

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Diseño e Implementación de un prototipo a bajo costo para la detección y alarma en ciclovías con intersecciones urbanas no controladas.

PROYECTO INTEGRADOR

Previo la obtención del Título de:

INGENIERO EN TELECOMUNICACIONES

Presentado por:

ALARCÓN TOMALÁ DANIEL ALEJANDRO

TORRES ORTEGA NICOLÁS MIGUEL

GUAYAQUIL - ECUADOR

2022

DEDICATORIA

Le dedicamos el presente proyecto a nuestras familias que nos han brindado su amor y apoyo incondicional ante las situaciones adversas que hemos vivido durante esta etapa universitaria. Sus consejos, guías y sacrificios han permitido que siempre avancemos en la vida, que nuestros anhelos y sueños no se pierdan, que ante las inconformidades e injusticias desviemos el coraje, la culpa o la desesperanza y lo transformemos en carácter y templanza. Por tal motivo es justo y meritorio que este logro que hemos alcanzado sea adjudicado a ellos.

Daniel Alejandro Alarcón Tomalá

Nicolás Miguel Torres Ortega

AGRADECIMIENTOS

Agradecemos primeramente a Dios, por permitirnos haber llegado hasta este momento tan importante de nuestra formación profesional, a nuestras familias por su apoyo incondicional, especialmente a nuestros padres, por ser los principales promotores de nuestros sueños, apoyándonos en los buenos y malos momentos. Finalmente, a nuestros amigos que nos han acompañado durante esta etapa, aportando a nuestra formación tanto profesional y ser humano.

Daniel Alejandro Alarcón Tomalá

Nicolás Miguel Torres Ortega

DECLARACIÓN EXPRESA

“Según los derechos de titularidad y de divulgación, conforme al reglamento de propiedad intelectual de la institución; **DANIEL ALEJANDRO ALARCÓN TOMALÁ** y **NICOLÁS MIGUEL TORRES ORTEGA** damos nuestro consentimiento para que la ESPOl realice la comunicación pública de la presente documentación por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

**DANIEL ALEJANDRO
ALARCÓN TOMALÁ**

**NICOLÁS MIGUEL
TORRES ORTEGA**

EVALUADORES

M.Sc. Washington Medina Moreira
PROFESOR DE LA ASIGNATURA



M.Sc. Félix Moncayo Rea
TUTOR DE LA ASIGNATURA

RESUMEN

Este proyecto consiste en diseñar e implementar un prototipo a bajo costo para la detección y alarma en ciclovías con intersecciones no controladas, mediante el despliegue de un conjunto de sensores controlados por unidades programables, con el fin de promover el respeto a los usuarios que hacen uso de la ciclovía y fomentar una participación ciudadana segura.

El desarrollo del proyecto se basa en la detección de los usuarios por medio de sensores ópticos LIDAR encargados de determinar parámetros físicos de distancias en los usuarios detectados y de esta forma, obtener la velocidad y tiempo que toma un ciclista en llegar a un cruce con la ciclovía a fin de habilitar una alarma visual, dirigida a los vehículos, como mecanismo de prevención.

Además, se incorpora un sistema de conteo de usuarios para obtener y presentar información del tráfico de usuarios que ha circulado durante el día. Este sistema establece una conexión con un servidor IoT para el registro de tránsito y generación de notificaciones de alerta en la ciclovía.

Como resultados del proyecto se obtuvo, un factor de calibración en el sensor que proporciona un error relativo del 2.33% con las distancias que recolecta el sensor LIDAR, una distancia máxima de detección de usuarios de 15m y el tiempo estimado en el que un usuario estará próximo un cruce. Por otra parte, se presentó el tráfico de usuarios que circulan en la ciclovía por medio de una interfaz gráfica, en donde dicha información se envió y almacenó en el servidor web de Ubidots.

Con los resultados obtenidos se pudo validar una correcta calibración del prototipo, obteniendo con éxito los parámetros físicos de los ciclistas para la activación de una alarma visual de prevención, cuando exista la proximidad de usuarios a una intersección. Además, el registro del tráfico de usuarios en un servidor permite evaluar el impacto social que conlleva la implementación de ciclovías.

Palabras claves: sensor, distancia, LIDAR, IoT, intersecciones

ABSTRACT

This project consists of designing and implementing a low-cost prototype for the detection and alarm of uncontrolled intersections in bike paths, through the deployment of a set of sensors controlled by programmable units, so that promote respect for users who make use of the bike paths and encourage safe citizen participation.

Project development is based on the detection of users by means of LIDAR optical sensors responsible for determining physical distance parameters of the detected users and thus obtain the speed and time it takes a cyclist to reach an intersection with the bike path to enable a visual alarm, aimed at the vehicles, as a preventive mechanism.

In addition, a user's counting system is incorporated to obtain and show information on user traffic that has circulated during the day. This system establishes a connection with an IoT server to recording traffic and generating alert notifications of bike path.

As results of the project we obtained a calibration factor in the sensor that provides a relative error of 2.33% with the distances collected by the LIDAR sensor, a maximum user detection distance of 15m and the estimated time in which a user will be close to an intersection. Moreover, the traffic of users circulating in the bike path was presented through a graphical interface where the information was sent to and stored on the Ubidots web server.

The results obtained it was possible to validate a correct calibration of the prototype, successfully obtaining the physical parameters of the cyclists for the activation of a visual alarm of prevention, when there is proximity of the user to an intersection. In addition, the registration of user traffic on a server makes it possible to evaluate the social impact of the implementation of bike paths.

Keywords: sensor, distance, LIDAR, IoT, intersection

ÍNDICE GENERAL

DEDICATORIA	2
AGRADECIMIENTOS	3
DECLARACIÓN EXPRESA.....	4
EVALUADORES.....	5
RESUMEN.....	VI
ABSTRACT	VII
ABREVIATURA	X
SIMBOLOGÍA.....	XI
ÍNDICE DE ILUSTRACIONES.....	XI
CAPITULO 1.....	14
1 INTRODUCCIÓN	14
1.1 Descripción del problema	15
1.2 Justificación del problema	16
1.3 Objetivos	17
1.3.1 Objetivo General.....	17
1.3.2 Objetivos Específicos.....	18
1.4 Estado del arte	18
1.5 Alcance.....	22
1.6 Metodología.....	23
CAPITULO 2.....	24
2 MARCO TEÓRICO	24
2.1 Tecnologías.....	25
2.1.1 Telemetría.....	25
2.1.2 Redes IoT	26
2.2 Sensores	26
2.2.1 Por principio de funcionamiento.....	27
2.2.2 Por tipo de Señal	27
2.2.3 Por parámetro físico.....	28
2.3 Microcontroladores y módulos electrónicos.....	32
2.3.1 Arduino	33
2.3.2 Entorno de desarrollo ESP32 y ESP8266.....	35
2.3.3 Módulo LCD.....	35
2.3.4 Módulo I2C.....	36

2.4	Protocolos de comunicación.....	36
2.4.1	Protocolo de Control de Transmisión TCP.....	37
2.4.2	Protocolo IP	37
2.4.3	Protocolo I2C	38
2.4.4	Protocolo UART.....	38
2.4.5	Protocolos 802.1X.....	39
2.5	Principios Físicos.....	40
2.5.1	Reflectividad y reflectancia	40
2.5.2	Refracción de la luz y Ángulo de incidencia.....	41
2.5.3	Dispersión de la luz.....	42
2.6	Modelos estadísticos	43
2.6.1	Modelo de regresión lineal	43
2.6.2	Modelo de regresión logística	44
2.6.3	Modelo lineal Mixto generalizado.....	44
2.6.4	Nodo de regresión Cox	44
2.7	Plataformas Webs	44
2.7.1	Ubidots.....	45
2.7.2	FireBase	46
2.7.1	ThingSpeak.....	46
CAPITULO 3.....		47
3	METODOLOGÍA	47
3.1	Escenario de Estudio.....	48
3.2	Diagrama de Bloques	50
3.3	Análisis de equipos seleccionados.....	53
3.3.1	Equipos seleccionados para la primera etapa	53
3.3.2	Equipos seleccionados para la segunda etapa.....	56
3.4	Construcción del prototipo.....	58
3.4.1	Algoritmo y funciones establecidas para la primera etapa.....	59
3.4.2	Diagrama de Flujo de la conexión a Internet para la primera etapa.....	62
3.4.3	Diagrama de Flujo de la conexión a Internet para la segunda etapa	64
3.4.4	Diagrama de Flujo del comportamiento del sensor infrarrojo para la segunda etapa	66
3.4.5	Diagrama de Conexiones	68
3.4.6	Implementación del prototipo.....	72
3.5	Presupuesto	80

CAPITULO 4.....	81
4 ANÁLISIS DE RESULTADOS.....	81
4.1 Rango de detección de distancias.....	82
4.2 Velocidad media de usuarios registrados.....	84
4.3 Tiempo de llegada de usuarios	86
4.4 Habilidad de alarma visual.....	86
4.5 Registro de Usuarios	88
4.5.1 Conectividad a la nube y registro de datos.....	89
4.5.2 Activación de alertas.....	90
CONCLUSIONES.....	93
RECOMENDACIONES	94
BIBLIOGRAFÍA.....	95
APÉNDICE	103

ABREVIATURA

INEC	Instituto Nacional de Estadística y Censo
ANT	Agencia Nacional de Transito
LOTTTSV	Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial
GADs	Gobiernos Autónomos Descentralizados
FIS	Sistema de Interferencia Difusa
LoRa	Long Range
LIDAR	Light Detection and Ranging
HTTP	Protocolo de Transferencia de Hipertexto
IDE	Entorno de Desarrollo Integrado
UART	Transmisor-Receptor Asíncrono Universal
SIP	Protocolo de Inicio de Sesión
HTML	Lenguaje de Marcas de Hipertexto

MQQT Transporte de Telemetría de cola de mensajes

SIMBOLOGÍA

Km	Kilómetros
m	Metros
s	Segundos
v	Velocidad
t	Tiempo
cm	Centímetros
h	Hora
kHz	Kilohertzio
mA	Miliamperio
V	Voltaje
mm	Milímetros
W	Watts
kB	Kilobyte

ÍNDICE DE ILUSTRACIONES

Ilustración 2.1 Esquema de comunicación de telemetría	25
Ilustración 2.2 Esquema básico de red IoT	26
Ilustración 2.3. Conexión de sensores digitales (A) y analógicos (B)	28
Ilustración 2.4 Sensores ópticos de barrera (A), Retro reflectivo (B) y Difuso (C)	30
Ilustración 2.5 Estructura de un microcontrolador	33
Ilustración 2.6 Módulos Electrónicos	33
Ilustración 2.7. Pantalla LCD 8x2.	36

Ilustración 2.8. Módulo I2C acoplado a pantalla LCD.	36
Ilustración 2.9 Encabezado del protocolo TCP	37
Ilustración 2.10. Bus de datos I2C.	38
Ilustración 2.11.Comunicacion UART	39
Ilustración 2.12. Protocolos Wifi IEEE 802.1	39
Ilustración 2.13. Reflectancia de la Luz.	40
Ilustración 2.14. Angulo de incidencia y refracción de la luz.	42
Ilustración 2.15. Prisma de Newton	42
Ilustración 2.16. Modelo de regresión lineal.	43
Ilustración 2.17. Herramientas IoT en Ubidots	45
Ilustración 2.18 Tableros de gestión y control de datos en Ubidots	45
Ilustración 2.19 Tablero de mensajería en la nube de FireBase	46
Ilustración 2.20 Canales de recopilación de datos en ThingSpeak	47
Ilustración 3.1 Carril de ciclovía ubicado en la Avenida Malecón en Guayaquil	48
Ilustración 3.2 Bosquejo de implementación de prototipo	49
Ilustración 3.3. Diagrama de bloques inicial	51
Ilustración 3.4. Diagrama de bloques general	51
Ilustración 3.5. Arduino Mega 2560	54
Ilustración 3.6. Sensor Leddar One de Leddar Tech.	55
Ilustración 3.7. Módulo Relevador 5v	55
Ilustración 3.8. Módulo Wifi-Bluetooth ESP-32	56
Ilustración 3.9. Sensor fotoeléctrico E18-D80NK.....	57
Ilustración 3.10. Panel de gestión de información en Ubidots	58
Ilustración 3.11 Declaración de librerías y señales de salida.	59
Ilustración 3.12 Cálculo de distancia, tiempo y velocidad de los usuarios.....	60
Ilustración 3.13 Condición para encender la alarma visual.	61
Ilustración 3.14. Diagrama de Flujo de la conexión a la red y notificaciones.....	62
Ilustración 3.15. Parámetros de red para la comunicación con Ubidots	63
Ilustración 3.16 Publicación de parámetro de velocidad en Ubidots.....	63
Ilustración 3.17 Configuración de la condición de alertas en Ubidots.....	63
Ilustración 3.18. Diagrama de Flujo de la conexión a la red y publicación de datos.....	64
Ilustración 3.19. Parámetros de red para la comunicación con Ubidots	65
Ilustración 3.20 Publicación de datos en el servidor IoT.....	65
Ilustración 3.21 Visualización de datos en el servidor IoT	66
Ilustración 3.22 Diagrama de Flujo para el registro de cantidad de usuarios	67
Ilustración 3.23 Algoritmo con condiciones para el conteo de usuarios	68
Ilustración 3.24 Diagrama de conexiones de la etapa de detección y alarma.	69
Ilustración 3.25 Diagrama de conexiones de la etapa de conteo de usuarios.....	71
Ilustración 3.26. Detección de objeto a 60 centímetros.	73
Ilustración 3.27 Detección de objeto a 4 metros.....	73
Ilustración 3.28. Muestras obtenidas por el sensor por cada medición.	74
Ilustración 3.29. Respuesta Lineal entre valores reales y experimentales.	74
Ilustración 3.30. Construcción de prototipo de detección y alarma.....	77
Ilustración 3.31. Detección de distancias por parte del sensor LIDAR	77
Ilustración 3.32 Alarma encendida con detecciones de distancia menor a 1 metro.	78
Ilustración 3.33 Construcción de prototipo de conteo de usuarios	78

Ilustración 3.34 Construcción de prototipo de conteo de usuarios	79
Ilustración 3.35 Funcionamiento del acumulador de registros.....	80
Ilustración 4.1 Implementación del prototipo	81
Ilustración 4.2 Distancias procesadas a medida que se aproxima un bicisuario.	82
Ilustración 4.3 Comparativa entre las distancias teóricas y experimentales	83
Ilustración 4.4 Alcance y detección de bicisuarios.....	84
Ilustración 4.5 Recolección de datos de velocidad	84
Ilustración 4.6 Velocidades procesadas a medida que se aproxima un bicisuario	85
Ilustración 4.7 Velocidad alcanzadas por bicisuarios	85
Ilustración 4.8 Tiempo de llegada a la intersección.	86
Ilustración 4.9 Activación de alarma cuando se satisface la condición de proximidad .	87
Ilustración 4.10 Activación de alarma cuando se satisface la condición de proximidad	87
Ilustración 4.11 Etapa de registro de usuarios.....	88
Ilustración 4.12 Detección y registro de usuarios	89
Ilustración 4.13 Configuraciones establecidas en la etapa de conteo	89
Ilustración 4.14 Configuraciones en la plataforma Ubidots de la etapa de conteo	90
Ilustración 4.15 Recepción de alertas por incumplimiento de límites de velocidad	91
Ilustración 4.16 Generación de alertas de velocidad	91
Ilustración 4.17 Información de alerta generada.....	92
Ilustración 4.18 Registro de velocidades censadas y eventos de alertas generadas	92

ÍNDICE DE TABLAS

Tabla 2.1. Especificaciones técnicas de sensor ultrasónico HC-SR04.....	30
Tabla 2.2 Especificaciones técnicas de sensor ultrasónico RU40U	30
Tabla 2.3 Comparativa entre modelos de sensores difusos	31
Tabla 2.4 Especificaciones técnicas de módulos sensor LIDAR de Leedar Tech	32
Tabla 2.5. Tabla comparativa entre las módulos Arduino Uno y Mega2560.	34
Tabla 2.6. Especificaciones técnicas de módulos ESP8266 – ESP32	35
Tabla 2.7. Reflectancia de la luz según el material	41
Tabla 3.1 Resultado de pruebas experimentales para la velocidad y tiempo de llegada	61
Tabla 3.2 Conexiones entre sensores, módulos y microcontrolador.	70
Tabla 3.3 Conexiones entre sensores, módulos y microcontrolador.	72
Tabla 3.4 Calibración con distancias entre 0 y 100 cm	75
Tabla 3.5 Calibración con distancias entre 1 y 5 m	76
Tabla 3.6. Detalle de los componentes para la construcción del prototipo	80

ÍNDICE DE ECUACIONES

Ecuación 2.1	40
Ecuación 3.1.....	60
Ecuación 3.2.....	60

CAPITULO 1

1 INTRODUCCIÓN

En los últimos años, las principales urbes del Ecuador presentan una gran demanda en proyectos que garanticen una movilidad sostenible, con la finalidad de incentivar a las personas en el uso de bicicletas y/o vehículos amigables con el medio ambiente. Hasta 2019, el 22.54% de los hogares en el país tenían un integrante del grupo familiar haciendo uso de este transporte según indica el Instituto Nacional de Estadística y Censos (INEC). [1]

Por otra parte, la presencia de la pandemia de coronavirus COVID-19 junto a las restricciones que conlleva como el distanciamiento social, capacidad de transportes y aforos reducidos, ha impulsado el ciclismo en las principales ciudades del país, como alternativa ideal para cubrir las necesidades de movilidad y cumplimiento de restricciones establecidas. Además, como indica [2], la pandemia ha impulsado el sector de manufactura y su economía debido a que la comercialización y producción de bicicletas ha incrementado en un 30%.

En términos de transporte y movilidad, las municipalidades buscan implementar nuevos espacios para ciclistas y vehículos de micromovilidad, sin embargo, la bicicleta representa una opción poco viable debido a que las ciudades no cuentan con la infraestructura y cultura necesaria para una movilización masiva y segura. [1]

Por este motivo, el presente proyecto propone una solución tecnológica de bajo costo, enfocada a la precaución y reducción de accidentes en ciclovías con la finalidad de impulsar este medio transporte de forma segura en las urbes.

1.1 Descripción del problema

Los accidentes que involucran bicicletas en el año 2020 han aumentado en un 60% respecto al año anterior, ocasionando el deceso de 23 ciclistas, cifra que supera al doble de víctimas mortales registradas en 2019 según indica la Agencia Nacional de Tránsito (ANT) en sus estadísticas. [3]

A pesar de que los accidentes de vehículos no motorizados no representan un porcentaje significativo en comparación con los automóviles, estos comparten causas similares. Como menciona [4], una conducción inapropiada implica la desatención del conductor con su entorno, sumado al incumplimiento de señales de prevención como disco pare, luz roja, cruce de caminos o ciclistas en vía; figuran entre las causas de mayor incidencia de siniestros en las vías.

ANT hace mención que las intersecciones o cruces de vías son considerados como escenarios de alta accidentabilidad y riesgo, además de esto, la imprudencia de los conductores representa un peligro a los que hacen uso de la ciclovía [5]. Como refiere [6] “en términos generales no hay mucho respeto para el ciclista, los carros estrechan la vía y quitan espacio; falta educación vial para conductores, peatones y ciclistas”.

Según comunica [7], la normativa referente a los derechos del ciclista indica, “derecho preferente de vía o circulación en los desvíos de avenidas y carreteras, cruce de caminos, intersecciones no señalizadas y ciclovías”, reglamento estipulado en la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial (LOTTTSV) que no se ejerce con rigurosidad.

Aunque se incentive el uso de vehículos de micromovilidad por parte de los Gobiernos Autónomos Descentralizados (GADs), a través del despliegue de ciclorrutas

como indica [7], éste no se ejecuta como tal, es irrespetado y utilizado para otros fines como parqueo de vehículos. [6]

Un estudio realizado por [8], indica que cerca de 300 muertes de ciclistas urbanos se han registrados entre el período 2004-2017, representando el 0.68% de mortalidad en términos de tránsito. Estos siniestros de tránsito se ubican en las principales urbes del país como Guayaquil o Quito debido a su densidad poblacional.

Según apunta [9], hasta 2015 cerca de 2 millones de personas en Ecuador recurren al uso de bicicletas, cifra que incrementó un 2.4% para el año siguiente, indicando que la actividad de ciclismo es de uso recurrente en la urbe y debido a la situación sanitaria, desde marzo del 2020 hasta la actualidad, el uso del transporte supera el 200%.

Básicamente, las obstrucciones ubicadas en carriles preferenciales a ciclistas desencadenan eventos críticos, ya que la infraestructura de ciclorrutas complementado con señaléticas de tránsito hacia los conductores no es suficiente debido a la imprudencia o carencias en su implementación. [10]

1.2 Justificación del problema

El proyecto se sustenta en la Ordenanza Municipal de movilidad en Guayaquil, que regula el uso de bicicletas y vehículos de micromovilidad como eje fundamental para el desarrollo sostenible de la urbe. Desarrollo que pretende menguar la contaminación del medio ambiente, fomentar a la actividad física, impulsar el uso de transportes ecológicos, reducir la saturación en las vías públicas y aminorar el consumo de energías.

Por otro lado, el artículo 3 resalta los aspectos fundamentales de la ordenanza, donde denota, el derecho de los usuarios al acceso de transportes no contaminantes

de forma segura y apropiada, para el desarrollo de actividades de ciclismo en espacios públicos que involucre la participación ciudadana. [11]

El desarrollo de este proyecto trae consigo beneficios en el tema de seguridad vial y movilidad, acorde a lo que busca impulsar los GAD(s) en las urbes. En primera instancia, se procura reducir el número de siniestros de tránsito en ciclovías, debido a que los accidentes que involucran bicicletas incrementan anualmente en un 60% [3].

A su vez, la funcionalidad de este sistema de alerta para los involucrados (vehículos de micromovilidad, automotores y agencias de tránsito), conlleva a la concientización en la conducción de vehículos y el respeto hacia las leyes de tránsito, las cuales figuran entre las causas de mayor incidencia de siniestros en las vías. [4]

Además, como indica [12], la ausencia de proyectos integrales para vías de vehículos de micromovilidad, evita que los ciclistas circulen con garantías y no fomenta a la práctica de este deporte. Para este efecto, el prototipo recolecta y presenta datos, como evidencia del uso de la ciclovía; información que incentiva a una participación ciudadana segura y la inclusión de proyectos tecnológicos relacionados a este tema.

Finalmente, el trabajo se alinea al objetivo 11 de desarrollo sostenible en Ecuador referente a ciudades y comunidades sostenibles como se detalla en [13], el cual resalta la administración y mejoras de espacios urbanos con la finalidad de que sea inclusiva y participativa.

1.3 Objetivos

1.3.1 Objetivo General

- Diseñar e implementar un prototipo a bajo costo para la detección y alarma en ciclovías con intersecciones no controladas.

1.3.2 Objetivos Específicos

- Procesar parámetros censados como velocidad, distancia y tiempo para la detección de proximidad de usuarios que usan ciclovías.
- Diseñar una interfaz de alertas en intersecciones cuando se identifique la proximidad de usuarios en la ciclovía.
- Diseñar un algoritmo de detección que permita la simplificación de tareas de detección en función de la lectura de los sensores.
- Implementar un servidor web mediante unidades programables que permita registrar la información de los sensores de forma asincrónica.
- Analizar los datos almacenados en el servidor para generar notificaciones dirigidas a entes reguladores sobre el incumplimiento de leyes de tránsito en el tema de ciclovías.
- Implementar un contador de usuarios de ciclovía que permita evaluar el impacto social que conlleva el uso de transportes amigables con el medio ambiente.

1.4 Estado del arte

Heijmans, una empresa establecida en Holanda, ha desarrollado un sistema de control orientada a la protección de las personas que recurren a la actividad de ciclismo, y de esta forma precautelar su entorno frente a intersecciones peligrosas o escenarios de alto riesgo. El proyecto denominado *BikeScout*, se basa en la habilitación de luces LED como señal de alerta a vehículos, cuando existe en sus alrededores ciclistas aproximándose. [14]

El sistema de detección se originó como un prototipo y a la actualidad se ha convertido en un proyecto a gran escala orientado a ciudades inteligentes, que lleva funcionando cerca de 8 meses con un presupuesto de 40 000 euros. Su funcionamiento consiste en el diseño de un radar situado a un rango de 50 m a la intersección más próxima, que capta parámetros de posición y velocidad estableciendo el tiempo de llegada del ciclista. [14]

Las urbes representan la mayor concentración de tráfico debido a su densidad poblacional, el cual ocasiona un incremento en la probabilidad de accidentes de tránsito. El trabajo que desarrolla [15], otorga un enfoque autónomo en la evaluación de ejes del flujo vehicular, como el volumen, la velocidad y los patrones de movimiento de vehículos, con la finalidad de obtener el mayor índice de precisión en la detección de objetos en intersecciones, que permitan un control de tráfico óptimo en tiempo real.

Algunas características de interés que contribuye al análisis del tráfico, se presentan en [16], mediante un sistema inteligente que permite el control de tráfico mediante estructuras de código abierto. El objetivo es una detección continua de imágenes a través de la conmutación de dispositivos de vídeo, que recopila en tiempo real movimientos en el entorno y en función de estas variables establece tiempos para el control de semáforos.

Según publica [17], la información del tránsito se recopila a partir de sensores ubicados en el asfalto y sus zonas laterales, en donde se efectúa un conteo de los vehículos que pasan por zonas con semaforización. El algoritmo se basa en una metodología genética ya que los eventos repetitivos tienden a mutar la selección de semáforos en los cruces, mientras se está desarrollando eventos de conteo de forma simultánea. A partir de la cantidad de vehículos que almacena cada estación, se organiza de forma descendientes el tráfico para otorgar prioridad de paso.

El trabajo [18], consiste en un sistema de advertencias para accidentes en donde se determina parámetros de distancia y velocidad del tráfico basado en un modelo de reglas difusas (FIS) que se complementa con el envío de señales de vibración a dispositivos incorporados en el manillar de la bicicleta. El mecanismo otorga una respuesta háptica del usuario con su entorno sin reducir el nivel de atención en la vía.

Para el proyecto [19], se recurre a mecanismos que proporcionan medidas de seguridad en intersecciones que pueden ser consideradas de riesgo. El trabajo tiene por objetivo, la posibilidad de sustituir dispositivos de señalización electrónica (semáforos), bajo el escenario de que todos los vehículos que hagan uso de las vías se comuniquen entre sí por medio de un sistema universal de sensores, con lo cual el distanciamiento y flujo vehicular será seguro y continuo.

Bajo este enfoque de direccionamiento de tráfico, se busca un alcance tanto para peatones y ciclistas, manteniendo una asignación dinámica de recorridos seguros, que habilita carriles disponibles en intersecciones, calles principales y secundarias. [19]

Para el autor [20], el factor de seguridad es una de las carencias principales en los ciclistas por tal motivo plantean una alternativa que puede ser incorporado por usuarios que desarrollan la actividad del ciclismo. El diseño de una prenda inteligente que emite alertas visuales al resto de usuarios, complementado con iluminación, conectividad a internet y detalles de impermeabilidad frente a escenarios climáticos y nocturnos.

Por otra parte [21], presenta un sistema de detección de objetos para vehículos que precautela la seguridad del conductor. El sistema se conforma por un conjunto de sensores de largo alcance que detectan obstáculos en el trayecto del vehículo en un rango de 20 a 40 metros y mediante una pantalla de cristal líquido (LCD) indica parámetros como distancia, tiempo, entre otros. A pesar de que su objetivo principal es

mantener una conducción más segura, busca reducir el índice de accidentes en peatones y ciclistas.

En comparación con el trabajo que presenta [22] [23], se implementa un sistema de sensores infrarrojos activos que permiten determinar parámetros de distancia y detección de objetos en su radio de acción, a través de la energía reflejada que se transmite hacia el sensor desde el objeto móvil. Desde otra perspectiva, [24] presenta una propuesta para el control de intersecciones basado en la incorporación de sistemas de sensores térmicos para el monitorio de conductores, peatones y ciclistas independiente de la cantidad de luz disponible.

Según el proyecto [25], el diseño de un detector por radar mediante tecnología microonda representa una alternativa viable para la detección o presencia de objetos en intersecciones, ya que su capacidad de recolección de datos como recuento y distancia, complementado con avances en aplicaciones de detección permite diferenciar a los ciclistas de automotores. Además, puede incorporar herramientas como prolongación de tiempo en señales de paso cuando se identifique ciclistas.

En la literatura que expone [26], se evalúa el alcance y viabilidad de envío de datos en una red con tecnología LoRa, con la finalidad de recopilar información de posición del ciclista hacia un servidor. La vinculación permite transmitir mensajes o notificaciones de alerta entre el ciclista y una estación base, para una respuesta inmediata en caso de accidentes.

Por otra parte [27], presenta un estudio de intersecciones peligrosas a través de modelos con métodos predictivos en colisiones, con el objetivo de identificar y clasificar cruces de alto nivel de accidentes.

El trabajo que presenta [28], sostiene la identificación de modelos y patrones de ocupación mediante redes de sensores con tecnología Wifi como alternativa inminente

del concepto de ciudades inteligentes. Su funcionamiento conlleva la instalación de sensores en diferentes cruces de vías y mediante un servidor llevar a cabo el monitoreo y detección de volumen de tráfico que permita mejorar la planificación de la urbe en los Países Bajos.

El diseño del prototipo a desarrollar presenta 3 diferencias respecto a las soluciones mencionadas: Primero, se diseña a partir de sensores ópticos LIDAR [29] que permiten recopilar datos precisos de manera rápida y con un sistema de bajo costo, ocasionando que el aspecto económico difiera significativamente comparado con proyectos similares que incorporan cámaras de video y sistemas de radar.

Segundo, el proyecto va dirigido como una herramienta de prevención para todos los usuarios de vehículos involucrados en el cruce de vías con el carril de ciclovía, el cual se diferencia de varios proyectos citados que van dirigidos para la seguridad de un solo vehículo o conductor.

Tercero, el trabajo a desarrollarse emplea lenguajes de programación *Processing* de código abierto, esto, sumado a que LIDAR se integra con otros tipos de tecnologías, permite futuras actualizaciones y mejoras del proyecto.

1.5 Alcance

El trabajo tiene por finalidad diseñar e implementar un prototipo preventivo orientado a ciclovías con intersecciones no controladas, el cual detecta la proximidad de bicicletas y vehículos de micromovilidad mediante un conjunto de sensores, unidades de microcontrol y módulos de gestión de información de bajo costo.

Los sensores se encargan de recolectar parámetros físicos de los ciclistas, como velocidad, distancia y tiempo, para posteriormente ser procesados por una unidad de

control que valide la proximidad a cruces de vías con la ciclorruta. Además, se implementa una interfaz visual mediante luces leds que funcionan en sincronía con el conjunto de sensores con la finalidad de emular una alerta visual, cuando exista la aproximación de un ciclista; la alerta es dirigida a los automotores que se acerquen a intersecciones con la ciclovía.

Como siguiente punto, se establece la comunicación con un servidor web mediante módulos de conectividad inalámbrica, con el objetivo de almacenar la información que proporcionan los sensores, en una base de datos. El servidor ofrece herramientas de análisis y monitoreo en tiempo real.

La configuración de alertas en el algoritmo de detección, mediante restricciones en los parámetros censados, permite identificar escenarios de uso no adecuado de ciclovías o reconocer vehículos no autorizados en esta vía, las cuales pueden ser compartidas a entes reguladores de tránsito para que ejecuten las respectivas medidas correctivas.

Finalmente, se implementa un módulo de despliegue informativo para presentar datos que se han almacenado en el servidor, en este caso, el número de ciclistas que circulan por día en un tramo de la ciclovía donde se establezca el prototipo.

1.6 Metodología

El prototipo de detección y alarma tiene un enfoque en ambientes exteriores ocasionando que el estudio del entorno sea el eje principal para el despliegue del conjunto de sensores. Por tal motivo, se selecciona como alternativa, sensores de tipo LIDAR debido a que otorga, al sistema de detección, características de mayor precisión a partir de la posición y distancia de los objetos de estudio en comparación con sensores de proximidad o infrarrojos estándares.

Por otra parte, la tecnología del proyecto se basa en telemetría identificando los parámetros a censar como velocidad y distancia, un canal inalámbrico para el envío de información de los sensores a un servidor y la unidad de microcontrol para la conmutación de las entradas digitales de los actuadores o sensores y la conectividad con servicios de estado de respuesta basados en protocolo de transferencia de hipertexto (HTTP).

La información que se recolecta de los sensores es procesada en un entorno digital interactivo (IDE) del microcontrolador, el cual emplea un lenguaje de programación de tipo *Processing* en Java con la finalidad de construir un algoritmo de vinculación entre módulos y sensores.

Además, se vincula la tecnología Wifi a través del módulo ESP8266 con el fin de establecer una comunicación vía internet para el envío de datos. Finalmente, la información se vincula a una plataforma IoT que permite visualizar, en tiempo real, los parámetros de estudio, evaluar el funcionamiento del prototipo y generar un análisis del volumen de tránsito en las ciclovías.

CAPITULO 2

2 MARCO TEÓRICO

El siguiente capítulo contiene conceptos relacionados con tecnologías de la información orientadas al diseño del proyecto, el cual otorga una idea del funcionamiento de unidades de control, sensores de proximidad y técnicas para la recopilación y transmisión de datos; además de las diversas plataformas que brindan servicios en la nube para la gestión del tráfico de información de forma remota.

2.1 Tecnologías

El concepto de tecnologías de la información ha sido introducido en actividades que realizan los usuarios a diario ya que tiene por objetivo la búsqueda de mecanismos que faciliten la interacción con el usuario, la reducción de complejidad en procesos y la transformación de la sociedad hacia la era digital. Por tal motivo, se detallan las siguientes características respecto a diversos tipos de tecnologías.

2.1.1 Telemetría.

La tecnología de telemetría se basa en la medición de parámetros y su respectiva transmisión de forma inalámbrica; esta tecnología permite realizar análisis de la información que se procesa a partir de los sensores, con el objetivo de monitorear el funcionamiento de un sistema, determinar errores o actualizar registros de nuevas entradas de información.

En el ámbito del internet de las cosas, el manejo de los dispositivos transductores se fundamenta en la tecnología de telemetría debido a que las mediciones de parámetros físicos se realizan de forma remota y operan en conjunto con unidades de control e interfaces electrónicas como se observa en la Ilustración 2.1. [30]



Ilustración 2.1 Esquema de comunicación de telemetría [31]

2.1.2 Redes IoT

Internet de las cosas (IoT) representa la vinculación de dispositivos físicos en una red con el objetivo de tener acceso a diversos recursos, servicios y sitios webs mediante internet desde cualquier localización. La tecnología IoT, representa una herramienta de comunicación entre los usuarios y las cosas debido a que es posible introducir procesos en objetos cotidianos como accesorios de automóviles, artefactos de cocina y control del hogar. [32]

La tecnología se orienta a la hiper-conexión de usuarios mediante la vinculación de herramientas en la nube, servicios móviles y el concepto de Big Data en un esquema integral como se observa en la Ilustración 2.2, para facilitar la transmisión y recopilación de datos que permita reducir la intervención del usuario y alcanzar una sincronización eficiente entre el mundo digital y el físico. [32]

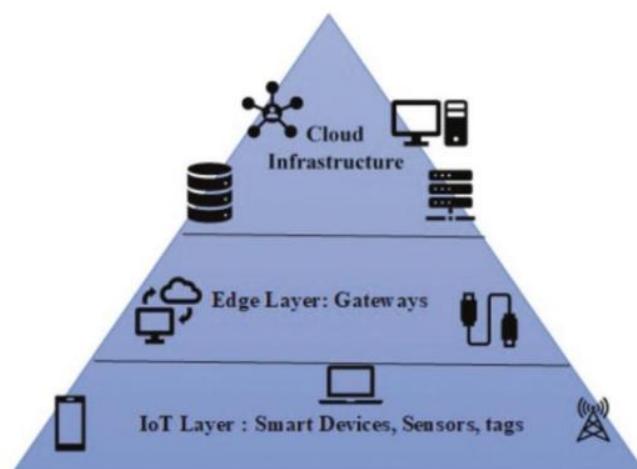


Ilustración 2.2 Esquema básico de red IoT [33]

2.2 Sensores

Se definen como dispositivos con capacidad de procesar parámetros físicos que faciliten realizar tareas de medición, control de señales o accionamiento manual de

funciones que establece el usuario [34]. A continuación, se detalla la clasificación de los sensores por principio de funcionamiento, por tipo de señal y por parámetro físico.

2.2.1 Por principio de funcionamiento

Los sensores por su funcionamiento pueden ser identificados en 2 grupos: sensores activos o sensores pasivos, los cuales son descritos a continuación:

2.2.1.1 Sensores Activos

Se denominan a sensores que incorporan, de forma externa, una fuente de energía para el funcionamiento o alimentación de su señal de salida como sensores de tipo RADAR, SONAR o termistores. [35]

2.2.1.2 Sensores Pasivos

Por otra parte, los de tipo pasivo o generadores se caracterizan por energizar la señal de salida a partir de la señal de entrada sin necesidad de incorporar una fuente externa. La señal eléctrica que se genera representa una respuesta ante un estímulo del entorno físico como termocuplas. [35]

2.2.2 Por tipo de Señal

Respecto a la representación de la señal de salida que se puede obtener, los transductores se clasifican en digitales o analógicos. Por consiguiente, se procede a detallar las características principales de los sensores por su tipo de señal:

2.2.2.1 Digitales

Hace referencia a las señales que asignan 2 posibles valores lógicos: nivel lógico alto (1) o nivel lógico bajo (0) como interruptores o detectores; los cuales ofrecen mayor fiabilidad y reduce considerablemente la complejidad de transmisión de datos en comparación con los sensores de tipo analógicos. Su conexión básica es visualizada en la Ilustración 2.3 A. [36]

2.2.2.2 Analógicos

Son dispositivos que se caracterizan por la variación de la señal de salida de forma continua como reguladores, dimmers, piezoeléctricos o sensores de temperatura en donde la conexión de sensores de este tipo se visualiza en la Ilustración 2.3 B. El rango de valores adquiridos es de gran utilidad para la esquematización, por parte del usuario, de acciones o funciones de acuerdo al valor procesado [36].

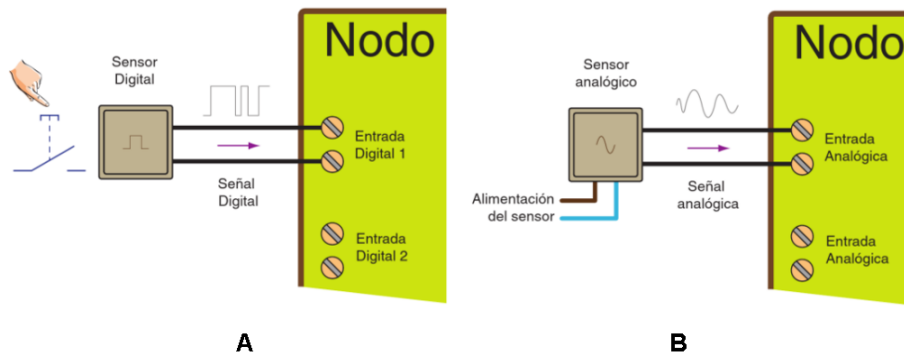


Ilustración 2.3. Conexión de sensores digitales (A) y analógicos (B) [36]

2.2.3 Por parámetro físico

Por otra parte, los sensores pueden ser identificados por las magnitudes físicas que se obtienen. A continuación, se describen diversos tipos de sensores por el parámetro físico de estudio como ultrasónicos, ópticos o láser:

2.2.3.1 Ultrasónicos

Son dispositivos orientados a determinar valores de distancias de diversos objetos que se encuentran en la periferia de detección del dispositivo. Los sensores ultrasónicos son introducidos con frecuencia en el ámbito automotriz, para realizar maniobras de entradas y salida de parqueos, el cual puede alcanzar distancia de detección de hasta 1.5 metros. [37]

Su funcionamiento se basa en la transmisión de sonido y su reflexión con los obstáculos que detecta como objetos planos o transparentes independientes de su forma o color. Además, son utilizados para la detección de nivel en tanques hasta 8 metros. La información que se presenta en la Tabla 2.1 y en la Tabla 2.2, corresponde a sensores ultrasónicos de bajo costo e industrial respectivamente. [37]

2.2.3.1 Ópticos

Los sensores que se definen en esta categoría utilizan la luz para determinar la presencia o ausencia de diversos objetos que pueden ser utilizados en la industria para obtener el tamaño de producción o implementados en sistemas domóticos como aperturas automáticas de puertas. [38]

Las aplicaciones de los sensores ópticos son de gran utilidad en la industria automotriz, alimenticia, transporte y manejo de materiales con capacidad de detectar objetos metálicos, plásticos, de madera, vidrio y pueden ser de tipo barrera, retro reflectivo o difuso como se observa en la Ilustración 2.4 A, 2.4B, 2.4C respectivamente.

Tabla 2.1. Especificaciones técnicas de sensor ultrasónico HC-SR04 [39]

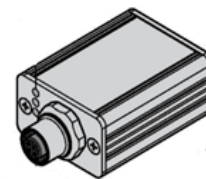
SENSOR HC-SR04



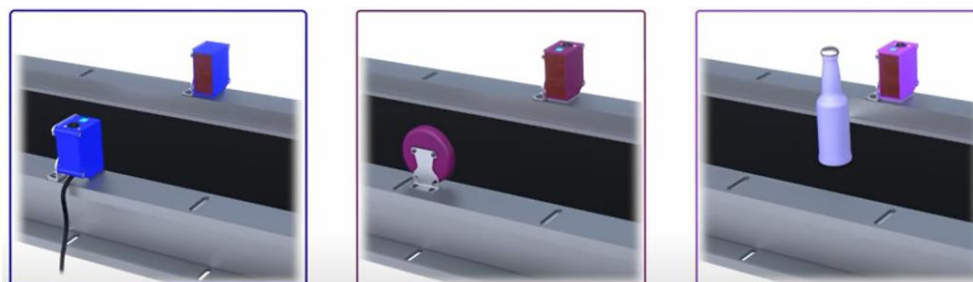
Alimentación	5 Vdc
Trigger	Pin configurado como entrada
Echo	Pin configurado como salida
GND	0 Vdc
Corriente	15 mA
Rango de detección	2 – 500 cm
Ángulo de detección	15°- 20°

Tabla 2.2 Especificaciones técnicas de sensor ultrasónico RU40U [40]

SENSOR RU-40U



Alimentación	15-30 Vdc
Frecuencia Ultrasonido	300 kHz
Protocolo	IO-Link
Tiempo de respuesta	60 ms
Corriente	150 mA
Rango de detección	25 – 400 mm
Resolución	0.5 mm



A



B

C

Ilustración 2.4 Sensores ópticos de barrera (A), Retro reflectivo (B) y Difuso (C) [41]

Los sensores ópticos difusos comparten características de detección similares con los de tipo barrera o retro reflectivo, sin embargo, la diferencia radica en su diseño debido a que incorporan el sistema transmisor y receptor en el mismo componente, en donde la luz viaja desde el emisor y al detectar un objeto, rebota hacia el sistema receptor. Algunas características y modelos de este tipo de sensor se describen en la Tabla 2.3.

Tabla 2.3 Comparativa entre modelos de sensores difusos [42] [43]

	PMD40DN	E18-D80NK	E3Z-D82
SENSOR			
Alimentación	10-30 Vdc	5 Vdc	12-24 Vdc
Corriente	300 mA	100 mA	100 mA
Rango de detección	40 cm	3-80 cm	1 m
Modo de salida	NPN	NPN	PNP

2.2.3.2 Láser

Para la detección de distancias de objetos se cuenta con sensores de tipo LIDAR en donde el rango del haz láser, que proporciona la tecnología de este sensor permite obtener una posición georreferenciada de un objeto. El despliegue de este dispositivo involucra aplicaciones a nivel terrestre estacionario, aerotransportado e inclusive mapeo móvil en 3D. [44]

La medición de parámetros es obtenida de LIDAR tipo *dispersómetro* para el análisis de propiedades de las nubes, *Doppler* para la estimación de vientos o método de predicción meteorológico y de *altimetría* para proporciona distancias de precisión entre sitios de referencia y localizaciones específicas. [45]

Por consiguiente, se detalla en la Tabla 2.4, las características de sensores de tipo LIDAR proporcionados por la empresa Leddar Tech.

Tabla 2.4 Especificaciones técnicas de módulos sensor LIDAR de Leddar Tech [46] [47]

	LeddarOne	Leddar M16	Leddar Vu8
SENSOR			
Interfaz	3.3 V UART	UART, USB, RS-485	UART, RS-485
Alimentación	5 Vdc	12-30 Vdc	12 Vdc
Longitud de onda	850 nm	905 nm – 940 nm	905 nm
Rango de detección	0-40m	165 m	185 m
Consumo	1.3 W	4 W	2 W
Haz	3° Difuso	19°	48°
Temperatura de operación	-45°C a 85°C	-45°C a 85°C	-45°C a 85°C
Velocidad de datos	140 Hz	100 Hz	100 Hz

2.3 Microcontroladores y módulos electrónicos

Los microcontroladores o microprocesadores son pequeños circuitos integrados programables capaces de realizar procesos digitales. Su estructura se basa en 4 partes fundamentales, como se indica en la Ilustración 2.5: primero, la memoria donde se almacenan los algoritmos programados, segundo, su unidad de procesamiento, la cual ejecuta los comandos según las funciones establecidas, tercero, los periféricos de soporte como convertidores digitales, temporizadores, contadores y, por último, las señales de entrada y salida que procesan los puertos de comunicación. [48]

Sus principales ventajas se basan en los tamaños reducidos, bajo consumo energético, compatibilidad y costes reducidos. Es por ello, que sus aplicaciones están orientadas en el ámbito de las telecomunicaciones y automatización de procesos. [48]

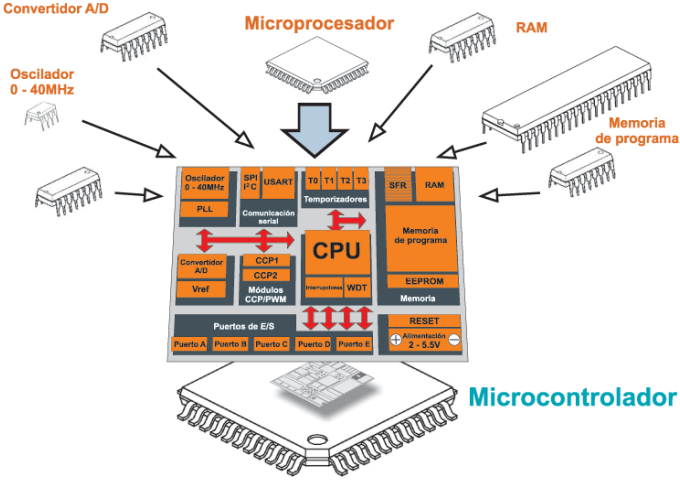


Ilustración 2.5 Estructura de un microcontrolador [49]

Por otra parte, los módulos electrónicos observados en la Ilustración 2.6, son elementos adaptables a los microcontroladores con el objetivo de realizar funciones específicas de forma eficiente, ahorrando recursos a los circuitos integrados. A continuación, se proporciona información de diferentes módulos electrónicos [50]



Ilustración 2.6 Módulos Electrónicos [51]

2.3.1 Arduino

La plataforma Arduino se desarrolla bajo placas electrónicas con microcontroladores programables que incorporan hardware y software de código



abierto. Todos sus dispositivos utilizan el microcontrolador ATMEL y manejan el lenguaje de programación IDE similar a C++.

Entre sus ventajas principales esta la versatilidad y flexibilidad de sus dispositivos, un lenguaje de programación de fácil comprensión, bajo costo y una comunidad ya establecida a nivel mundial que brinda soporte y documentación de sus múltiples aplicaciones.

Entre sus dispositivos más utilizados se tiene al Arduino Uno y el Mega2560, los cuales incorporan entradas y salidas para señales analógica y digitales, puertos PWM, comunicación serial UART, protocolo de sesión SIP, botón de reinicio, conexión USB entre otras más, que permiten interactuar de forma sencilla en el campo de la electrónica [52].

En la Tabla 2.5, se describe las placas correspondientes a Arduino Uno y Arduino Mega2560.

Tabla 2.5. Tabla comparativa entre las módulos Arduino Uno y Mega2560. [53] [54]

Modelo	Arduino Mega2560 	Arduino UNO 
Microcontrolador	ATmega2560	Atmega328p
Voltaje de entrada	7-12 V	7-12 V
Puertos Digitales	54 pines	14 pines
Puertos Analógicos	16 pines	6 pines
Frecuencia de reloj	16 MHz	
Almacenamiento	256 KB	32 KB
Memoria Estática (SRAM)	8 KB	2 KB
EPROM	4 KB	1 KB
Precio	\$29.63	\$22.64

2.3.2 Entorno de desarrollo ESP32 y ESP8266

Las placas de desarrollo ESP integran un funcionamiento muy similar al entorno de desarrollo Arduino, sin embargo, la diferencia radica en que poseen un chip híbrido con comunicación Wifi y Bluetooth como se detalla en las especificaciones de la Tabla 2.6. Además, presentan ventajas respecto a su capacidad de procesamiento y adaptación a condiciones climáticas extremas, siendo aplicadas en distintos entornos industriales [55].

2.3.3 Módulo LCD

Las pantallas LCD brindan la funcionalidad de presentar información que procesan diversos microcontroladores de forma visual, como se observa en la Ilustración 2.7. Los caracteres son asignados en una matriz que según sus dimensiones implica la ubicación en pantalla de la información, entre los más utilizados tenemos la LCD8x2 y LCD16x2. [56]

Tabla 2.6. Especificaciones técnicas de módulos ESP8266 – ESP32 [57]

	ESP32	ESP8266
Microcontrolador	Tensilica X36 2 núcleos	Tensilica XL106 1 núcleos
Voltaje de entrada	2.2-3.6 V	3-3.6 V
Puertos de Modulación por anchura de pulso (PWM)	16 pines	8 pines
Puertos Genéricos (GPIO)	36 pines	17 pines
Memoria Flash	16 MiB	16 MiB
SRAM	512 KB	160 KB
Frecuencia de reloj	160MHz a 240 MHz	80MHz a 160 MHz
Temperatura	-40°C a 125°	
Bluetooth	Si	No
Wifi	802.11 b/g/n	



Ilustración 2.7. Pantalla LCD 8x2. [56]

2.3.4 Módulo I2C

Establecen comunicaciones seriales bajo el protocolo de circuitos integrados (I2C), generalmente es un complemento para las pantallas LCD debido a que permiten vincular los pines de salida de la pantalla a través de un solo bus de datos por medio de 2 puertos, un sistema de reloj (SCL) y un sistema de datos (SDA) como se visualiza en la Ilustración 2.8. Esto representa una gran ventaja para el microcontrolador ya que se reduce considerablemente conexiones entre pines de entradas digitales. [58]



Ilustración 2.8. Módulo I2C acoplado a pantalla LCD. [58]

2.4 Protocolos de comunicación

Los protocolos de comunicación hacen referencia a una agrupación de reglas que garantizan la transferencia de datos digitales de forma segura y eficiente, entre un nodo emisor y receptor. Entre los protocolos más importantes tenemos:

2.4.1 Protocolo de Control de Transmisión TCP

Representa uno de los protocolos más importantes en el ámbito de las telecomunicaciones, debido a que el tráfico de información se controla, ordena y transporta en la red de internet. Su funcionamiento se basa en la identificación del usuario y el servidor a donde desea conectarse, segmenta en pequeños paquetes, la información para asignar etiquetas, la cual indica la secuencia de cómo se deben recuperar los datos [59].

En la Ilustración 2.9, se observa el formato de cabecera que utiliza TCP para segmentar y encapsular los datos a transmitir.

ENCABEZADO TCP



Ilustración 2.9 Encabezado del protocolo TCP [60]

2.4.2 Protocolo IP

El protocolo de internet representa un concepto fundamental en las comunicaciones, su funcionamiento radica en el arreglo de los datos a transmitir, por medio de una cabecera que indica el origen, el tipo de datos y el destino. Este protocolo funciona en conjunto con TCP brindando la señalización y transporte de datos en toda la red de Internet. [61]

2.4.3 Protocolo I2C

Este protocolo permite enviar y recibir datos por medio de 2 líneas de transmisión, la línea SDA orientada al paso exclusivo de información y la línea SCL que transporta una señal de reloj para la sincronización como se visualiza en la Ilustración 2.10. Cabe mencionar que este tipo de comunicación es Half-Dúplex síncrona, dependiendo de la señal de reloj tanto para codificar y decodificar los bits de interés y no puede enviar y recibir datos en un mismo instante. [62]

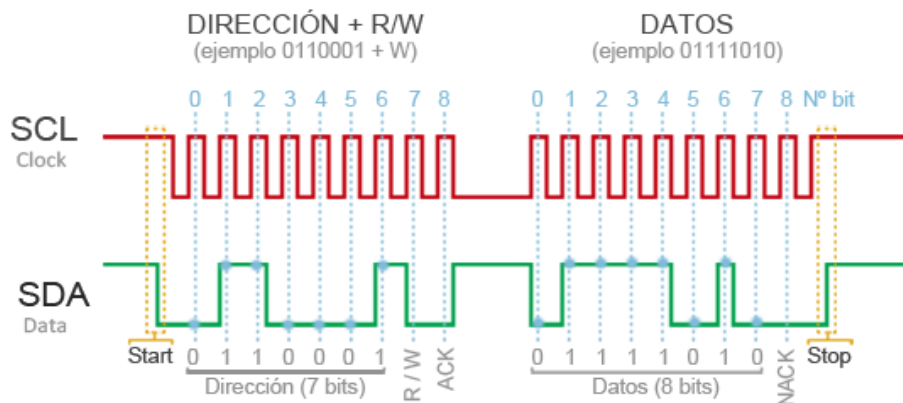


Ilustración 2.10. Bus de datos I2C. [62]

2.4.4 Protocolo UART

A diferencia de I2C, este protocolo maneja el transporte de datos de forma asíncrona, por lo que sólo se tiene en consideración los canales de transmisión y recepción. Al mantener una comunicación en paralelo se pueden transmitir mayor cantidad de bits por ciclo de reloj, además de enviar y recibir datos en un mismo instante, generando una comunicación sea Full-Dúplex. En la actualidad, la mayoría de los puertos de conexión trabajan bajo este protocolo, como, por ejemplo, USB, HDMI, VGA, entre otros. [62]

En Ilustración 2. 11, se observa los puertos y conexiones establecidas para llevar a cabo una comunicación de tipo UART.

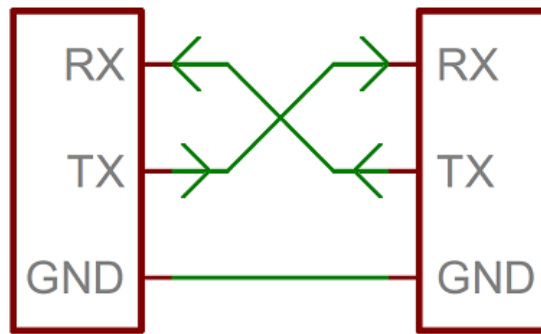


Ilustración 2.11. Comunicación UART [63].

2.4.5 Protocolos 802.1X

Son conjuntos de reglas que rigen las comunicaciones en redes inalámbricas, desde su primera aparición con el tradicional 802.11 ha mantenido una evolución exponencial en cuanto a la velocidad de transmisión, ancho de canal, frecuencia y número de usuarios. La Ilustración 2.12 que se presenta a continuación, otorga un resumen de los diversos protocolos de Wifi. [64]

Protocolo	Frecuencia	Ancho del canal	MIMO	Velocidad de datos máxima (en teoría)
802.11ax	2,4 o 5 GHz	20, 40, 80, 160 MHz	Usuario múltiple (MIMO-MU)	2,4 Gbps
802.11ac wave2	5 GHz	20, 40, 80, 160 MHz	Usuario múltiple (MIMO-MU)	1,73 Gbps
802.11ac wave1	5 GHz	20, 40, 80 MHz	Un solo usuario (SU-MIMO)	866,7 Mbps
802.11n	2,4 o 5 GHz	20, 40 MHz	Un solo usuario (SU-MIMO)	450 Mbps
802.11g	2,4 GHz	20 MHz	No se aplica	54 Mbps
802.11a	5 GHz	20 MHz	No se aplica	54 Mbps
802.11b	2,4 GHz	20 MHz	No se aplica	11 Mbps
Tradicional 802.11	2,4 GHz	20 MHz	No se aplica	2 Mbps

Ilustración 2.12. Protocolos Wifi IEEE 802.1 [64]

2.5 Principios Físicos

Los principios físicos de la materia son todas las leyes y conceptos en la cual se fundamentan los distintos tipos de ingeniería y su amplio campo de aplicaciones. Entre algunos ejemplos tenemos:

- Cinemática
- Óptica
- Termodinámica
- Electricidad y magnetismo
- Sonido

2.5.1 Reflectividad y reflectancia

La reflectividad es una propiedad intrínseca de todos los materiales, se define como una relación entre la amplitud de la onda que incide en el material con la que se refleja, como se ve en la Ilustración 2.13. Por otra parte, la reflectancia hace un análisis similar, pero relaciona las potencias de las ondas electromagnéticas involucradas como se expresa en la Ecuación 2.1. [65]

$$R = \frac{\text{Potencia Reflectada } (P_r)}{\text{Potencia Incidente } (P_i)} \quad (\text{Ecuación 2.1})$$

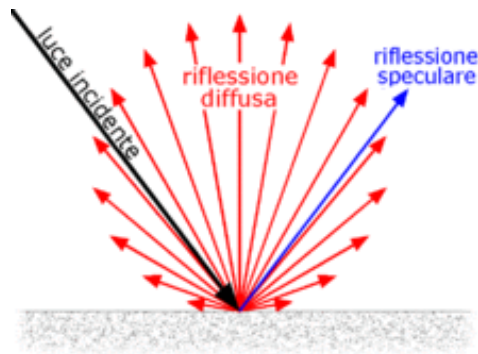


Ilustración 2.13. Reflectancia de la Luz. [66]

En la tabla 2.7, se observa los valores que toma la reflectividad para distintos tipos de materiales, en donde la onda incidente es de origen infrarroja.

Tabla 2.7. Reflectancia de la luz según el material [67]

Tipo de material	Reflectividad luz infrarroja %
Acero inoxidable	400
Aluminio	140
Papel	80
Arcilla	70
Arena	50
Plástico	40
Asfalto	17
Lava	8
Neopreno	4
Caucho	8

2.5.2 Refracción de la luz y Ángulo de incidencia.

La velocidad de la luz, durante su trayectoria, se caracteriza por ser constante, sin embargo, cuando experimenta un cambio en sus parámetros de velocidad y dirección por efecto de las propiedades del medio que atraviesa, se produce el fenómeno de refracción de la luz. [68]

Por otro parte, la apertura angular que forma el rayo que incide en un material con su normal, introduce el concepto de ángulo de incidencia, mientras que el ángulo de refracción corresponde a la apertura angular que se forma del rayo refractado y dicha normal; esta definición se aprecia en la Ilustración 2.14. [68]

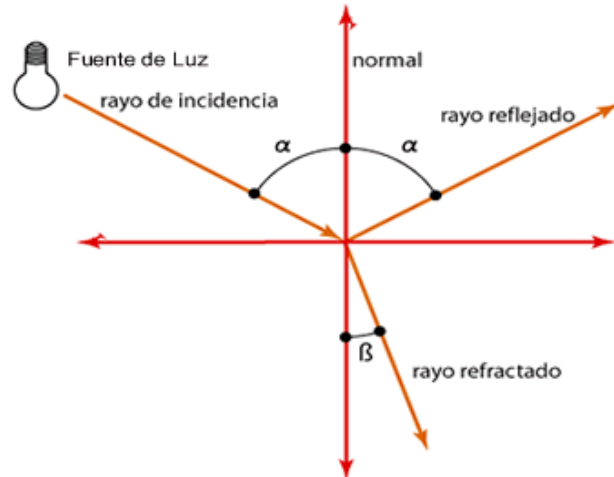


Ilustración 2.14. Angulo de incidencia y refracción de la luz. [68]

2.5.3 Dispersión de la luz

Descubierto por Newton en 1665, el hacer incidir la luz sobre un prisma, ocasiona la descomposición de colores debido a la refracción de la luz en diferentes ángulos como se indica en la Ilustración 2.15, lo que implicaba distintas longitudes de onda. La diferenciación de los colores observados, permite categorizarlos por colores primarios como azul, rojo, amarillo y colores secundarios como violeta, verde y naranja, los cuales fueron segmentados según sus longitudes de onda y unificados en lo que hoy se conoce como espectro visible. [69]

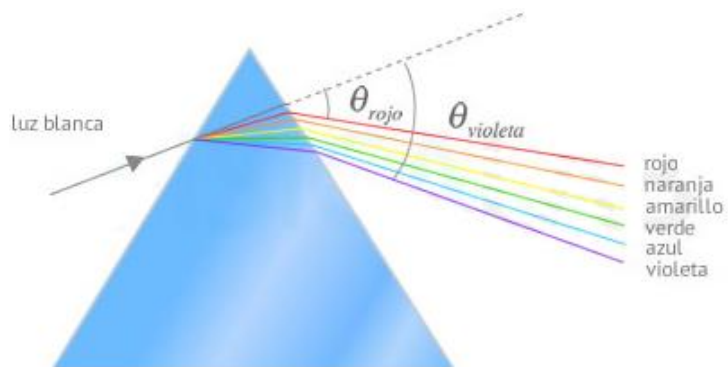


Ilustración 2.15. Prisma de Newton [69].

2.6 Modelos estadísticos

Los modelos estadísticos utilizan procesos matemáticos para cifrar y codificar datos. Sus tipos van desde modelados sencillos que permiten dar soluciones de forma rápida hasta técnicas más avanzadas para predecir resultados, cabe mencionar que no siempre el modelo más desarrollado es el óptimo para cualquier tipo de circunstancia, siempre dependerá del tipo de situación y las variables de estudio.

Entre los modelos se tiene:

- Modelo de regresión Lineal
- Regresión logística
- Nodo PCA Factorial
- Análisis de discriminante
- Nodo de regresión Cox
- Regresión lineal mixto generalizado.

2.6.1 Modelo de regresión lineal

El modelo de regresión representa una de las técnicas de mayor uso estadístico debido a que permite la clasificación de registros en función de información de entrada alfanumérica para así obtener una línea recta generalizada que permita minimizar el error y predecir futuros valores, tal como se observa en la Ilustración 2.16. [70]

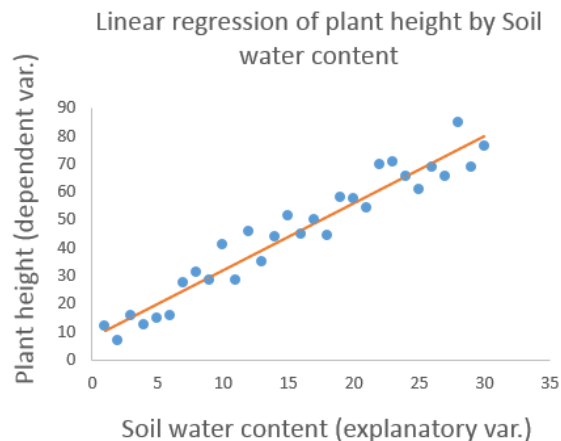


Ilustración 2.16. Modelo de regresión lineal. [70]

2.6.2 Modelo de regresión logística

Este modelo nace a partir del modelo de regresión lineal, donde utiliza todos los datos procesados y añade dos campos, la predicción y probabilidad asociada al modelo. Se orienta generalmente en catalogar valores de entrada no numéricos.

2.6.3 Modelo lineal Mixto generalizado

A diferencia del modelo de regresión lineal, este se orienta a casos que no se predicen mediante una distribución normal, se relaciona entre factores y co-variables mediante relaciones específicas. [71]

2.6.4 Nodo de regresión Cox

Orientado a modelos de supervivencia que dependen del tiempo, su finalidad es proporcionar una función de probabilidad que permita predecir un evento en un instante de tiempo t . [71]

2.7 Plataformas Webs

Basados en lenguajes de programación como Hipertexto HTML, Java o de estilo de cascada CSS, las plataformas webs comprenden una interfaz visual simplificada orientada al usuario con el fin de otorgar herramientas intuitivas para ejecutar procesos, visualizar información almacenada en sus servidores o aplicar acciones en función de datos de entrada y/o salida. A continuación, se detalla algunas características que ofrecen las plataformas webs como Ubidots, FireBase y ThingSpeak. [72]

2.7.1 Ubidots

La plataforma Ubidots se define como una herramienta IoT fundamentada en el desarrollo de aplicaciones y su producción como se observa en la Ilustración 2.17, el cual proporciona la vinculación de dispositivos externos de control y procesamiento de variables a través de su motor de análisis con la finalidad de generar paneles de gestión de datos, reportes, condiciones de eventos preestablecidos por el usuario o la visualización de las variables creadas en la nube. [73]

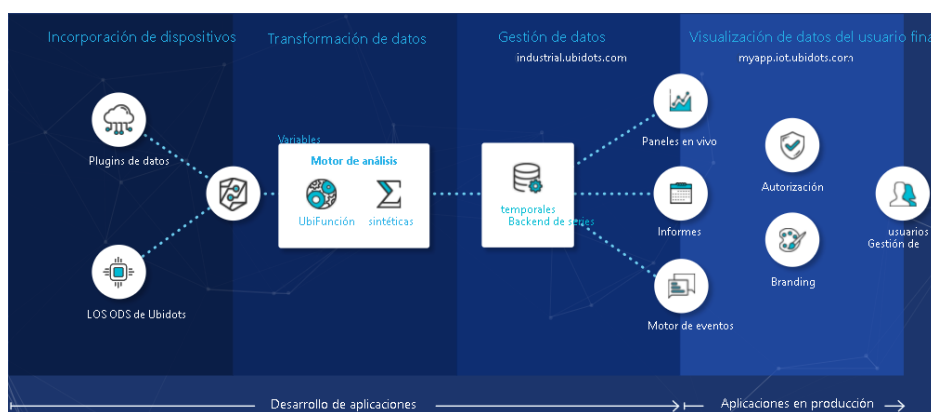


Ilustración 2.17. Herramientas IoT en Ubidots [73]

La Ilustración 2.18 corresponde a los tableros de monitoreo y control en tiempo real o programado que proporciona el servidor Ubidots en donde es posible gestionar dispositivos o analizar información mediante gráficos, tablas o visualizadores con codificación de Hipertexto HTML. [73]

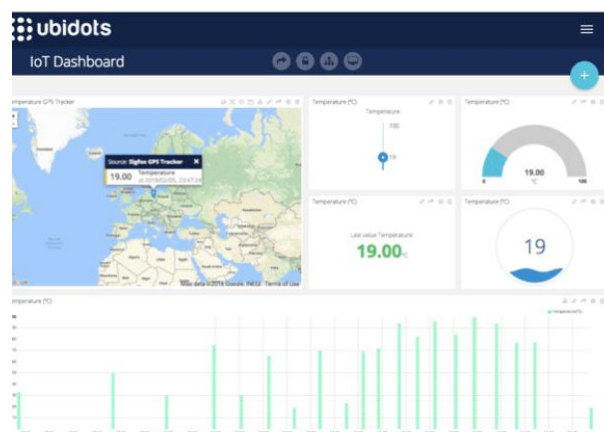
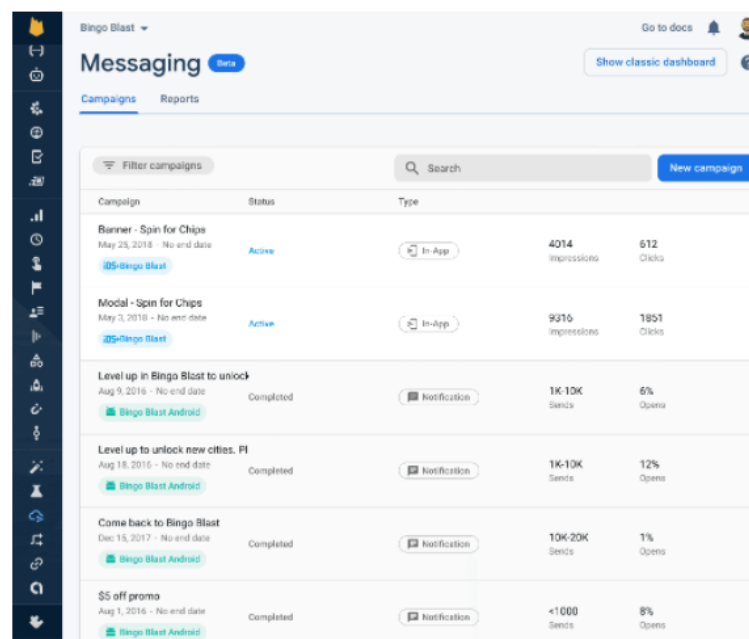


Ilustración 2.18 Tableros de gestión y control de datos en Ubidots [73]

2.7.2 FireBase

FireBase es una plataforma que vincula servicios de Google para el desarrollo de aplicaciones, asignando servidores de tipo *backend* para la gestión y almacenamiento de información desde unidades de control externas. FireBase, permite monitorear en tiempo real los dispositivos registrados, revisar su funcionamiento, reportar fallas y generar informes como se observa en la Ilustración 2.19. [74]



The screenshot shows the Firebase Messaging dashboard for a project named 'Bingo Blast'. The page is titled 'Messaging' and has tabs for 'Campaigns' and 'Reports'. A table lists several campaigns with their status, type, and performance metrics.

Campaign	Status	Type	Impressions	Clicks
Banner - Spin for Chips May 25, 2016 - No end date Bingo Blast	Active	In-App	4014	612
Modal - Spin for Chips May 3, 2016 - No end date Bingo Blast	Active	In-App	9316	1851
Level up in Bingo Blast to unlock Aug 9, 2016 - No end date Bingo Blast Android	Completed	Notification	1K-10K Sends	6% Opens
Level up to unlock new cities. FI Aug 18, 2016 - No end date Bingo Blast Android	Completed	Notification	1K-10K Sends	12% Opens
Come back to Bingo Blast Dec 15, 2017 - No end date Bingo Blast Android	Completed	Notification	10K-20K Sends	1% Opens
\$5 off promo Aug 1, 2016 - No end date Bingo Blast Android	Completed	Notification	<1000 Sends	8% Opens

Ilustración 2.19 Tablero de mensajería en la nube de FireBase [74]

Los servidores se caracterizan por introducir un desarrollo de software (SDK) con la finalidad de incorporar librerías de interacción con entornos como Unity, C++, iOS o Android para ejecutar la lectura o escritura de información en tiempo real, programar notificaciones de procesos y verificar TOKEN de autenticación en usuarios. [74]

2.7.1 ThingSpeak

Representa una herramienta IoT orientada al análisis de datos en la nube mediante la creación y distribución de canales. La Ilustración 2.20, presenta canales en

ThingSpeak que procesan de forma continua la información que se envía desde sensores externos, ejecutando decisiones a medida que los datos son registrados en la nube.

Se encuentra vinculada con funcionalidades y productos de Matlab con el objetivo de analizar la información almacenada y establecer eventos o decisiones programables sin necesidad de efectuar configuraciones en servidores. [75]

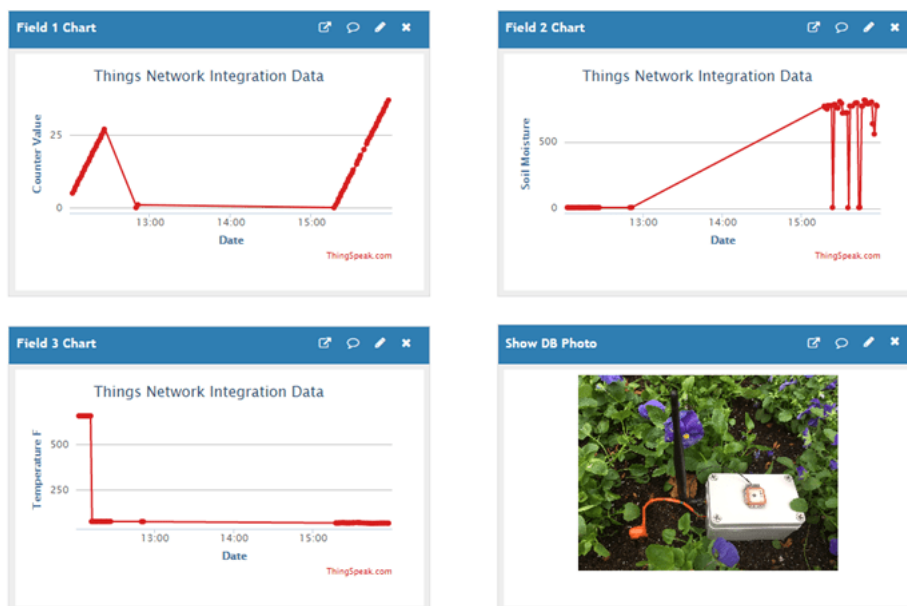


Ilustración 2.20 Canales de recopilación de datos en ThingSpeak [75]

CAPITULO 3

3 METODOLOGÍA

El presente capítulo detalla el proceso de construcción del prototipo de detección y alarma en función de la propuesta de solución orientado a intersecciones no controladas en ciclovías, en donde se describen las respectivas instrucciones y

funciones relevantes del algoritmo; así como también, los diagramas de bloques y diagramas de flujos del prototipo.

Desde el análisis preliminar del escenario identificado, se busca determinar la viabilidad de ejecución del proyecto conforme se avance en la lectura del capítulo, con el objetivo de presentar pruebas en escenarios iniciales que ratifiquen su funcionamiento.

3.1 Escenario de Estudio

A partir de la Ilustración 3.1, se identifica el escenario en el que se fundamenta el proyecto, localizada en la ciudad de Guayaquil, representa un escenario de riesgo para los usuarios que hacen uso de la ciclovía debido a que, al ser una intersección no controlada, las maniobras de conductores que circulan de forma transversal no pueden ser predecibles.



Ilustración 3.1 Carril de ciclovía ubicado en la Avenida Malecón en Guayaquil

Por tal motivo, se presenta el bosquejo de la propuesta en la Ilustración 3.2, en donde mediante unidades programables, sensores y dispositivos emisores de luz como los LED se pretende desarrollar el prototipo de detección y alarma.

El prototipo busca la detección de los usuarios que circulan por la ciclovía mediante el cálculo continuo de distancias, y a partir de este parámetro determinar el tiempo estimado de llegada a la intersección o vía transversal para generar una señal que habilite una alarma visual.

Por otra parte, se pretende incorporar un sistema de conteo de usuarios con la finalidad de obtener información de las rutas que más frecuentan los vehículos de micromovilidad; los datos son gestionados y almacenados en una plataforma IoT en la nube para el monitoreo en tiempo real o realizar seguimientos o análisis de tránsito, a partir de histogramas de registros.

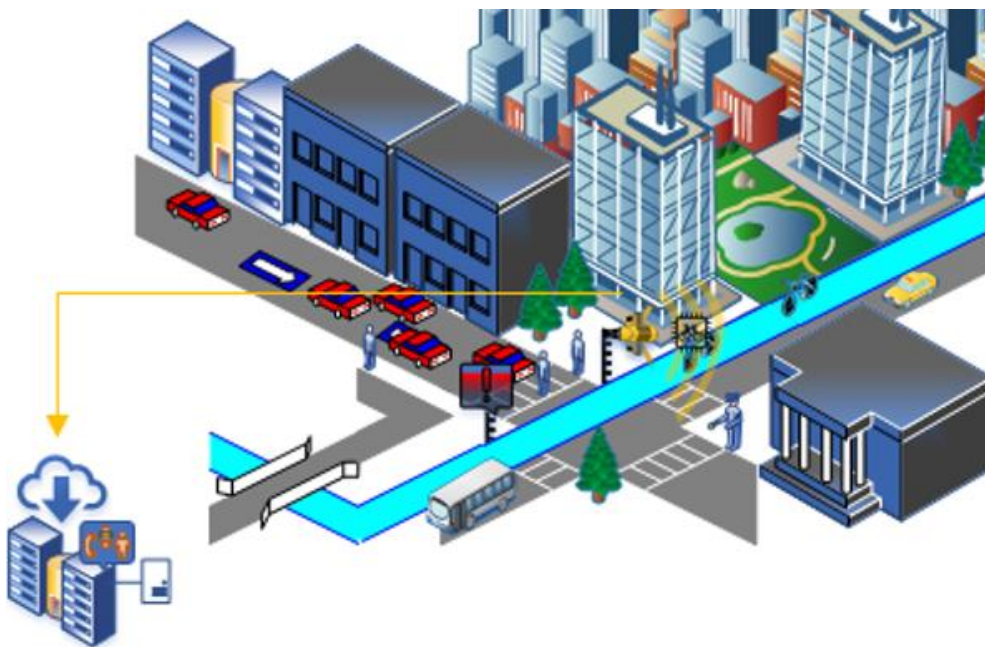


Ilustración 3.2 Bosquejo de implementación de prototipo

3.2 Diagrama de Bloques

El prototipo a desarrollarse cuenta con 3 fases con la finalidad de alcanzar los objetivos que se han planteado. En la Ilustración 3.3, se presenta cada fase de interés que permita segmentar el proyecto y establecer tareas específicas; las fases de estudio son detalladas a continuación:

- **Fase 1: Extracción de datos.**

Consiste en la incorporación de sensores activos o pasivos para obtener parámetros de medición y análisis como respuesta para la fase de acondicionamiento de las señales, a través de una configuración externa con el dispositivo sensor y una unidad de control.

- Etapa de acondicionamiento: Previo al estudio de la fase 2, se tiene el acondicionamiento de las señales de entrada que capta el sensor, con la finalidad de procesar y gestionar los resultados en subsistemas digitales según corresponda.

- **Fase 2: Procesamiento de datos.**

Mediante unidades programables se asignan algoritmos y operaciones en torno a los valores que se obtienen de la etapa anterior, para la ejecución de funciones específicas y de esta forma establecer variables de salida direccionadas a la fase 3.

- Almacenamiento: los valores que se obtienen de las funciones detalladas en el entorno de desarrollo son recopilados y procesados por módulos configurables para comunicación vía internet hacia un servidor.

- **Fase 3: Presentación de datos y alarma**

La última fase corresponde a la visualización y monitorio de la información que se ha gestionado y almacenado en el servidor, con la finalidad de enviar las variables de salida hacia interfaces visuales externas y habilitar condiciones de alarma.



Ilustración 3.3. Diagrama de bloques inicial

A partir del diagrama de bloques que se presenta en la Ilustración 3.3, se procede a detallar cada fase del proyecto describiendo las características y herramientas a incorporar en el prototipo, tal como se presenta en la Ilustración 3.4.

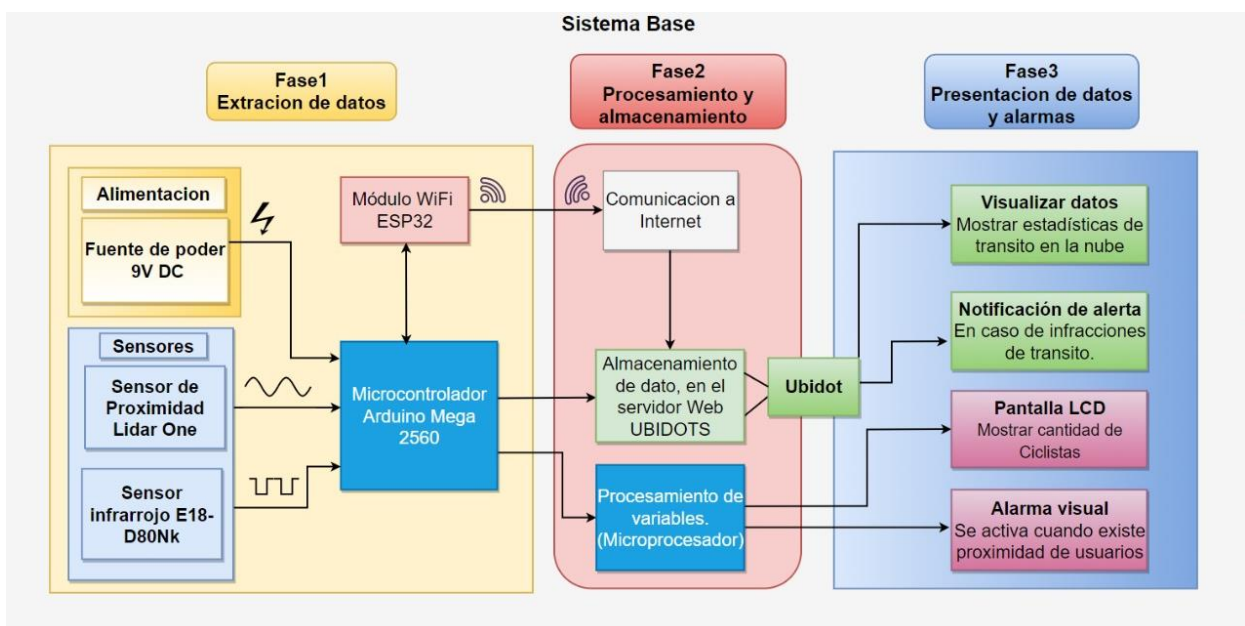


Ilustración 3.4. Diagrama de bloques general

La **fase 1** posee una etapa de alimentación de energía mediante un convertidor AC-DC que permita suministrar un voltaje y corriente adecuada para los módulos en base a las especificaciones técnicas del dispositivo, para energizar sensores y unidades de microcontrol. Por otra parte, se recurre al uso de sensores de proximidad tipo LIDAR para la medición de distancias y sensores infrarrojos para detección de objetos.

Estos dispositivos se interconectan con la unidad de microcontrolador *Mega 2560* para el respectivo acondicionamiento de señal, complementado con la vinculación de un módulo Wifi ESP32, el cual mediante el protocolo TCP/IP, permita establecer una sesión de comunicación entre anfitriones de forma bidireccional.

Respecto al procesamiento y almacenamiento de la **fase 2**, el módulo ESP32 establece una comunicación vía internet, para esto se tiene que determinar parámetros de red como SSID y modos de autenticación con el servidor mediante *Token*, el cual vincula la plataforma IoT con los datos procesados.

La unidad programable dispone de un entorno de programación *Processing* en donde se introducen funciones para la extracción de parámetros físicos, que permitan calcular valores de distancia y tiempo. Esta información se almacena en el servidor mediante configuraciones del módulo de internet y algoritmos de telemetría.

Por último, la **fase 3** muestra la información que se ha procesado de las fases previas, presentando por pantalla LCD, la cantidad de usuarios que han utilizado la ciclovía, habilitar una alarma visual cuando se detecte la aproximación de usuarios, generar alertas cuando se identifique a usuarios que no respeten los límites de velocidad establecidas en la ciclovía y mostrar datos estadísticos almacenados en el servidor web.

3.3 Análisis de equipos seleccionados.

El análisis efectuado se divide en 2 etapas; la primera etapa corresponde a los equipos encargados de la alarma y detección de usuarios y la segunda etapa encargada de presentar, mediante una interfaz gráfica, la cantidad de usuarios y redireccionar la información hacia una plataforma IoT.

3.3.1 Equipos seleccionados para la primera etapa

A continuación, se describen los equipos de elección para el diseño de la primera etapa del prototipo.

3.3.1.1 Arduino Mega 2560 V3

La tarjeta de microcontrol elegida para el diseño del proyecto es Arduino, la cual ofrece una amplia flexibilidad en lo que se refiere a hardware y software. Ahora bien, el microcontrolador seleccionado es el Mega 2560 V3 debido a las siguientes características:

- Su capacidad de memoria de 256 Kb permite el almacenamiento de códigos extensos y múltiples librerías.
- Posee múltiples entradas digitales y analógicas, además de 4 puertos de transmisión (Tx) -recepción (Rx), idóneas para comunicaciones de tipo UART.
- Los puertos de alimentación de 5V y 3.3V compatibles con la mayoría de sensores y módulos electrónicos.
- Consumo energético de 93 mA/h bajo.

En la Ilustración 3.5, se presenta el Arduino Mega 2560 V3:

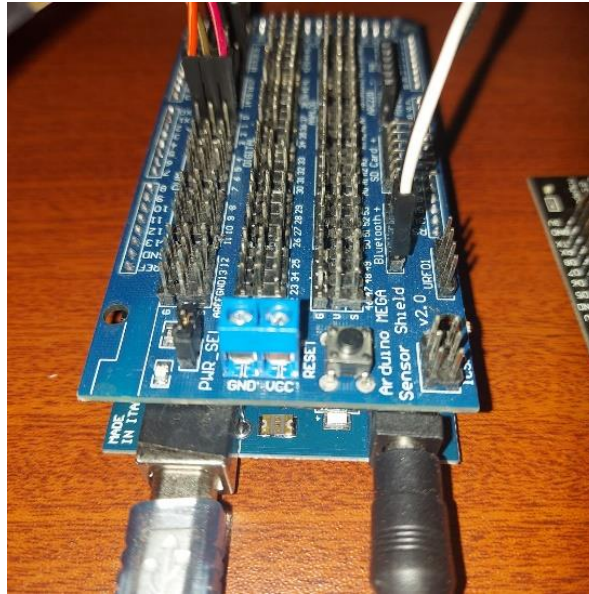


Ilustración 3.5. Arduino Mega 2560

3.3.1.2 Sensor de Proximidad

Luego de realizar un estudio respecto a la calidad y precio de distintos tipos de sensores de proximidad, se elige el sensor LIDAR, el cual ofrece ventajas en su gama de productos, ya que su precio es relativamente menor en comparación con sensores de tipo radar.

El campo de aplicaciones del sensor LIDAR, se enfoca en la seguridad en vehículos, automatización y mapeo de superficies, las cuales van acorde al tipo de aplicación del proyecto (Seguridad vial). El modelo de elección, ofrece las siguientes ventajas:

- Su rango de detección va de 0 a 40 metros, distancia suficiente para detectar la proximidad de objetos lejanos.
- Haz de luz con apertura angular de 3 grados, para abarcar todo carril de ciclovía y no obtener falsas detecciones con los vehículos que se aproximen de forma paralela.
- Soporta enlaces tipo UART facilitando la comunicación con microcontroladores.
- Alimentación menor o igual a 5V y de bajo consumo energético.

El sensor LIDAR de la compañía Leddar Tech, se observa en la Ilustración 3.6:

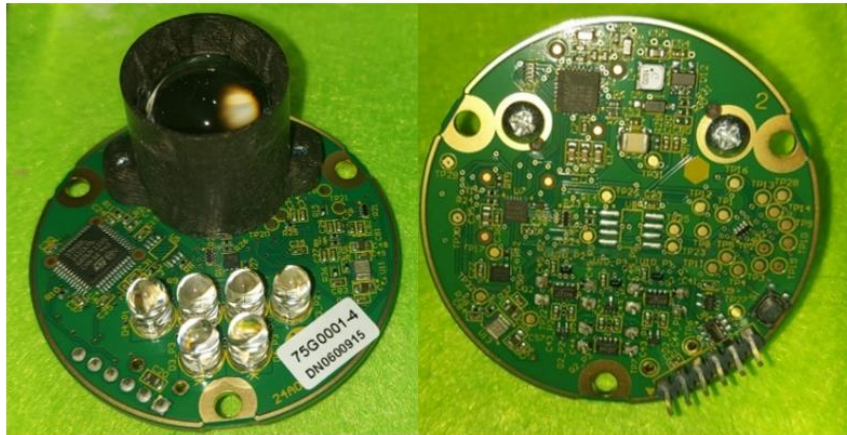


Ilustración 3.6. Sensor Leddar One de Leddar Tech.

3.3.1.3 Módulo Relevador 5v

El dispositivo electrónico de la Ilustración 3.7, permite controlar mecanismos por medio de una señal de entrada, ideal para el encendido de luces y equipos eléctricos. Sus características son las siguientes:

- Señal de control entre 3.3 y 5 voltios.
- Límite de voltaje DC 30V/10A y AC 250V/15A.
- Dimensiones de 3.4cm x 2.6cm x 1.9cm.

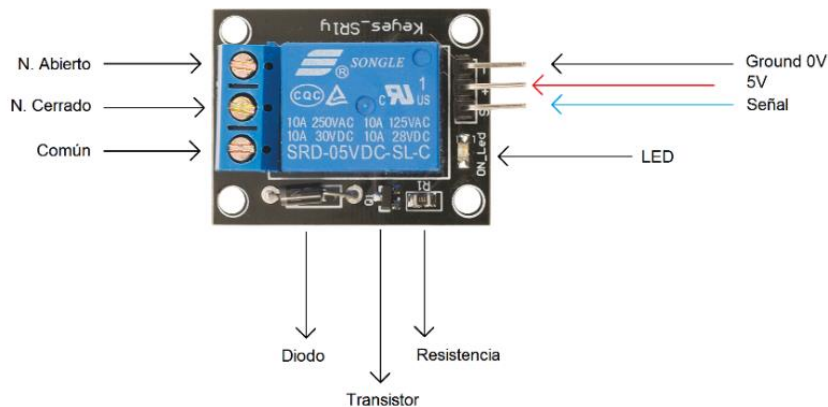


Ilustración 3.7. Módulo Relevador 5v [76]

3.3.2 Equipos seleccionados para la segunda etapa.

Para la siguiente sección, se detallan los equipos de elección para el diseño de la segunda etapa del prototipo correspondiente al registro de usuarios que hacen uso de las ciclovías.

3.3.2.1 Módulo ESP32

La elección del módulo de comunicación inalámbrica, se debe a la estructura de sus núcleos ya que utiliza uno de manera exclusiva para la ejecución de procesos Wifi. Además, los puertos de propósito general (GPIO) permite la asignación de entradas o salidas para establecer convertidores digitales- analógicas o viceversa sin la necesidad de incorporar recursos o módulos adicionales y las alternativas de conexión a internet pueden ser compatible con protocolos de mensajería IoT MQTT o HTTP. En la Ilustración 3.8, se visualiza el dispositivo que se ha descrito, el cual presenta las siguientes características:

- Consumo de Corriente de 150 miliamperios al establecer conexión a Internet
- Alimentación de 5 Voltios al establecer conexión a Internet
- Procesamiento de 240 MHz en sus núcleos
- Protocolo de comunicación inalámbrica 802.11

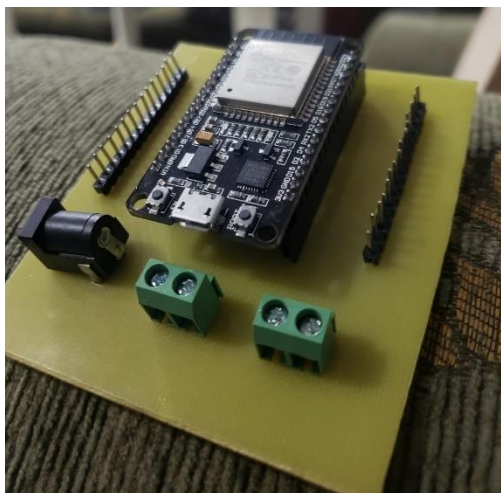


Ilustración 3.8. Módulo Wifi-Bluetooth ESP-32

3.3.2.2 Sensor E18-D80NK

Es un sensor fotoeléctrico de tipo retro reflectivo difuso, el cual no necesita de un panel reflexivo para su operación ya que cada vez que un objeto es detectado, éste actúa como un reflector, devolviendo el haz de luz emitido hacia el sistema receptor incorporado en el mismo sensor.

El sensor, visualizado en la Ilustración 3.9, otorga niveles de voltaje según la lógica de operación digital cuando identifica la detección de obstáculo en su campo de acción. Entre sus características principales se tiene:

- Alimentación de 5 Voltios.
- Rango de detección regulable entre 3- 80 centímetros.
- Led indicador de detección.
- Señal de salida tipo NPN.
- Tiempo de respuesta de 2 milisegundos.
- Consumo de 20 miliamperios



Ilustración 3.9. Sensor fotoeléctrico E18-D80NK.

3.3.2.3 Plataforma Ubidots

La plataforma Ubidots destaca de las demás opciones de gestión de datos en la nube debido a la compatibilidad con diversos módulos y tarjetas microcontroladoras que se han elegido previamente. La creación e inclusión de bibliotecas y funciones en un algoritmo permite establecer comunicación entre los dispositivos anfitriones.

Además, desde la plataforma IoT de la Ilustración 3.10, se visualizan los datos procesados en tiempo real, permitiendo el diseño de gráficas en función de la información almacenada y configurar condiciones sobre parámetros como mensajería de alertas IoT.

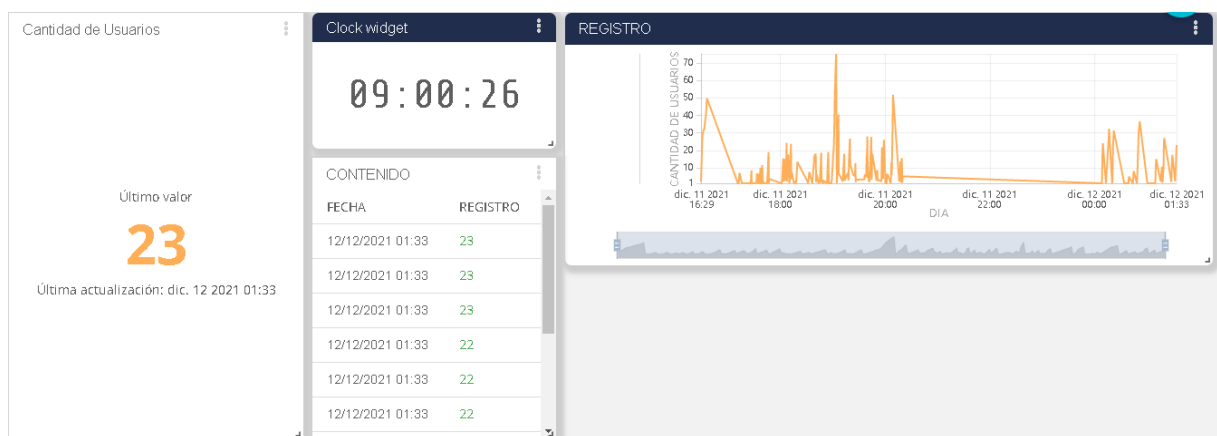


Ilustración 3.10. Panel de gestión de información en Ubidots

3.4 Construcción del prototipo.

En la sección 3.4, se describe el **algoritmo** empleado para las 2 etapas del prototipo: la primera etapa correspondiente a la detección-alarma de usuarios y la segunda etapa referente al conteo de usuarios de usuarios y su conexión a internet para el procesamiento de información.

Además, se presentan las configuraciones y conexiones establecidas con los equipos seleccionados para el diseño del prototipo.

3.4.1 Algoritmo y funciones establecidas para la primera etapa

En la Ilustración 3.11, se muestra el código inicial de esta etapa, donde se declara la librería compatible con el sensor LIDAR junto a las variables de interés para obtener los parámetros de distancia, velocidad, tiempo en los usuarios de la ciclovía que se aproximen a la intersección. Por otro lado, se tiene los pines para las señales de salida, que permita activar un led como identificador de que el equipo se encuentre en funcionamiento.

```
.
#include <Leddar.h>
#include <SoftwareSerial.h>
LeddarOne Leddar1(115200, 1);
#define rxPin 2
#define txPin 3
int SENAL = 13;
int LED = 8;
unsigned int Distancia1 = 0;
unsigned int Distancia2 = 0;
int velocidad =0 ;
float tiempo=0;
int contador=0;
SoftwareSerial Serial2(rxPin, txPin);
void setup()
{
  Serial.begin(115200);
  Serial2.begin(115200);
  Leddar1.init();
  pinMode(SENAL, OUTPUT);
  pinMode(LED, OUTPUT);
  pinMode (txPin , OUTPUT);
}
```

Ilustración 3.11 Declaración de librerías y señales de salida.

En la Ilustración 3.12, se da inicio al sensor LIDAR, obteniendo muestras de distancias cada 400 milisegundos. Estas distancias son almacenadas en las variables

distancia 1 (d_1) y distancia 2 (d_2), las cuales permiten obtener la velocidad instantánea (v) por medio de la Ecuación 3.1:

$$v = \frac{d_2 - d_1}{t_2 - t_1} \quad \text{Ecuación 3.1}$$

Donde la diferencia de tiempo $t_2 - t_1$, corresponde al intervalo en que se toma las muestras de distancia, 400 milisegundos. Cabe mencionar, que el cálculo de la velocidad se debe multiplicar por el factor de calibración calculado anteriormente, obteniendo resultados con unidades en kilómetros/hora.

Además, se calcula el tiempo en que el usuario llegará a la intersección por medio de la Ecuación 3.2, donde se utiliza la variable de distancia 2 (d_2), la cual es la muestra de distancia más próxima a la intersección, dividida para la velocidad instantánea del usuario u objeto.

$$t = \frac{d_2}{v} \quad \text{Ecuación 3.2}$$

```
void loop() {

char Resultado = Leddar1.getDetections();
if (Resultado >= 0)
{
// digitalWrite(LED, HIGH);
Distancia1 = (Leddar1.Detections[0].Distance);
Leddar1.clearDetections();
delay(400); // Muestras de distancia cada 50 milisegundos
Leddar1.getDetections();
Distancia2 = Leddar1.Detections[0].Distance;
velocidad=(Distancia1-Distancia2)*0.00875685;
tiempo= ((Distancia2/1027.766342)/(velocidad/3.6));
Serial.print("Velocidad="+ String(velocidad));
Serial.print("Tiempo de llegada="+ String(tiempo));
Serial.print("distancia2="+ String(Distancia2/1027.766342));
Serial.print("distancia1="+ String(Distancia1/1027.766342));

// ...
}
```

Ilustración 3.12 Cálculo de distancia, tiempo y velocidad de los usuarios.

Para establecer la condición de encendido de la alarma se realizaron pruebas experimentales, donde se tomó distintas muestras de tiempo y velocidad para una bicicleta que se aproxime a un cruce de calles con la ciclovía.

Con las pruebas realizadas, se obtuvo un tiempo promedio en el que ciclista llega al cruce correspondiente a 5.3376 segundos como se presenta en la Tabla 3.1, por lo cual, la condición para activar la alarma visual será cuando un ciclista se encuentre a un tiempo menor de 5 segundos en llegar al punto de interés.

Tabla 3.1 Resultado de pruebas experimentales para la velocidad y tiempo de llegada

Distancia 15m	Tiempo de llegada s	velocidad m/s	velocidad km/h
Promedio de muestras	5,3376	2,844435481	10,23996773

Cuando esta condición se cumple, se envía una señal de activación al módulo relevador como se observa en la Ilustración 3.13, que permitirá que las luces parpadeen cada medio segundo.

```

if ((tiempo>0 && tiempo < 5)//Se activa la alarma si el obstaculo llegara
al cruce en menos de 5 segundos
{
    for (int i=0; i<6; i++) //La alarma se mntendra activa por un tiempo
aproximado de 5 segundos
    {
        digitalWrite(SENAL, LOW);
        delay(400);
        digitalWrite(SENAL, HIGH);
        delay(400 );
    }
}
else if (tiempo =0 && tiempo > 5)//La alarma se mantiene apagada si el
usuario detectado se encuentra a mas de 5 segundos
//de llegar al cruce
{
    digitalWrite(SENAL, LOW);
}
}
delay(25);
}

```

Ilustración 3.13 Condición para encender la alarma visual.

Por otro lado, este algoritmo de alarma se encuentra condicionado a activarse si la velocidad del usuario es mayor a 0 km/h, evitando así, un encendido innecesario para un objeto que se encuentre detenido en el rango de detección de sensor. Además,

se envían los datos de velocidad por la comunicación Serial2 hacia el microcontrolador ESP8266, para posteriormente ser almacenados en la plataforma Ubidots.

3.4.2 Diagrama de Flujo de la conexión a Internet para la primera etapa

El diagrama de flujo que se presenta en la Ilustración 3.14, detalla 3 sub-procesos de funcionamiento para la etapa de detección; la primera corresponde a la configuración de parámetros de red para el reconocimiento del módulo y su respuesta con el servidor como se observa en la Ilustración 3.15.

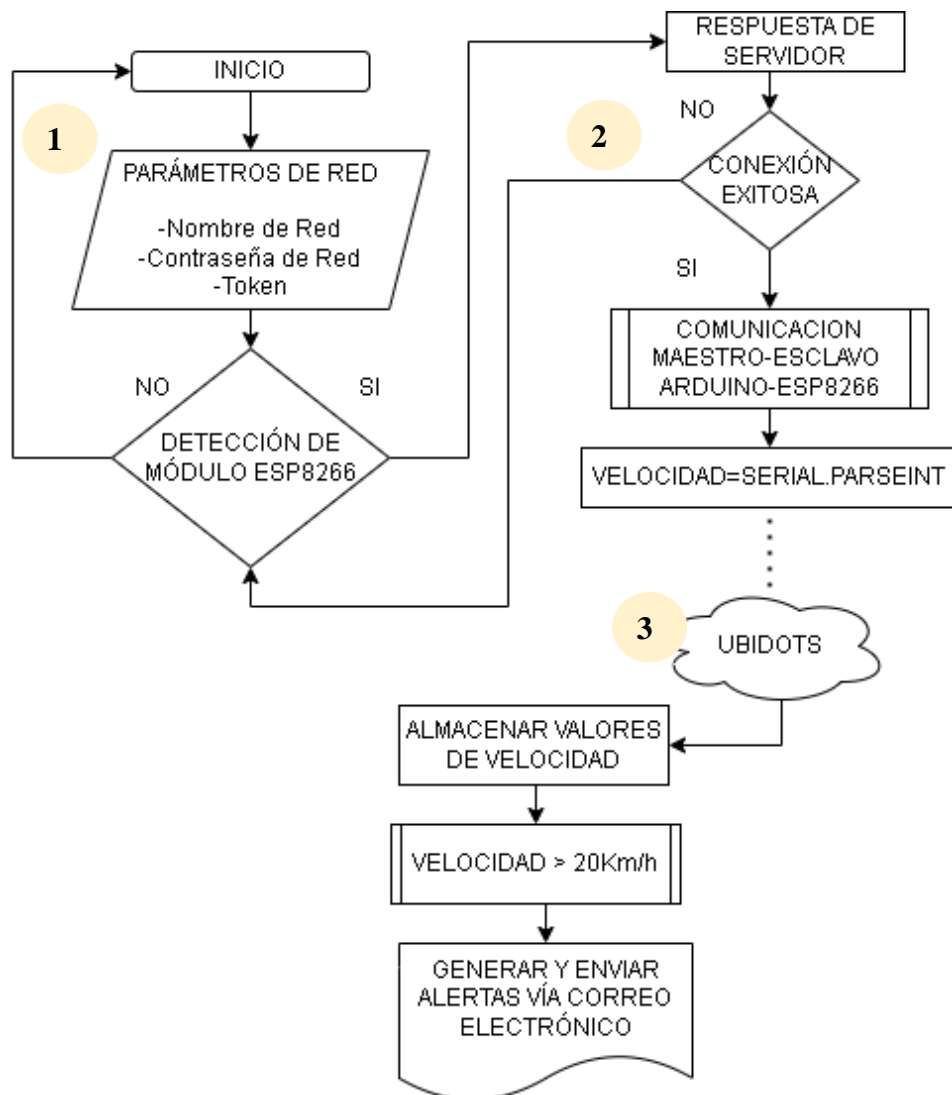


Ilustración 3.14. Diagrama de Flujo de la conexión a la red y notificaciones

```

#define DEVICE "ecfab08218a"
#define VARIABLE "Alerta de velocidad"
#define ID_TEMP "61aa4f391d847252b558635b"
#define TOKEN "BBFF-czomeTZKnviDusr6z1McDvt6PtSkO"
#define WIFISSID "NETLIFE-LAPRINCESA"
#define PASSWORD "G1971N1996S2006#" /

```

Ilustración 3.15. Parámetros de red para la comunicación con Ubidots

El segundo subproceso involucra la transferencia de información que se presenta en el monitor serial entre la unidad de microcontrol y el módulo de comunicación a internet ESP8266 con la finalidad de almacenar y publicar los datos en la plataforma Ubidots, como se presenta en la Ilustración 3.16, correspondiente al tercer subproceso del diagrama de flujo.

```

if (Serial.available())
{
  //Serial.write(Serial.read());
  velocidad=Serial.parseInt();
  ubidots.add(VARIABLE, velocidad);
  ubidots.send();
  Serial.println(velocidad);
  delay(50);
}

```

Ilustración 3.16 Publicación de parámetro de velocidad en Ubidots

A partir de los datos de velocidades censadas de los usuarios que se almacenan en Ubidots, es posible establecer condiciones. Como se visualiza en la Ilustración 3.17, se define una condición para las velocidades en donde si se cumple este escenario, se procede a generar una alerta y enviarla vía correo electrónico.

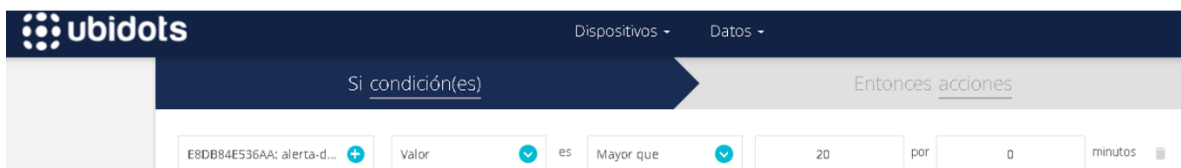


Ilustración 3.17 Configuración de la condición de alertas en Ubidots

3.4.3 Diagrama de Flujo de la conexión a Internet para la segunda etapa

De forma análoga al diagrama de flujo que se presenta en la Ilustración 3.14, se tiene un diagrama de conexión a internet para el conteo de usuarios, el cual contiene 3 sub-procesos para su funcionamiento:

El primero corresponde a la configuración de parámetros de red para el reconocimiento del módulo y su respuesta con el servidor, la segunda involucra el proceso de detección del sensor infrarrojo cuando exista un cambio en los niveles lógicos (Alto o Bajo) y por último la comunicación y publicación de los datos en la plataforma IoT como se observa en la Ilustración 3.18.

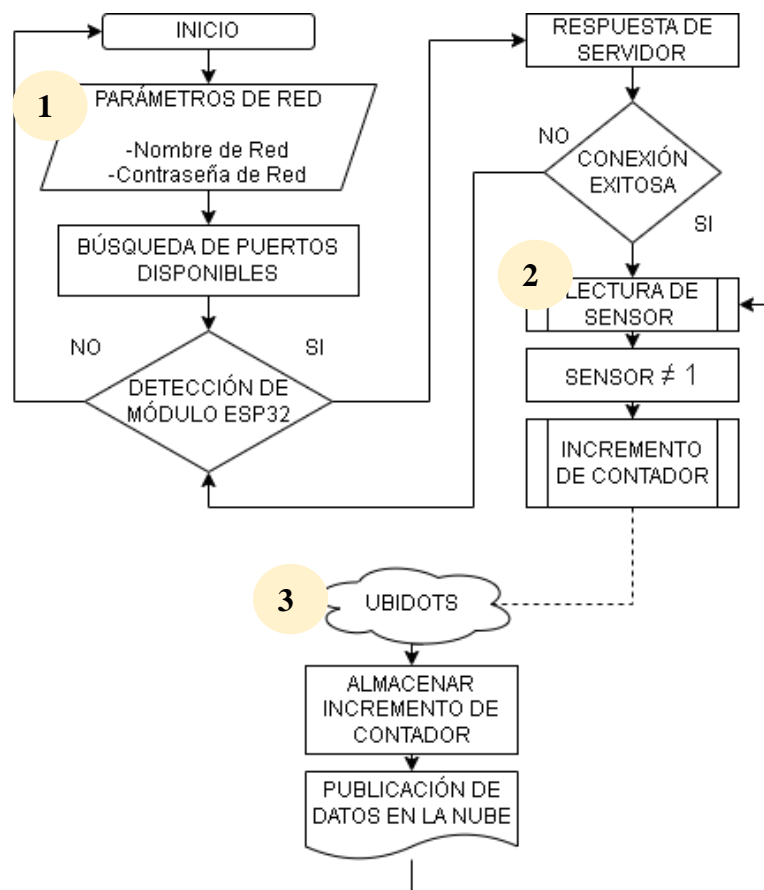


Ilustración 3.18. Diagrama de Flujo de la conexión a la red y publicación de datos

La primera etapa necesita definir la información básica de la red y de la plataforma Ubidots como se observa en la Ilustración 3.19 en donde se define el nombre de la red, la contraseña de la red, un identificador de autenticación de solicitud denominado *Token* y el canal o variable en donde se pretende almacenar los datos procesados desde el módulo ESP32.

```
namespace {
  const char * WIFISSID = "NETLIFE-LAPRINCESA";
  const char * PASSWORD = "G1971N1996S2006#";
  const char * TOKEN = "BBFF-czomeTZKnviDusr6z1MsCDvt6PtSk0";
  const char * VARIABLE_LABEL_1 = "Numero";
  const char * DEVICE_LABEL = "ESP8266";
  const int PUBLISH_FREQ = 1000;
}
```

Ilustración 3.19. Parámetros de red para la comunicación con Ubidots

La segunda etapa se enfoca en el comportamiento del sensor infrarrojo y su interpretación en el algoritmo en donde, una vez que se ha establecido conexión con la red, comienza con las lecturas de niveles de lógica digital.

La última etapa del diagrama de flujo de la Ilustración 3.18, corresponde a la publicación de los datos cuando se satisfaga la condición de identificación de valores lógicos bajos equivalente a la detección de objetos. La Ilustración 3.20, detalla las funciones de adición y publicación de cada información válida para la condición del algoritmo en los canales o variables creados en la plataforma Ubidots.

```
if(digitalRead(pin) != 1)
{
  conteo++;
  while(digitalRead(pin) != 1)
  {
    ubidots.add(VARIABLE_LABEL_1, conteo);
    ubidots.publish(DEVICE_LABEL);
    delay(100);
  }
}
```

Ilustración 3.20 Publicación de datos en el servidor IoT

Ubidots agrega, de forma continua, el registro de los usuarios detectados durante el día en donde, los datos pueden ser representados mediante gráficos históricos visualizados en la Ilustración 3.21.

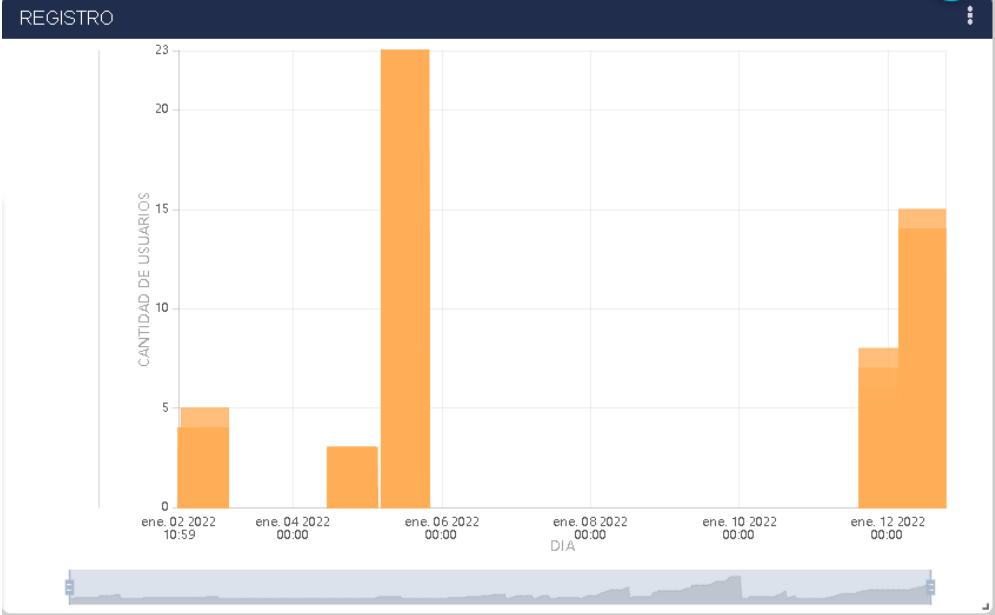


Ilustración 3.21 Visualización de datos en el servidor IoT

3.4.4 Diagrama de Flujo del comportamiento del sensor infrarrojo para la segunda etapa

Respecto al sistema de conteo de usuarios, se establecen 2 subprocesos: una correspondiente al registro de usuarios en tiempo real que utilizan el carril y la restante, al total de usuarios que se acumulan al finalizar cada día.

El diagrama de flujo de la Ilustración 3.22, representa el funcionamiento del algoritmo empleado, en donde la funcional básica, es el conteo de usuarios cada vez que el sensor detecta una variación en sus respuestas digitales.

Si el sensor infrarrojo detecta un usuario, la unidad programable cambia la lógica de detección, es decir, mientras se encuentra en funcionamiento sus lecturas son valores digitales altos (1) y cuando existe la presencia de ciclista, las lecturas digitales

cambian su valor digital a bajo (0) generando un incremento interno en el sistema, cada vez que identifique este comportamiento.

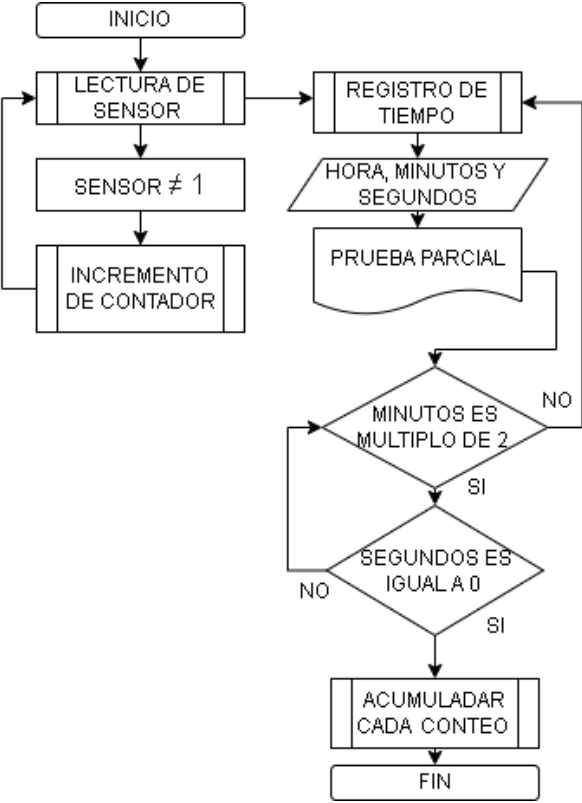


Ilustración 3.22 Diagrama de Flujo para el registro de cantidad de usuarios

Para el desarrollo del acumulador de registros por día y su respectiva visualización, se incorpora condiciones en función del módulo reloj (RTC) y las variables de horas, minutos y segundos como se observa en la Ilustración 3.23.

El módulo RTC permite la configuración de la hora inicial desde que se encuentra el funcionamiento el dispositivo y mediante las librerías que se incorporan en el algoritmo es posible almacenar información de los parámetros de horas, minutos y segundos.

La condición para el acumulador del número de ciclistas totales, se basa en que cada vez que el reloj registre la hora 00:00:00, el acumulador guarde los valores que se han detectado en tiempo real, correspondiente a la cantidad usuarios que han circulado durante el día y posteriormente resetee dicho contador.

```

if(digitalRead(pin)!=1){
  conteo++;

  while(digitalRead(pin)!=1){
    delay(1000);
  }
}
if (conteo <10 && conteo >0 || conteo == 0){
  lcd.setCursor(11,0);
  lcd.print("0");
  lcd.setCursor(12,0);
  lcd.print(conteo);
}
else {
  lcd.setCursor(11,0);
  lcd.print(conteo);
}

if (now.minute() %2 ==0 && now.second() == 0){
  totales+=conteo;
  conteo =0;
  lcd.setCursor(11,1);
  lcd.print(totales);
}

```

Ilustración 3.23 Algoritmo con condiciones para el conteo de usuarios

3.4.5 Diagrama de Conexiones

A continuación, se detallan las respectivas conexiones físicas efectuadas en la construcción del prototipo detallando los módulos, tarjetas de control y puertos utilizados para la primera etapa de detección de distancias mediante sensores LIDAR y la segunda etapa correspondiente al conteo de usuarios mediante sensores infrarrojos.

3.4.5.1 Diagrama de Conexiones de la primera etapa.

En la Ilustración 3.24, se observa la estructura y conexiones de la etapa de detección y alarma, donde por medio del sensor LIDAR se enviará información de distancia de los usuarios en la ciclovía; dicha información es procesada por el algoritmo programado en el microcontrolador para posteriormente ejecutar funciones o acciones en equipos secundarios.

La primera función, se trata de un led que se mantendrá encendido siempre y cuando el sensor detecte obstáculos, lo que permite corroborar si el equipo se mantiene activo o inactivo.

Como siguiente punto, se tiene el módulo relevador, el cual activa el paso de energía a la tira led cuando se cumplan las condiciones de distancia, velocidad y tiempo de los usuarios que se aproximen. Finalmente se tiene la conexión entre el Arduino Mega y la ESP8266 con el objetivo de poder enviar los datos procesados hacia internet.

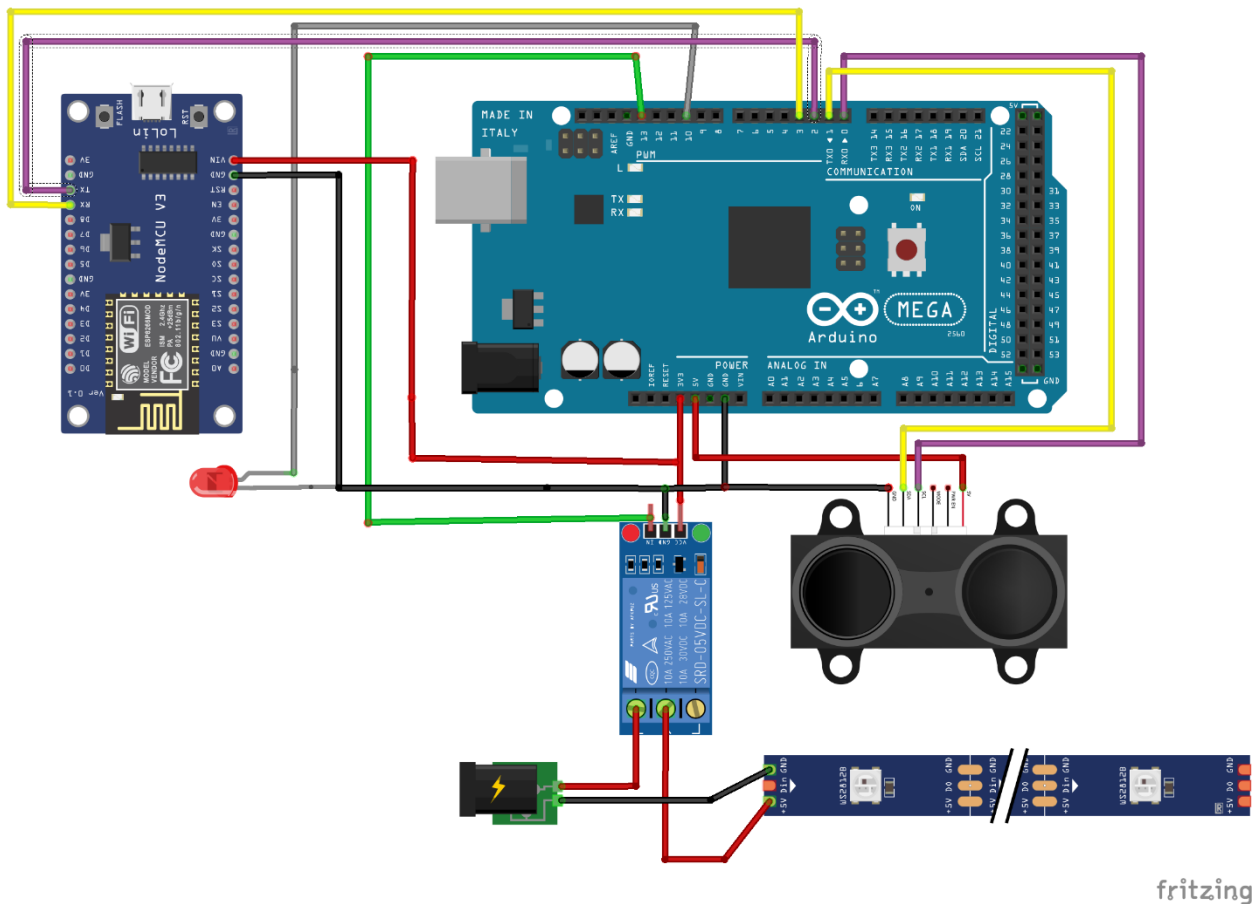


Ilustración 3.24 Diagrama de conexiones de la etapa de detección y alarma.

En la Tabla 3.2, se tiene las respectivas conexiones entre los equipos de la primera etapa, donde tanto el módulo relevador y el sensor LIDAR serán alimentados por medio de los puertos de 5V del Arduino Mega y la ESP8266 por el puerto de 3.3V.

Por otra parte, la tira led tiene una alimentación independiente al microcontrolador de Arduino, la cual es controlada por la señal GPIO13 que activa o desactiva el por el módulo relevador.

Además, se tiene una comunicación tipo UART desde la Arduino Mega hacia el sensor LIDAR y la ESP8266 por medio de los puertos TX/RX. Finalmente, el puerto de reinicio en el sensor se mantendrá inactivo debido a que su activación implicaría un nuevo cálculo del factor de calibración.

Tabla 3.2 Conexiones entre sensores, módulos y microcontrolador.

	Pines	Microcontrolador
Módulo Relevador	VCC	5V
	GND	GND
	Vi	GPIO13
	NA	12V
Módulo LEDDAR	VCC	5V
	GND	GND
	RESET	-
	TX	RX1
	RX	TX1
Led	Ánodo	GPIO10
	Cátodo	GND
Tira Led	-	Relevador
	VCC	NC
	GND	GND
	3V	3.3V
ESP8266	GND	GND
	TX	GPIO2
	RX	GPIO3
	Alimentación	6V/600mA

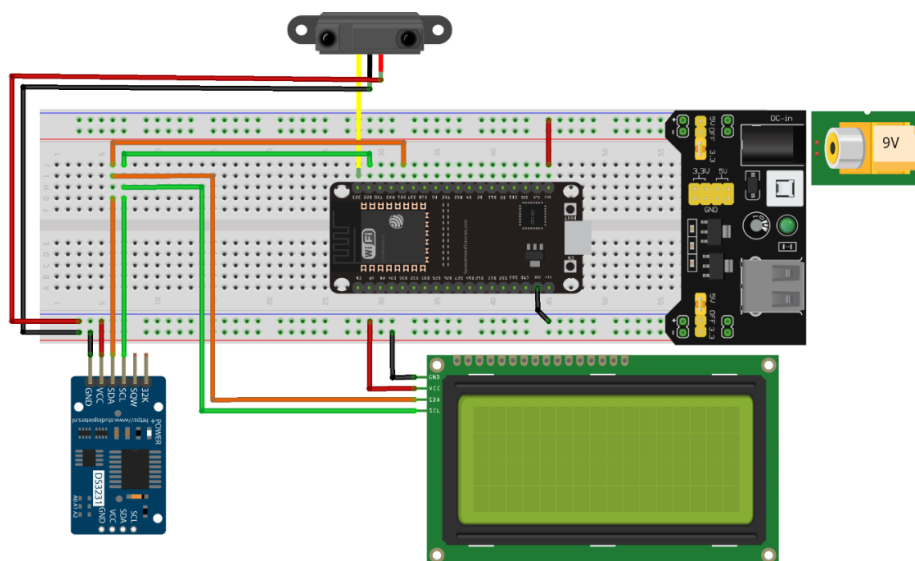
3.4.5.2 Diagrama de Conexiones de la segunda etapa.

En la Ilustración 3.25, se observa la estructura y conexiones de la etapa de conteo de usuarios, donde se utiliza una fuente de alimentación externa que permita suministrar un voltaje de operación, en función de las específicas técnicas de los equipos o módulos, con la finalidad de no causa averías de los mismos.

El módulo de control corresponde al equipo ESP32, el cual permite procesar las lecturas del sensor infrarrojo a través de sus puertos de entradas/salidas digitales. Cada vez que el sensor infrarrojo detecte un usuario, el algoritmo del ESP32 obtiene un nivel lógico bajo (0), el cual se incrementa en uno por cada vez que se presente este escenario y los datos son visualizados en la pantalla LCD.

El almacenamiento en la nube se debe al protocolo TCP/IP que maneja la ESP32 y las librerías de Ubidots incorporadas en el algoritmo como publicación de datos mediante el estándar de mensajes IoT (MQTT).

Como siguiente punto, se incorpora un módulo de reloj (RTC), para agregar la función de registro de usuarios totales, el cual acumula los valores que se obtienen a diario. La comunicación I2C de la pantalla LCD y del módulo RTC permite simplificar las conexiones físicas entre el módulo de reloj y la ESP32.



fritzing

Ilustración 3.25 Diagrama de conexiones de la etapa de conteo de usuarios.

En la Tabla 3.3, se tiene las respectivas conexiones entre los equipos de la segunda etapa donde, tanto el módulo RTC, la pantalla LCD y el sensor infrarrojo, serán alimentados por medio de los puertos de 5V de la fuente de alimentación externa. Por otra parte, el microcontrolador ESP32 tiene un voltaje de operación de 3.3V y la señal de control con el sensor se establece en el puerto GPIO23, para el envío de valores lógicos (altos o bajos).

Tabla 3.3 Conexiones entre sensores, módulos y microcontrolador.

	Pines	Microcontrolador
Módulo LCD	VCC	5V
	GND	GND
	SCL	D22
	SDA	D21
Módulo RTC	VCC	5V
	GND	GND
	SCL	D23
	SDA	D21
ESP32	Alimentación	3.3V/225mA
Sensor Infrarrojo	VCC	5V
	GND	GND
	SALIDA	D23

3.4.6 Implementación del prototipo.

Con la incorporación del sensor LIDAR se prevé, la recolección de valores de velocidad en función de la medición de distancia sin embargo se debe realizar previamente una calibración del dispositivo con la finalidad de trabajar con datos que conlleven el mínimo error posible.

3.4.6.1 Calibración de Sensor LIDAR

Por tal motivo, se registraron distancias teóricas y muestras experimentales bajo los dos siguientes escenarios. El primero, caracterizado por trabajar con distancias cortas entre un rango de [0 – 100] cm como se visualiza en la Ilustración 3.26.



Ilustración 3.26. Detección de objeto a 60 centímetros.

El segundo escenario corresponde a la medición de distancias entre [1-5] metros como se visualiza en la Ilustración 3.27.

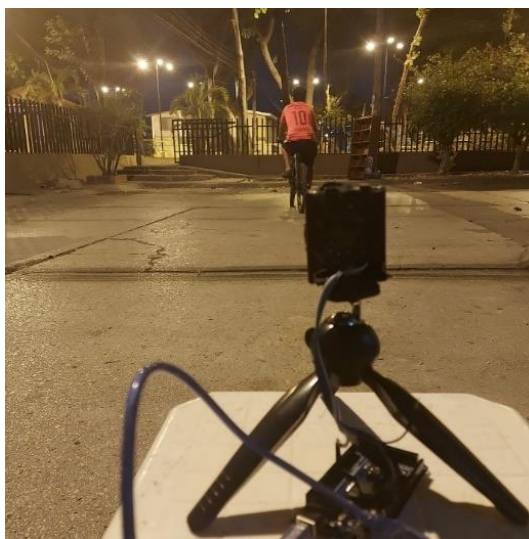


Ilustración 3.27 Detección de objeto a 4 metros

En la Ilustración 3.26 y 3.27, se observan los dos escenarios para la calibración del sensor LIDAR, en donde las muestras de recolección en cada medición son procesadas por el microcontrolador a través del puerto serie, como se visualiza en la Ilustración 3.28.

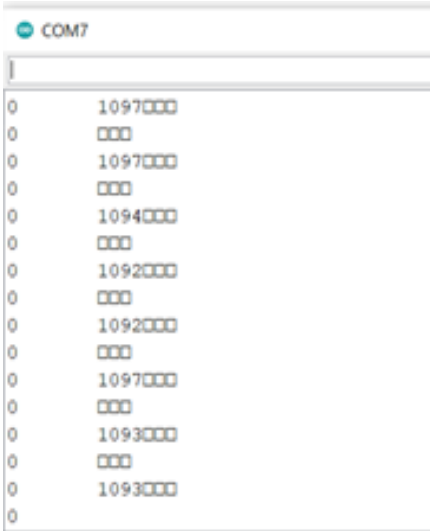


Ilustración 3.28. Muestras obtenidas por el sensor por cada medición.

Para realizar la calibración, se dio inicio con el registro de las muestras censadas para luego obtener una muestra promedio. Estas muestras promedio se linealizaron respecto a los valores reales, obteniendo así pendientes de la ecuación de una recta que relaciona el valor real versus el valor medido, el proceso aplicado se representa en Ilustración 3.29.

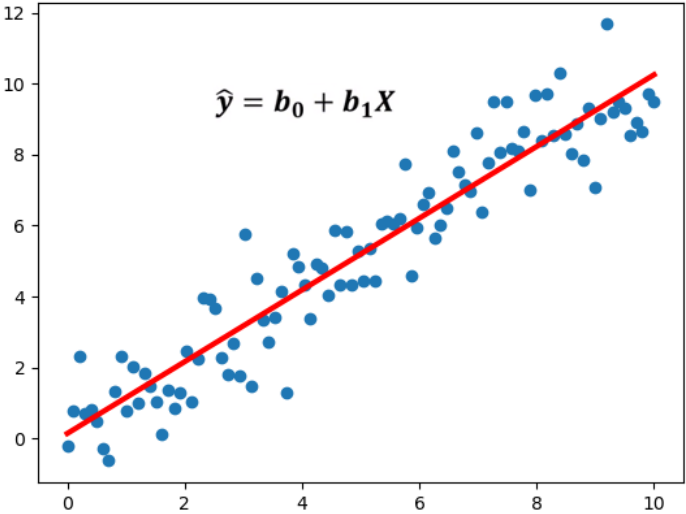


Ilustración 3.29. Respuesta Lineal entre valores reales y experimentales. [77]

Una vez obtenida las pendientes por cada muestra, se calculó un promedio de pendientes, el cual aproxima una sola función lineal entre los valores teóricos y experimentales, permitiendo que los valores obtenidos por el LIDAR se aproximen a los valores reales.

En la Tabla 3.4, se observan los datos calculados según el procedimiento descrito, donde se obtuvo 2 factores de calibración, uno promediando todas las pendientes obtenidas llamado factor de calibración 1 y otro omitiendo las muestras que presentaban un mayor margen de error a la hora de linealizar.

Tabla 3.4 Calibración con distancias entre 0 y 100 cm

Distancia Teórica [cm]		20	25	30	35	40	45	50	55	60	65	70	
Muestras Sensor	1	300	356	399	453	500	550	595	642	699	764	809	
	2	298	358	397	452	499	548	596	644	698	763	810	
	3	299	357	396	451	498	549	594	643	700	762	811	
	4	297	359	398	450	497	551	593	-	697	765	808	
	5	-	-	-	454	496	-	592	-	700	-	-	
Promedio de muestras		298,5	357,5	397,5	452	498	549,5	594	643	698,8	763,5	809,5	
Pendiente promedio de muestras		14,925	14,3	13,25	12,91	12,45	12,21	11,88	11,69	11,65	11,75	11,56	
Factor de Calibración 1 12,59803747	Corrección	23,69	28,38	31,55	35,88	39,53	43,62	47,15	51,04	55,47	60,6	64,26	Promedio
	Error	18,471%	13,51%	5,18%	2,51%	1,18%	3,07%	5,70%	7,20%	7,55%	6,76%	8,21%	7,21%
Factor de Calibración 2 12,01292652	Corrección	-	-	-	37,63	41,46	45,74	49,45	53,53	58,17	63,56	67,39	
	Error	-	-	-	7,50%	3,64%	1,65%	1,11%	2,68%	3,05%	2,22%	3,73%	3,20%
Distancias Largas		100	200	300	400	500	600	700	800	900	1000	1100	
Corrección		115	185	324	413	475	525	-	-	-	-	-	

El promedio de error obtenido en la linealización para el factor 1 es del 7.21% mientras que el factor 2 se redujo aproximadamente a la mitad con un valor de 3.20%, esto gracias a que se omitió las muestras que presentaban mayor error.

Ahora bien, se realizó pruebas con distancias superiores a los 100 cm donde los valores corregidos presentaban un error considerable, por lo cual se procedió a calcular un nuevo factor de corrección, pero con muestras de distancias superiores a los 100 cm. Las muestras tomadas fueron en el rango de [1 – 5] metros con saltos de 1 metro entre cada muestra.

El factor de calibración obtenido con distancias superiores a 1 metro en la Tabla 3.5, presentó un promedio de error menor al 1% en la corrección de las muestras del

sensor, porcentaje que representa una mejora de un 6% de error respecto a los factores obtenidos con distancias pequeñas, teniendo distancias aproximadas a las reales. Por tal motivo, se lo eligió como factor de calibración para el sensor.

Tabla 3.5 Calibración con distancias entre 1 y 5 m

Distancia Teórica [cm]	100	200	300	400	500	
Muestras Sensor	1077	2176	3357	4379	5419	
	1105	2170	3357	4375	5412	
	1098	2181	3355	4384	5419	
	1087	2185	3354	4389	5433	
	1088	2196	3356	4391	5429	
	1086	2159	3356	4386	5418	
	1091	2170	3354	4382	5415	
	1097	2200	3354	4376	5416	
	1097	2178	3357	4384	5417	
	1094	2175	3356	4401	5419	
	1092	2175	3357	4412	5422	
	1092	2177	3355	4420	5419	
	1097	2178	3355	4413	5428	
	1093	2189	3355	4388	5411	
	1093	2174	3354	4365	5408	
	Promedio de muestras	1092,4667	2178,8667	3355,4667	4389,6667	5419
Pendiente promedio de muestras	10,924667	10,894333	11,184889	10,974167	10,838	10,96321111
Corrección	99,64842	198,74347	306,06605	400,39972	494,28949	Promedio error
Error	0,35%	0,63%	2,02%	0,10%	1,14%	0,85%

3.4.6.2 Pruebas Parciales

En este apartado se procede a mostrar la estructura y funcionamiento parcial de los equipos, tanto para la etapa de detección como la encargada de presentar el número de usuarios y enlazar los datos hacia una plataforma web.

En Ilustración 3.30, se observa la estructura del prototipo de detección con sus puertos de alimentación, comunicación serial, y conector de la alarma led; luego se comprobó su funcionamiento, en donde se obtiene una medida aproximada de 2.31 metros mostrada por el monitor serial de la aplicación de Arduino, observado en Ilustración 3.31.

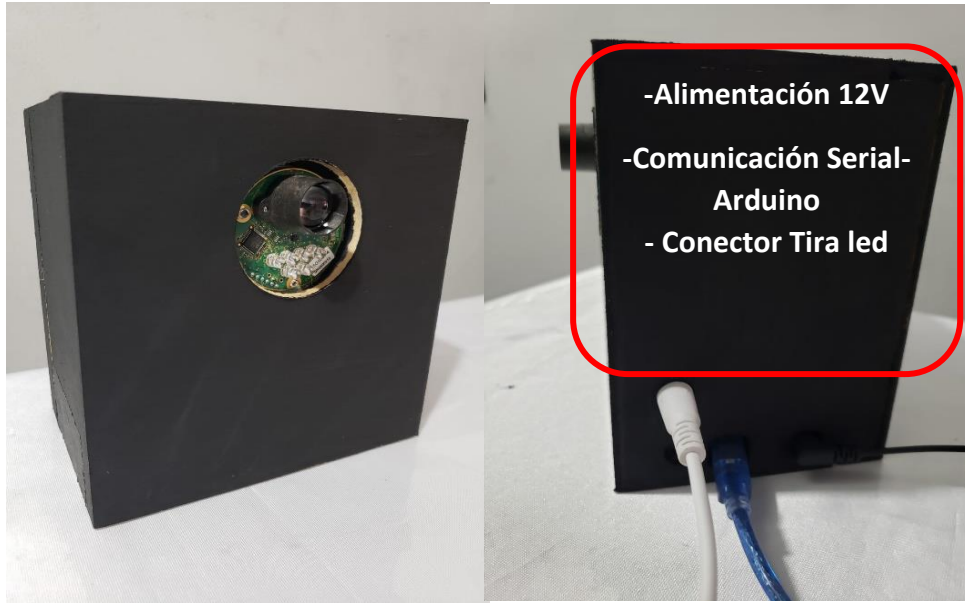


Ilustración 3.30. Construcción de prototipo de detección y alarma.

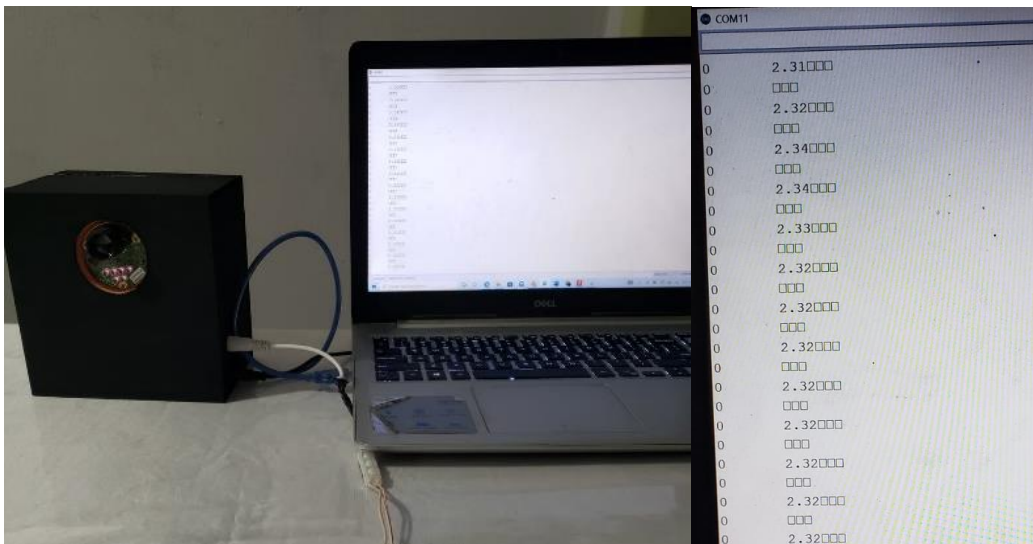


Ilustración 3.31. Detección de distancias por parte del sensor LIDAR

Para corroborar la activación de la alarma se procedió a configurar la condición de encendido para distancias menores a 1 metro. En Ilustración 3.32, se observa la activación de la tira led.

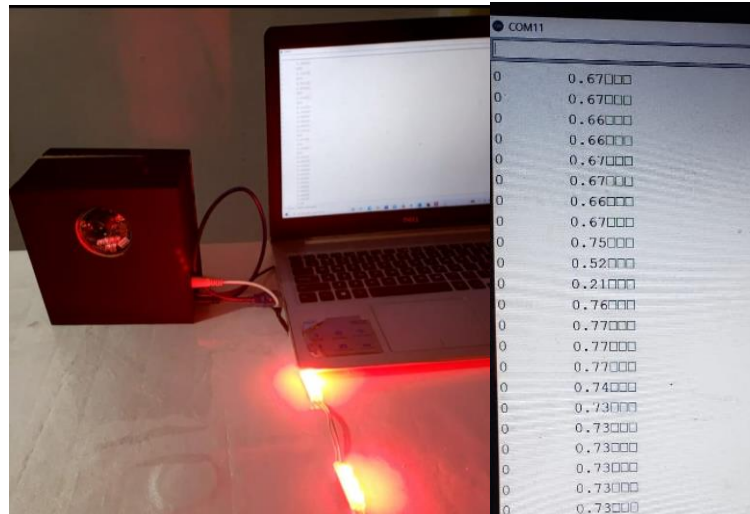


Ilustración 3.32 Alarma encendida con detecciones de distancia menor a 1 metro.

Por otra parte, en la Ilustración 3.33, se presenta el diseño del conteo de usuarios en donde se han incorporado dispositivos como fuente de alimentación, módulo de microcontrol ESP32 para la conectividad a internet, visualización de información en pantalla LCD con comunicación I2C que simplifique la cantidad de conexiones físicas, módulo de reloj para ejecutar el acumulador de registros por día y el respectivo sensor infrarrojo para la detección.



Ilustración 3.33 Construcción de prototipo de conteo de usuarios

Se procede a verificar el funcionamiento del algoritmo que se ha incorporado en el microcontrolador, en donde la Ilustración 3.34, muestra el respectivo conteo cada vez que el sensor devuelve un valor lógico bajo ya que es cuando se identifica a un usuario.



Ilustración 3.34 Construcción de prototipo de conteo de usuarios

La incorporación del módulo de reloj en el prototipo permite reducir el consumo energético del sistema debido a que se encarga de funciones de tiempo que por defecto debe realizar el microcontrolador.

En Ilustración 3.35, se puede observar que cada vez que los minutos sean divisibles para 2 y los segundos tomen un valor de cero, el contador se reinicia y el último valor es registrado en una variable de acumulación; este ejemplo permite dar una noción de la condición que se debe introducir para realizar un acumulador por día en donde el mismo comportamiento debe ejecutarse sobre las variables de hora y segundo.



Ilustración 3.35 Funcionamiento del acumulador de registros

3.5 Presupuesto

Para la construcción del prototipo se ha tenido en consideración el aspecto económico en la adquisición de cada componente utilizado. La Tabla 3.6 detalla los productos que se han requerido para la construcción y funcionamiento del sistema con sus respectivos precios. Cabe mencionar que el sensor LIDAR puede modificar su modelo y precio en función de las características de alcance y detección.

Tabla 3.6. Detalle de los componentes para la construcción del prototipo

Concepto	Cantidad	Costos por Unidad	Costos Totales
Sensor LIDAR	1	\$ 175.97	\$ 175.97
Sensor Infrarrojo	1	\$ 80	\$ 80
Arduino ATmega2560	2	\$ 19	\$ 38
Módulo ESP32	1	\$ 14	\$ 14
Módulo ESP8266	1	\$ 12	\$ 12
Módulo LCD	1	\$ 10	\$ 10
Módulo RTC	1	\$ 3	\$ 3
Módulo Fuente de Alimentación	1	\$ 5	\$ 5
Módulo Relevador 5V	1	\$ 5	\$ 5
Placa PCB ESP32	1	\$ 15	\$ 15
Soporte 3D LIDAR	1	\$ 6	\$ 6
Tira Led	1	\$ 6	\$ 6
Cajas	2	\$ 10	\$ 20
Total			\$ 400

CAPITULO 4

4 ANÁLISIS DE RESULTADOS

El presente capítulo detalla los resultados del prototipo que se ha diseñado, en donde mediante pruebas y simulaciones en campo, se obtienen tablas y gráficos con la finalidad de analizar el funcionamiento del sistema, a partir de los parámetros procesados en la detección de proximidad de usuarios y el registro en la nube de la cantidad de ciclistas censados.

En la Ilustración 4.1, se visualiza las componentes del sistema correspondientes a la etapa de detección de proximidad distancia identificada en el recuadro 1, la activación de la alerta visual ubicada en el recuadro 2 y la etapa de conteo de bicicletas para el envío de la información a la plataforma IoT identificada en el recuadro 3.



Ilustración 4.1 Implementación del prototipo

4.1 Rango de detección de distancias.

A partir de la Tabla 3.4 descrito en el Capítulo 3, correspondiente a las muestras obtenidas para la calibración del equipo de medición de distancias, se obtuvo un factor de corrección que permite un ajuste de los datos experimentales versus las distancias reales.

En la Ilustración 4.2, se tiene las distancias que se han procesado y corregido mediante el factor que se incorpora en el algoritmo de detección. A medida que un biciusuario se aproxime a la intersección el equipo recolecta, de forma continua, dichos valores de distancias que permiten efectuar cálculos de velocidad media para la activación de la alarma visual.

Timestamp	Distance
19:23:00.463	14.18
19:23:01.484	14.19
19:23:02.486	14.19
19:23:02.533	14.19
19:23:03.507	14.19
19:23:03.554	14.21
19:23:04.559	14.19
19:23:05.581	14.19
19:23:06.604	14.19
19:23:07.653	14.18
19:23:08.662	14.18
19:23:09.670	14.13
19:23:09.717	14.12
19:23:10.698	14.12

Timestamp	Distance
19:21:57.763	0.00
19:21:58.806	2.69
19:22:00.665	3.11
19:22:01.450	3.66
19:22:02.296	4.45
19:22:03.126	5.24
19:22:03.941	6.05
19:22:04.740	6.72
19:22:05.586	7.40
19:22:06.385	7.99
19:22:07.242	8.55
19:22:08.044	9.13
19:22:09.067	9.91
19:22:09.114	10.81
19:22:10.089	10.81

Ilustración 4.2 Distancias procesadas a medida que se aproxima un biciusuario.

La Ilustración 4.3, establece una comparativa entre los valores de distancias teóricas y las pruebas realizadas en campo. Se observa que cuando no existe un factor de corrección adecuado para la detección continua de distancias, los resultados

difieren de las distancias reales ocasionando un funcionamiento no adecuado del equipo sin embargo cuando se configura un factor correctivo existe gran similitud entre los valores reales y experimentales.

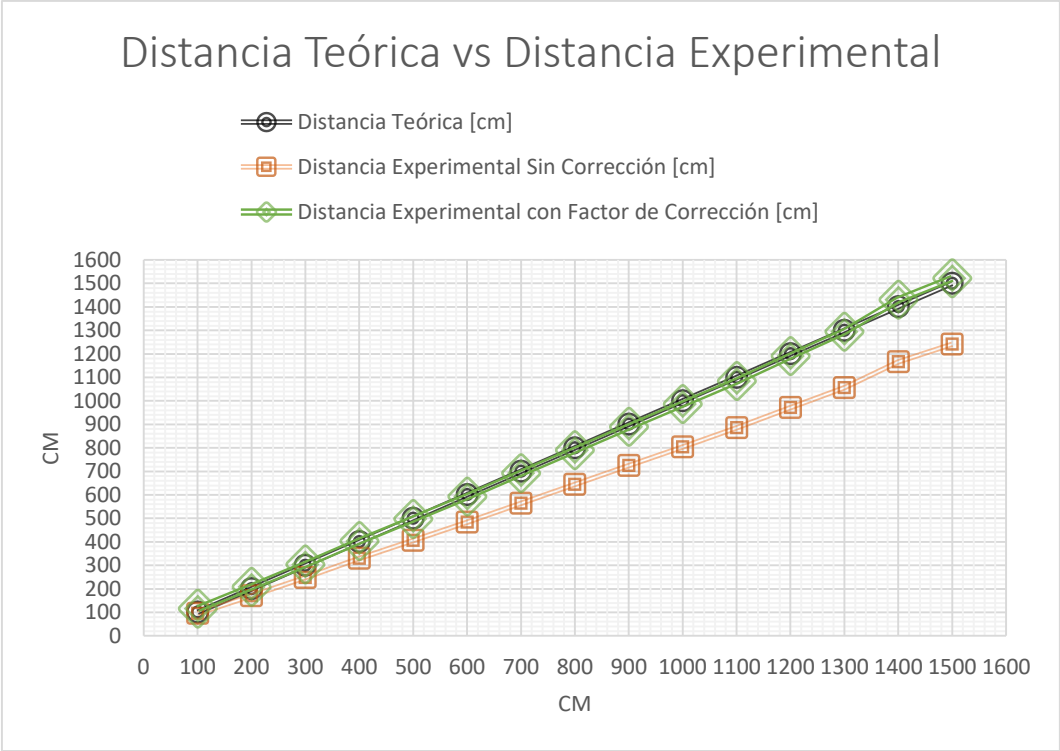


Ilustración 4.3 Comparativa entre las distancias teóricas y experimentales

El alcance de detección de distancias registrado durante las pruebas de campo, como se visualiza en la Ilustración 4.4, es menor a 15 metros debido a que un biciusuario no ocupa una gran área que permita que el haz proveniente del sensor se refleje y sea captado por el esquema receptor del sensor.

Sin embargo, se tiene en consideración que objetos que ocupan una mayor área como es el caso de vehículos, el sensor tendría un alcance de detección de aproximadamente 35 metros.



Ilustración 4.4 Alcance y detección de bicisuarios

4.2 Velocidad media de usuarios registrados.

En la Ilustración 4.5 se presenta al equipo tomando muestras de velocidad desde una distancia de 15m, la cual según las pruebas de distancias resultó ser la máxima posible en objetos medianos como bicicletas o Scooter.



Ilustración 4.5 Recolección de datos de velocidad

A partir de la Ilustración 4.6, se observan los valores procesados de velocidad en las pruebas realizadas. Además, se tiene el registro de las dos distancias detectadas, las cuales se reducen a medida que se aproximen al cruce.

```

---
Velocidad=13Tiempo de llegada=2.59distancia2=9.36distancia1=10.81
Velocidad=12Tiempo de llegada=2.24distancia2=7.45distancia1=8.83
Velocidad=13Tiempo de llegada=1.51distancia2=5.45distancia1=6.96
Velocidad=14Tiempo de llegada=0.85distancia2=3.32distancia1=4.94
Velocidad=6Tiempo de llegada=1.38distancia2=2.29distancia1=3.07
Velocidad=4Tiempo de llegada=1.38distancia2=1.53distancia1=2.04
Velocidad=1Tiempo de llegada=4.85distancia2=1.35distancia1=1.50

```

Ilustración 4.6 Velocidades procesadas a medida que se aproxima un bicusuario

A partir de la Ilustración anterior, se tomaron muestras de velocidad alcanzada por los ciclistas en 15 pruebas realizadas, las cuales se visualizan en la Ilustración 4.7 en donde se obtuvo un rango de velocidades que van desde 1Km/h hasta 14 Km/h. Por otra parte, el promedio de velocidades que se obtiene es de aproximadamente 9 Km/h, el cual se asemeja al rango de velocidades que puede alcanzar ciclistas no profesionales.

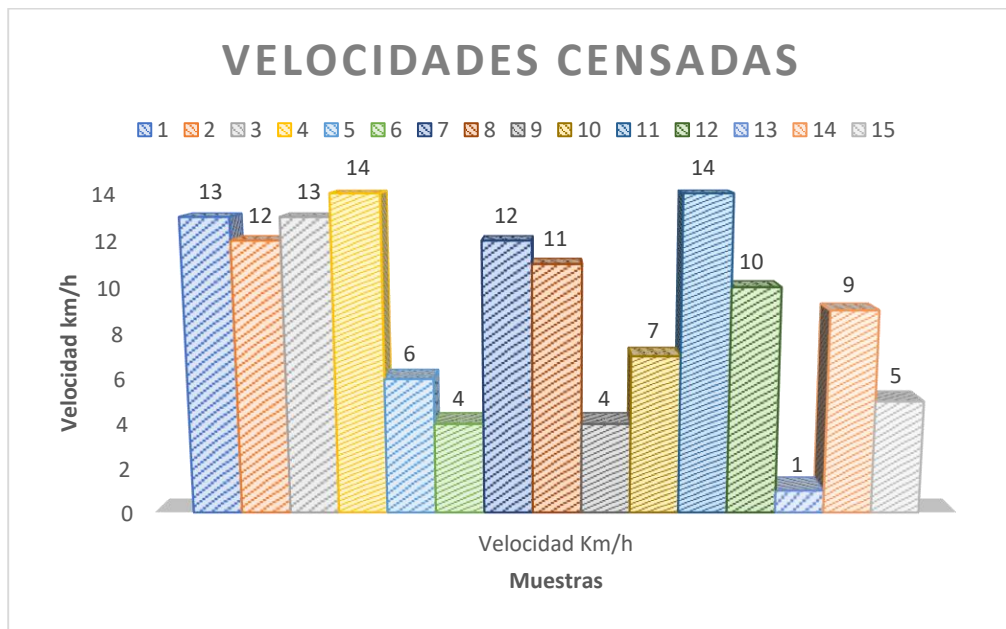


Ilustración 4.7 Velocidad alcanzadas por bicusuarios

4.3 Tiempo de llegada de usuarios

Una vez obtenida la distancia y la velocidad de los usuarios detectados, se puede calcular el tiempo que tardarán en llegar a la intersección. En la Ilustración 4.8, se puede observar un resumen de todos los parámetros detectados y según la velocidad que lleve el usuario se presentará el tiempo estimado en llegar al cruce. Cabe mencionar que cada vez que este tiempo sea menor a 5 segundos, la alarma se habilitará.

```
Velocidad=13Tiempo de llegada=2.59distancia2=9.36distancia1=10.81□□□□  
□□□  
Velocidad=12Tiempo de llegada=2.24distancia2=7.45distancia1=8.83□□□□  
□□□  
Velocidad=13Tiempo de llegada=1.51distancia2=5.45distancia1=6.96□□□□  
□□□  
Velocidad=14Tiempo de llegada=0.85distancia2=3.32distancia1=4.94□□□□  
□□□  
Velocidad=6Tiempo de llegada=1.38distancia2=2.29distancia1=3.07□□□□  
□□□  
Velocidad=4Tiempo de llegada=1.38distancia2=1.53distancia1=2.04□□□□  
□□□  
Velocidad=1Tiempo de llegada=4.85distancia2=1.35distancia1=1.50□□□□  
□□□
```

Ilustración 4.8 Tiempo de llegada a la intersección.

4.4 Habilitación de alarma visual.

En Ilustración 4.9, se presenta el funcionamiento de la alarma visual en la señalética, la cual se activa una vez que los usuarios se encuentren a 5 segundos próximo al cruce de vías. Cada vez que se cumple la condición, la alarma se mantiene en funcionamiento durante 6 segundos, permitiendo que una vez que el usuario se encuentre por detrás del sensor, la advertencia siga activa, hasta atravesar el tramo de ciclovía que se encuentra con la intersección.



Ilustración 4.9 Activación de alarma cuando se satisface la condición de proximidad

En la Ilustración 4.10, se observa el caso en donde el usuario se encuentra dentro del radio de detección y a 5 segundos próximos al cruce, pero este se detiene completamente.

En este caso, la alarma solo estará activa por 6 segundos, luego de eso se apagará hasta que el objetivo vuelva a estar en movimiento, evitando así falsas alarmas de objetos o usuarios que no se encuentran en movimiento dentro del rango de detección.

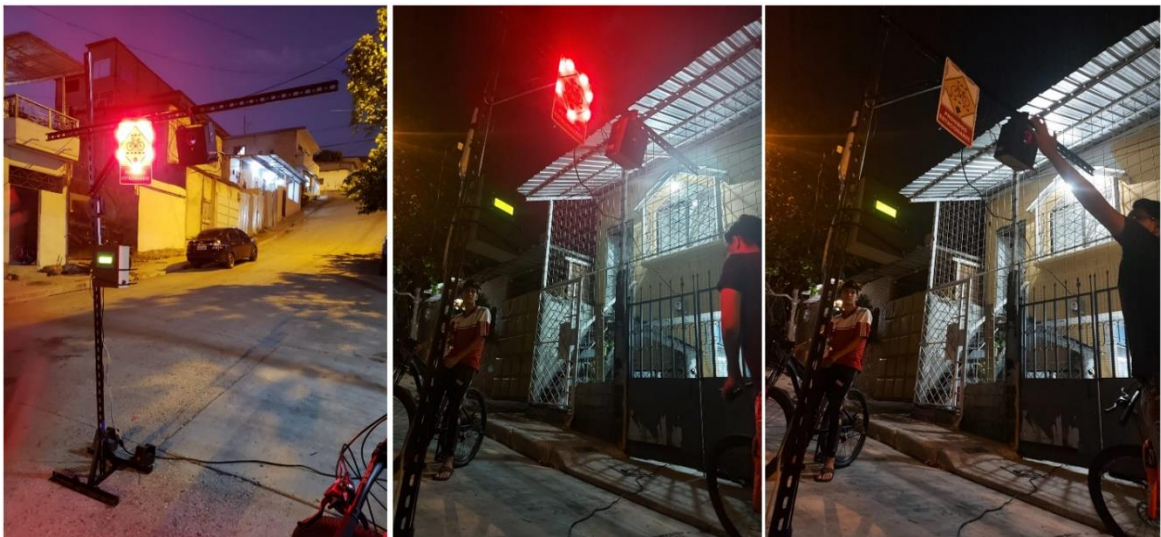


Ilustración 4.10 Activación de alarma cuando se satisface la condición de proximidad

4.5 Registro de Usuarios

La etapa de registro de usuarios presenta información correspondiente a la cantidad de usuarios que se han registrado durante el día y el total que se ha acumulado desde que se encuentra en funcionamiento el prototipo mediante un módulo LCD como se observa en la Ilustración 4.11.



Ilustración 4.11 Etapa de registro de usuarios

El sensor infrarrojo, visualizado en la Ilustración 4.12, permite la detección de usuarios debido a las lecturas digitales que proporciona a la unidad de control ESP32. Cuando un ciclista circula frente su campo de acción, las lecturas modifican su estado y este comportamiento permite realizar incrementos en las secciones de conteo y acumulación que se visualizan en el módulo LCD.

Finalmente, se muestra el formato de reloj en 24 horas, donde cada vez que se registre la hora 00:00:00, se reinicia el conteo por día de usuarios y se acumula al total.



Ilustración 4.12 Detección y registro de usuarios

4.5.1 Conectividad a la nube y registro de datos.

Cuando la etapa de conteo de bicisusuarios se encuentra en funcionamiento, el equipo establece configuraciones al obtener información de la red y la plataforma IoT como se visualiza en la Ilustración 4.13. Se tiene en consideración, una dirección de solicitud-respuesta con el servidor y la búsqueda de puertos disponibles para la autenticación de clientes que faciliten las conexiones con diferentes redes.

```

COM5
09:17:46.643 -> .....WiFi connected
09:17:50.669 -> IP address:
09:17:50.669 -> 192.168.100.134
09:17:50.669 -> broker:industrial.api.ubidots.com
09:17:50.669 -> brokerPort:1883
09:17:50.669 -> 94:B9:7E:E9:4E:9C
09:17:50.669 -> BBFF-czometZKnviDusr6z1MsCDvt6PtSk0
09:17:50.669 -> Attempting MQTT connection...connected

```

Ilustración 4.13 Configuraciones establecidas en la etapa de conteo

El registro de los usuarios censados es procesado y almacenados en la plataforma IoT de Ubidots en donde se identifica registros de la cantidad de ciclistas que transitan por día y la cantidad total almacenada desde que se encuentra en funcionamiento el equipo.

A partir de la Ilustración 4.14, se visualiza la generación de un gráfico de la cantidad de usuarios que se han censado por día con la finalidad de identificar las rutas de mayor preferencia de los usuarios y evaluar el impacto social que conlleva la implementación de ciclovías.

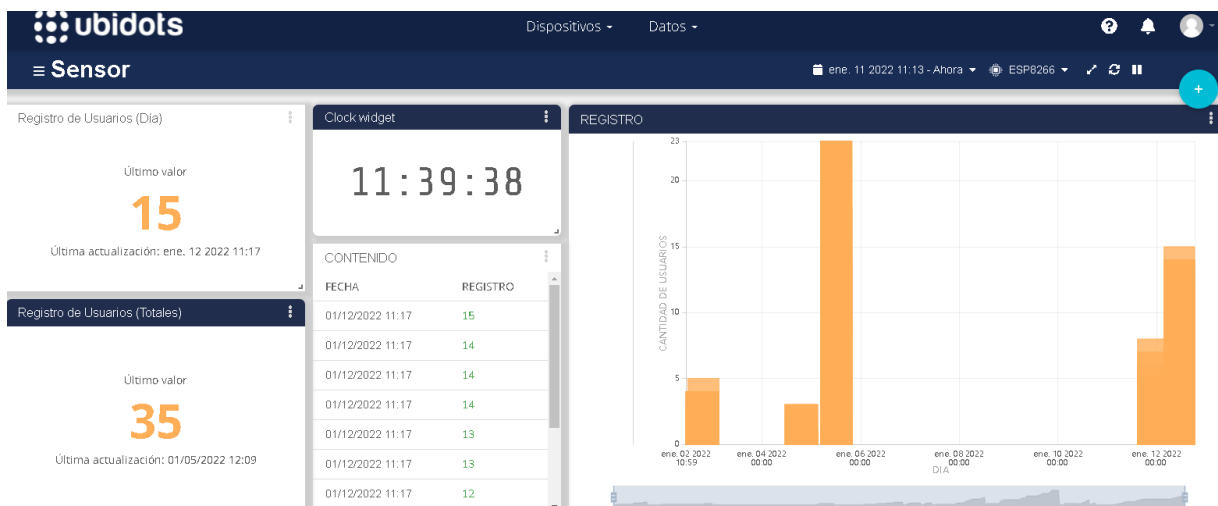


Ilustración 4.14 Configuraciones en la plataforma Ubidots de la etapa de conteo

4.5.2 Activación de alertas

A partir de los valores de velocidad que se censan, se establece una función para generar alertas orientadas al incumplimiento de los límites de velocidad establecidas en el carril de la ciclovía. La Ilustración 4.15, muestra la recepción de alertas cada vez que se detecten usuarios que superen los 20 Km/h, límite establecido por la ordenanza municipal de Guayaquil para la circulación en ciclovías.

De	Asunto	Recibido
NU Notifications Ubidots	alerta-de-velocidad Detectada El sistema de control de alerta-de-velocidad ha registrado un valor ...	11:43
NU Notifications Ubidots	alerta-de-velocidad Detectada El sistema de control de alerta-de-velocidad ha registrado un valor ...	11:42
NU Notifications Ubidots	alerta-de-velocidad Detectada El sistema de control de alerta-de-velocidad ha registrado un valor ...	11:42
Ayer		
NU Notifications Ubidots	alerta-de-velocidad Detectada El sistema de control de alerta-de-velocidad ha registrado un valor ...	Dom 17:02
NU Notifications Ubidots	alerta-de-velocidad Detectada El sistema de control de alerta-de-velocidad ha registrado un valor ...	Dom 17:01

Ilustración 4.15 Recepción de alertas por incumplimiento de límites de velocidad

Para la verificación de la generación de alertas y envío de notificaciones vía correo electrónico, se realizó pruebas en donde vehículos se aproximen al prototipo, como se visualiza en la Ilustración 4.16, con una velocidad superior a 20 Km/h correspondiente al límite de circulación permitida en ciclovías.



Ilustración 4.16 Generación de alertas de velocidad

En la Ilustración 4.17, se muestra el contenido de la alerta generada en donde se identifica el parámetro de velocidad censado y la fecha registrada en la plataforma IoT de Ubidots.

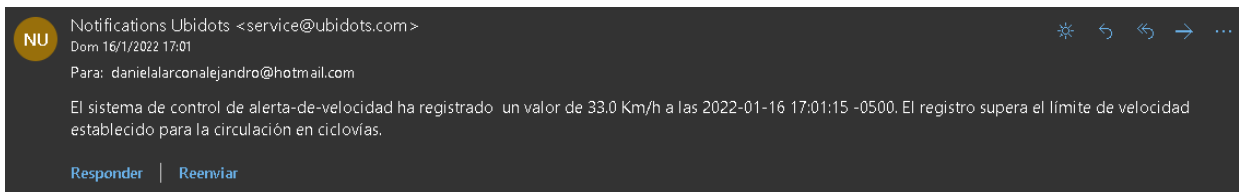


Ilustración 4.17 Información de alerta generada

Los datos de velocidad que se han censado de la etapa de detección son almacenados en tablas en donde se proporciona información del valor de velocidad que se carga en la nube y su respectiva fecha de publicación. Los eventos de generación y envío de alerta son registrados en la plataforma Ubidots como se visualiza en la Ilustración 4.18.

FECHA	VALOR
2022-01-16 17:01:59 -05:00	8.00
2022-01-16 17:01:55 -05:00	0.00
2022-01-16 17:01:52 -05:00	4.00
2022-01-16 17:01:49 -05:00	0.00
2022-01-16 17:01:46 -05:00	2.00
2022-01-16 17:01:38 -05:00	0.00
2022-01-16 17:01:35 -05:00	11.00
2022-01-16 17:01:18 -05:00	0.00
2022-01-16 17:01:15 -05:00	33.00
2022-01-16 17:01:04 -05:00	0.00

Registro del evento		
Si Múltiples variables entonces Email		
Tipo	Fecha	Mensaje
Actualización evento	2022-01-17 12:00:51 -05:00	Event "Si Múltiples variables entonces Email" updated
Correo electrónico	2022-01-17 11:43:42 -05:00	Email sent to danielalarconalejandra@hotmail.com
Correo electrónico	2022-01-17 11:42:44 -05:00	Email sent to danielalarconalejandra@hotmail.com
Correo electrónico	2022-01-17 11:42:02 -05:00	Email sent to danielalarconalejandra@hotmail.com
Correo electrónico	2022-01-16 17:02:05 -05:00	Email sent to danielalarconalejandra@hotmail.com
Correo electrónico	2022-01-16 17:01:16	Email sent to danielalarconalejandra@hotmail.com

Listo

Ilustración 4.18 Registro de velocidades censadas y eventos de alertas generadas

CONCLUSIONES

- Se logró procesar parámetros físicos de velocidad, distancia y tiempo de los biciusuarios censados con la finalidad de validar su proximidad. El prototipo detectó usuarios a una distancia máxima de 15m y un rango de velocidades alcanzados por los usuarios de hasta 14 Km/h.
- El factor de corrección de 10,2776 obtenido en las pruebas experimentales para las muestras de distancia, proporcionó un error relativo del 2.33% en comparación con las distancias teóricas, validando una correcta calibración del equipo.
- Se implementó una interfaz de alerta visual en señaléticas de prevención para intersecciones o cruces con ciclovías, la cual se habilita con éxito cada vez que un usuario que se movilice en este carril, se encuentre a 5 segundos próximos al cruce.
- Como resultado de las pruebas experimentales, se obtuvo un tiempo promedio de 5 segundos de llegada de los ciclistas a un cruce desde su detección a 15 metros, estableciendo un intervalo adecuado para la activación de la alarma que permita notificar a los vehículos que se aproximan biciusuarios hacia la intersección.
- Se diseñó un algoritmo que incorpora funciones de detección, registro y publicación en la nube de los biciusuarios que transitan por la ciclovía, a partir de las lecturas de sensores digitales. Su diseño se caracterizó por la simplificación de procesos debido a que las bibliotecas utilizadas facilitan la interconexión entre módulos y microcontroladores.
- Se logró diseñar una etapa de registro de usuarios de forma exitosa, mediante la incorporación de sensores infrarrojos, permitiendo almacenar el tráfico de usuarios que ha circulado durante el día, abarcando un rango de detección

aceptable en función de las dimensiones para carriles unidireccionales de ciclovía que rigen en el país.

- Se logró establecer una conexión entre el prototipo y un servidor web mediante unidades programables que permiten la comunicación a internet. El servidor registra y monitorea en tiempo real parámetros físicos y tráfico de los ciclistas, en donde dicha información puede ser segmentada y visualizada en gráficos estadísticos.
- Se estableció eventos de alertas en el servidor web en función del límite de velocidad establecido por la ordenanza municipal en ciclovías, generando notificaciones dirigidas a entes reguladores vía correo electrónico en el que se detalla la velocidad, fecha y hora de la infracción.
- Finalmente, se implementó y diseñó un prototipo de detección y alarma en ciclovías a bajo costo, el cual puede ser aplicada en múltiples tramos de la ciclovía con intersecciones urbanas no controladas con la finalidad de promover una participación ciudadana segura.

RECOMENDACIONES

- Si se desea incrementar el rango de detección de distancias, se recomienda implementar la versión LIDAR M16 debido a las mejoras en los parámetros de alcance (100 metros), la mayor apertura en su campo visión (10° - 19°) y la detección multiobjetivo con discriminación lateral.
- Se sugiere ubicar un enrutador próximo al sistema de conteo de usuarios con la finalidad de brindar umbrales de intensidad de señal aceptables, para que el envío de datos a la nube se realice sin pérdidas de paquetes.

- Se aconseja que el soporte en donde se ubique el prototipo de detección, tenga una altura mínima de 2 metros con la finalidad de evitar conflictos en el paso de ciclistas.
- Se debe tener en consideración realizar un mantenimiento preventivo de forma periódica de las unidades programables y sensores con la finalidad de que la calibración de los equipos mantenga la precisión adecuada de los parámetros censados.

BIBLIOGRAFÍA

- [1] Instituto Nacional de Estadística y Censos, «Ecuador en Cifras,» Noviembre 2020. [En línea]. Available: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Encuestas_Ambientales/Hogares/Hogares%202019/BOL_TEC_AMB_ESPN_D_2019_11.pdf. [Último acceso: 02 Noviembre 2021].
- [2] Diana Holguín, «Negocios de bicicletas en Ecuador reportan hasta 300% más ventas,» Vistazo, Guayaquil, 2020.
- [3] Agencia Nacional de Tránsito, «Estadísticas Siniestros de Tránsito,» ANT, Guayaquil, 2021.
- [4] El Universo, «Informes,» *Ciclistas están en riesgo por falta de ciclovías y desacato de normas en Ecuador*, p. 1, 20 Julio 2019.
- [5] Instituto Ecuatoriano de Normalización , «Reglamento Técnico Ecuatoriano,» Agencia Nacional de Tránsito, Quito, 2011.
- [6] El Telégrafo, «Guayaquil tiene déficit de aceras y ciclovías,» *El telégrafo*, p. 1, 05 Abril 2019.
- [7] A. N. d. Tránsito, «LEY ORGANICA DE TRANSPORTE TERRESTRE TRANSITO Y SEGURIDAD VIAL». Ecuador Patente Registro Oficial Suplemento 398, 31 Diciembre 2014.

- [8] S. Cordovez, E. Ortiz-Prado, E. Vasconez, F. Andrade, K. Simbaña-Rivera, L. Gómez-Barreno y R. McIlroy, «Bicycling-Related Mortality in Ecuador: A Nationwide Population-Based Analysis from 2004 to 2017,» *Sustainability*, vol. 5906, nº 11, p. 13, 2021.
- [9] F. Larrea, «En Ecuador han muerto 300 ciclistas en la calle entre 2004 y 2017,» *Primicias*, p. 1, 02 Agosto 2021.
- [10] B. Moncada, «Guayaquil: Ciclista de 9 años, arrollado al salir de la ciclovía por un auto aparcado,» *Diario Expreso*, p. 1, 06 Junio 2021.
- [11] Gobierno Autónomo Descentralizado Municipal de Guayaquil, «Municipalidad de Guayaquil,» 1 Julio 2020. [En línea]. Available: <https://www.guayaquil.gob.ec/wp-content/uploads/Documentos/Gacetas/Periodo%202019-2023/Gaceta%2018.pdf>. [Último acceso: 19 Octubre 2021].
- [12] El Universo, «Ciclismo en Guayaquil,» *El Universo*, p. 1, 15 Abril 2016.
- [13] Organismo de las Naciones Unidas, «Objetivos de Desarrollo Sostenible,» Programa de las Naciones Unidas para el Desarrollo, 13 Julio 2021. [En línea]. Available: <https://www.ec.undp.org/content/ecuador/es/home/sustainable-development-goals.html>. [Último acceso: 25 Octubre 2021].
- [14] HEIJMANS INFRA B.V., «Verkeersanalyse Bikescout Westland,» 19 Septiembre 2020. [En línea]. Available: <https://www.fietsberaad.nl/Kennisbank/Verkeersanalyse-Bikescout-Westland>. [Último acceso: 18 Octubre 2021].
- [15] K. Khazukov, V. Shepelev, T. Karpeta, S. Shabiev, I. Slobodin, I. Charbadze y I. Alferova, «Real-time monitoring of traffic parameters,» *Journal Big Data*, vol. 7, nº 84, p. 20, 2020.
- [16] J. A. L. Castro, «Desarrollo de un sistema inteligente de control de tráfico con software de código abierto en sistemas embebidos,» Junio 2017. [En línea]. Available: <https://ciateq.repositorioinstitucional.mx/jspui/handle/1020/91>. [Último acceso: 19 Octubre 2021].

- [17] F. Mejía, G. Torres y E. Villa, «Intelligent traffic light model using a Genetic Algorithm,» *INFOCIENCIA*, vol. 10, nº 1390-339X, p. 8, 2016.
- [18] J. V. Brummelen, B. Emran, K. Yesilcimen y H. Najjaran, «Reliable and Low-cost Cyclist Collision Warning System for Safer Commute on Urban Roads,» *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, vol. SMC.2016.7844814, nº 16656746, p. 6, 2016.
- [19] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli y D. Helbing, «Revisiting Street Intersections Using Slot-Based Systems,» *PLoS ONE*, vol. 3, nº e0149607, p. 11, 17 Mayo 2016.
- [20] L. J. A. AHUMADA, «DISEÑO E IMPLEMENTACIÓN DE UNA CHAQUETA PARA CICLISTAS,» 16 Junio 2019. [En línea]. Available: <https://repositorio.ecci.edu.co/bitstream/handle/001/1020/Trabajo%20de%20grado.pdf?sequence=1&isAllowed=y>. [Último acceso: 18 Octubre 2021].
- [21] C. Padilla y J. illagran, «DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO DE OBJETOS EN CARRETERA DURANTE LA CONDUCCIÓN, BAJOCONDICIONES ADVERSAS,» *Magazine Mechanics Energy Innovation and Future*, vol. IV, nº 4, p. 6, 29 Noviembre 2015.
- [22] W. Á. Romero Guzmán, «Prototipo de Semáforo Inteligente en intersecciones,» Abril 2013. [En línea]. Available: <https://biblioteca.epn.edu.ec/cgi-bin/koha/opac-detail.pl?biblionumber=11995>. [Último acceso: 18 Octubre 2021].
- [23] M. Hevia Villanueva, «MOB-IOT: Análisis inteligente de la movilidad,» 18 Julio 2019. [En línea]. Available: <https://digibuo.uniovi.es/dspace/handle/10651/53452>. [Último acceso: 18 Octubre 2021].
- [24] FLIR, «Intelligent Transportation Systems,» FLIR, 14 Julio 2013. [En línea]. Available: https://www.flirmedia.com/MMC/CVS/Traffic/IT_0004_EN.pdf. [Último acceso: 19 Octubre 2021].
- [25] J. Gilpin, T. Bonnell, M. Fralick, K. Paulson y L. Zefting, «Bicycle Detection,» Alta Planning, Arkansas, 2021.

- [26] D. G. Moscoso Montenegro y D. F. Romero Guaycha, «DESARROLLO DE UN PROTOTIPO DE RADIOBALIZA PARA CICLISTAS, INTEGRADO EN LA BICICLETA Y EL CASCO, CON LA CAPACIDAD DE DETECCIÓN DE IMPACTOS EN LA CABEZA Y EMISIÓN DE ALERTAS,» Julio 2020. [En línea]. Available:
https://rraae.cedia.edu.ec/Record/UPS_baeaf33108f69fd996d03f508f9a18eb.
[Último acceso: 19 Octubre 2021].
- [27] C. Chung, P. Koonce, P. Eun y J. Cabezuela, «Bellevue trialling new software to identify dangerous intersections,» Bellevue Vision Zero Summit, Bellevue , 2019.
- [28] D. Kyritsis, «The identification of road modality and occupancy patterns by Wi-Fi monitoring sensors as a way to support the “Smart Cities” concept,» Architecture and The Built Environment, 20 Abril 2017. [En línea]. Available:
<https://repository.tudelft.nl/islandora/object/uuid:68a1d7d2-3828-4df5-9c05-63a2511188e4?collection=education>. [Último acceso: 17 Octubre 2021].
- [29] LeddarTech, «LeddarOne,» LeddarTech, 24 Agosto 2021. [En línea]. Available:
https://leddartech.com/app/uploads/dlm_uploads/2021/04/Spec-Sheet_LeddarOne_V10.0_EN.pdf. [Último acceso: 26 Octubre 2021].
- [30] T. Denton, «Sistemas eléctrico y electrónico del automóvil,» de *Tecnología automotriz: mantenimiento y reparación de vehículos* , Mexico, Marcombo, 2016, p. 744.
- [31] Diego Aranda, «Sistema de Telemetria,» de *Plataformas Arduino y Raspberry Pi*, Buenos Aires, Manuales Users, 2014, p. 320.
- [32] Oracle, «Internet of Things (IoT),» Oracle, 13 Abril 2021. [En línea]. Available:
<https://www.oracle.com/ar/internet-of-things/>. [Último acceso: 01 Diciembre 2021].
- [33] M. Alam, K. A. Shakil y S. Khan, «Resource Management Techniques for Cloud-Based IoT Environment,» de *Internet of Things (IoT) Concepts and Applications*, Cham, Springer Nature Switzerland, 2020, p. 515.
- [34] PCE Instruments, «PCE Ibérica S.L,» PCE Ibérica S.L, 30 Septiembre 2014. [En línea]. Available: <https://www.pce-iberica.es/instrumentos-de->

- medida/sistemas/sensores.htm. [Último acceso: 11 27 2021].
- [35] R. P. Areny, «Introducción a los sistemas de medidas,» de *Sensores y Acondicionadores de Señal*, Barcelona, Marcombo, 2003, p. 474.
- [36] J. C. Martín, «Sensores,» de *CPI - Instalaciones domóticas*, España, Editex, 2009, p. 106 .
- [37] E. Zabler, «Técnica del automóvil BOSCH,» de *Los sensores en el automóvil*, España, ROBERT BOSCH GMBH, 2000, p. 148.
- [38] A. O. Baturone, «Robótica,» de *Manipuladores y Robots Móviles*, Sevilla, Marcombo, 2001, p. 422.
- [39] LEANTEC, «HC-SR04,» Junio 2019. [En línea]. Available: <https://leantec.es/wp-content/uploads/2019/06/Leantec.ES-HC-SR04.pdf>. [Último acceso: 01 Diciembre 2021].
- [40] TURCK Inc, «Sensor ultrasónico RU40U-M18E-LIU2PN8X2T-H1151,» TURCK Inc, 03 Noviembre 2021. [En línea]. Available: https://www.turck.de/datasheet/_es/edb_1610024_esl_es.pdf. [Último acceso: 07 Diciembre 2021].
- [41] A. Stykemain, «Photoelectric Sensor,» RealPars, 21 Junio 2021. [En línea]. Available: <https://realpars.com/photoelectric-sensor/>. [Último acceso: 2021 Diciembre 1].
- [42] Unit Electronics, «e E18-D80NK Infrared Distance Ranging Sensor,» Abril 2019. [En línea]. Available: <https://uelectronics.com/wp-content/uploads/2019/04/E18-D80NK-ETC.pdf>. [Último acceso: 28 Noviembre 2021].
- [43] OMRON, « Standard Photoelectric Sensors,» Julio 2005. [En línea]. Available: <https://www.tme.eu/Document/5cf828984171c2e5a7f57d494b362769/E3Z-SERIES-EN.pdf>. [Último acceso: 1 Diciembre 2021].
- [44] GeoSpatial, «LiDAR,» GeoSpatial, 11 Marzo 2021. [En línea]. Available: <https://www.geospatial.com.co/servicios/tecnologia-lidar.html>. [Último acceso:

Noviembre 30 2021].

- [45] J. A. Sobrino, «LiDAR,» de *Teledetección*, Valencia, Guada Impresores S.L., 2000, p. 449.
- [46] LeddarTech, «LiDAR Vu8,» Abril 2021. [En línea]. Available: https://leddartech.com/app/uploads/dlm_uploads/2021/04/Spec-Sheet_Leddar-Vu8_V4.1_EN-1.pdf. [Último acceso: 26 Noviembre 2021].
- [47] Leddar Tech, «LeddarOne,» Abril 2021. [En línea]. Available: https://leddartech.com/app/uploads/dlm_uploads/2021/04/Spec-Sheet_LeddarOne_V10.0_EN-1.pdf. [Último acceso: 26 Noviembre 2021].
- [48] E. M. Perez, *Microcontroladores PIC: Sistema Integrado para el Autoaprendizaje*, Barcelona: Marcombo, 2007.
- [49] M. Verle, «MIKROE,» 05 Noviembre 2021. [En línea]. Available: <https://www.mikroe.com/ebooks/microcontroladores-pic-programacion-en-basic/introduccion-al-mundo-de-los-microcontroladores>. [Último acceso: 30 Noviembre 2021].
- [50] E. M. Perez, «Tecnologías de Circuitos Digitales,» de *Sistemas Electrónicos Digitales*, Barcelona, Marcombo, 2008, p. 849.
- [51] E. Embajadores, «Módulos Electrónicos,» *Electrónica Embajadores*, 12 Enero 2021. [En línea]. Available: <https://www.electronicaembajadores.com/es/Subfamilias/Productos/LCZB/modulos-electronicos/zigbee-802-15-4>. [Último acceso: 30 Noviembre 2021].
- [52] Ruben Beiros Mosquera, «MCU,» de *Arduino, Electrónica y Programación*, España, Marcombo, 2018, p. 216 .
- [53] Arduino Store, « Arduino Uno Rev3,» *Arduino* , 01 Septiembre 2021. [En línea]. Available: <https://store.arduino.cc/products/arduino-uno-rev3>. [Último acceso: 30 Noviembre 2021].
- [54] Arduino Store, «Arduino Mega 2560 Rev3,» *Arduino*, 30 Agosto 2021. [En línea]. Available: <https://store.arduino.cc/products/arduino-mega-2560-rev3>. [Último

acceso: 30 Noviembre 2021].

- [55] Espressif, «ESP Product Selector,» Espressif, 20 Julio 2020. [En línea]. Available: <https://www.espressif.com/en/products/socs/esp32>. [Último acceso: 30 Noviembre 2021].
- [56] WinStar, «Pantalla LCD Alfanumérica,» WinStar, 11 Junio 2017. [En línea]. Available: <https://www.winstar.com.tw/es/products/character-lcd-display-module/8x2-lcd.html>. [Último acceso: 30 Noviembre 2021].
- [57] Espressif, «ESP8266EX,» Espressif, 17 Mayo 2016. [En línea]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Último acceso: 08 Diciembre 2021].
- [58] Eneka, «Módulo interfaz serial I2C,» Eneka, 10 Septiembre 2018. [En línea]. Available: <https://www.eneka.com.uy/robotica/modulos-comunicacion/m%C3%B3dulo-interfaz-serail-i2c-detail.html>. [Último acceso: 30 Noviembre 2021].
- [59] kionetworks, «<https://www.kionetworks.com/>,» [En línea]. Available: <https://www.kionetworks.com/blog/data-center/protocolos-de-comunicaci%C3%B3n-de-redes>. [Último acceso: 30 Noviembre 2021].
- [60] Cisco , «Descripción General de TCP-CCNA 1,» Cisco CCNA , 10 Julio 2020. [En línea]. Available: <https://ccnadesdecero.es/descripcion-general-tcp/>. [Último acceso: 08 Diciembre 2021].
- [61] U. I. d. Valencia, «Protocolo IP,» VIU, 25 Agosto 2016. [En línea]. Available: <https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-y-como-funciona-el-protocolo-ip>. [Último acceso: 30 Noviembre 2021].
- [62] Arduino, «Protocolo I2C,» Arduino, 04 Mayo 2020. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/>. [Último acceso: 30 Noviembre 2021].
- [63] Silvia Palazzi, «Fab Academy,» Santa Chiara Fab Lab, 20 Noviembre 2018. [En línea]. Available:

- <http://archive.fabacademy.org/fabacademy2017/fablabtoscana/students/207/exercise15.html>. [Último acceso: 30 Noviembre 2021].
- [64] INTEL, «Potocolos de Wi-Fi y velocidades de datos,» INTEL, 01 Abril 2021. [En línea]. Available: <https://www.intel.la/content/www/xl/es/support/articles/000005725/wireless/legacy-intel-wireless-products.html>. [Último acceso: 08 Diciembre 2021].
- [65] J. M. Miranda, Ingeniería de Microondas, Madrid.: Prentice Hall, 2002.
- [66] «kripkit.com,» [En línea]. Available: <https://kripkit.com/reflectancia/>. [Último acceso: 1 Diciembre 2021].
- [67] C. G. Garcia, «Sistema de deteccion preventiva de obstaculos en vehiculos.,» Madrid., 2019.
- [68] J. M. Fallas, «Instituto Tecnológico de Costa Rica,» 06 Junio 2015. [En línea]. Available: <http://funes.uniandes.edu.co/8122/1/pag1.htm>. [Último acceso: 1 Diciembre 2021].
- [69] J. L. Fernández, «Dispersion de la Luz,» Fisicalab , 09 Agosto 2020. [En línea]. Available: <https://www.fisicalab.com/apartado/dispersion-luz>. [Último acceso: 1 Diciembre 2021].
- [70] XLSTAT, «Modelado Estadístico,» XLSTAT, 03 Septiembre 2018. [En línea]. Available: <https://www.gmsl.it/wp-content/uploads/2018/09/Articolo-XLSTAT-Modellazione-statistica.pdf>. [Último acceso: 01 Diciembre 2021].
- [71] IBM, «Modelos estadísticos,» IBM, 05 Julio 2021. [En línea]. Available: <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=nodes-statistical-models>. [Último acceso: 1 Diciembre 2021].
- [72] P. R. Canfranc, Las webs y demás plataformas en red., Quito: Fundación Telefonica, 2019.
- [73] Daniel Raynaud, «Ubidots,» Ubidots, 20 Mayo 2018. [En línea]. Available: <https://ubidots.com/platform/>. [Último acceso: 26 Noviembre 2021].

- [74] Google , «FireBase,» Google Cloud, Octubre 2014. [En línea]. Available: <https://firebase.google.com/docs/reference>. [Último acceso: 30 Noviembre 2021].
- [75] MathWorks, «ThingSpeak,» MathWorks, 14 Abril 2021. [En línea]. Available: https://thingspeak.com/pages/learn_more. [Último acceso: 30 Noviembre 2021].
- [76] C. Ruiz, «Open Lanuza,» 17 Enero 2018. [En línea]. Available: <https://openlanuza.com/utilizando-un-rele-en-arduino/>. [Último acceso: 12 Diciembre 2021].
- [77] P. N. Roldán, «Modelo de regresión,» Economipedia, 15 Septiembre 2016. [En línea]. Available: <https://economipedia.com/definiciones/modelo-de-regresion.html>. [Último acceso: 11 Diciembre 2021].
- [78] INTEL, «Protocolos de Wi-Fi,» INTEL, 01 Abril 2021. [En línea]. Available: <https://www.intel.la/content/www/xl/es/support/articles/000005725/wireless/legacy-intel-wirele>. [Último acceso: 30 Noviembre 2021].

APÉNDICE

```
//Codigo desarrollado por Nicolas Torres - Daniel Alarcon
//ETAPA DE DETECCION Y ALARMA
//Librerias necesarias
#include <Leddar.h>
#include <SoftwareSerial.h>
LeddarOne Leddar1(115200, 1);
//Se definen las señales de interes
#define rxPin 2
#define txPin 3
int SENAL = 13; //Pin que habilita la alarma visual
int LED = 8; //Pin que habilita el led que indica el
funcionamiento
unsigned int Distancia1 = 0; //Variable para almacenar la primera distancia
detectada
unsigned int Distancia2 = 0; //Variable para almacenar la segunda distancia
detectada
int velocidad = 0 ; //Variable para almacenar la velocidad calculada
float tiempo=0; //Variable para almacenar el tiempo de llegada a
la interseccion
int contador=0;
SoftwareSerial Serial2(rxPin, txPin); //Puerto serial para enviar informacion
hacia la ESP8266
void setup()
```

```

{
  //Se inicia las conexiones seriales
  Serial.begin(115200);
  Serial2.begin(115200);
  Leddar1.init();          //Se inicializa el sensor Lidar
  //Se define las señales como pines de salida
  pinMode(SENAL, OUTPUT);
  pinMode(LED, OUTPUT);
  pinMode (txPin , OUTPUT);
}

void loop(){

char Resultado = Leddar1.getDetrections(); //Matriz de deteccion del sensor
if (Resultado >= 0)
{
  digitalWrite(LED, HIGH); //Se activa el led cada vez que el sensor detecte
un obstaculo
  Distancial = (Leddar1.Detections[0].Distance); //Se almacena la primera
distancia detectada
  Leddar1.clearDetrections(); //Se limpia la matriz de
deteccion
  delay(400); //Se toma la 2da Muestra de
distancia cada 400 milisegundos
  Leddar1.getDetrections();
  Distancia2 = Leddar1.Detections[0].Distance;
  velocidad=(Distancial-Distancia2)*0.00875685; //Velocidad calculada como la
diferencia de las distancias obtenidas
  //dividida para el tiempo entre cada muestra. Se multiplica por el factor
de calibracion.
  tiempo= ((Distancia2/1027.766342)/(velocidad/3.6)); //El tiempo calculado es
en funcion de la velocidad calculada y la segunda distancia obtenida
  //debido a que es la mas cercana al cruce
  Serial.print("Velocidad="+ String(velocidad));
  Serial.print("Tiempo de llegada="+ String(tiempo));
  Serial.print("distancia2="+ String(Distancia2/1027.766342));
  Serial.print("distancial="+ String(Distancial/1027.766342));

if (velocidad >0 && velocidad <70) // Se limita el sensor velocidades de 0 a
70 km/h
{

  Serial2.println(velocidad); //Se envia la variable velocidad al puerto
serial2 conectado a la ESP8266

  if ((tiempo>0 && tiempo < 5)) //Se activa la alarma si el obstaculo llegara
al cruce en menos de 5 segundos
  {
    for (int i=0; i<6; i++) //La alarma se mntendra activa por un tiempo
aproximado de 5 segundos
    {
      digitalWrite(SENAL, LOW);
      delay(400);
      digitalWrite(SENAL, HIGH);
      delay(400 );
    }
  }

  else if (tiempo =0 && tiempo > 5) //La alarma se mantiene apagada si el
usuario detectado se encuentra a mas de 5 segundos
  //de llegar al cruce

```



```

    {
        digitalWrite(SENAL, LOW);
    }

}
delay(25);
}

//Codigo desarrollado por Nicolas Torres - Daniel Alarcon
//ETAPA DE CONTEO DE USUARIOS Y CONEXIÓN CON SERVIDOR
//Librerias necesarias

#include "UbidotsEsp32Mqtt.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (0x27, 20, 4); // si no te sale con esta direccion
puedes usar (0x3f,16,2) || (0x20,16,2)
#include <RTClib.h>
RTC_DS3231 rtc;
#define pin 23
int conteo;
int totales;

//Parametros de red y de servidor

namespace {
    const char * WIFISSID = "NETLIFE-ALARCON";
    const char * PASSWORD = "0818ALEALARCON0423";
    const char * TOKEN = "BBFF-czomeTZKnviDusr6z1MsCDvt6PtSk0";
    const char * VARIABLE_LABEL_1 = "Numero";
    const char * VARIABLE_LABEL_2 = "usuarios";
    const char * DEVICE_LABEL = "ESP8266";
    const int PUBLISH_FREQ = 1000;
}
Ubidots ubidots(TOKEN);

void setup() {
    Serial.begin(115200);
    pinMode(pin, INPUT); //Pin de sensor en modo de entrada
    Wire.begin();
    lcd.init(); //Inicialización de pantalla LCD
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print("AL DIA: ");
    lcd.setCursor(0,1);
    lcd.print("TOTAL: ");
    rtc.begin(); //Inicialización de reloj
    //rtc.adjust(DateTime(__DATE__, __TIME__));
    ubidots.connectToWifi(WIFISSID, PASSWORD); //Conexion a establecer con la red
    ubidots.setup();
    ubidots.reconnect();
}

void loop() {
    DateTime now = rtc.now();
    //Configuracion de formato de conteo de usuarios

    lcd.setCursor(6,3);
    if ( (now.hour() < 10) && (now.hour() > 0) || (now.hour() == 0) ){

```

```

        lcd.print("0");
        lcd.setCursor(7,3);
        lcd.print(now.hour());
    }
    else{
        lcd.print(now.hour());
    }

    lcd.print(':');
    lcd.setCursor(9,3);
    if ( (now.minute() < 10) && (now.minute() > 0) || (now.minute() == 0) ){
        lcd.print("0");
        lcd.setCursor(10,3);
        lcd.print(now.minute());
    }
    else{
        lcd.print(now.minute());
    }

    lcd.print(':');
    lcd.setCursor(12,3);
    if ( (now.second() < 10) && (now.second() > 0) || (now.second() == 0) ){
        lcd.print("0");
        lcd.setCursor(13,3);
        lcd.print(now.second());
    }
    else{
        lcd.print(now.second());
    }
}

if(digitalRead(pin)!=1){
    conteo++; //Incremento de contador cada vez que detecte usuarios

    while(digitalRead(pin)!=1){ //Mantener comportamiento cuando exista
deteccion
        delay(1000);
    }
}

//Agregar los datos en el canal del servidor
ubidots.add(VARIABLE_LABEL_1, conteo);
//Publicar los datos en el canal del servidor
ubidots.publish(DEVICE_LABEL);

if (conteo <10 && conteo >0 || conteo == 0){
    lcd.setCursor(11,0);
    lcd.print("0");
    lcd.setCursor(12,0);
    lcd.print(conteo);
}
else {
    lcd.setCursor(11,0);
    lcd.print(conteo);
}

//Condicion de acumulador de registro total de usuarios detectados
if (now.hour()==0 && now.minute()==0 && now.second() == 0){
    totales+=conteo;
    conteo =0;
    lcd.setCursor(11,1);
}

```

```
        lcd.print(totales);
    }
    //Agregar los datos en el canal del servidor
    ubidots.add(VARIABLE_LABEL_2,totales);

    //Publicar los datos en el canal del servidor
    ubidots.publish(DEVICE_LABEL);

}
```