ser frente a la cámara para poder detectar el rostro*	☒	☐	☐		
StRS 2	*La altura a la cual se ubique la cámara debe basarse en la estatura promedio de los estudiantes de CIERCOM de la Universidad Técnica del Norte*	☒	☐	☐		
REQUERIMIENTOS OPERACIONALES						
StRS 3	*Se requiere transmitir los datos a una plataforma en la web por lo cual necesariamente debe estar conectado a internet*	☒	☐	☐		
StRS 4	*Se requiere de una cámara y un sistema embebido para monitorear a un solo estudiante*	☒	☐	☐		
REQUERIMIENTOS DE USUARIOS						
StRS 5	*El usuario no debe interferir ni mover la cámara de la posición definida por el administrador del sistema*	☒	☐	☐		
StRS 6	*Los usuarios que deseen conocer los datos de desconcentración serán únicamente los permitidos por el administrador del sistema*	☒	☐	☐		
StRS 7	*Los resultados obtenidos de los estimadores de desconcentración serán tratados con confidencialidad para posteriormente tomar las medidas respectivas en busca de mejorar el proceso de aprendizaje*	☒	☐	☐		

Fuente: Autoría

3.6 Selección de Hardware y Software del sistema

Una vez definidos los requerimientos del sistema, se procede a seleccionar el hardware y software necesarios para el sistema. Para lo cual se realizará una valoración a cada una de las posibles opciones tanto en sistemas embebidos y cámaras de alta resolución como en software de programación y de procesamiento de imágenes, las opciones con una mayor valoración serán las utilizadas para la implementación del sistema.

3.6.1 Selección de Software

En primer lugar, se dará inicio a la selección del software debido a que las opciones de software que permiten realizar la detección de objetos y el procesamiento de imágenes son limitadas, a partir del software se buscará un hardware que sea compatible y que esté disponible en el mercado, se analizarán únicamente opciones Open Source.

El formato de la tabla 8 presentada a continuación indica en la primera columna las opciones de software existentes en el mercado, la siguiente columna contiene los requerimientos que deben ser analizados y se indica con la abreviatura correspondiente a que tabla anterior hace referencia, finalmente tenemos la columna de valoración total donde se indica el puntaje obtenido por cada una de las opciones de software. Para la valoración se ha definido de la siguiente manera: se califica con un valor de 1 si cumple el requerimiento y un valor de 0 si no cumple el requerimiento indicado. La tabla 8 posee en la parte inferior una sección en la que se explicará la elección realizada y su respectiva justificación.

En el caso del software de procesamiento de imágenes se han encontrado dos opciones que se adaptan a los requerimientos antes señalados y estas serán analizadas a continuación en la tabla 8:

Tabla 8. Selección Software de Tratamiento de Imágenes

SOFTWARE TRATAMIENTO DE IMÁGENES	**REQUERIMIENTOS** (ver tabla 5)		**VALORACIÓN TOTAL**
	SRSH 1	**SRSH 2**	
OpenCV	1	1	2
SimpleCV	1	0	1

Cumple

1 - Cumple

0 - No cumple

Elección:

A pesar de que el software SimpleCV es de distribución libre y puede ser compatible con un sistema embebido, no se conoce con exactitud que disponga de clasificadores previamente diseñados para la detección ojos y boca, mientras que el software OpenCV dispone de más de 500 funciones para la detección de objetos y específicamente dispone de clasificadores de ojos y boca, tanto en imágenes o video streaming. Los clasificadores de tipo Haar con los que cuenta OpenCV han sido implementados en diversos sistemas de detección, reconocimiento y seguimiento objetos, mejorando el tiempo de respuesta de los sistemas y reduciendo el margen de error de los mismos. Por las razones antes indicados se selecciona OpenCV para el desarrollo del sistema de monitoreo facial.

Fuente: Autoría

Para la selección del software de programación se encontraron disponibles dos opciones que son compatibles con OpenCV (software seleccionado en la tabla 8) y que además son de tipo Open Source, estas opciones serán analizadas en la tabla 9. El formato de la tabla es igual al explicado en la sección anterior.

Tabla 9. *Selección Software de Programación*

SOFTWARE PROGRAMACIÓN	**REQUERIMIENTOS** (ver tabla 5)		**VALORACIÓN TOTAL**
	SRSH 3	**SRSH 4**	
Python	1	1	2
Java	1	0	1

Cumple

1 - Cumple

0 - No cumple

Elección:

Basados en los requerimientos de software detallados en la tabla 6 se analizaron estas dos opciones para el software de programación, obteniendo la mayor valoración Python. Este software además de cumplir con los requerimientos analizados tiene un tiempo de ejecución menor a otros lenguajes de programación debido a que es un lenguaje interpretado o de script (Python Software Foundation, s.f.), esta característica especial representa una ventaja para el sistema de monitoreo pues permitirá que el tratamiento de imágenes se efectué en menor tiempo que al emplear otro tipo de software.

Fuente: Autoría

3.6.2 Selección de Hardware

Para conocer cuál es la mejor opción de hardware se deben tener en cuenta dos aspectos importantes, en primer lugar, la decisión estará basada en la selección de software establecida en el apartado anterior y en segundo lugar estará basada en los requerimientos ya analizados en la tabla 6. En la parte de hardware se seleccionará un sistema embebido, a partir de esta elección se buscará una cámara que sea compatible, estos dos dispositivos serán los encargados de captar las imágenes y procesarlas, respectivamente.

Durante la elección del sistema embebido se analizó varias opciones disponibles en el mercado, de éstas se escogieron 3 que son las que más se adaptan a las necesidades del sistema. El formato de la tabla para la selección del sistema embebido es similar al detallado en la selección del software, la única diferencia es que para este caso se tiene en cuenta tres requerimientos y por lo tanto la valoración será mayor. Al final de la tabla se indica cual ha sido el sistema embebido seleccionado y los argumentos que respaldan esta decisión.

En la tabla 10 se realiza la valoración de cada uno de los sistemas embebidos disponibles tomando en cuenta que cumpla con los requerimientos de hardware indicados con anterioridad:

Tabla 10. Selección Sistema Embebido

HARDWARE / **SISTEMA EMBEBIDO**	**REQUERIMIENTOS** (ver tabla 5)			**VALORACIÓN TOTAL**
	SRSH 5	**SRSH 6**	**SRSH 7**	
RASPBERRY PI 2 MODELO B	1	1	1	**3**
INTEL GALILEO	0	1	1	**2**
PC DUINO	1	0	1	2

1- Cumple
0- No cumple

Elección:

Se opta por la placa de desarrollo Raspberry Pi 2 debido a que cumple con los requerimientos de arquitectura correspondientes a SRSH 5, SRSH 6 y SRSH 7, obteniendo una valoración mayor a las otras opciones existentes en el mercado. Uno de las principales características de la placa seleccionada es su procesador, el mismo que tiene capacidad para el tratamiento de imágenes requerido por el

sistema de monitoreo. A continuación, se presentan las principales características de la placa. (Raspberry Pi Foundation, s.f.)

- CPU ARM de 700 MHz
- 512 MB de SDRAM
- 10/100 Ethernet RJ45
- 4 x USB 2.0
- HDMI (1.4) y el compuesto RCA
- Acceso a configuración mediante SSH
- Ejecuta sistemas operativos basados en LINUX

Fuente: Autoría

Basados en el sistema embebido seleccionado, se buscó en el mercado las opciones de cámaras compatibles, en este caso podemos optar por cámaras USB o por las cámaras propias del Raspberry Pi 2 Modelo B. En la tabla 11 se presentan 3 opciones de cámaras disponibles en el mercado, el formato de la tabla y la valoración se realizan de manera similar a lo antes mencionado en la selección de software y hardware:

Tabla 11. Selección de Cámara para el sistema

HARDWARE CÁMARA	**REQUERIMIENTOS**		**VALORACIÓN TOTAL**
	SRSH 8	**SRSH 9**	
Canyon CNR-WCAM820	0	1	1
Creative VF0610	0	1	1
Raspberry Pi camera module v1	1	1	2

1- Cumple

0- No cumple

Elección:

Una vez analizadas las opciones existentes en el mercado respecto a cámaras se optó por el Raspberry Pi Camera Module v1 debido a su compatibilidad con la placa Raspberry Pi 2 modelo B que fue el sistema embebido seleccionado. Además, el tamaño reducido de esta cámara permitirá que el sistema tenga un diseño compacto y facilitará la implementación del mismo cumpliendo con los requerimientos especificados en la tabla 5.

Fuente: Autoría

3.7 Diseño del sistema

De la información recolectada en los apartados anteriores se hizo un análisis que permitió definir las directrices para el diseño del sistema, a continuación, se presentan los criterios tomados en cuenta para el desarrollo e implementación del sistema de monitoreo facial:

- El sistema basa su funcionamiento en la detección de aspectos fisiológicos de cansancio, la razón de este criterio se debe a que en la tabulación de encuestas se obtuvo que la principal causa de desconcentración es la falta de horas de descanso. Los aspectos relacionados al cansancio fueron citados del trabajo de (Vandna Saini & Rekha Siani, 2014).
- Al ser un objetivo de este trabajo brindar los estimadores de desconcentración se procederá al diseño de la aplicación en una plataforma para IoT que permita visualizar adecuadamente los datos.

- Dentro del diseño del sistema se debe considerar las limitaciones de los usuarios analizadas en el apartado 3.4 de este trabajo, por lo cual se realizará más adelante un análisis de la ubicación adecuada del sistema para que éste pueda adaptarse a las distintas características de los usuarios.

Como parte del diseño del sistema se muestra a continuación el diagrama de bloques y diagrama de flujo que guiará el funcionamiento y los procesos para poder desarrollar adecuadamente los scripts y la aplicación en la plataforma IoT.

3.7.1 Diagrama de Bloques del Sistema

El diagrama que se presenta a continuación es el que comprende las fases de diseño del sistema, está formado por 5 bloques los cuales a su vez están contienen varios subprocesos, se han planteado los bloques dependiendo de las funciones que cada uno debe desarrollar. En la figura 5 se puede observar cada uno de ellos.

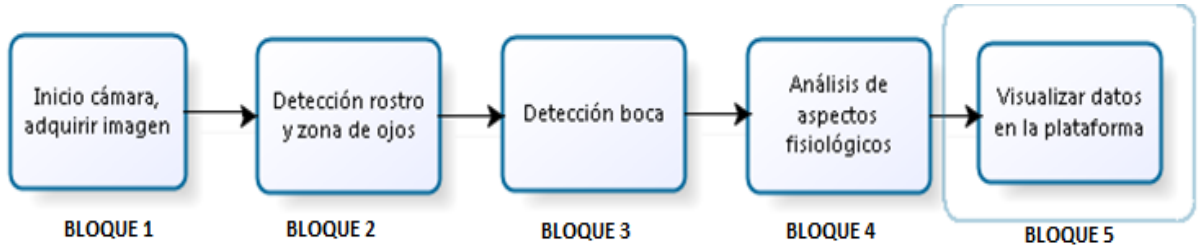


Figura 5. Diagrama de Bloques del Sistema

Fuente: Autoría

Para dar inicio con la adquisición de datos se inicia con el bloque 1, este es el encargado de que la cámara se inicie y empiece a adquirir imágenes, para ello debe desarrollarse un script en Python para arrancar la cámara. En el bloque 2 se inicia el trabajo de OpenCV, aquí

se detecta el rostro y se crea una zona de interés mejor conocida como ROI. Las regiones ROI, de sus siglas en inglés “Region of Interest”, son funciones de OpenCV que permiten enfocarnos específicamente en un lugar de la imagen para sacar de éste sus características más importantes. La ROI determinada para nuestra imagen permite enfocar la parte superior del rostro y detectar los ojos y por consiguiente el número de parpadeos. El siguiente bloque se encargará de la detección de la boca y posteriormente de contar el número de bostezos. Con los datos obtenidos en el bloque 2 y bloque 3 se realiza el proceso de análisis de aspectos fisiológicos para determinar si el sujeto monitoreado presenta rasgos de cansancio y está perdiendo la desconcentración. Finalmente, los datos del bloque 4 deben ser subidos a la plataforma IoT, esto se realiza en el bloque 5.

3.7.1.1. Bloque 1: Inicio de cámara y adquisición de la imagen.

- **Habilitación de la cámara**

Una vez conectado el módulo Pi Camera a la placa Raspberry Pi se debe tomar en cuenta que la cámara se encuentra en un estado de reposo debido a que aún no ha sido habilitada. Para poder utilizar el módulo Pi Camera v1 se requiere finalizar el estado de reposo y habilitar la cámara, procedemos a acceder desde el terminal y ejecutar el comando que se muestra a continuación.

sudo apt-get install python-picamera

Al ejecutar el comando se despliega la pantalla que se muestra en la figura 6:

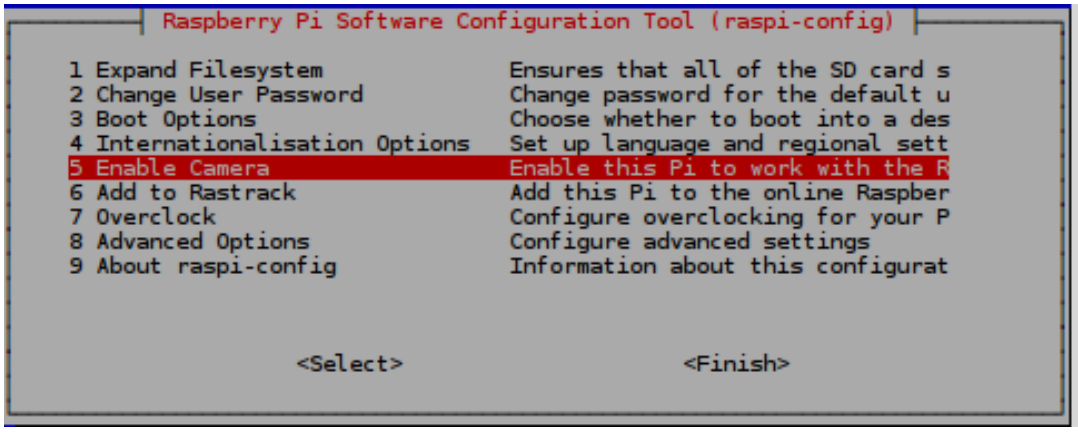


Figura 6. Habilitación del módulo Pi Camera en el Raspberry Pi

Fuente: Autoría

Damos enter en la opción Enable Camera y nos aparece una pantalla como la que se indica en la figura 7:

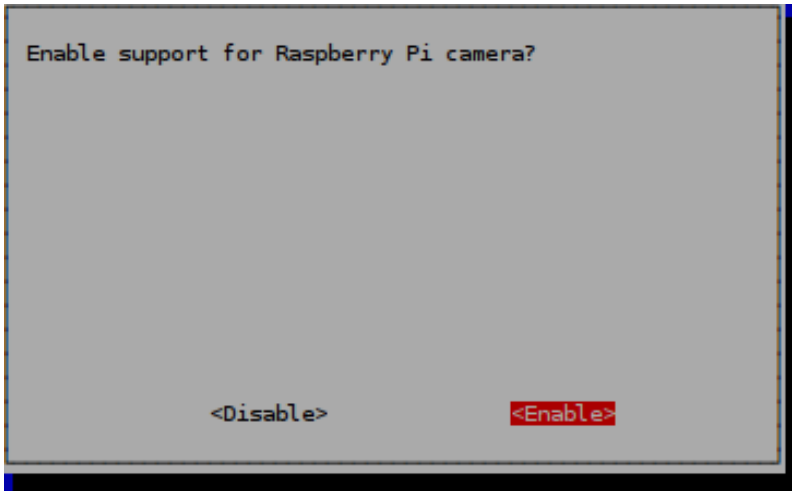


Figura 7. Pantalla de habilitación del módulo Pi Camera

Fuente: Autoría

Una vez habilitada la cámara desde el terminal del Raspberry Pi, se debe incluir en los scripts las siguientes librerías de Python: **time** y **picamera.** El uso de la librería **time** nos permitirá indicar un tiempo específico de grabación de un video, o un lapso de tiempo entre el cual se desea capturar una fotografía, por otra parte, al incluir la librería **picamera** estamos

llamando directamente al módulo de la cámara. A continuación, dentro del script se debe llamar a la librería **picamera** y especificar un nombre para la cámara con lo cual ya podemos configurar el brillo, saturación, nitidez, resolución, velocidad de obturación entre los principales. Cuando ya se ha creado un objeto con el nombre de la cámara, se puede empezar a tomar fotografías o grabar videos, el comando para crear este objeto es el siguiente:

with picamera.PiCamera() as "*ejemplo*":

En la tabla 12 se indican los principales comandos de configuración del módulo PiCamera, utilizando como nombre de la cámara "ejemplo":

Tabla 12. *Configuraciones del módulo PiCamera*

Comando de configuración	Descripción de la configuración
ejemplo.start_preview() **ejemplo.**stop_preview()	Estos comandos permiten visualizar en tiempo real la imagen capturada por el módulo PiCamera
ejemplo.capture**('nombre.jpg')**	Este comando nos permite tomar una fotografía desde la PiCamera, el parámetro que debemos configurar es el nombre con el que se desea guardar la imagen y el formato que puede ser jpg, png, bmp, raw, rgb, dependiendo de la necesidad del usuario.
ejemplo.start_recording**('video.h264')** **ejemplo.wait_recording(20)**	El comando start_recording permite grabar un video desde la cámara, se debe

ejemplo.stop_recording()	especificar el tiempo de grabación en segundos (wait_recording) y finalmente con stop_recording detenemos la filmación.
ejemplo.resolution **= (2592, 1944)**	El sensor de la cámara ofrece una resolución de 2592x1944 píxeles, sin embargo para utilizar menor espacio de almacenamiento se puede reducir la resolución según se requiera.
ejemplo.led= False	Este comando se utiliza en caso de que se requiera apagar el led para que al momento de capturar imágenes se lo realice de manera más discreta
ejemplo.shutter_speed= 300000	Si se desea controlar la velocidad de obturación lo podemos realizar con este comando, únicamente se requiere especificar el tiempo en microsegundos
ejemplo.brightness= 60	El brillo se puede ajustar asignando un valor de 0 al 100

Fuente: Autoría

- **Adquisición de imágenes**

El primer paso para el desarrollo del sistema es adquirir la imagen, se pretende que la imagen posea buena iluminación, no exista oclusión, deformaciones de ningún tipo, ni cambios de escala, si la imagen adquirida cumple con estas características de calidad el

reconocimiento de los patrones será exitoso. La imagen adquirida desde la cámara será una imagen RGB.

Para cumplir con estas características se debe tener muy en cuenta el funcionamiento de la cámara, la cual tiene como objetivo transformar señales luminosas en señales analógicas. Una cámara posee dos partes bien identificadas, por una parte, tenemos el sensor que es un chip completamente sensible a la luz, es el encargado de captar las características de una imagen en forma de señales luminosas para luego convertirlas a señales analógicas, y por otra parte tenemos la parte óptica que se encarga de proyectar los elementos a una distancia focal adecuada.

El módulo PiCamera v1 que ha sido la cámara seleccionada para este proyecto posee una resolución de imagen de 5 Megapíxeles (2592 x 1944 píxeles) y una velocidad de transferencia de imágenes de 1080p: 30fps y 780p: 60fps. El módulo de la cámara se conecta al Raspberry Pi a través del conector CSI diseñado específicamente para este periférico. Para el conocimiento de mayores especificaciones técnicas se adjunta el Datasheet de la cámara en la sección de ANEXOS de este trabajo. (ver Anexo 3)

- **Pre procesamiento de imágenes**

Una vez realizado el proceso de adquisición de la imagen RGB se procede a una etapa de Pre-procesamiento, lo que se busca en esta etapa es eliminar el ruido de la imagen original generado por cambios de iluminación o degradación afectando a la calidad de la misma. Con el Pre-procesamiento también es posible eliminar ciertos objetos dentro de nuestra imagen que no son de interés para el estudio, mejorando las etapas posteriores de segmentación y

clasificación. (Vera Meaurio & Martínez Jara, 2016). A continuación, se detallan las técnicas que deben ser aplicadas durante la etapa de Pre-procesamiento de las imágenes:

- *Conversión de RGB a escala de grises:* La conversión a escala de grises es el equivalente de la luminancia de una imagen, dentro de las funciones de OpenCv se cuenta con la función cvtColor, ésta puede realizar diferentes tipos de transformaciones:
    - CV_BGR2GRAY
    - CV_RGB2GRAY
    - CV_GRAY2BGR
    - CV_GRAY2RGB

Dependiendo de los canales requeridos. OpenCV emplea el siguiente principio para la conversión a escala de grises el cual está basado en la ecuación de Luminancia (Ey):

$$\text{RGB[A] to Gray:} \quad Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

En la figura 8 se muestra un ejemplo de la conversión a escala de grises empleando la función cvtColor con el siguiente comando:

```
cvtColor(src, bwsrc, CV_RGB2GRAY)
```

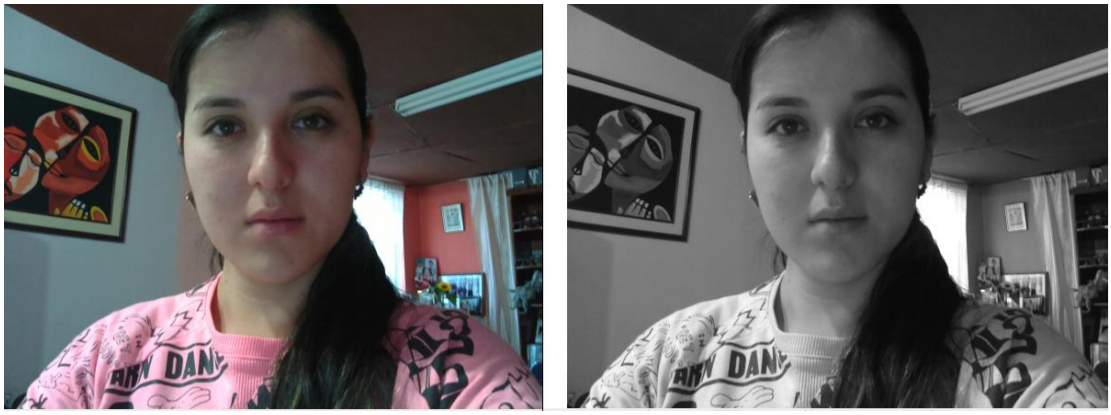


Figura 8. Conversión de imagen a escala de grises

Fuente: Autoría

- *Ecualización del histograma:* En el sitio web oficial de OpenCv se presenta la siguiente definición: *"La ecualización es un método que mejora el contraste en una imagen, con el fin de estirar el rango de intensidad."* (OpenCV documentation, 2014). Dada una imagen se busca agrupar los píxeles de manera que se normalizan los niveles de grises, aumentando el contraste entre las zonas oscuras y las zonas claras.

El comando empleado en OpenCv para ecualizar una imagen se presenta a continuación:

```
cv2.equalizeHist(src, dst )
```

En la figura 9 y figura 10 se muestra una imagen convertida a escala de grises a la cual se aplica la ecualización:

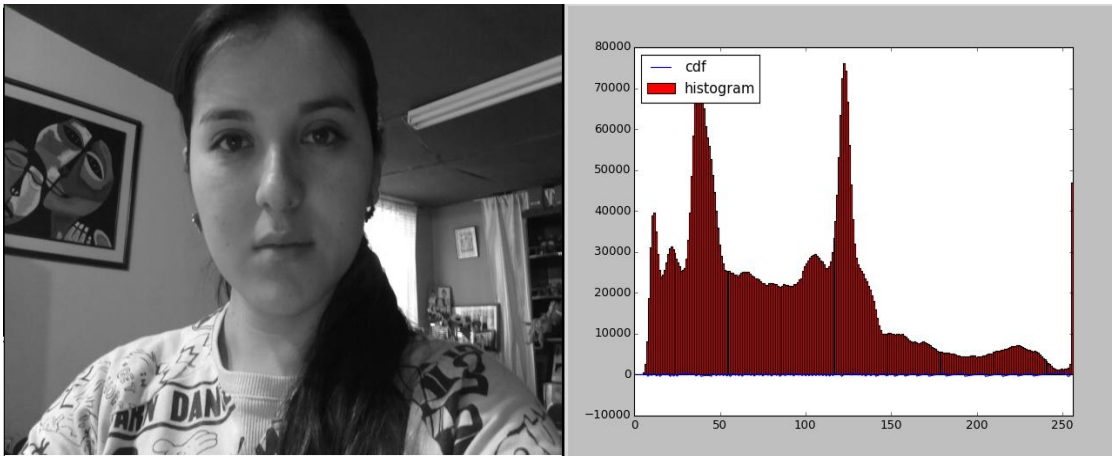


Figura 9. Ecualización del histograma.

Fuente: Autoría

Figura 10. Imagen original (izq) junto a la imagen ecualizada (der)

Fuente: Autoría

- *Binarización de la imagen:* La aplicación de binarización a una imagen permite clasificar los distintos valores de los píxeles, OpenCv cuenta con la función cv2.threshold, esta función se utiliza normalmente para conseguir únicamente dos niveles de una imagen de escala de grises, resulta muy útil para eliminar un ruido. A continuación, en la figura 11 se presenta un ejemplo de binarización propuesto en la página web oficial del software.

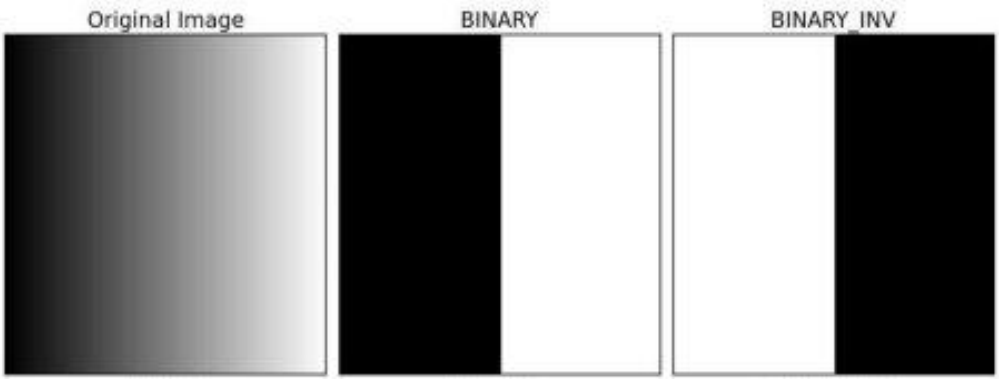


Figura 11. Ejemplo de imagen binarizada

Fuente: (OpenCV documentation, 2014)

3.7.1.2. Bloque 2: Detección de rostro y zona de ojos.

Para iniciar el desarrollo del bloque 2, se empieza detectando el rostro, para el reconocimiento facial OpenCV emplea un conjunto de bloques también conocidos como clasificadores, con el fin de reconocer distintos tipos de formas. Estos clasificadores son los Haar-Like Features, a continuación, se muestran en la figura 13:

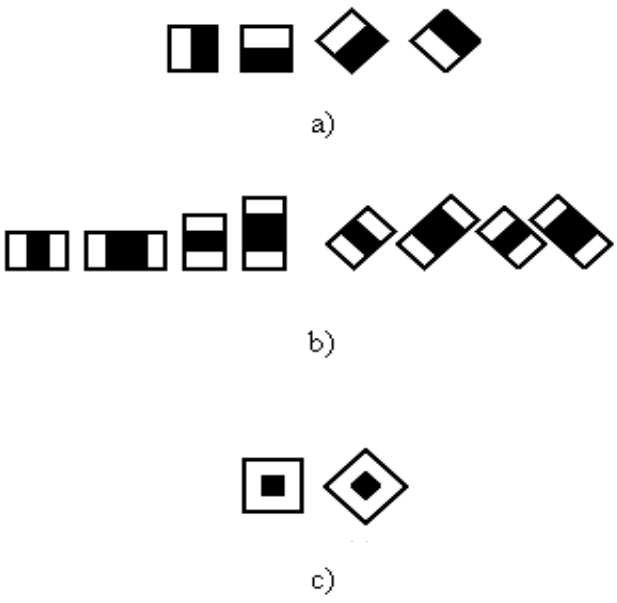


Figura 12. Haar Like Features utilizados para la detección de objetos

Fuente: Autoría

El algoritmo de detección facial usado en OpenCV es el propuesto por Paul Viola y Michael Jones que utiliza una variante del AdaBoost, y emplea a la vez los clasificadores Haar-Like Features, cada característica se superpone sobre la imagen ubicándolas de distintas maneras combinándolas hasta aproximarse a la imagen de un rostro. (Vivas)

Al emplear la herramienta OpenCV y a la vez Python se realizó el siguiente script que permite detectar un rostro y enmarcarlo dentro de un rectángulo. Se empleó la función cv2.CascadeClassifier y el clasificador que proporciona OpenCV denominado *faces.xml,* con esto conseguimos definir una zona ROI que será dentro de la cual se aplique el clasificador para ojos y boca. La estructura del script se muestra a continuación:

```
#agregamos las librerías necesarias para que funcione OpenCV en Python
import io
import picamera
import cv2
import numpy

#creamos un espacio de memoria para guarder las imágenes
stream = io.BytesIO()

#creamos el objeto para llamar a la cámara, definimos la resolución y el formato
with picamera.PiCamera() as camera:
   camera.resolution = (320, 240)
   camera.capture(stream, format='jpeg')

buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8)

#creamos una imagen empleando OpenCv
image = cv2.imdecode(buff, 1)

#se carga el HaarCascade de detección de rostro
face_cascade = cv2.CascadeClassifier('/home/pi/Desktop/GoPiGoLocal/faces.xml')

#se convierte a escala de grises
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

#se procede a buscar los rostros en la imagen
 faces = face_cascade.detectMultiScale(gray, 1.1, 5)
```

```
#una vez detectado el rostro se dibuja un rectángulo al contorno
for (x,y,w,h) in faces:
    cv2.rectangle(image,(x,y),(x+w,y+h),(255,0,0),2)

#se guarda el resultado
cv2.imwrite('result.jpg',image)
```

Como resultado del script de Python se muestran en la figura 13 la detección del rostro enmarcado en un rectángulo de color verde.

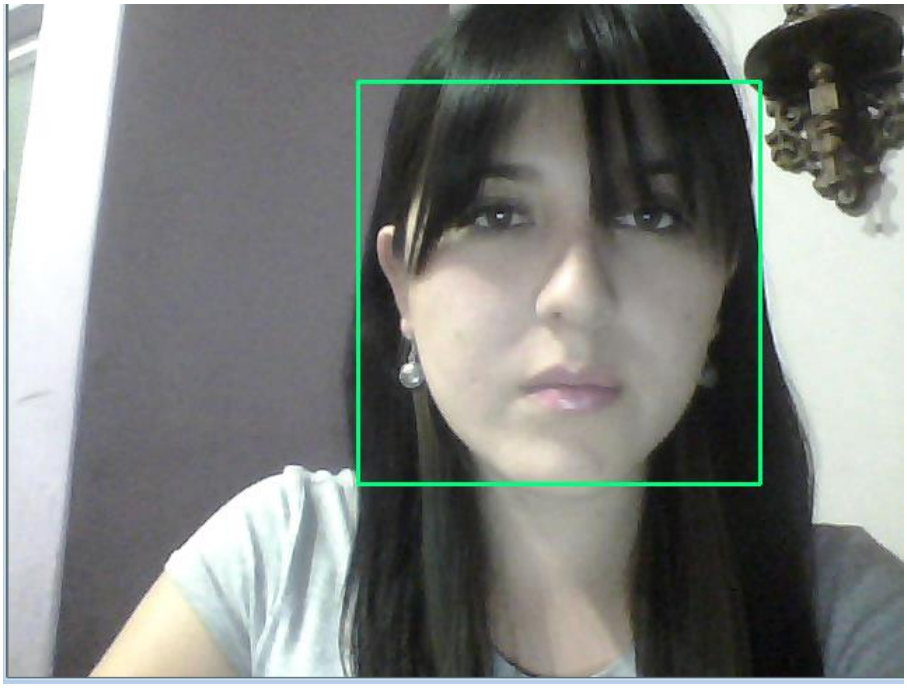


Figura 13. Prueba de detección de rostro

Fuente: Autoría

Aplicando el mismo script se realizaron las pruebas para conocer si el programa tiene la capacidad de reconocer varios rostros en una sola imagen, los resultados se muestran en la figura 14:

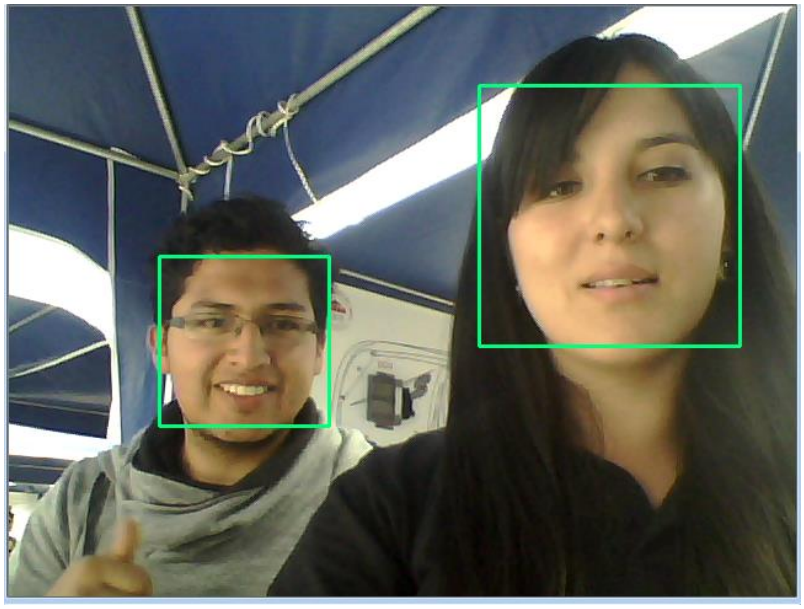


Figura 14. Detección de varios rostros

Fuente: Autoría

De igual manera se probó la detección de rostro a una distancia mayor a 1 metro, los resultados obtenidos se muestran en la figura 15:

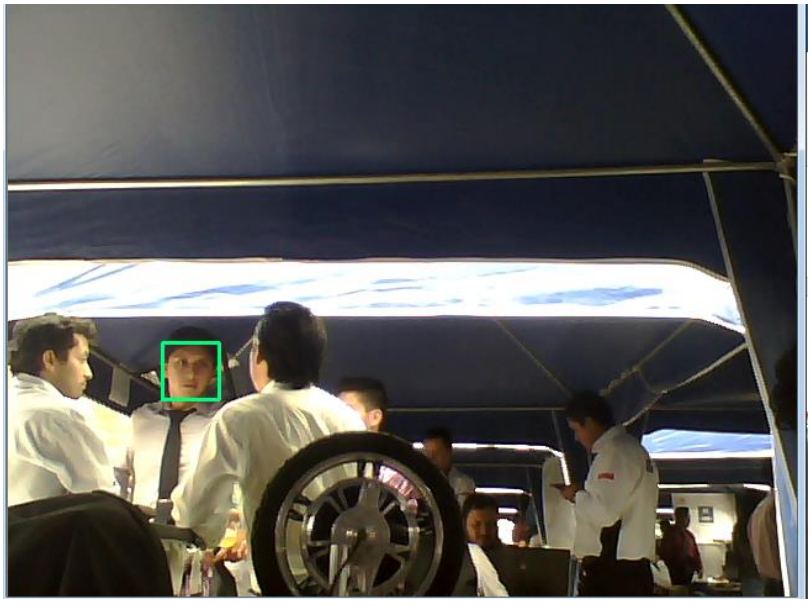


Figura 15. Detección de rostros en diferentes escalas

Fuente: Autoría

El inconveniente que se presentó durante las pruebas de detección de rostro, es que al ubicarse la cámara en una sala con mucha iluminación, el sistema presenta errores, además al no estar el rostro frontalmente no se marca el rectángulo que identifica la sección del rostro como a continuación se indica en la figura 16. Estos errores deben ser tomados en cuenta para el sistema final.

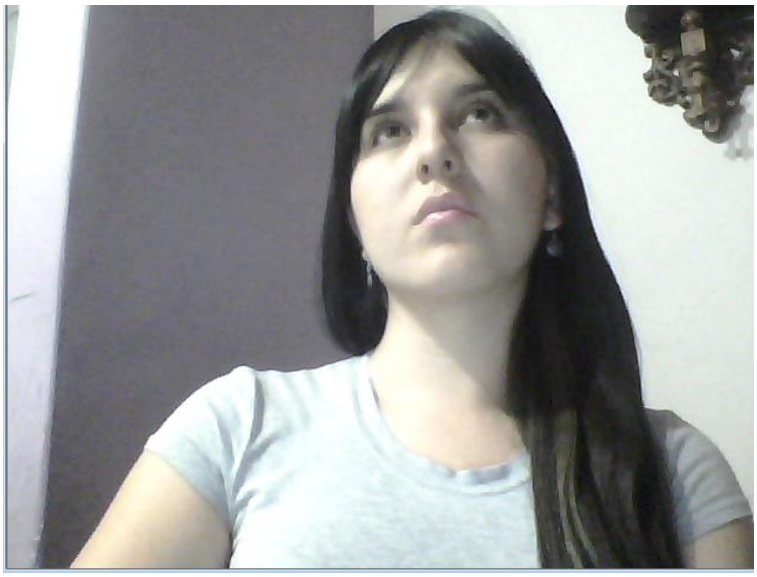


Figura 16. Error en la detección de rostro

Fuente: Autoría

Continuando con el desarrollo del bloque 2, se presenta el siguiente paso del algoritmo que es detectar la región de los ojos, para ello se utiliza el script anterior en donde ya se tiene enmarcado el rostro. El script para la detección de ojos y rostro es el siguiente:

```
#agregamos las librerías necesarias para que funcione OpenCV en Python
import io
import picamera
import cv2
import numpy

stream = io.BytesIO()
with picamera.PiCamera() as camera:
        camera.resolution = (320, 240)
        camera.capture(stream, format='jpeg')
```

```
buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8)
image = cv2.imdecode(buff, 1)

face_cascade = cv2.CascadeClassifier('/home/pi/Desktop/GoPiGoLocal/faces.xml')
eye_cascade=cv2.CascadeClassifier('/home/pi/Desktop/GoPiGoLocal/haarcascade_eye_tree_eyeglasses.xml')
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray, 1.1, 5)
for (x,y,w,h) in faces:
        cv2.rectangle(image,(x,y),(x+w,y+h),(0,255,0),2)
        roi_gray = gray [y:y+h, x:x+w]
        roi_color= image[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color, (ex,ey),(ex+ew, ey+eh), (0,255,0),2)

cv2.imwrite('result.jpg',image)
```

La parte fundamental del script es el uso de los clasificadores Haar-Like Features para detectar los ojos y el rostro, la estructura para el uso del clasificador en el script hace el llamado a una función denominada *CascadeClassifier* la cual requiere como parámetro la ubicación exacta del archivo que contiene el código del clasificador. Los archivos de varios clasificadores están disponibles en la Internet únicamente se necesita descargarlos y guardarlos en la misma carpeta que los scripts de monitoreo. A continuación, se muestra la estructura del clasificador de rostro y de ojos:

- Clasificador para detección de rostro

  face_cascade = **cv2.CascadeClassifier**('/home/pi/Desktop/GoPiGoLocal/faces.xml')

- Clasificador para detección de ojos

  eye_cascade=**cv2.CascadeClassifier**('/home/pi/Desktop/GoPiGoLocal/haarcascade_eyes.xml')

Los resultados obtenidos con la ejecución del script se pueden observar en la figura 17, aquí podemos observar que tanto el rostro como los ojos han sido detectados y enmarcados utilizando rectángulos:

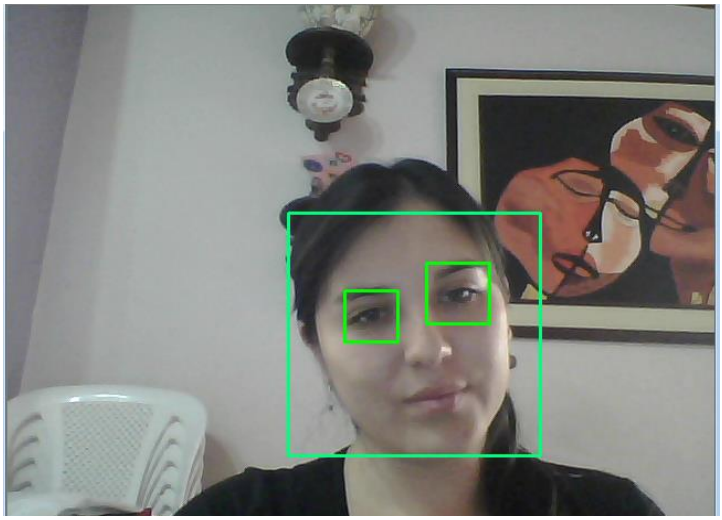


Figura 17. Detección de rostro y ojos

Fuente: Autoría

Durante el diseño se consideró que el sistema debe emplearse también para estudiantes que usen lentes, por lo cual se realizó la prueba de detección usando lentes y el resultado fue negativo, pues el sistema no reconoció adecuadamente el rostro ni los ojos. El resultado de esta prueba se presenta en la figura 18.

Figura 18. Error de reconocimiento al usar lentes

Fuente: Autoría

Para superar este inconveniente y mejorar el sistema para que sea aplicable tanto para personas con lentes como sin ellos, se puede emplear otro tipo de clasificador que está diseñado específicamente para este propósito, este clasificador es una variación al clasificador de ojos inicialmente diseñado por OpenCV y presenta una estructura similar a lo indicado anteriormente por lo cual se llamará a la función *CascadeClassifier* y se indicará la ubicación del archivo .xml. La estructura del clasificador para personas que usen lentes es:

eye_cascade = cv2.CascadeClassifier('haarcascade_eye_tree_eyeglasses.xml')

La prueba realizada al ejecutar el script incluyendo el nuevo clasificador se puede observar en la figura 19, donde a pesar del uso de lentes se logró la detección de rostro y ojos.

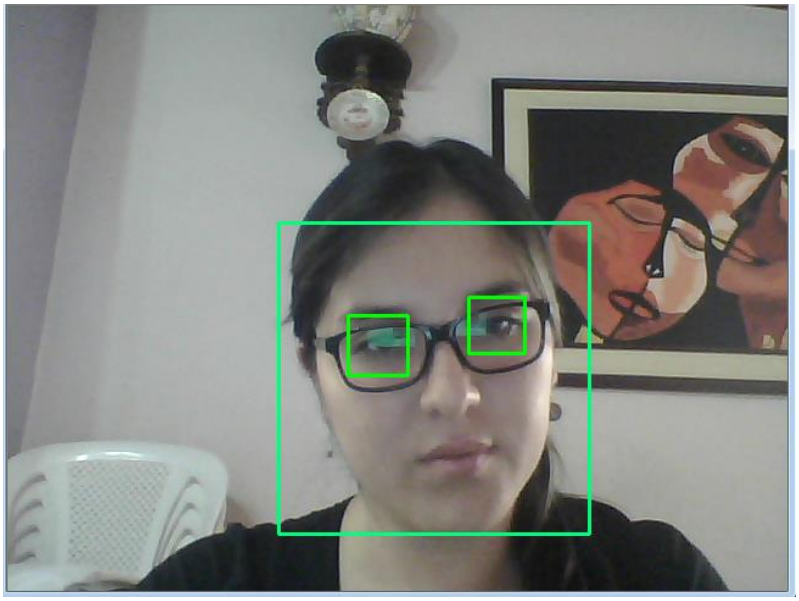


Figura 19. Detección de rostro y ojos, usando lentes

Fuente: Autoría

3.7.1.3. Bloque 3: Detección de boca.

El siguiente bloque a desarrollar corresponde a la detección de boca, el objetivo es conocer si el estudiante ha bostezado o no. Para continuar con el desarrollo del algoritmo se utilizará el script ya indicado en el apartado anterior en el que ya tenemos detectado ojos y rostro, esto se lo realiza porque se requiere que todos los detectores funcionen simultáneamente.

El script está diseñado para que una vez que ha encontrado la boca, encierre esta área dentro de un rectángulo. Tal como se realizó para la detección de rostro y ojos, se utilizará un clasificador propio de OpenCV que permite la detección de boca. El archivo "mouth.xml" está almacenado en la misma carpeta raíz del script y emplea la función *CascadeClassifier* que ya se empleó en scripts anteriores. A continuación, se muestra el script:

```
#agregamos las librerías necesarias para que funcione OpenCV en Python
import io
import picamera
import cv2
import numpy
import time

stream = io.BytesIO()

with picamera.PiCamera() as camera:
   camera.resolution = (320, 240)
   camera.capture(stream, format='jpeg')


buff = numpy.fromstring(stream.getvalue(), dtype=numpy.uint8)

image = cv2.imdecode(buff, 1)

# cargamos los clasificadores
face_cascade = cv2.CascadeClassifier('/home/pi/Desktop/GoPiGoLocal/faces.xml')
eye_cascade=cv2.CascadeClassifier('/home/pi/Desktop/GoPiGoLocal/haarcascade_eye_tree_eyeglas
ses.xml')
mouth_cascade=cv2.CascadeClassifier('/home/pi/Desktop/GoPiGoLocal/mouth.xml')
```

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

faces = face_cascade.detectMultiScale(gray, 1.1, 5)

print "Found "+str(len(faces))+" face(s)"

#dibujamos un rectángulo alrededor del rostro dectectado
 for (x,y,w,h) in faces:
    cv2.rectangle(image,(x,y),(x+w,y+h),(0,255,0),2)
    roi_gray = gray [y:y+h, x:x+w]
    roi_color= image[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color, (ex,ey),(ex+ew, ey+eh), (0,255,0),2)
    mouth= mouth_cascade.detectMultiScale(roi_gray)
    time.sleep(5)
    for (mx,my,mw,mh) in mouth:
        cv2.rectangle(roi_color, (mx,my),(mx+mw, my+mh), (0,255,0),2)

cv2.imwrite('result.jpg',image)
```

Los resultados que se obtuvieron con la implementación del script se pueden apreciar en la figura 20:

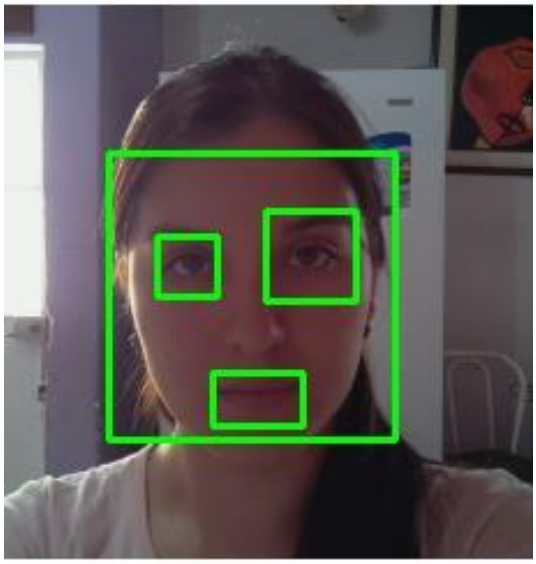


Figura 20. Detección de rostro, ojos y boca

Fuente: Autoría

3.7.1.4. Bloque 4: Análisis de aspectos fisiológicos.

Cuando ya se ha implementado el script de detección de ojos y boca se procede a hacer un análisis más detallado de las características de cada uno. Con la región de los ojos ya enmarcados, se procede a ejecutar el contador de pestañeo y con la región de la boca se detectará el bostezo.

Para implementar los contadores en Python se utiliza la palabra reservada *while* que nos permite realizar ciclos, ejecutando el código múltiples veces. El funcionamiento del ciclo *while* se basa en una comparación lógica entre una variable y un estado, que puede ser mayor, menor, o igual y cuyo resultado será un valor verdadero o falso, basado en ese valor se ejecutarán las siguientes instrucciones del script.

El primer contador se ha denominado *pestaneo,* tendrá un valor incial de 0 y será la variable encargada de guardar el número de pestañeos realizados durante 1 minuto, una vez transcurrido este tiempo el contador vuelve a tomar el valor de 0. Se estima que la variable puede tomar valores desde 5 a 20 según la información analizada de Häkkänen *et al*. (1999). Los valores de esta variable serán almacenados en un archivo *.csv.*

El segundo contador se denomina bostezo, tendrá un valor inicial de 0 y será la variable encargada de almacenar el número de bostezos realizados durante 1 minuto, al finalizar este tiempo, el contador se encera y vuelve a iniciar el conteo. El valor que puede tomar este contador dependerá de la fatiga que presente el sujeto monitoreado (Muchnik, Finkielman, Seemeniuk, & de Aguirre, s.f.). Este valor será almacenado en el mismo archivo .csv con el contador de pesteñeo.

Como se había indicado en la definición de los contadores, se habló de que la duración del contador debe ser de 1 minuto, para esto se utiliza la librería de Python llamada *time*, con esta librería podemos declarar variables globales de inicio y fin de tiempo de duración permitiéndonos recolectar datos del monitoreo minuto a minuto.

Asimismo, se indicó que los contadores serán almacenados en un archivo .csv, este archivo es un tipo de documento que permite representar los datos en un formato parecido a una tabla, facilitando la accesibilidad a la información. Para crear un archivo .csv desde Python se debe importar el módulo **csv,** este módulo contiene operaciones de lectura y escritura, para el caso del script de monitoreo facial se emplean csv.writer y csv.writerow.

3.7.1.5. Bloque 5: Visualizar datos en la plataforma.

El objetivo principal de este trabajo se resume en este bloque, donde todos los datos recolectados son presentados como información clara que el usuario puede interpretar, aquí se visualizarán los estimadores de desconcentración en la plataforma en Internet, que en este caso es la plataforma Ubidots. Los datos que ya han sido almacenados en un archivo .csv serán subidos en tiempo real desde el mismo sistema embebido a través del uso de un script de Python.

El primer paso para la creación de la aplicación será dirigirnos a la página principal de Ubidots (https://ubidots.com), damos click en la opción Sign Up y nos solicita ingresar la información personal para crear la cuenta: nombre de usuario, correo electrónico y contraseña. Tal como se indica en la figura 21.

Figura 21. Página principal de la plataforma Ubidots

Fuente: Autoría

Una vez creada la cuenta, nos dirigimos a la pestaña Sources para generar una nueva fuente de datos con sus respectivas variables, damos click en Add Data Source y nombramos a nuestra Data Source según requiera el usuario, en este caso se ha nombrado como Sistema de Monitoreo Facial, la creación puede apreciarse en la figura 22.

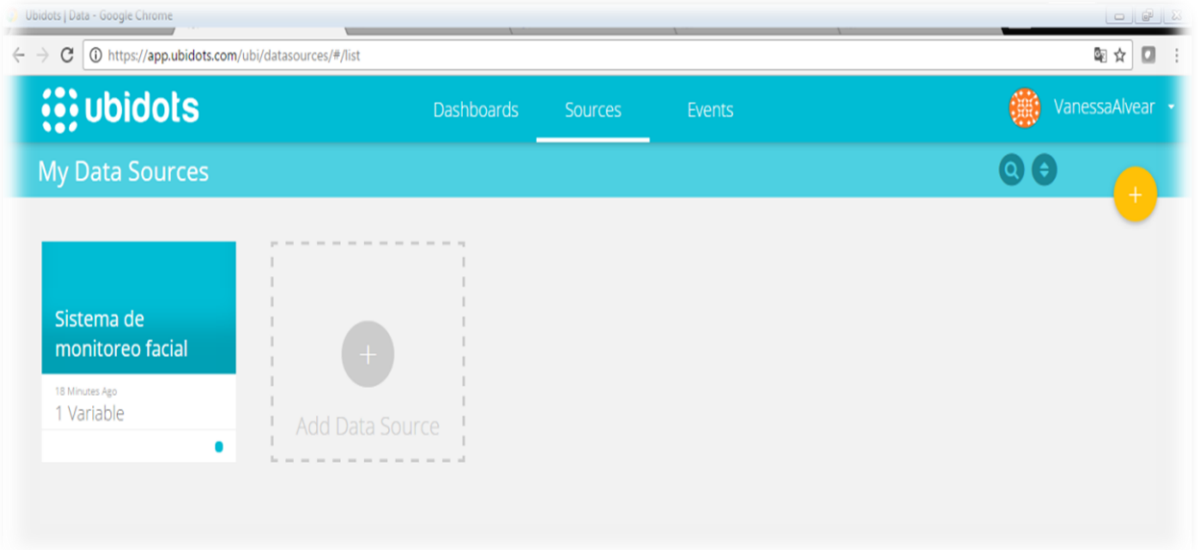


Figura 22. Creación de data source en la plataforma Ubidots

Fuente: Autoría

Dentro del Data Source creado vamos a iniciar las variables que se desea visualizar, en este caso se ha creado una variable para el contador de pestañeo y una para el contado de bostezo, la creación de variables puede observarse en la figura 23:

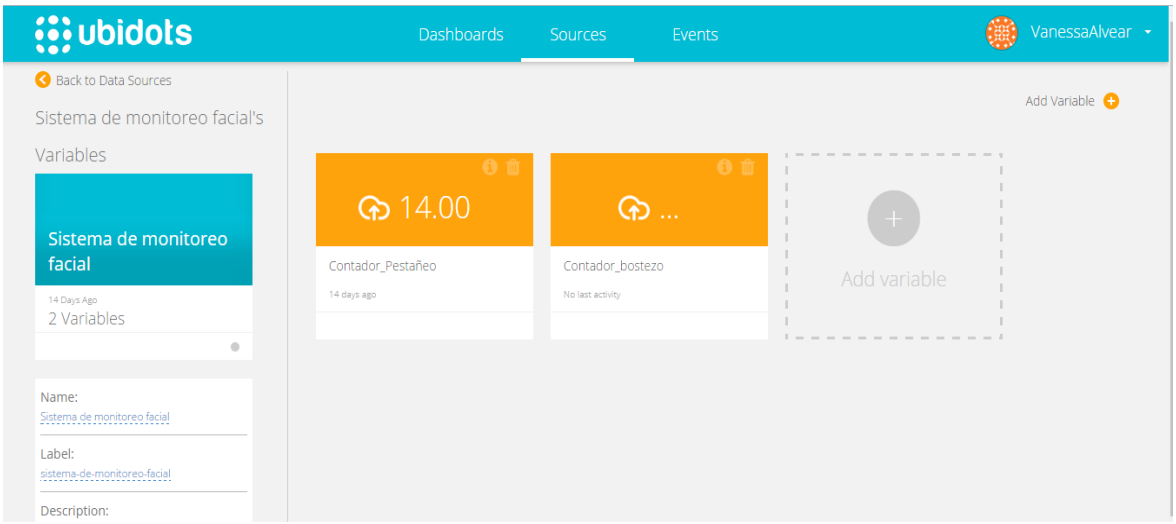


Figura 23. Creación de variables en la plataforma IoT

Fuente: Autoría

A cada variable creada se le asignará un ID, este identificador es el que permitirá el envío de datos desde nuestro Raspberry Pi, este valor se especifica dentro del script siendo un valor único e irrepetible para cada variable en cada Data Source creado. Otro identificador que se debe tener en cuenta es el API Key, valor que estará asignado de manera única para cada cuenta creada en la plataforma.

Después de haber ejecutado el script desde el Raspberry Pi, podemos ir visualizando los datos en tiempo real, brindando un entorno gráfico y amigable que permite una mejor interpretación de la información. Dentro de las opciones para visualizar los datos tenemos las siguientes: Datos en bruto, Promedio de los datos, Dato mínimo, Dato máximo, y la sumatoria.

En la figura 24 se presentan los datos correspondientes a la variable creada en Ubidots que representa al contador de pestañeo, se puede observar que los datos son subidos minuto a minuto a la plataforma:

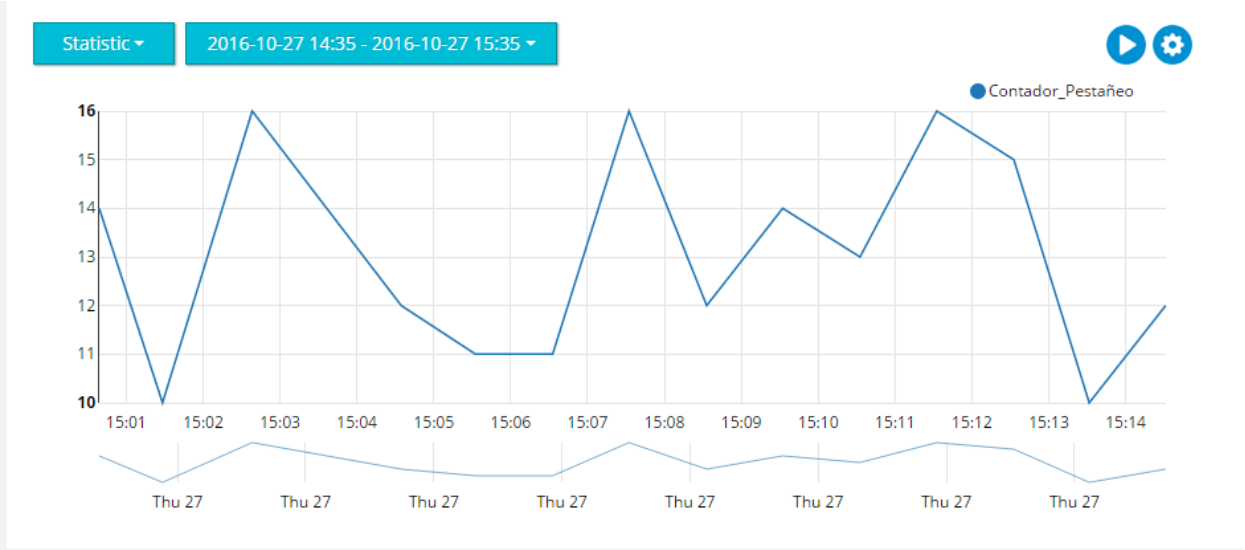


Figura 24. Datos del contador de pestañeo visualizados en la plataforma Ubidots

Fuente: Autoría

Los datos obtenidos en la variable de Contador de Bostezo pueden visualizarse en la plataforma en tiempo real, aquí se indican los valores mínimos y valores máximos que pued tomar la variable y el tiempo en el que ha sido subido cada uno de los datos, la figura 25 presenta estos resultados.

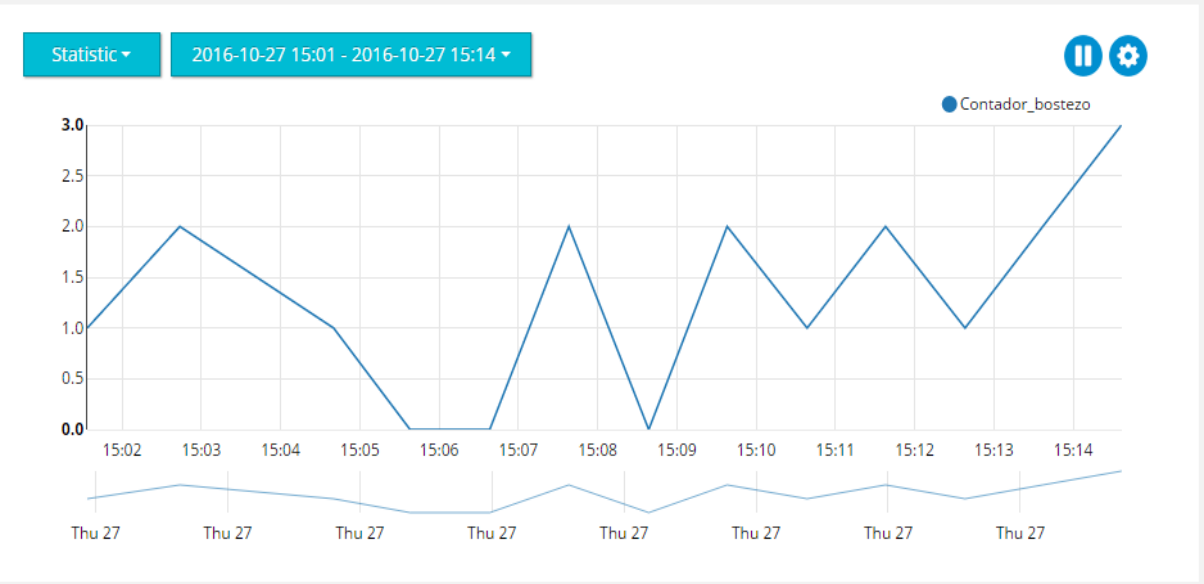


Figura 25. Datos del contador de bostezo visualizados en la plataforma Ubidots

Fuente: Autoría

Para conseguir una mejor visualización de los datos la plataforma Ubidots nos permite crear dashboards, aquí podremos relacionar nuestras variables para tener un mejor entendimiento de los datos obtenidos. Las opciones en cuanto a dashboards que nos permite crear la plataforma son: gráficas, métricas, mapas, tablas, indicadores y control.

En la figura 26 se muestra un dashboard empleando la opción de gráficas, se observa que en el eje vertical izquierdo tenemos la escala de valores para el contador de pestañeo, mientras que en el eje vertical izquierdo se ubica la escala de valores para el contador de bostezos, en el eje horizontal se observa la hora en la que han sido subidos los datos a la plataforma.

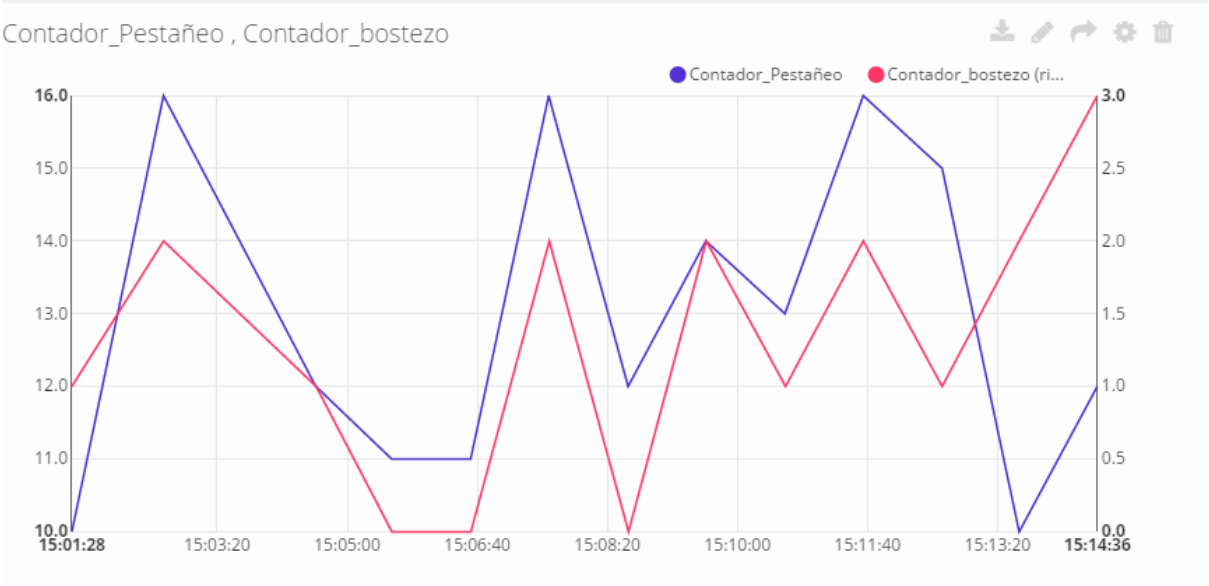


Figura 26. Dashboard visualizado con gráfica de líneas

Fuente: Autoría

3.8 Diagrama de Flujo del Sistema.

El diagrama de flujo desarrollado para el sistema de monitoreo ha sido el que ha guiado la parte de programación del script, aquí se muestran las condicionantes de cada variable, como funcionarán los contadores y la detección de la desconcentración del estudiante. En la figura 27 se presenta este diagrama:

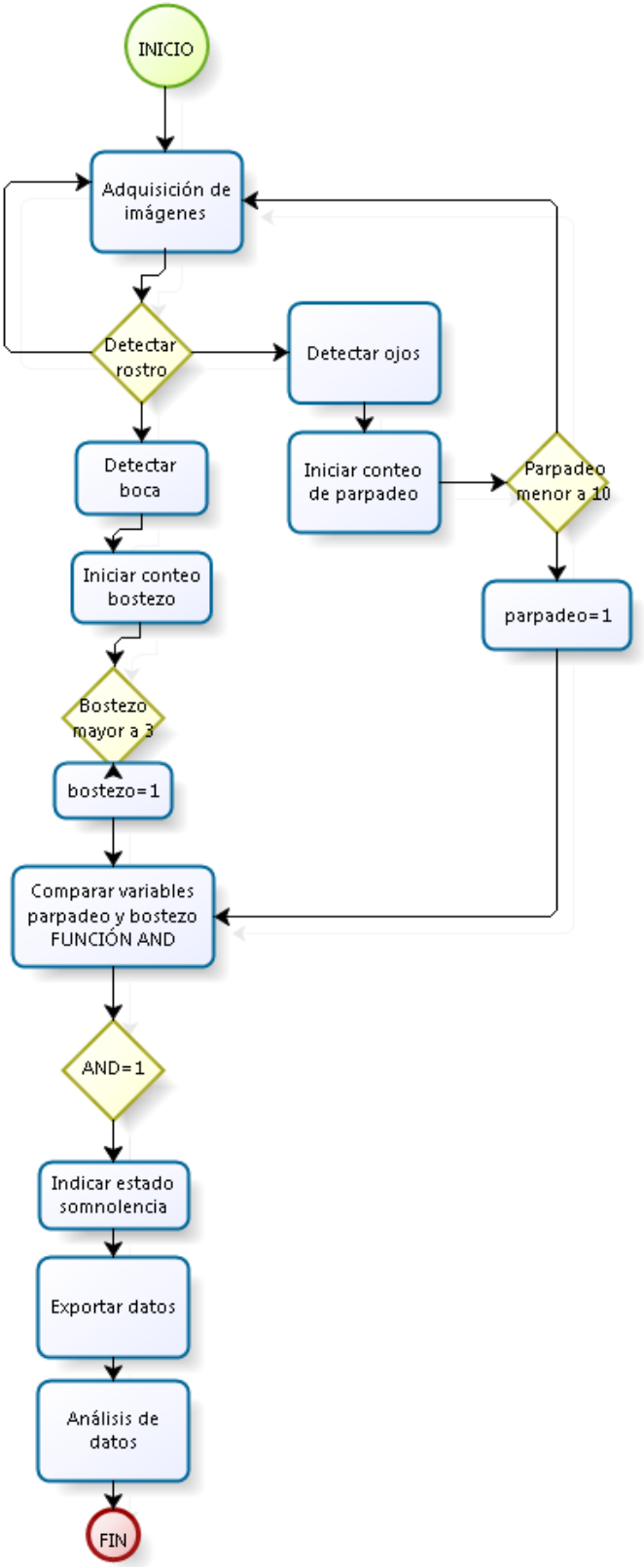


Figura 27. Diagrama de flujo del sistema

Fuente: Autoría

3.8 Diagrama Circuital

A continuación, en la figura 28 se presenta el diagrama circuital del dispositivo, se debe tomar en cuenta que para el funcionamiento del sistema únicamente se requiere el sistema embebido Raspberry Pi 2 modelo B, el Raspberry camera module v1 y la fuente de alimentación micro USB.

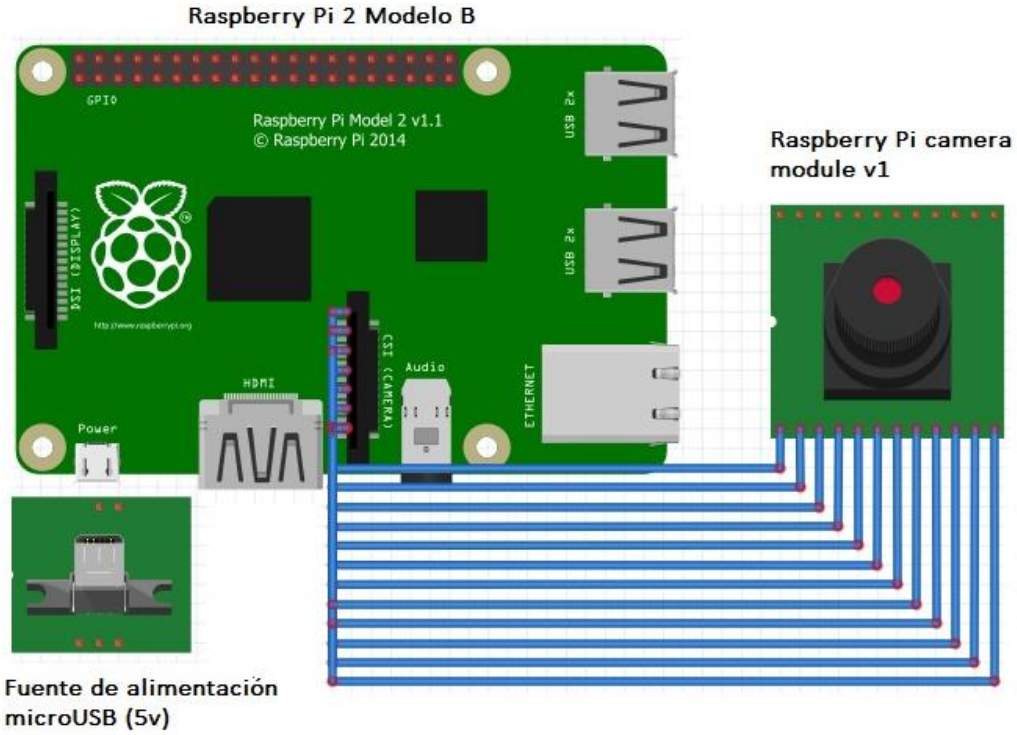


Figura 28. Diagrama Circuital del dispositivo

Fuente: Autoría

3.10 Análisis de Ubicación de la Cámara.

Para el análisis de ubicación de la cámara dentro del aula de clase se ha considerado las limitaciones ya tratadas con anterioridad como son que la cámara debe estar ubicada frontalmente para que pueda detectar los ojos correctamente, adicionalmente se debe considerar que la cámara debe estar a una altura específica para que tenga un buen enfoque y pueda realizar la detección.

Tomando en cuenta que los estudiantes presentan características físicas distintas, relacionadas principalmente al género y a la etnia de la cual son parte, se debe analizar qué estatura promedio tienen los estudiantes de CIERCOM de la UTN. Para ello se buscó información específica sobre talla, pero no existen datos al respecto dentro de las bases de datos de estudiantes de la universidad. Una vez analizada esta situación se procedió a buscar información más general, optando por basarnos en la estatura promedio de los ecuatorianos.

La Organización Mundial de la Salud (OMS) realizó en el 2014 un estudio para determinar la estatura promedio en un sinnúmero de países, incluyendo Ecuador. Los datos de la OMS coinciden con estudios realizados a nivel nacional, estos resultados se muestran en la tabla 13.

Tabla 13. Promedio de Estatura de los ecuatorianos

Etnia	**Estatura según el género**	
	Hombres	Mujeres
Indígenas	1,55mts	1,45mts
Afro ecuatorianos	1,80mts	1,65mts
Mestizos	1,68mts	1,55mts
PROMEDIO	1,68mts	1,55mts

Fuente: (Organización Mundial de la Salud, 2014)

A continuación, se debe analizar de igual manera la altura del mobiliario de las aulas de la Universidad Técnica del Norte, en este caso y basándose en estudios realizados para determinar las dimensiones funcionales con las que se debe contar para que no existan daños musculoesqueléticos, las dimensiones correctas para una mesa de trabajo son detalladas en la tabla 14:

Tabla 14. Dimensiones funcionales para una mesa de trabajo

	NORMA
Altura de la mesa	70-72
Profundidad útil	>60
Anchura útil	>120
Altura libre debajo de la mesa	>65
Anchura libre debajo de la mesa	>60
Profundidad libre debajo de la mesa (rodillas)	>45
Profundidad libre debajo de la mesa (pies)	>60

Fuente: (Servicios de Prevención de Riesgos Laborales (SEPRUMA), 2004)

En cuanto a las medidas funcionales para las sillas de trabajo de las aulas de clase se presenta la siguiente información en la tabla 15:

Tabla 15. Medidas funcionales para la silla de trabajo

	REGULABLE
Altura del asiento	38-54
Profundidad del asiento	40-44
Anchura del asiento	>40
Inclinación del asiento	-5° a 5°
Altura del apoyo lumbar	>35
Altura del borde superior del respaldo	>50
Radio lumbar	40
Ángulo asiento-respaldo	90° - 120°

Fuente: (Servicios de Prevención de Riesgos Laborales (SEPRUMA), 2004)

Haciendo una relación entre la altura promedio de los ecuatorianos con la altura promedio del mobiliario de la Universidad Técnica del Norte se determina que la altura a la que debe estar ubicado el Raspberry Pi camera module v2 para que pueda detectar tanto a hombres

como a mujeres es a una altura de 25cm sobre la mesa de trabajo, es decir a 95cm aproximadamente del suelo.

3.11 Análisis de consumo de corriente del sistema.

La placa del Raspberry Pi 2 modelo B requiere ser alimentado mediante un suministro micro USB, lo que nos indica que maneja un voltaje de 5V y un consumo de corriente que dependerá de los periféricos que requiera el sistema, en el caso del Sistema de Monitoreo Facial el único componente adicional a usar es el módulo PiCamera conectado a la entrada CSI de la placa.

El datasheet proporcionado por la página oficial www.raspberrypi.org indica que el módulo de la cámara consume alrededor de 250mA, por lo que con una fuente que suministre 1000 mA el sistema puede funcionar correctamente.

3.12 Costo del proyecto

El costo de los materiales utilizados en el sistema de monitoreo facial se detalla a continuación en la tabla 16:

Tabla 16. Descripción de costos del proyecto

Descripción	Cantidad	Valor Unit.	Valor Total
Raspberry Pi 2 Modelo B	1	$67,00	$67,00
Módulo Pi Camera	1	$30,00	$30,00
Micro SD Kingston 16 GB (clase 10)	1	$15,00	$15,00

Cable de red 3m	1	$3,00	$3,00
Cable HDMI 3m	1	$6,00	$6,00
Mouse USB	1	$10,00	$10,00
Teclado USB	1	$12,00	$12,00
Caja plástica de PVC	1	$2,00	$2,00
		TOTAL	$145,00

Fuente: Autoría

CAPÍTULO IV

Pruebas y Conclusiones

El capítulo 4 de este trabajo contempla la fase de pruebas realizadas con el sistema de monitoreo facial, las cuales han sido realizadas con diferentes estudiantes y en horarios indistintos de manera que permita conocer la relación entre los resultados del monitoreo y los horarios de clase de los alumnos. Se presenta una fase de pruebas por cada bloque desarrollado en la fase de diseño, una fase de pruebas preliminares, una fase de pruebas finales y un apartado con el análisis de resultados de las mismas.

Una vez realizadas las pruebas, se concluye sobre los resultados obtenidos al finalizar el proyecto, indicando como el sistema ha cumplido los objetivos plasmados en un inicio y como se superaron las limitaciones durante el desarrollo del proyecto. Se presentan recomendaciones generales para la implementación del sistema e implementaciones similares.

4.1. Pruebas del sistema

Culminada la etapa de diseño se procede a realizar la fase de pruebas para validar el funcionamiento del sistema. El objetivo principal de dichas pruebas es verificar que el sistema realice un monitoreo facial exitoso a estudiantes con características físicas diferentes y durante horarios de clase distintos.

Las pruebas verificarán el correcto funcionamiento en situaciones controladas para posteriormente analizar los resultados. Se comprende como situaciones controladas

escenarios tales como aulas de clase donde se cuenta con iluminación adecuada, espacios libres de obstáculos frente a los estudiantes, mobiliario de acuerdo a lo indicado en la tabla 15. Con el desarrollo de estas pruebas se pretende encontrar posibles errores que se presenten en el momento de la implementación del sistema.

Para la verificación del sistema se proponen dos tipos de pruebas, de funcionalidad y de usabilidad. Las pruebas de funcionalidad serán las encargadas de examinar si el sistema cumple con todos los requerimientos y objetivos señalados en el diseño, en este tipo de pruebas se llevará el sistema a tareas límites, como el funcionamiento por un tiempo mayor al previsto de un periodo académico. Por otra parte, las pruebas de usabilidad serán las que permitan conocer que tan fácil e intuitivo puede resultar el manejo del sistema para los distintos usuarios, tanto para el uso del sistema de monitoreo como para el ingreso a la plataforma de Ubidots.

4.1.1 Pruebas Preliminares del Sistema

Conforme se diseñó el sistema se requirió realizar pruebas para comprobar el funcionamiento de cada bloque, por lo tanto además de las pruebas finales se realizaron otras cuyo objetivo fue comprobar el funcionamiento correcto del algoritmo del sistema de monitoreo facial. El desarrollo de las pruebas se realizó acorde al cronograma presentado a continuación, la tabla 17 indica el tipo de prueba, los sujetos de prueba, lugar de desarrollo de las pruebas, los resultados esperados y la duración de las mismas.

Tabla 17. Cronograma de pruebas del Sistema de Monitoreo Facial

CRONOGRAMA DE PRUEBAS				
TIPO DE PRUEBA	**SUJETOS DE PRUEBA**	**LUGAR DE DESARROLLO DE LAS PRUEBAS**	**RESULTADOS ESPERADOS**	**DURACIÓN**
BLOQUE 1: Comprobación de la calidad de imágenes captadas por el módulo PiCamera en las aulas de clase, considerando la cantidad de luz en los diferentes horarios y periodos académicos		Aulas de clase de la Facultad de Ingeniería en Ciencias Aplicadas	Se espera comprobar que la calidad de las imágenes captadas sea la óptima para ser procesadas por OpenCV	2 semanas, en distintos horarios con diferente cantidad de luz
Comprobación de la ubicación de la cámara	Estudiantes de CIERCOM	Aulas de clase de la Facultad de Ingeniería en Ciencias Aplicadas	Se busca comprobar que la ubicación propuesta en apartados anteriores sea la	2 semanas

			adecuada tomando en cuenta la forma de sentarse de los alumnos	
BLOQUE 2: Detección de rostros	Estudiantes de CIERCOM	Aulas de clase de la Facultad de Ingeniería en Ciencias Aplicadas	Se requiere conocer el porcentaje de efectividad del sistema al detectar rostros con las distintas características físicas de los estudiantes	2 semanas
BLOQUE 2: Detección de ojos	Estudiantes de CIERCOM	Aulas de clase de la Facultad de Ingeniería en Ciencias Aplicadas	Se busca verificar que el sistema sea capaz de detectar los ojos indistintamente de si el estudiante use lentes o no	4 semanas
BLOQUE 3: Detección de boca	Estudiantes de CIERCOM	Aulas de clase de la Facultad de Ingeniería en Ciencias Aplicadas	Se pretende confirmar que la detección de la boca se realice simultáneamente a la detección de los ojos	2 semanas

BLOQUE 4: Comprobación de contadores de bostezos y pestañeos	Estudiantes de CIERCOM	Aulas de clase de la Facultad de Ingeniería en Ciencias Aplicadas	Se verificará que los contadores programados en el sistema funcionen correctamente	4 semanas
BLOQUE 5: Visualización de los datos recolectados por el sistema de monitoreo en la plataforma IoT	Estudiantes de CIERCOM	Aulas de clase de la Facultad de Ingeniería en Ciencias Aplicadas	Se verificará que la información del monitoreo pueda ser visualizada en la plataforma IoT	2 semanas
BLOQUE 5: Pruebas de manejo de la plataforma IoT		Aulas de clase de la Facultad de Ingeniería en Ciencias Aplicadas	Se busca determinar si la plataforma posee un entorno amigable para el usuario y	4 semanas

Fuente: Autoría

En el Bloque 1 se contemplaron dos tipos de pruebas tal como se detallan en la tabla 16. En la primera prueba se obtuvo un 6% de error debido a malas condiciones de iluminación dentro de las aulas de clase especialmente en horas de la mañana donde existe excesiva cantidad de luz.

El porcentaje de error de la segunda prueba fue de un 10%, ocasionado por distintos obstáculos que pueden presentarse dentro del aula, tales como equipos y materiales ubicados en los escritorios e incluso la presencia de otros estudiantes interfiriendo entre el dispositivo y el estudiante a ser monitoreado. La tabla 18 presenta el resumen de las pruebas del Bloque 1:

Tabla 18. Resultados de las pruebas del bloque 1

Pruebas	Nro pruebas	Pruebas Exitosas	Pruebas erróneas	% Aciertos	% Fallos
Prueba 1 Comprobación de la calidad de las imágenes	100	94	6	94,00%	6,00%
Prueba 2 Comprobación de la ubicación de la cámara	100	90	10	90,00%	10,00%
TOTAL	**200**	**184**	**16**	**92,00%**	**8,00%**

Fuente: Autoría

Las pruebas para el Bloque 2 comprendieron la detección de rostros y detección de ojos. Estas pruebas fueron realizadas con estudiantes de CIERCOM en aulas de clase y laboratorios, el porcentaje de éxito de estas pruebas fue de un 95,5% que nos indica que el porcentaje de falsos positivos que pueden presentarse en el sistema son mínimos, dándole mayor efectividad al monitoreo facial. Los resultados obtenidos se presentan en la tabla 19.

Tabla 19. Resultados de las pruebas del bloque 2

Pruebas	Nro pruebas	Pruebas Exitosas	Pruebas erróneas	% Aciertos	% Fallos
Prueba 1 Detección de rostros	600	575	25	95,83%	4,17%
Prueba 2 Detección de ojos	400	380	20	95,00%	5,00%
TOTAL	**1000**	**955**	**45**	**95,50%**	**4,50%**

Fuente: Autoría

La prueba del bloque 3 correspondió a la detección de la boca. El porcentaje de éxito fue de 92,75% que aseguró la efectividad del sistema, el resumen de los resultados obtenidos se muestra en la tabla 20.

Tabla 20. Resultados de las pruebas del Bloque 3

Pruebas	Nro pruebas	Pruebas Exitosas	Pruebas erróneas	% Aciertos	% Fallos
Prueba 1 Detección de boca	400	371	29	92,75%	7,25%
TOTAL	400	371	29	**92,75%**	**7,25%**

Fuente: Autoría

Para las pruebas del Bloque 4 correspondientes a la comprobación de los detectores de bostezo y pestañeo se realizaron pruebas durante 1 semana con distintos estudiantes de CIERCOM. Los resultados de estas pruebas se muestran a continuación en una tabla resumen por cada día de monitoreo realizado, además se presentan las gráficas estadísticas que contienen los datos de los contadores de bostezo y pestañeo. El formato de la tabla presenta algunos datos informativos como el día de realización de la prueba, lugar, sujetos de prueba, hora de monitoreo, tiempo de monitoreo y observaciones. La tabla 21 presenta los resultados del monitoreo del día 1.

Tabla 21. Descripción de la prueba del Bloque 4, Día 1

PRUEBA BLOQUE 4: SISTEMA DE MONITOREO FACIAL		
Día: Lunes	Localización:	Universidad Técnica del Norte, Edificio FICA, Laboratorio Fibra Óptica
	Sujetos de prueba:	Estudiantes de CIERCOM de sexto semestre
	Hora en que se realizó el monitoreo:	Durante la mañana que es el periodo académico de los estudiantes
	Tiempo de monitoreo:	10 minutos cada estudiante
Observaciones: El sistema de monitoreo funcionó correctamente y detectó el rostro, ojos y boca de cada uno de los estudiantes.		

Fuente: Autoría

En la figura 29 se muestran los resultados obtenidos en la prueba correspondiente a la tabla 20:

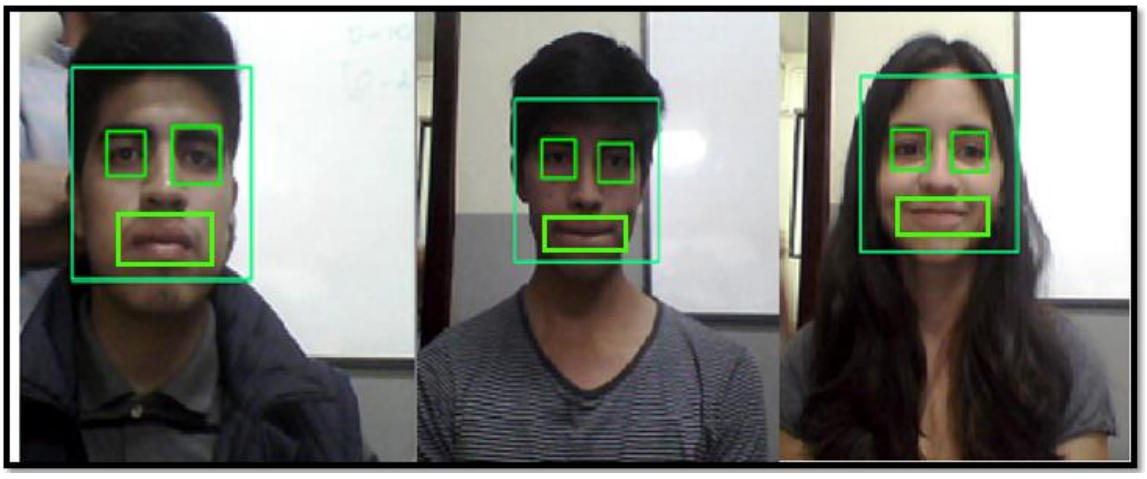


Figura 29. Prueba Bloque 4, Día 1

Fuente:Autoría

En la figura 30 se presenta un diagrama de dispersión que permite ver la relación entre el número de pestañeos y el número de bostezos obtenidos el primer día de pruebas del Bloque 4. En la gráfica el eje X representa la escala de tiempo en que se ha realizado el monitoreo, mientras que el eje Y es el encargado de mostrar los valores de pestañeos y de bostezos.

Se observa que para cada valor de pestañeo existe un valor de bostezo en el mismo instante de tiempo, esto ocurre a que el sistema almacena un valor de cada contador por minuto.

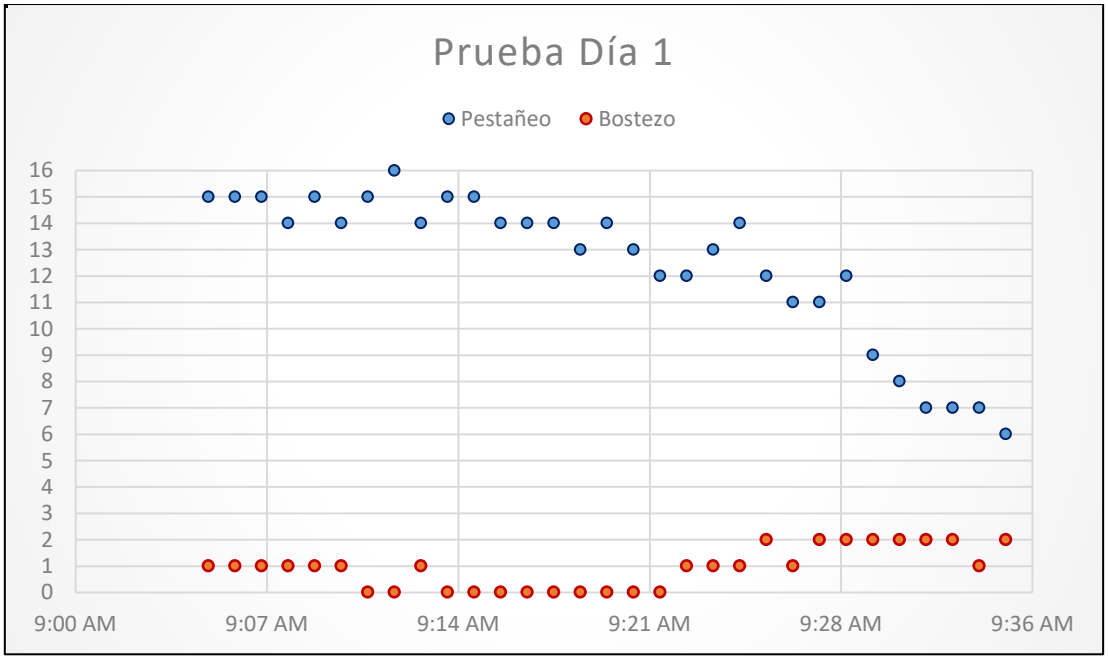


Figura 30. Datos obtenidos prueba Día 1

Fuente: Autoría

Con el mismo formato detallado en la tabla anterior, se presentan los resultados obtenidos en el segundo día de desarrollo de las pruebas del Bloque 4, el detalle de esta prueba se observa en la tabla 22:

Tabla 22. Descripción de la prueba del Bloque 4, Día 2

PRUEBA BLOQUE 4: SISTEMA DE MONITOREO FACIAL		
Día: Martes	Localización:	Universidad Técnica del Norte, Edificio FICA, Laboratorio Fibra Óptica
	Sujetos de prueba:	Estudiantes de CIERCOM de sexto semestre
	Hora en que se realizó el monitoreo:	Durante la mañana que es el periodo académico de los estudiantes
	Tiempo de monitoreo:	10 minutos cada estudiante
Observaciones: En esta prueba hubo un estudiante que usaba lentes, a pesar de que la detección de rostro fue inmediata la detección de ojos tomó alrededor de 1 minuto en funcionar correctamente. Estas irregularidades serán corregidas para las pruebas finales.		

Fuente: Autoría

Los resultados de la prueba del Bloque 4 en el segundo día se visualizan en la figura 31:

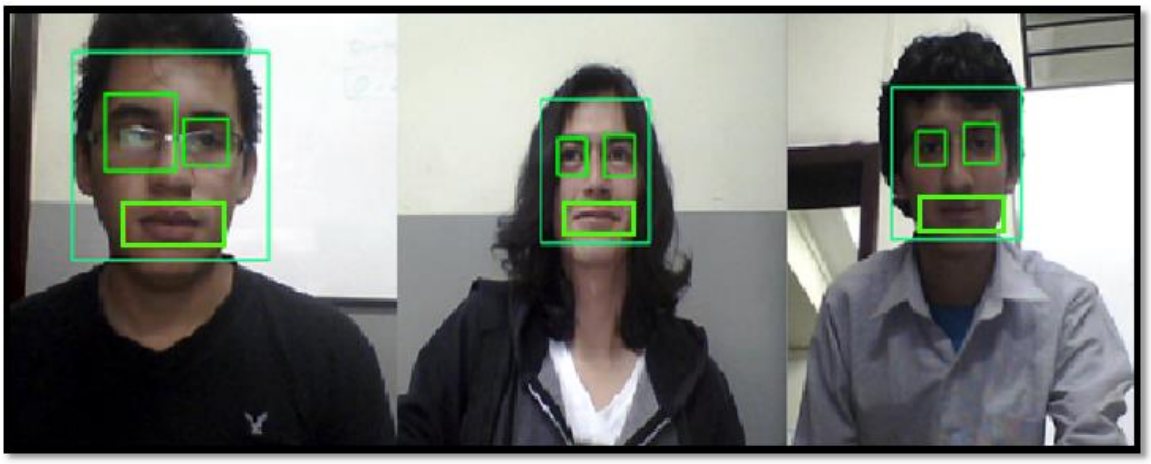


Figura 31. Prueba Bloque 4, Día 2

Fuente: Autoría

Con el formato señalado en el diagrama de dispersión anterior se presenta en la figura 32 los datos correspondientes a la prueba del Bloque 4 del segundo día:

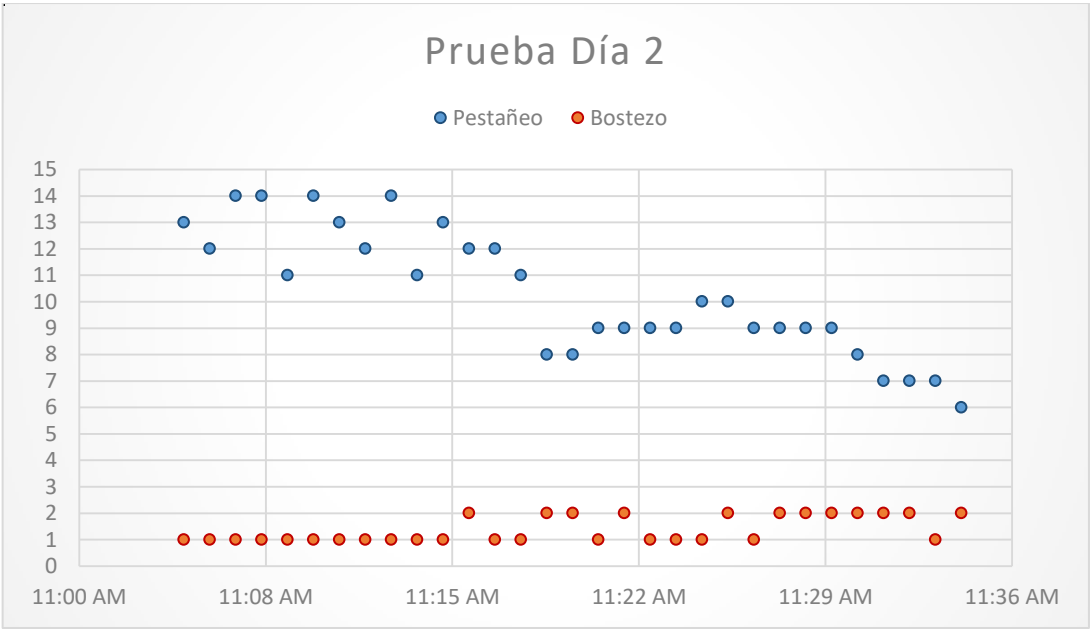


Figura 32. Datos obtenidos prueba Día 2

Fuente: Autoría

Continuando con la fase de pruebas se presenta en la tabla 23 con la descripción del tercer día de pruebas del sistema de monitoreo facial:

Tabla 23. Descripción de la prueba del Bloque 4, Día 3

PRUEBA BLOQUE 4: SISTEMA DE MONITOREO FACIAL		
Día: Miércoles	Localización:	Universidad Técnica del Norte, Edificio FICA, Laboratorio Fibra Óptica
	Sujetos de prueba:	Estudiantes de CIERCOM de distintos semestres
	Hora en que se realizó el monitoreo:	Durante la mañana que es el periodo académico de los estudiantes
	Tiempo de monitoreo:	10 minutos cada estudiante
Observaciones: En esta prueba se contó con dos estudiantes que usaron lentes, sin embargo la detección de rostro y ojos fue inmediata.		

Fuente: Autoría

La figura 33 presenta los resultados del monitoreo a los estudiantes de CIERCOM en el día 3:

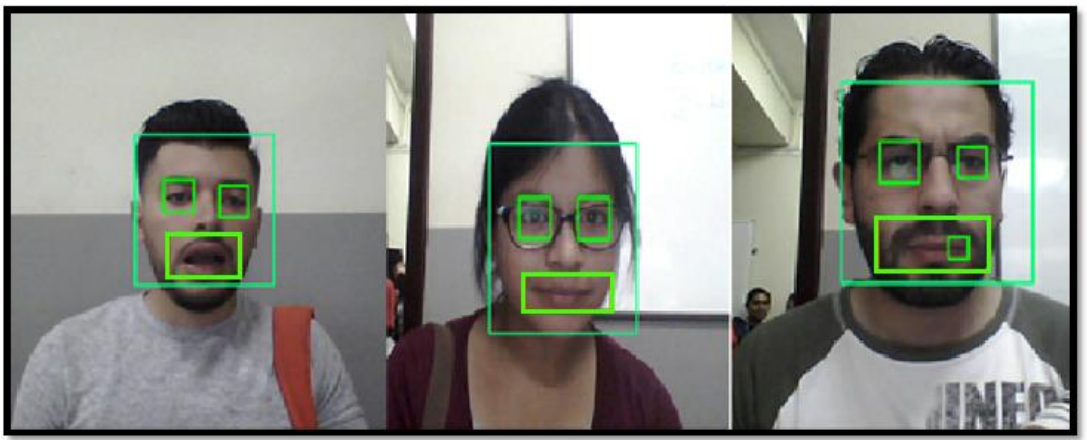


Figura 33. Prueba Bloque 4, Día 3

Fuente: Autoría

Siguiendo el formato señalado en los diagramas de dispersión anteriores se presenta en la figura 34 los datos de la prueba del Día 3:

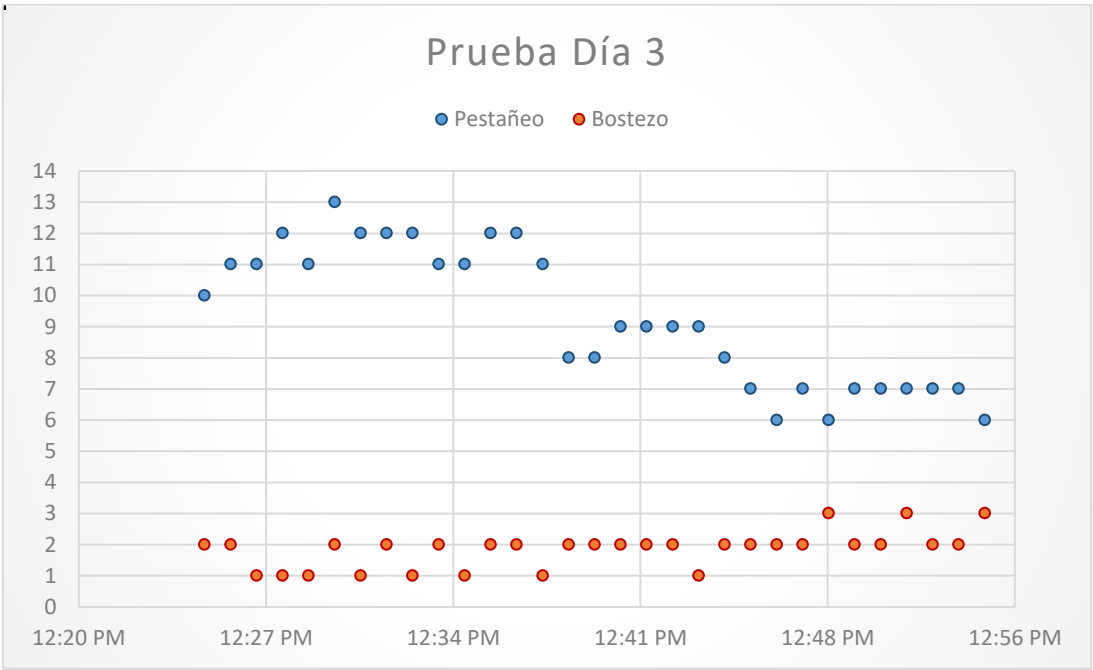


Figura 34. Datos obtenidos prueba Día 3

Fuente: Autoría

La tabla 24 indica la información correspondiente al día 4 de las pruebas del sistema de monitoreo facial:

Tabla 24. Descripción de la prueba del Bloque 4, Día 4

PRUEBA 4: SISTEMA DE MONITOREO FACIAL		
Día: Jueves	Localización:	Universidad Técnica del Norte, Edificio FICA, Laboratorio Fibra Óptica
	Sujetos de prueba:	Estudiantes de CIERCOM de sexto semestre
	Hora en que se realizó el monitoreo:	Durante la mañana que es el periodo académico de los estudiantes
	Tiempo de monitoreo:	10 minutos cada estudiante
Observaciones: La detección funcionó correctamente en todas las pruebas		

Fuente: Autoría

Los resultados de la figura 35 muestran la detección de rostro, ojos y boca de los estudiantes monitoreados el día 4.

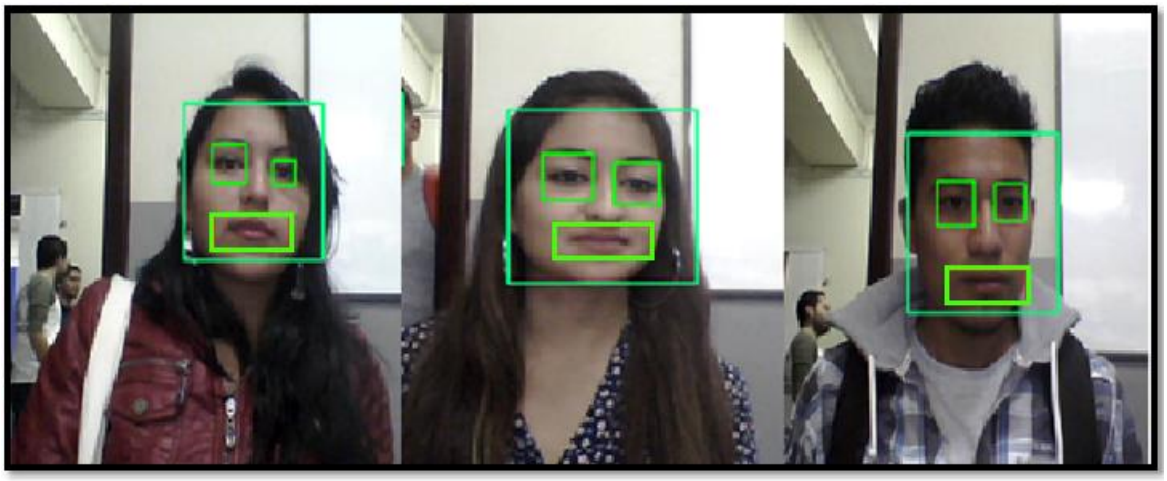


Figura 35. Prueba Bloque 4, Día 4

Fuente: Autoría

El diagrama de dispersión que se presenta en la figura 36 muestra los datos obtenidos en la prueba del Bloque 4, día 4, esta figura conserva el formato anteriormente.

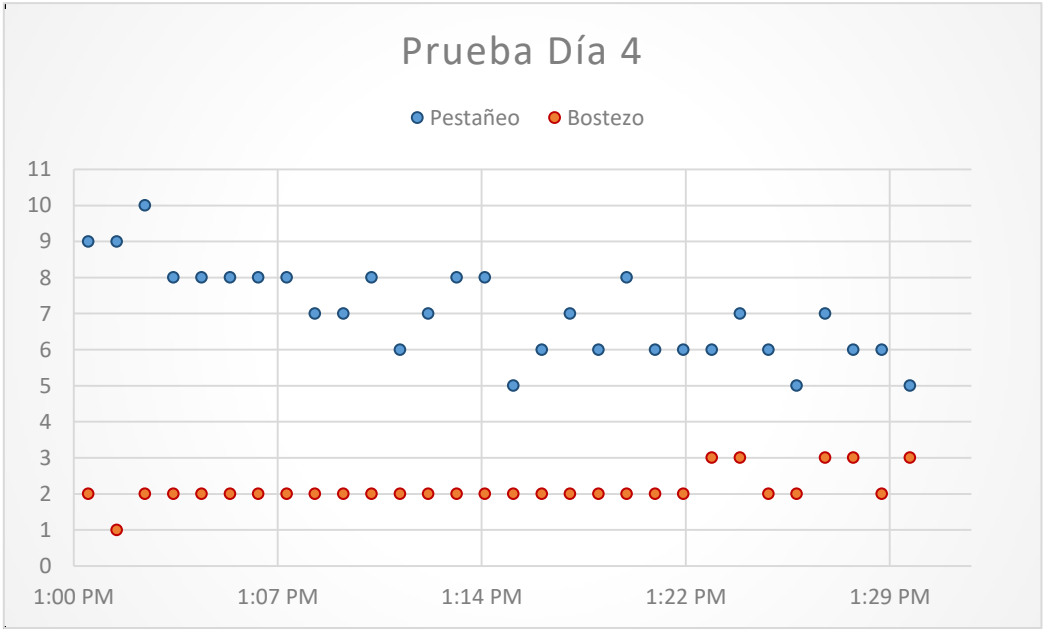


Figura 36. Datos obtenidos en la prueba Día 4

Fuente: Autoría

La tabla 25 presenta la información del día 5 de la fase de pruebas del Bloque 4:

Tabla 25. Descripción de la prueba del Bloque 4, Día 5

PRUEBA 4: SISTEMA DE MONITOREO FACIAL		
Día: Viernes	Localización:	Universidad Técnica del Norte, Edificio FICA, Laboratorio Fibra Óptica
	Sujetos de prueba:	Estudiantes de CIERCOM de sexto semestre
	Hora en que se realizó el monitoreo:	Durante la mañana que es el periodo académico de los estudiantes
	Tiempo de monitoreo:	10 minutos cada estudiante
Observaciones: La detección funcionó correctamente en todas las pruebas		

Fuente: Autoría

La detección de factores fisiológicos del día 5 de la fase de pruebas preliminares se observan en la figura 37:

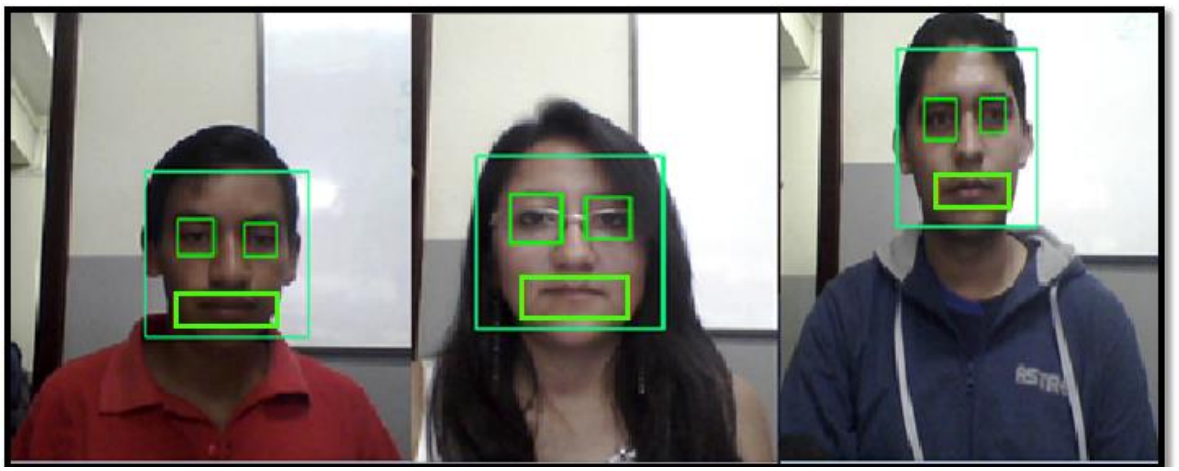


Figura 37. Prueba preliminar día 5

Fuente: Autoría

El diagrama de dispersión correspondiente al último día de pruebas del Bloque 4 se presenta en la figura 38.

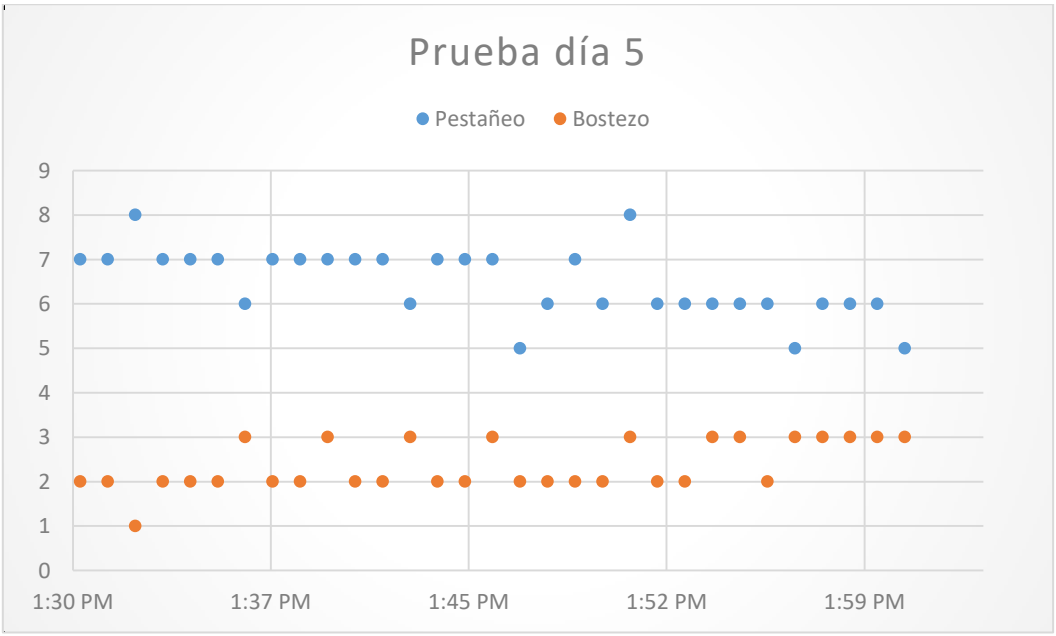


Figura 38. Datos obtenidos en la prueba Día 5

Fuente: Autoría

Para finalizar la etapa de pruebas de cada bloque, se procedió a realizar las pruebas de visualización de datos en la plataforma Ubidots, las gráficas mostradas en la plataforma son las que se detallan en la figura 39 y figura 40 con el contador de pestañeos y el contador de bostezos respectivamente.

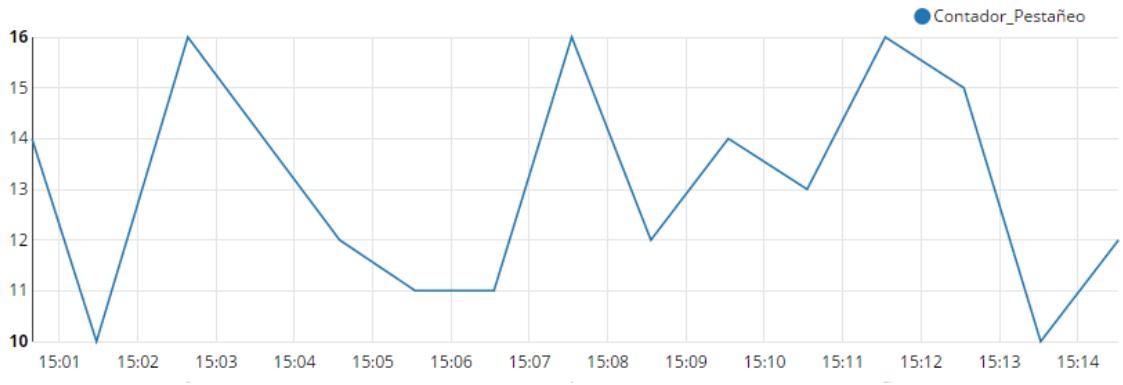


Figura 39. Contador de Pestañeo, gráfica de Ubidots

Fuente: Autoría

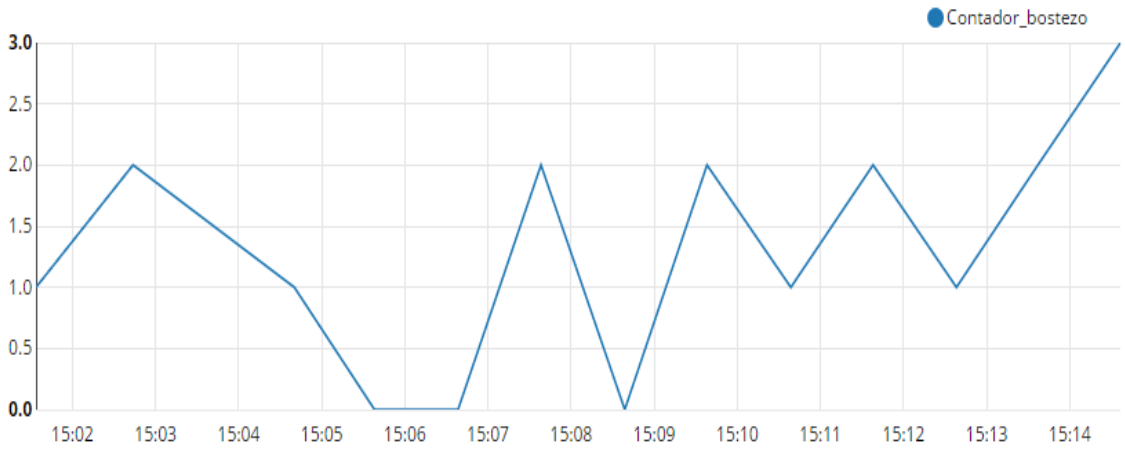


Figura 40. Contador de bostezo, gráfica de Ubidots

Fuente: Autoría

Al emplear una versión gratuita de Ubidots no se cuenta con todas las herramientas de representación de datos por lo cual al visualizar el Dashboard de los datos solo se pueden emplear gráficas con líneas que en ocasiones no permiten concebir claramente la relación que

guardan los datos, por esta razón se puede optar por descargar los archivos .csv que contienen los datos subidos a la nube y éstos pueden ser analizados empleando la herramienta Microsoft Excel. En la figura 41 se presenta el Dashboard generado en la plataforma Ubidots.

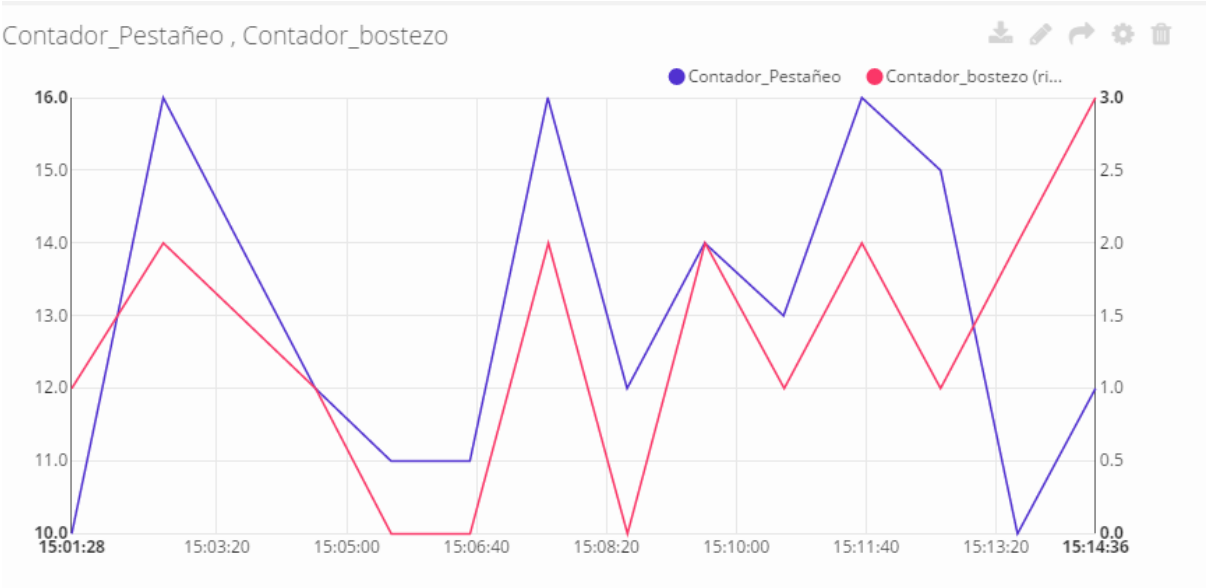


Figura 41. Dashboard generado en la plataforma Ubidots

Fuente: Autoría

4.1.2 Pruebas Finales del Sistema

Las pruebas finales del sistema fueron realizadas con estudiantes de séptimo semestre de CIERCOM, en este monitoreo se tomaron datos durante todo el periodo de clases comprendido en dos horas, de este monitoreo se obtuvo una base de datos con más de 1000 datos de contadores de pestañeo y bostezo. A continuación, en la figura 42 se muestra un diagrama de dispersión que contiene en el eje vertical los datos de los contadores, en este caso no se ha incluido el eje de tiempo debido a que el monitoreo se realizó en distintas horas del día, sin embargo se puede observar que para cada valor de pestañeo existe su respectivo valor de bostezo.

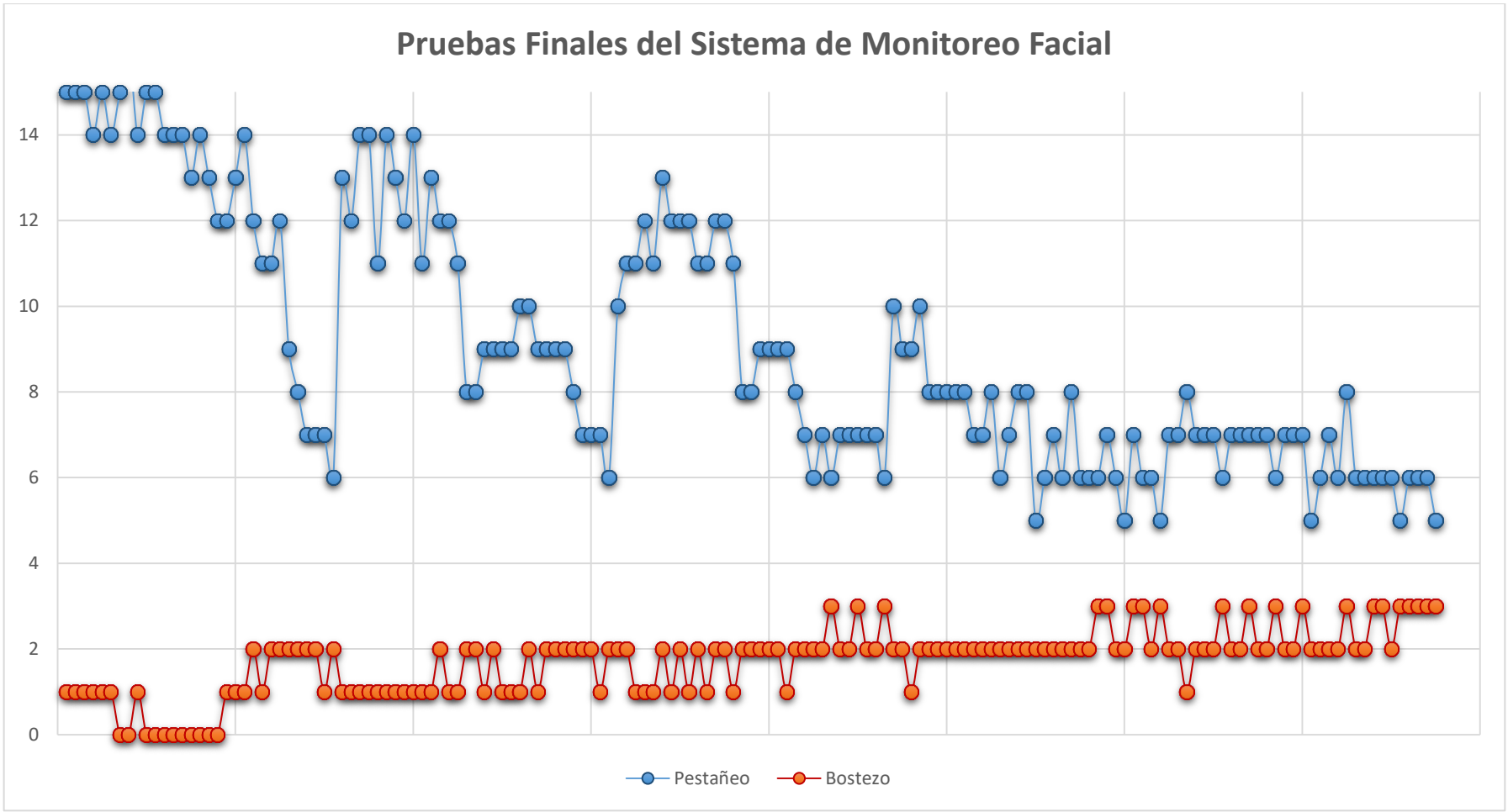


Figura 42. Diagrama de dispersión de datos de las pruebas finales del sistema

Fuente: Autoría

4.1.3 Análisis de resultados obtenidos en las pruebas

En los datos obtenidos predominó una tendencia que indica que en los horarios de 9h00 a 12h00 horas del día la distancia entre el número de pestañeos y el número de bostezos es mayor, conforme transcurren los periodos académicos a partir de las 12h30 se observó que el contador de bostezos aumentaba y el de pestañeos disminuía, por lo tanto la distancia entre las dos variables se reduce. En el monitoreo a partir de las 12h30 no se registran valores mayores a 10 pestañeos y el valor de los bostezos sube hasta 3, lo que muestra claramente como una muestra de cansancio, los valores críticos fueron los obtenidos a partir de la 13h00 donde el cansancio se hizo más evidente hasta el hecho de perder por completo la concentración. El análisis de estos resultados se basa en las figuras 30,32,34,36,38 y se confirmó con la gráfica de resumen presentada en la figura 42.

La fase de pruebas preliminares y finales permitió conocer los posibles errores del sistema y como solucionarlos, también mostró la efectividad de detección de los rasgos fisiológicos de los estudiantes monitoreados, pero principalmente demostró que existe una tendencia de desconcentración mayor a partir de las 12h30, horario que podría ser reestructurado para evitar la falta de concentración en los alumnos.

4.2 Conclusiones

- Se desarrolló un sistema electrónico con aplicación IoT para el monitoreo facial que brinda estimadores de desconcentración de estudiantes universitarios dentro del aula, empleando herramientas de visión artificial y desarrollado sobre una placa Raspberry Pi 2.

- Se obtuvieron las bases teóricas y conceptos necesarios relacionados principalmente a Internet de las Cosas y Visión Artificial, los fundamentos teóricos adquiridos respaldaron los criterios de diseño establecidos para el sistema de monitoreo facial.
- Se encontró que las principales causas de desconcentración de los estudiantes de CIERCOM del semestre Abril-Agosto de 2016 están relacionados a la falta de horas de descanso, estos resultados sustentaron la realización de este proyecto que permite el monitoreo de desconcentración de los estudiantes dentro del aula de clases.
- Se empleó la norma IEEE 29148 como directriz para conocer los requerimientos de arquitectura, de software y hardware y requerimientos de usuarios necesarios para el desarrollo del sistema de monitoreo facial
- Se seleccionó el hardware y software adecuados para el sistema de monitoreo facial, basados en los requerimientos del sistema y en un análisis de los productos existentes en el mercado, consiguiendo como resultado que el Raspberry Pi 2 Modelo B, el módulo Pi Camera v1, el software OpenCV y Python fueron las opciones idóneas que permitieron el correcto desarrollo del sistema.
- Se diseñó el sistema de monitoreo facial empleando herramientas de visión artificial que permitieron detectar factores fisiológicos como pestañeo y bostezo. El sistema se basó en clasificadores tipo Haar proporcionados por OpenCV que permitieron la detección de rostro, ojos y boca.
- Se realizó una fase de pruebas para determinar el correcto funcionamiento del sistema, para ello se monitoreó a varios estudiantes de CIERCOM de la Universidad Técnica del Norte, comprobando que la detección rostro, ojos y boca fue exitosa y los contadores de pestañeo y bostezo funcionaron satisfactoriamente.
- Se utilizó la plataforma libre Ubidots para subir los datos adquiridos por el sistema de monitoreo facial a la Internet donde pueden ser visualizados y analizados por los

usuarios, permitiéndoles conocer los estimadores de desconcentración de los estudiantes durante los distintos periodos académicos.

- Durante la fase de pruebas finales se recolectaron datos que indicaron una mayor tendencia de desconcentración de los estudiantes en los horarios que comprenden a partir de las 12 pm, y teniendo la tendencia más baja en los horarios de 8 am a 11 am.

4.3 Recomendaciones

- El sistema de monitoreo facial debe ser ubicado dentro de las aulas de clase en un lugar donde no interfiera con las actividades normales tanto de los estudiantes como del docente evitando cualquier molestia en el periodo académico.
- Se recomienda que la cámara tenga movilidad permitiendo monitorear a diferentes estudiantes a la vez, sin necesidad de estar haciendo ajustes a la ubicación, para ello se podrían implementar servos motores que permitan enfocar en diferentes ángulos y adquirir mayor número de imágenes para ser procesadas.
- Se recomienda adquirir la versión pagada de Ubidots debido a que esta proporciona mejores representaciones estadísticas y una mejor manera de visualizar y relacionar los datos obtenidos.
- Para el desarrollo de aplicaciones de visión artificial se recomienda emplear la herramienta OpenCV debido a que cuenta con más de 500 funciones para la detección de objetos, colores y características específicas, además de ser compatible con varios lenguajes de programación permitiendo el desarrollo de robustos sistemas de seguridad, control de calidad y rastreo de personas, entre los principales.
- Es recomendable el uso de la placa Raspberry Pi para el desarrollo de sistemas electrónicos, debido a que cuentan con características de procesamiento y

almacenamiento de datos, además de sus distintos periféricos que pueden convertirlo en un mini computador portátil cuenta con un sistema operativo intuitivo y que permite que se desarrollen desde aplicaciones sencillas hasta robustos sistemas de información.

BIBLIOGRAFÍA

Alwakeel, S., Alhalabi, B., Aggoune, H., & Alwakeel, M. (2015). A Machine Learning Based WSN System for Autism Activity Recognition. *2015 IEEE 14th International Conference on Machine Learning and Applications*, (pp. 771-776). Miami, FL, USA .

Ansari, A. N., Sedky, M., Sedky, M., & Tyagi, A. (2015). An Internet of things approach for motion detection using Raspberry Pi. *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things, ICIT 2015*, (pp. 131-134).

Carriots. (2016). *Carriots*. Retrieved from https://www.carriots.com/

Cazorla, M. (n.d.). Robótica y Visión Aritificial. Universidad de Alicante.

Cisco IBSG. (2011). *Internet de las cosas, Cómo la próxima evolución de Internet lo cambia todo.*

Collina, M., Bartolucci, M., Vanelli, A., & Corazza, G. (2014). Internet of Things Application Layer Protocol Analysis over Error and Delay prone Links. *7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop*, (pp. 398-404).

Deepthi , R., & Sankaraiah, S. (2011). Implementation of mobile platform using Qt and OpenCV for image processing applications. *IEEE Conference on Open Systems (ICOS2011)*, (pp. 284 - 289). Langkawi, Malaysia.

Deng, L., & Li, X. (2013). Machine Learning Paradigms for Speech Recognition: An Overview. *IEEE Transactions on Audio, Speech, and Language Processing*, 1060 - 1089.

Dlodlo, N., & Kalezhi, J. (2015). The internet of things in agriculture for sustainable rural development. *Emerging Trends in Networks and Computer Communications (ETNCC), 2015 International Conference on*, (pp. 13-18).

Fan, Z., Chen, Q., Kalogridis, G., Tan, S., & Kaleshi, D. (2012). The Power of Data: Data Analytics for M2M and Smart Grid. *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*, (pp. 1-8). Berlin.

Fernández García, N. L. (n.d.). *Introducción a la Visión Artificial.* Córdoba: Universidad de Córdoba.

Fisher, & Navarro. (1994).

García, M. E. (2009). *Diseño e implementación de una herramienta de detección facial.* México.

GNU & Free Software Foundation. (n.d.). *El sistema operativo GNU*. Retrieved from http://www.gnu.org

González Castellanos, R. (2000). *Principios Básicos de Escalado.*

Häkkänen, H., Summala, H., Partinen, M., Tiihonen, M., & Silvo, J. (1999). Blink Duration as an Indicator of Driver Sleepiness in Professional Bus Drivers. *SLEEP, 22*(6), 798-802.

INEC. (2013). *Mujeres y Hombres del Ecuador en cifras III.*

ISO/IEC/IEEE. (2011). *Systems and software engineering -- Life cycle processes -- Requirements engineering.* Switzerland.

Itseez. (n.d.). *OpenCV*. Retrieved from OpenCV: http://opencv.org/

Jiménez Encalada, J. C. (2015). *Implementación de técnicas de identificación de objetos aplicadas al reconocimiento de rostros en video vigilancia del SIS-ECU-911*. Cuenca.

Kim, J., Lee, J., Kim, J., & Yun, J. (2014). M2M Service Platforms: Survey, Issues, and Enabling Technologies. *IEEE Communications Surveys & Tutorials*, pp. 61 - 76.

Loureiro, R. (2015, Junio 12). Retrieved from http://openaccess.uoc.edu/webapps/o2/bitstream/10609/42812/6/rloureiroTFC0615memoria.pdf

Luzuriaga, J., Cano, J. C., Calafate, C., Manzoni, P., Pérez, M., & Boronat, P. (2015). Handling Mobility in loT applications using the MQTT protocol . *Internet Technologies and Applications (ITA)*, (pp. 245 - 250). Wrexham, United Kingdom.

Marengoni, M., & Stringhini, D. (2011). High Level Computer Vision using OpenCV. *24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials*, (pp. 11-24).

Miró, E., Iáñez, M. Á., & Cano-Lozano, M. C. (2002). Patrones de sueño y salud. *International Journal of Clinical and Health Psychology*, 301-326.

MQTT. (n.d.). *MQTT.org*. Retrieved from MQTT.org: http://mqtt.org

Muchnik, S., Finkielman, S., Seemeniuk, G., & de Aguirre, M. (n.d.). Retrieved from http://www.alfinal.com/neurofisiologia/boztezo.php

Open Source Hardware Association. (n.d.). *OSHWA*. Retrieved from http://www.oshwa.org

Open Source Initiative. (n.d.). *Open Source Initiative*. Retrieved from https://opensource.org/

OpenCV documentation. (2014). *OpenCV*. Retrieved from OpenCV: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html

Organización Mundial de la Salud. (2014). *Organización Mundial de la Salud*. Retrieved from http://www.who.int/childgrowth/standards/height_for_age/es/

Plauska, I., & Damaševičius, R. (2014). Educational Robots for Internet-of-Things Supported Collaborative Learning. *Springer International Publishing Switzerland 2014*, 346–358,.

Python Software Foundation. (n.d.). *python*. Retrieved from python: https://www.python.org/

Raspberry Pi Foundation. (n.d.). *Raspberry Pi Foundation*. Retrieved from https://www.raspberrypi.org/

Raspbian.org. (n.d.). *Raspbian*. Retrieved from https://www.raspbian.org/

Robologs. (2014, Mayo 26). Retrieved from http://robologs.net/2014/05/26/reconocimiento-facial-con-opencv-webcam/

Servicios de Prevención de Riesgos Laborales (SEPRUMA). (2004, Diciembre). Principales Requisitos de diseño para evitar los problemas musculoesqueléticos en las personas que realizan trabajos en oficinas y despachos. Málaga, España.

Standford University, Cousera. (n.d.). *Coursera*. Retrieved from Coursera: https://www.coursera.org/learn/machine-learning

Tianbo, Z. (2012). The Internet of Things Promoting Higher Education Revolution. *Fourth International Conference on Multimedia Information Networking and Security (MINES),*, (pp. 790-793). Nanjing, China .

Vandna Saini, & Rekha Siani. (2014). Driver Drowsiness Detection System and Techniques: A Review. *International Journal of Computer Science and Information Technologies*, 4245-4249.

Vera Meaurio, D. A., & Martínez Jara, E. A. (2016). Conteo de personas en imagen y video mediante la técnica de Viola-Jones a través de clasificadores Haar utilizando software libre. *FPUNE Scientific*.

Vivanco, M. (2005). *Muestreo Estadístico Diseño y Aplicaciones.* Santiago de Chile: Editorial Universitaria S.A.

Vivas, A. (n.d.). *Desarrollo de un sistema de reconocimiento facial.*

Zhao, W., Chellappa, R., Rosenfeld, A., & Phillips, P. (2003). *Face Recognition: A Literature Survey.*

ANEXOS

ANEXO 1: Formato de encuesta aplicada a los estudiantes de CIERCOM de la UTN

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CIERCOM

ENCUESTA DIRIGIDA A ESTUDIANTES DE 4to A 9no SEMESTRE

El objetivo de la presente encuesta es determinar las posibles causas de la falta de concentración de los estudiantes dentro del aula de clases.

- Marque con una X el semestre que se encuentra cursando:

☐ Cuarto ☐ Quinto ☐ Sexto

☐ Séptimo ☐ Octavo ☐ Noveno

- Marque con una X el número de créditos que usted tiene este semestre:

☐ Mayor a 28 créditos ☐ de 20 a 28 créditos

☐ de 10 a 20 créditos ☐ menos de 10 créditos

- Indique. En un día normal ¿cuántas horas dedica a estudiar?

___ horas

- Indique. En un día normal ¿cuántas horas dedica a descansar?

___ horas

¿Cuáles cree usted que son causas de la falta de concentración durante sus horas clase?

☐ Falta de horas de descanso

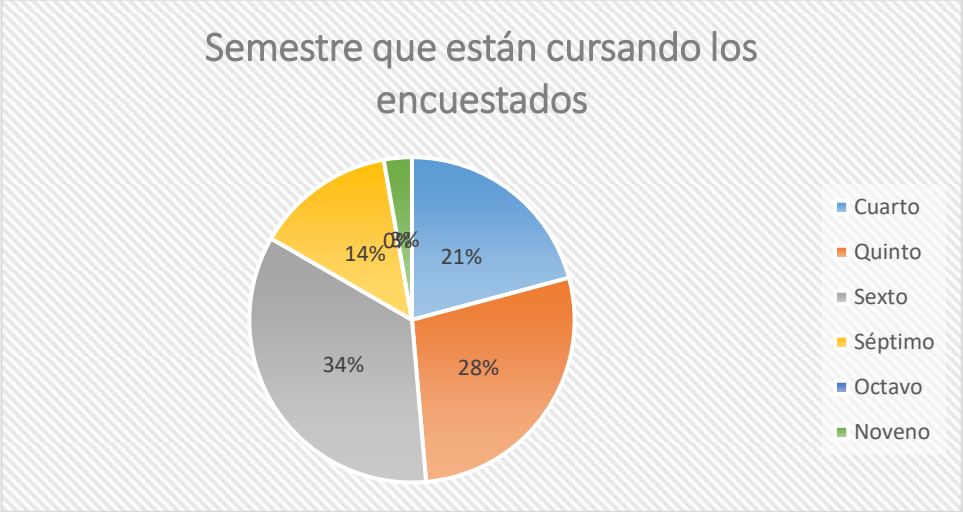
☐ Horarios inadecuados

☐ Metodologías de enseñanza inapropiadas

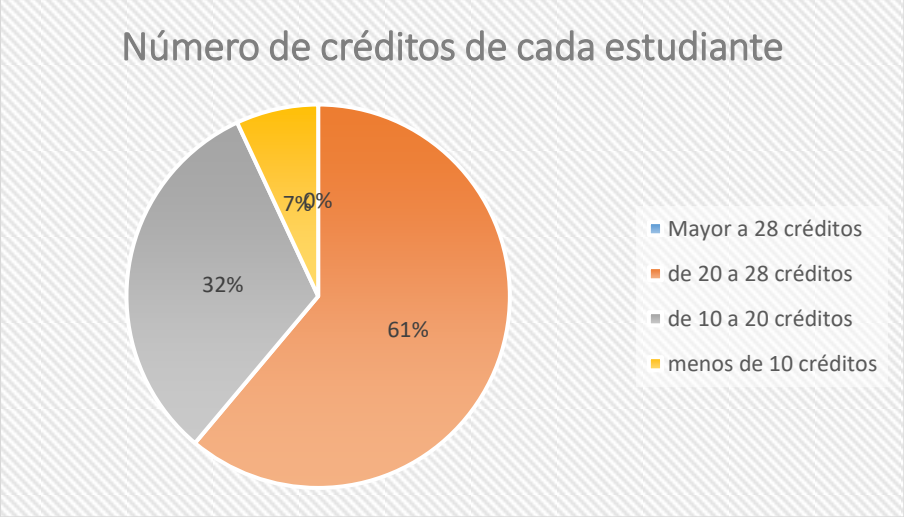
☐ Falta de interés en la materia

ANEXO 2: Tabulación de resultados de encuestas aplicadas a estudiantes de CIERCOM

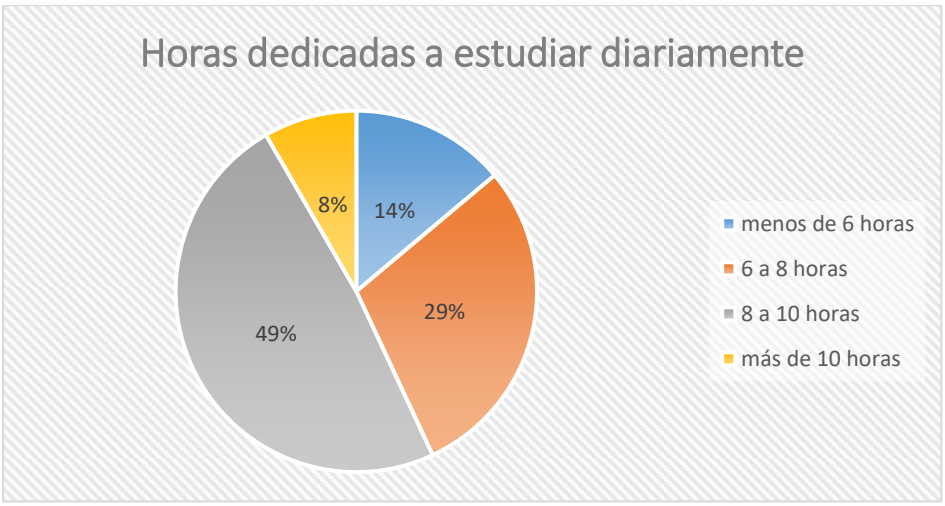
En la primera pregunta realizada se pidió a los encuestados que identifiquen el semestre que se encuentran cursando, en este caso el 34% de la muestra se encuentra cursando 6to semestre, siguiendo con el 28% que cursa quinto semestre, un 21% corresponde a cuarto semestre, un 14% a séptimo semestre y finalmente un 3% que se encuentra cursando 9no semestre. Estos resultados se muestran en el siguiente gráfico.



Continuando con la encuesta, los estudiantes indicaron el número de créditos en los que están matriculados el semestre Abril-Agosto 2016, la razón de esta pregunta se debe a que dependiendo del número de créditos varía el tiempo que el estudiante dedica tanto a actividades académicas como actividades de descanso. En este caso se obtuvo que un 61% de encuestados tienen un número de créditos entre 20 y 28, un 32% de estudiantes tienen entre 10 y 20 créditos, un 7% tienen más de 28 créditos y un 1% de la muestra tiene menos de 10 créditos.

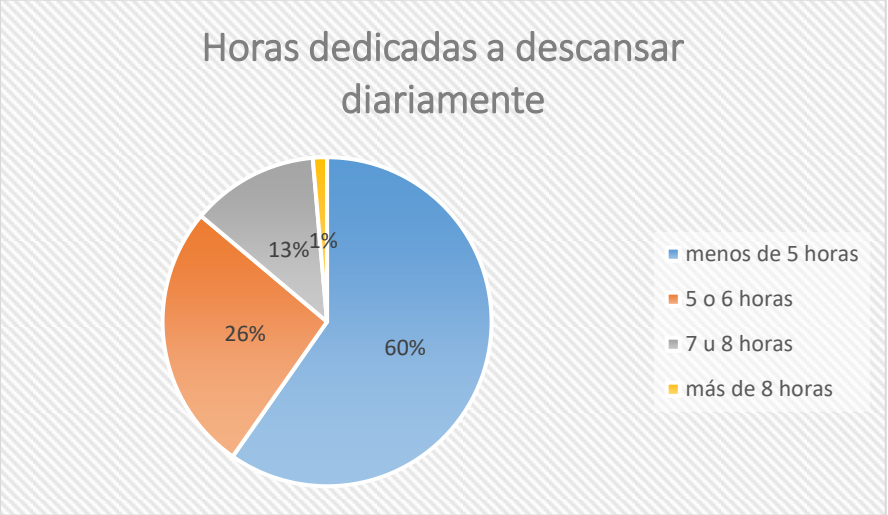


La siguiente pregunta que se realizó fue respecto a las horas que los estudiantes dedican a estudiar cada día, en un día normal el 49% de estudiantes dedica de 8 a 10 horas a estudiar, el 29% de encuestados respondió que dedica entre 6 u 8 horas, el 14% indicó que dedica más de 10 horas diarias y un 8% dedica menos de 6 horas a sus actividades académicas.

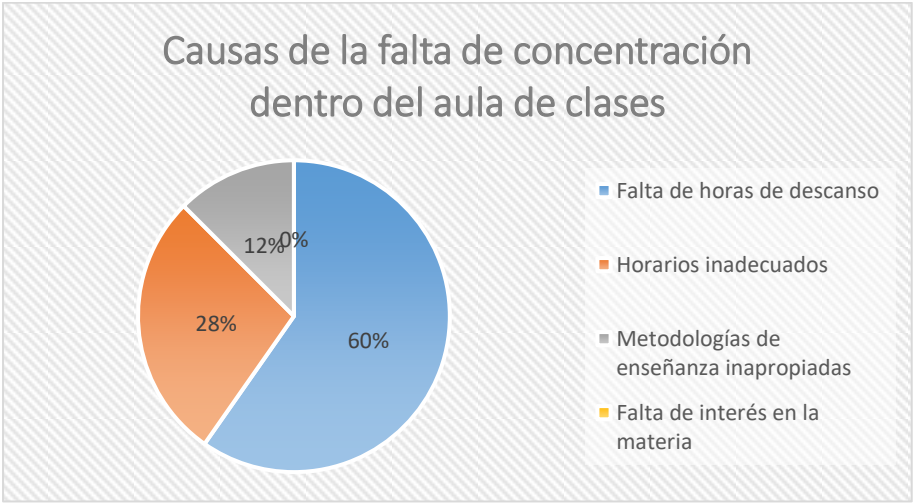


Una vez examinadas las horas de estudio se procedió a analizar las horas dedicadas a descansar de las que disponen los estudiantes obteniendo que un 60% de la muestra dedica

menos de 5 horas para descansar, un 26% dedica entre 5 y 6 horas, un 13% entre 7 u 8 y tan solo 1% de los estudiantes dedica más de 8 horas para descansar.



Finalmente se preguntó a los encuestados cuáles consideran que son las causas que generan la falta de concentración dentro del aula de clases a lo que respondieron de la siguiente manera: un 60% identificó como causa la falta de horas de descanso dentro de su rutina diaria, un 28% indicó que los horarios inadecuados pueden ser la razón de la falta de concentración, un 12% indicó que las metodologías de enseñanza inapropiadas y ningún encuestado consideró que la falta de interés en la materia sea un motivo para la falta de concentración.



ANEXO 3: Datasheet Raspberry Pi Camera Module V

Raspberry Pi

CAMERA MODULE

Product Name	**Raspberry Pi Camera Module**
Product Description	High definition camera module compatible with the Raspberry Pi model A and model B. Provides high sensitivity, low crosstalk and low noise image capture in an ultra small and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector designed specifically for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor.
RS Part Number	**775-7731**
Specifications	
Image Sensor	Omnivision 5647 CMOS image sensor in a fixed-focus module with integral IR filter
Resolution	5-megapixel
Still picture resolution	2592 x 1944
Max image transfer rate	1080p: 30fps (encode and decode) 720p: 60fps
Connection to Raspberry Pi	15 Pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2)
Image control functions	Automatic exposure control Automatic white balance Automatic band filter Automatic 50/60 Hz luminance detection Automatic black level calibration
Temp range	Operating: -30° to 70° Stable image: 0° to 50°
Lens size	1/4"
Dimensions	20 x 25 x 10mm
Weight	3g

Accessories

▲ Raspberry Pi Model B - **756-8308**

▲ Camera case **784-6193**

▲ 8GB SD card pre-programmed with NOOBS - **779-6770**

▲ Expansion board **772-2974**

▲ WiFi dongle **760-3621**

▲ 10400mAh Li-Ion battery pack **775-7517**

▲ Raspberry Pi user guide **768-6686**

RS

www.rs-components.com/raspberrypi