



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN ELECTRONICA Y COMUNICACIONES**

**Tema:**

---

**SISTEMA INTELIGENTE DE SEGURIDAD Y ALERTAS PARA BICICLETAS  
EN EL ECUADOR BASADO EN IOT**

---

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la  
obtención del título de Ingeniero en Electrónica y Comunicaciones

**ÁREA:** Electrónica y Comunicaciones

**LÍNEA DE INVESTIGACIÓN:** Sistemas Electrónicos

**AUTOR:** Diego Ismael Pazmiño Pérez

**TUTOR:** Ing. Ana Pamela Castro Martin, Mg.

**Ambato - Ecuador**

**septiembre – 2022**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA INTELIGENTE DE SEGURIDAD Y ALERTAS PARA BICICLETAS EN EL ECUADOR BASADO EN IOT , desarrollado bajo la modalidad Proyecto de Investigación por el señor Diego Ismael Pazmiño Pérez, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2022.

-----  
Ing. Ana Pamela Castro Martin, Mg.  
TUTOR

## AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA INTELIGENTE DE SEGURIDAD Y ALERTAS PARA BICICLETAS EN EL ECUADOR BASADO EN IOT es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022.



Diego Ismael Pazmiño Pérez

C.C. 1803537883

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Diego Ismael Pazmiño Pérez, estudiante de la Carrera de Electrónica y Comunicaciones , de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA INTELIGENTE DE SEGURIDAD Y ALERTAS PARA BICICLETAS EN EL ECUADOR BASADO EN IOT, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con el señor presidente del Tribunal.

Ambato, septiembre 2022.

-----  
Ing. Carlos Sánchez Rosero  
PRESIDENTE SUBROGANTE DEL TRIBUNAL

-----  
Ing. Mg. Julio Enrique Cuji Rodriguez  
PROFESOR CALIFICADOR

-----  
Ing. Mario Geovanny Garcia Carillo  
PROFESOR CALIFICADOR

## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2021.



Diego Ismael Pazmiño Pérez

C.C. 1803537883

AUTOR

Tabla de contenido	
CAPITULO I.....	1
MARCO TEÓRICO.....	1
1.1 Tema de Investigación .....	1
1.2 Antecedentes Investigativos.....	1
1.2.1 Contextualización del problema.....	3
1.2.2 Fundamentación teórica .....	5
Tecnología en bicicletas.....	5
Bici-conectada.....	5
Bici-inteligente.....	5
E-Bike .....	5
Clases de bicicletas .....	5
Bicicletas de montaña (MTB).....	6
Bicicleta de ruta con manubrio drop-bar.....	6
Bicicleta híbrida .....	7
Sistema de alerta .....	7
Internet de las cosas (IoT).....	7
Cloud Computing.....	8
Fog Computing.....	8
Edge Computing.....	9
Arquitectura Sistemas IoT.....	9
Modelos de comunicación de IoT.....	11
Comunicaciones dispositivo a dispositivo .....	11
Comunicaciones dispositivo a la nube .....	11
Modelo dispositivo a puerta de enlace .....	11
Modelo de intercambio de datos a través del back-end .....	11
Tarjeta de desarrollo.....	12
GNSS (Global Navigation Satellite System) .....	12
GPS (Sistema de Posicionamiento Global ).....	12
Estructura del G.P.S. ....	13

Segmento Espacial .....	14
Segmento de Control.....	14
Segmento de Usuario .....	15
Código C/A y señales GPS .....	15
Origen de error del GPS.....	15
Tiempo.....	16
Ionósfera y Tropósfera .....	16
Efecto Multitrayectoria.....	16
Red de comunicación móvil.....	17
Primera Generación.....	17
Segunda Generación.....	17
Generación 2.5.....	18
Tercera Generación .....	19
Cuarta Generación.....	19
5ta Generación.....	21
Tabla Comparativa Red móvil .....	21
Tarjeta universal de Circuito Integrado (UICC) .....	22
Operadoras Móviles .....	22
Modelo cliente servidor.....	23
Cliente .....	23
Servidor.....	24
Hosting .....	24
Lenguajes y entornos de programación.....	24
Php.....	24
phpMyAdmin .....	25
Kotlin.....	25
Android studio .....	25
1.3 Objetivos .....	25
1.3.1 Objetivo general.....	25
1.3.2 Objetivo específico.....	26
CAPÍTULO II .....	27
METODOLOGÍA .....	27
2.1. Materiales.....	27
2.2. Métodos.....	28

Modalidad de la Investigación .....	28
Recolección de Información .....	28
Procesamiento y Análisis de Datos .....	29
Desarrollo del Proyecto.....	29
CAPÍTULO III .....	31
RESULTADOS Y DISCUSIÓN.....	31
3.1 Análisis y discusión de los resultados.....	31
Desarrollo de la propuesta.....	31
Análisis de los dispositivos de seguridad.....	31
Problema de seguridad .....	34
Requerimiento del sistema .....	34
Tecnologías de comunicación IoT .....	36
Etapas del sistema IoT .....	37
Etapa de Adquisición .....	38
Diseño de la etapa de adquisición .....	39
Encendido del dispositivo.....	39
Comunicaciones internas TTGO .....	40
Activación del modo Parking .....	41
Autenticación y alarma.....	42
Etapa de Almacenamiento.....	45
Diseño de la etapa de almacenamiento.....	46
Transmisión y conexión con el servidor.....	46
Etapa de aplicación .....	49
Funcionamiento tercera etapa.....	50
Diseño de mapas en Android Studio con Kotlin .....	56
Hardware del prototipo.....	60
Selección de la plataforma IoT .....	60
Descripción de la placa de desarrollo IoT.....	62
Selección de sensores.....	63
Sensor Biométrico .....	63
Sensor de inclinación.....	64
Selección de la operadora móvil .....	65
Esquema de conexión del prototipo .....	67
Software del sistema .....	68

Selección Hosting.....	68
Pruebas de funcionamiento .....	69
Testeo de funcionamiento Red Móvil.....	70
Testeo funcionamiento GPS .....	74
Prueba de funcionamiento del Sistema .....	75
Consumo de energía del prototipo .....	79
Presupuesto del proyecto.....	81
CAPITULO IV .....	83
CONCLUSIONES Y RECOMENDACIONES.....	83
4.1 Conclusiones.....	83
4.2 Recomendaciones.....	84
C. MATERIALES DE REFERENCIA.....	85
Bibliografía .....	85

## Índice de tablas y figuras.

Figura 1 Bicicleta de enduro .....	6
Figura 2 Bicicleta de ruta .....	6
Figura 3 Bicicleta Rígida .....	7
Figura 4 Estructura cloud computing .....	8
Figura 5 Arquitectura 4 capas IoT .....	9
Figura 6 Estructura GPS .....	13
Figura 7 Estaciones de monitoreo .....	14
Figura 8 Error GPS .....	16
Figura 9 Objetivo GPRS .....	18
Figura 10 Arquitectura red LTE .....	20
Figura 11 Sistema Inteligente de seguridad y alertas para bicicletas .....	35
Figura 12 Tecnologías dedicadas al IoT .....	36
Figura 13 Estructura del sistema de seguridad a desarrollar.....	38
Figura 14 Algoritmo código t-sim7600 .....	39
Figura 15 Variables del boceto definidas .....	40
Figura 16 Comunicación Serial T-SIM7600 .....	41
Figura 17 Control de encendido de alarma .....	41
Figura 18 Sistema de autenticación .....	42
Figura 19 Método llamada de emergencia mediante comandos AT .....	43
Figura 20 Adquisición de datos a enviar latitud longitud e imei .....	43
Figura 21 Solicitud cliente mediante http para el envío de datos .....	44
Figura 22 Métodos del GPS on-off .....	45
Figura 23 Estructura etapa de Almacenamiento .....	46
Figura 24 Direcciones del hosting y APN .....	47
Figura 25 Credenciales para conexión placa y dispositivo.....	47
Figura 26 Tablas entidad relación .....	48
Figura 27 Definición variables a recibir en la base de datos phpMyAdmin .....	48
Figura 28 Inserción datos en tablas phpMyAdmin .....	48
Figura 29 Datos t-sim en phpMyAdmin.....	49
Figura 30 Algoritmo de la tercera etapa.....	50
Figura 31 Interfaz móvil .....	51
Figura 32 Usuarios registrados .....	52
Figura 33 Obtención ubicaciones cada 30s .....	52
Figura 34 Actualización cada 30s de ubicaciones en la interfaz móvil .....	53
Figura 35 función getDatos con RequestQueue .....	54
Figura 36 Parámetros necesario para obtener los datos dependiendo el usuario .....	55
Figura 37 Respuesta y conversión de datos del formato json .....	55
Figura 38 Diseño de Marker para ubicaciones en Google Maps Apis .....	56
Figura 39 Creación proyecto APIs GOOGLE .....	56
Figura 40 Selección API a utilizar.....	57
Figura 41 Clave API.....	57
Figura 42 Clave api dentro de la aplicación ControlBike. ....	58
Figura 43 Marker de posición en mapas .....	59
Figura 44 Compatibilidad con dispositivos Android .....	59
Figura 45 Placa de desarrollo (IOT) T-SIM 7600 G-H .....	63

Figura 46 Radio bases SMA Elaborado por ARCOTEL .....	66
Figura 47 Diagrama de conexión del sistema IoT .....	67
Figura 48 Montaje del sistema en la bicicleta Marin .....	68
Figura 49 Comprobación de la placa IoT con la PC .....	70
Figura 50 Comando AT +COPS.....	71
Figura 51 Comando AT+CNMP. ....	71
Figura 52 Comando AT+CNAOP. ....	72
Figura 53 Cobertura red 4G de Claro. ....	72
Figura 54 Directiva preprocesada de la red móvil. ....	73
Figura 55 Datos de la red móvil por parte de la placa IoT .....	74
Figura 56 Código de la directiva preprocesada del GPS. ....	74
Figura 57 Datos del GPS que se obtiene por parte de la placa IoT, con la representación de la ubicación.....	75
Figura 58 Sensor Biométrico. ....	76
Figura 59 Encendido del prototipo.....	76
Figura 60 Alarma desactivada. ....	77
Figura 61 Sistema de alerta activado. ....	77
Figura 62 Notificaciones del sistema activado. ....	78
Figura 63 Datos recibidos del prototipo en la BD.....	79
Tabla 1 Sistemas de geolocalización .....	12
Tabla 2 Tabla comparativa red móvil. ....	22
Tabla 3 Frecuencias para 3g , 4g en Ecuador .....	23
Tabla 4 Comparación de dispositivos inteligentes para bicicletas en Ecuador .....	33
Tabla 5 Cuadro comparativo de tecnologías IoT .....	37
Tabla 6 Comparativa placas de desarrollo IoT.....	62
Tabla 7 Comparación de sensores biométricos.....	64
Tabla 8 Comparación sensores inclinación.....	65
Tabla 9 Comparación Hosting .....	69
Tabla 10 Consumo de energías de los módulos utilizados .....	80

## RESUMEN EJECUTIVO

Actualmente existen varios medios de transporte y debido a las recientes condiciones que ha vivido la humanidad alrededor del mundo con la pandemia, el uso de la bicicleta para la movilidad dentro y fuera de la ciudad está ganando atención. Estos vehículos ofrecen muchos beneficios para la salud, desde la actividad física hasta la prevención de infecciones causadas por reuniones masivas cuando se utilizan medios de transporte convencionales. Con un aumento del 700% en su uso en Ecuador desde 2019, las bicicletas representan el cuarto medio de transporte más utilizado en el país. Sin embargo, el delito de robo de bicicletas que aumentó significativamente debido a que el robo de bicicletas representa un mercado viable y sostenible para los delincuentes, una bicicleta de gama media cuesta entre \$ 1,000 y \$ 9,000, los delincuentes la venden a la mitad o incluso menos, por lo tanto, tener que invertir en una bicicleta de alto rendimiento no es seguro.

El presente proyecto desarrolla un sistema de seguridad basado en internet de las cosas para la mitigación de gran parte de la problemática anteriormente expuesta. El funcionamiento se basa en 3 etapas la primera; la autenticación del usuario siempre que requiera hacer uso del vehículo mediante un sensor de huellas. En caso de detectar un comportamiento anormal injustificado mediante un sensor de inclinación, el dispositivo empezará a emitir alertas hacia el usuario registrado en el servicio, la segunda etapa es el almacenamiento de la información esta guarda los datos del dispositivo en una base de datos utilizando un Hosting, para luego enviar a la tercera etapa donde el propietario mediante un smartphone puede monitorear la ubicación de la bicicleta en tiempo real.

Gracias a la implementación en cada una de las etapas, permite la validación de usuarios, el almacenamiento de datos en la nube y monitoreo desde un dispositivo móvil, cumpliendo así con la arquitectura IoT solucionando así un problema de seguridad ante el robo de bicicletas.

**Palabras clave:** IOT, bicicleta, nube, seguridad, autenticación.

## **Abstract**

Currently there are several means of transportation and due to the recent conditions that humanity has experienced around the world with the pandemic, the use of bicycles for mobility inside and outside the city is gaining attention. These vehicles offer many health benefits, from physical activity to preventing infections caused by mass gatherings when using conventional means of transportation. With a 700% increase in its use in Ecuador since 2019, bicycles represent the fourth most used means of transport in the country. However, the crime of bike theft has increased significantly because bike theft represents a viable and sustainable market for criminals, a mid-range bike costs between \$1,000 and \$9,000, criminals sell it for half or less even, therefore having to invest in a performance bike is not safe.

This project develops a security system based on the internet of things to mitigate a large part of the aforementioned problems. The operation is based on 3 stages the first; user authentication whenever you need to use the vehicle through a fingerprint sensor. In case of detecting an unjustified abnormal behavior by means of a tilt sensor, the device will start to issue alerts to the user registered in the service, the second stage is the storage of the information, this saves the data of the device in a database using a Hosting, to then send to the third stage where the owner can monitor the location of the bicycle in real time through a smartphone.

Thanks to the implementation in each of the stages, it allows the validation of users, the storage of data in the cloud and monitoring from a mobile device, thus complying with the IoT architecture, thus solving a security problem in the event of bicycle theft.

**Keywords:** IOT, bicycle, cloud, security, authentication.

## **CAPITULO I**

### **MARCO TEÓRICO**

#### **1.1 Tema de Investigación**

Sistema inteligente de seguridad y alertas para bicicletas en el Ecuador basado en IoT

#### **1.2 Antecedentes Investigativos**

La investigación fue realizada a nivel nacional e internacional en los repositorios de diferentes universidades y revistas con proyectos relacionados con sistemas IoT y bicicletas inteligentes los mismos se detallan a continuación:

En el artículo “Categorizan bicycling enviroments using GPS-based public bicyle speed data” en el año 2015 y presentado por Jeong Joo como parte del equipo de investigación de Hanyang University, Korea, se expone el resultado de su estudio donde se propone la metodología para categorizar entornos de ciclismo en la búsqueda de aumentar el uso de bicicletas. El estudio se llevó a cabo utilizando sistemas de posicionamiento global en bicicletas públicas, el monitoreo realizado durante varias semanas y permitió hacer un registro estadístico extrayendo el nivel de entornos donde se emplean rutas de ciclismo con los datos de aceleración y velocidad que se obtienen del usuario, toda la información encontrada fue usada para hacer monitoreo del tráfico de bicicletas [1].

En el año 2016, Dr Frauke Behrendt presenta el articulo “Why cycling matters for Smart Cities. Internet of Bicycles for Intelligent Transport” donde destaca el desarrollo de un sistema SEMS (SMART E-BIKE MONITORING SYSTEM). Es una plataforma de código abierto para la adquisición de datos de uso de bicicletas eléctricas que pueden monitorear la ubicación, datos de control del ciclista y otra entrada de sensor personalizado en tiempo real, funcionando desde la batería de la bicicleta eléctrica sin

intervención del ciclista siendo la evolución de una bicicleta convencional a “Smart e-Bike” aplicando IoT , Smart Cities y Smart Mobility en Brighton (UK). [2] El resultado de la interacción entre los 80 participantes fue notable en una encuesta posterior, ya que los integrantes ingresaron al sitio web del sistema donde visualizaron todos los trayectos, tiempos de toda la flota. Lo cual impulsa al desarrollo de dispositivos IoT a bajo costo para el ciclismo. [2]

En el 2017 la empresa ORANGE BUSINESS publica el artículo “ Sherlock connects GPS-based anti-theft device for bicycles to global IoT network from Orange Business Services” [3] establecen la solución del mercado de accesorios incorpora un dispositivo de poco tamaño antirrobo, que está diseñado para ocultarse en el manillar de cualquier bicicleta y una aplicación móvil. El dispositivo integra un módulo GPS para la localización y un módulo GPRS para la conectividad Internet y Bluetooth de baja energía. Lo entregan con dos años de conectividad de red incluido el precio de compra. [3]

Por otra parte, en el Ecuador, la Universidad del Norte desarrolla una investigación en el año 2018 denominada “Plataforma para experimentación naturalística en bicicletas: Software de adquisición y tratamiento de datos”, en la cual buscan la unificación de la bicicleta y la tecnología de la información en una investigación realizada en Ibarra. Se presentó una aplicación para la adquisición y tratamiento de variables naturalísticas obtenidas desde la bicicleta, para esto establecieron requerimientos funcionales y no funcionales sobre el sistema a partir de las necesidades que demanda la aplicación utilizando protocolos como TCP-IP y SSH para la manipulación de los datos obtenidos desde la bicicleta. [4]

En el 2019 se desarrolla la investigación en la Universidad Técnica de Ambato con el nombre “Sistema autónomo de monitoreo de señales fisiológicas con gestión de emergencias para seguridad vial de ciclistas amateur” [5], realizada por Mario X. Bonilla. El trabajo presenta un sistema de emergencia para monitorear el estado de salud del ciclista con datos que puede ser visualizados en una HMI situadas en la estación fija secundaria del sistema, utilizando un sistema wearable en la indumentaria del conductor

con la cual procesa datos como temperatura, oximetría de pulso y frecuencia cardiaca, en el caso que sufra un accidente. [5]

Por otra parte, en Santiago de Chile el proyecto bottom-up Stgo2020 realizada por Martín Tironi Rodó y Matías Valderrama Barragán ante la cuestión por definir los lugares para instalar infraestructura vial, diseñan un diminuto dispositivo de self-tracking llamado RUBI. Este permite registrar los recorridos cotidianos de ciclistas que voluntariamente aportan colaborando con el estudio, gracias a esto los creadores del dispositivo pueden crear mapas y visualizaciones del uso de la bicicleta en la ciudad. [6]

### **1.2.1 Contextualización del problema**

En el Ecuador el uso de la bicicleta ha aumentado favorablemente como una alternativa de transporte saludable y sustentable con el medio ambiente, las estadísticas del INEC revelan la cantidad de personas que comenzaron a utilizar la bicicleta desde el año 2015 al 2016 pasando de 1.872.729 a 2.481.343 personas, por otra parte, como medio de transporte más utilizado en Ecuador ocupa el cuarto lugar [1]. Según la OMS en Ecuador, el uso de la bicicleta ha aumentado especialmente en Quito, impulsado por pandemia COVID-19 este aumento se reflejó en un 700%, a razón que desplazarse en bicicleta disminuye la posibilidad de contagio en contraste con el transporte público [2].

Las acciones públicas que benefician a los ciclistas a nivel nacional conducen principalmente a la provisión de infraestructura. Este es esencialmente el caso de la construcción de ciclovías de turismo y ocio que conecta ciudades y zonas rurales [3]. Las ciudades que disponen de espacios activos para realizar ciclo paseos actualmente son: Loja, Quito, Cuenca Guayaquil y Ambato. Sin embargo, las ciclovías no son el único punto de atracción para el turismo, sobre este mecanismo, también se han inaugurado Bike Parks en diferentes partes del Ecuador el más reconocido por el momento se encuentra en el Parque Nacional Cotopaxi donde la visita de turistas nacionales y extranjeros es

constante. No obstante, este medio de transporte presenta ciertos inconvenientes de inseguridad y accidentes para el ciclista.

Por otra parte, el delito de robo de bicicletas aumenta proporcionalmente, para cualquier persona ajena a los deportes, es posible que se sorprenda al saber que el valor de una bicicleta de enduro comienza en \$600 hasta los \$6000 según los costos publicados por una de las tiendas referentes en el Ecuador “bikehouse” [4], dichos precios son reducidos en un 50% en la comercialización clandestina.

Entre otros peligros está el irrespeto de la norma de distanciamiento entre el automóvil y el ciclista. Según la Agencia Nacional de Tránsito de Ecuador documentó que desde el 2017 hasta la fecha han fallecido 42 ciclistas al no ser socorridos rápidamente y desde junio del 2019 hasta diciembre del 2020 ,739 ciclistas resultaron lesionados solo en Quito. De acuerdo con la Facultad de Salud de la Universidad de las Américas (UDLA), en Ecuador desde el 2001 hasta el 2017 se registró un total de 317 muertes. El 84% de los fallecimientos se dieron en zonas urbanas y el 16 % en las rurales. [5]

Al mismo tiempo, el concepto de Internet de las cosas (IoT) representa la visión de que Internet se expandirá al mundo real, conectando los elementos físicos al mundo virtual y haciendo que la informática sea realmente ubicua. Según el informe Cisco Annual Internet Report, la cantidad de dispositivos conectados a redes IP será más del triple de la población mundial habrá 3.6 dispositivos en red per cápita para 2023. [6] Es notable el crecimiento de dispositivos conectados a la red, sin embargo, en Ecuador lo más cercano de tecnología para bicicletas es un ciclo computador el cual permite guardar rutas, tiempos en algunos casos es compatible con bandas de ritmo cardiaco mediante bluetooth, mas no ofrece seguridad ante accidentes o robos como alertas o alarmas en caso de no ser utilizada por el dueño mediante una conexión remota.

Por lo tanto, se busca generar una solución mediante una bicicleta inteligente diseñando e implementando un dispositivo que rastrea, monitorea el estado de la bicicleta, verifica el usuario y lo alerta en caso de robo. La información generada será enviada al usuario

utilizando IoT un concepto que beneficia a muchos ciclistas ya que podrán revisar dichos datos en cualquier parte del mundo mediante un teléfono inteligente.

### **1.2.2 Fundamentación teórica**

#### **Tecnología en bicicletas**

##### **Bici-conectada**

Su funcionalidad depende del teléfono inteligente o de la conexión integrada a la bicicleta que el usuario proporciona a través del teléfono inteligente, [7] transmitiendo información sobre ubicación, ruta, navegación, condiciones de la carretera, condiciones del tráfico, seguridad.

##### **Bici-inteligente**

Integra sensores para la seguridad del ciclista como la sensorización de neumáticos, frenos, velocidad, desgaste, etc. Interconectadas a un control operativo general. [8]

##### **E-Bike**

Es una bicicleta con asistencia al pedaleo, la potencia generada ya no depende solo del desempeño del ciclista, viene acompañada del motor eléctrico que esta implementado en la manzana trasera o en el eje de las bielas dependiendo de la marca de bicicleta. [9]

#### **Clases de bicicletas**

Los diseños de las bicicletas están categorizados según su funcionalidad dependiendo el tipo de terreno para el cual van a ser utilizadas es importante establecer el mercado al cual

se puede llegar con el dispositivo, ya que el costo debe ser mucho menor a la inversión de la bicicleta.

### **Bicicletas de montaña (MTB)**



Figura 1 Bicicleta de enduro [10]

Las bicicletas de montaña son diseñadas para fuera de ruta por lo que son bastante resistentes por ende tienen un marco robusto, manubrio recto y alto. Soportan Impactos. [10] Su precio puede variar entre los \$3.000 a \$9000

### **Bicicleta de ruta con manubrio drop-bar**



Figura 2 Bicicleta de ruta [10]

Están equipadas con un marco delgado, llantas delgadas, manubrio drop-bar que permite diferentes posturas. Su ventaja es ser muy ligera y veloz perfecta para grandes distancias. [10] Su precio va desde los \$1000(usadas) hasta \$15000 (profesionales).

### **Bicicleta híbrida**



Figura 3 Bicicleta Rígida [10]

Poseen mecanismos livianos son versátiles ya que cuentan con extras para la montaña o ruta, tienen un manubrio recto. [10] Su precio va desde los \$500 a \$1300

### **Sistema de alerta**

Un sistema de alerta es un conjunto de componentes electrónicos conectados a una red , que trabajan entre sí para evitar el robo y otras posibles intrusiones ilegales.

### **Internet de las cosas (IoT)**

Internet de las cosas (IoT) es una nueva arquitectura global basada en Internet que facilita el intercambio de bienes y servicios entre las redes de la cadena de suministro y tiene un impacto significativo en la seguridad y confidencialidad de las partes interesadas. [11]

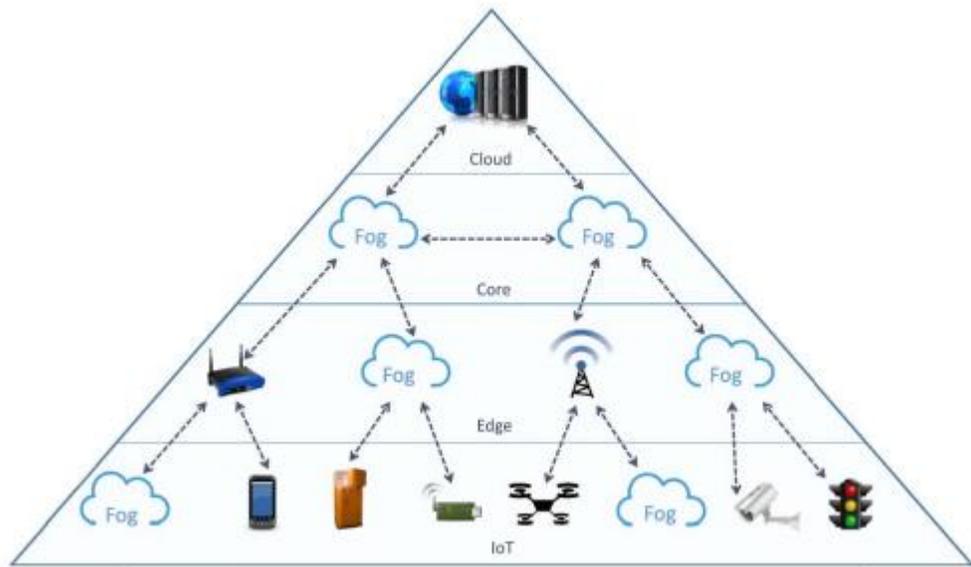


Figura 4 Estructura cloud computing [12]

## Cloud Computing

La nube no es una ubicación, sino una forma de administrar los recursos de TI que reemplazan las máquinas locales y los centros de datos privados con infraestructura virtual. En este modelo, los usuarios acceden a la máquina virtual, la red y los recursos de almacenamiento disponibles en línea a través de un proveedor remoto. Esto es especialmente útil para las organizaciones que necesitan expandir o contratar rápidamente su infraestructura para satisfacer la demanda cambiante. [13]

## Fog Computing

El modelo facilita la implementación de aplicaciones y servicios distribuidos, conscientes de la latencia, y consta de nodos FOG (físicos o virtuales), que residen entre dispositivos finales inteligentes y servicios centralizados (en la nube). [14]

## Edge Computing

La capa más externa de la nube representa la capa del borde. Edge es una colección de recursos espontáneos controlados por humanos, como computadoras centros de nano datos y tabletas. [15]

## Arquitectura Sistemas IoT

Cada año salen más dispositivos que hacen posible esta tecnología, al combinarse con IoT permiten recopilar, analizar y enviar datos a la red. Al implementar sistemas IoT para diferentes contextos de aplicación, se suele hacer uso de una arquitectura compuesta por 5 capas, se sabe que las arquitecturas no son un estándar por el momento, este proyecto presenta una arquitectura de 4 capas como se puede observar en la figura 5.

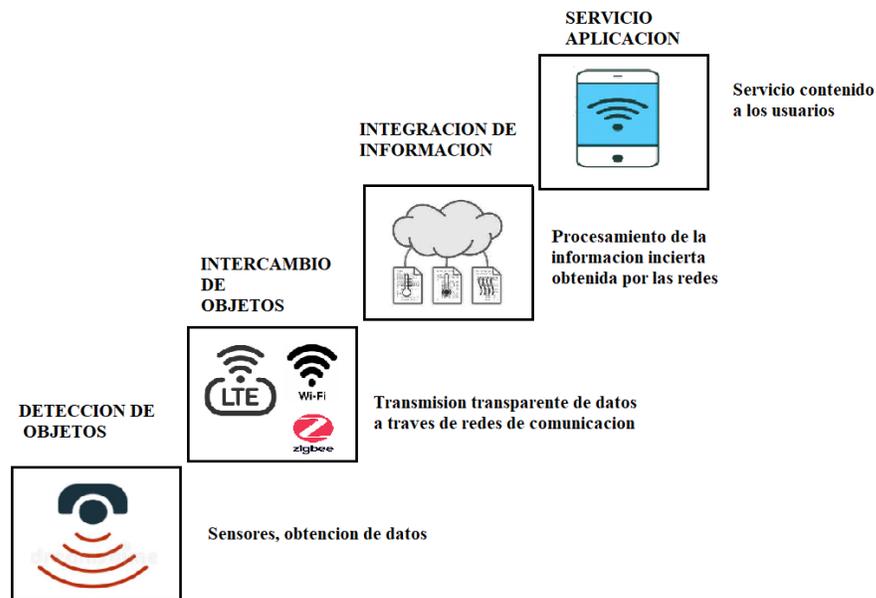


Figura 5 Arquitectura 4 capas IoT  
Elaborado por el investigador

## **Detección de Objetos**

En la primera capa los objetos conectados en red con sensores inteligentes empiezan a enviar información sobre sí mismos y su entorno a su centro de comunicaciones central en el microcontrolador. Conectar cosas (darles sentido y abrirles una conexión a Internet para que puedan enviarles sus datos) representa el comienzo de la evolución del IoT. [11]

## **Intercambio de objetos**

Transmisión de datos fluida a través de la red de comunicación por medio de tecnologías como 4G, Sigfox, LoraWan y N-biot. La capa de comunicación admite la conexión de dispositivos. Existen varios protocolos de comunicación entre los dispositivos y la nube. Los tres protocolos más conocidos y utilizados son:

- HTTP/HTTP
- MQTT 3
- CoAP (Constrained application protocol)

## **Integración de Información**

El procesamiento de información incierta obtenida de la red, filtrando datos no deseados e integrando información clave en conocimiento útil para el servicio y los usuarios finales. [11]

## **Servicio aplicación**

Este es el nivel más alto de abstracción en el que el usuario y el software del sistema interactúan entre sí. Da servicios de contenido a los usuarios

## **Modelos de comunicación de IoT**

Según el Comité de Arquitectura de Internet (IAB), publicó un artículo tutorial sobre redes de objetos inteligentes (RFC 7452). Describe cuatro modelos de comunicación utilizados en dispositivos IoT. [16]

### **Comunicaciones dispositivo a dispositivo**

El modelo de comunicación dispositivo a dispositivo representa dos o varios dispositivos que se conectan o comunican directamente sin utilizar un servidor de aplicaciones intermediario. Estos dispositivos se comunican sobre muchos tipos de redes, entre ellas las redes IP o la Internet. Sin embargo, para establecer comunicaciones directas de dispositivo a dispositivo, muchas veces se utilizan protocolos como Bluetooth, Z-Wave o ZigBee. [16]

### **Comunicaciones dispositivo a la nube**

En un modelo de comunicación de dispositivo a la nube, el dispositivo de la IoT se conecta directamente a un servicio en la nube. [16]

### **Modelo dispositivo a puerta de enlace**

Un software de aplicación corriendo en un dispositivo de puerta de enlace local, que actúa como intermediario entre el dispositivo y el servicio en la nube y provee seguridad y otras funcionalidades tales como traducción de protocolos o datos. [16]

### **Modelo de intercambio de datos a través del back-end**

Se refiere a una arquitectura de comunicación que permite que los usuarios exporten y analicen datos de objetos inteligentes de un servicio en la nube en combinación con datos de otras fuentes. Esta arquitectura soporta “el deseo del usuario de permitir que terceros accedan a los datos subidos por sus sensores” [16]

## Tarjeta de desarrollo

Las tarjetas de desarrollo son placas que poseen un microcontrolador y un hardware, para facilitar la creación y el abaratamiento de circuitos tanto para principiantes como para expertos. [17] La mayor parte de las aplicaciones de IoT requieren simplemente añadir un sensor a un objeto físico.

## GNSS (Global Navigation Satellite System)

De manera general se nombra a los Sistemas de Navegación por satélite que suministra un posicionamiento geoespacial con cobertura global. Las señales GNSS son proporcionadas por cuatro sistemas carácter global (GPS, GLONASS, GALILEO y BeiDou) y dos regionales (QZSS o IRNSS), a menudo complementado con sistemas de aumentación, basado en satélites (SBAS), como EGNOS. [18]

Constelación	GLONASS	GPS	BEIDOU	GALILEO
Pertenencia	Federación Rusa	EE. UU.	República Popular China	Unión Europea
# Satélites	24	31	35	30

*Tabla 1 Sistemas de geolocalización*

*Elaborado por el investigador*

## GPS (Sistema de Posicionamiento Global )

El sistema de posicionamiento global (GPS) fue desarrollado por el Departamento de Defensa de los Estados Unidos de América como “Sistema de Navegación mediante Tiempo y Distancia”, NAVSTAR [19], y fue fabricado para proveer capacidad de navegación a las fuerzas militares de EE. UU. en aire, mar y tierra. Esto produjo un beneficio colateral, el sistema en una aplicación en el área civil, incluyendo recreación control vehicular, navegación aérea y marítima y levantamientos topográficos. Actualmente en el Ecuador esta tecnología es ofrecida por diversas empresas de

telecomunicaciones como son CNT, Claro, Movistar y Tuenti, ofertando planes prepago y pospago, lo cual convierte en una tecnología de bajo costo para aplicaciones de rastreo. [19]

### **Estructura del G.P.S.**

El GPS está conformado por tres segmentos: satélites(espacial), estaciones terrenas (control)y los receptores(usuario). [20]Las 5 estaciones de la tierra están distribuidas a distancias similares alrededor del Ecuador, y tienen como objetivo monitorear el estado de los satélites, realizar pequeños ajustes en sus orbitas y calcular las posiciones de los satélites.

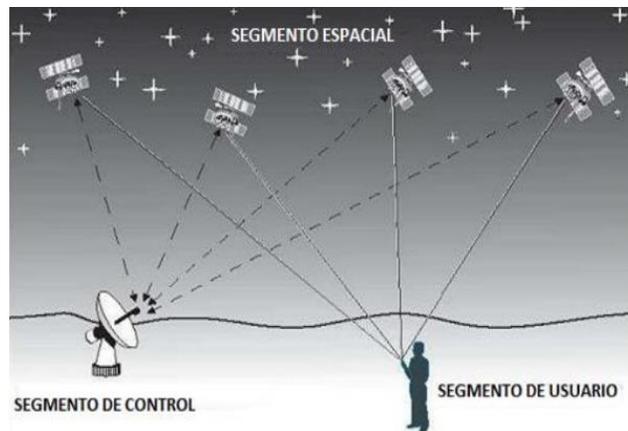


Figura 6 Estructura GPS [19]

Esta información es transmitida a los satélites, los cuales a su vez la transmiten a los receptores en tierra al menos una vez al día. El tercer componente es el usuario quien recibe las señales enviadas por los satélites a través de un receptor equipado con una antena. [19] La estructura de funcionamiento de GPS se muestra en la Figura 6.

## Segmento Espacial

La fuerza Espacial de EE. UU. Tiene 31 satélites GPS durante más de una década y se compromete a mantener la disponibilidad de su servicio al 95% del tiempo con 24 satélites GPS operativos. [21]

El segmento espacial GPS consiste en una constelación de satélites que transmiten señales de radio a los usuarios, estos vuelan en la órbita terrestre (MEO), a una altitud aproximada de 20200 Km. [21]

## Segmento de Control

Una red global de estaciones terrestres rastrea los satélites GPS sus funciones principales realizar análisis, monitorear sus transmisiones y enviar comandos y datos a la constelación. [21]

Las estaciones de monitoreo tienen coordenadas conocidas por su alta precisión y están equipadas con un receptor GPS de doble frecuencia L1 / L2 y un reloj de cesio. [22] El segmento de control está constituido por 10 estaciones de monitoreo (NGA) los cuales están distribuidos de la siguiente manera:

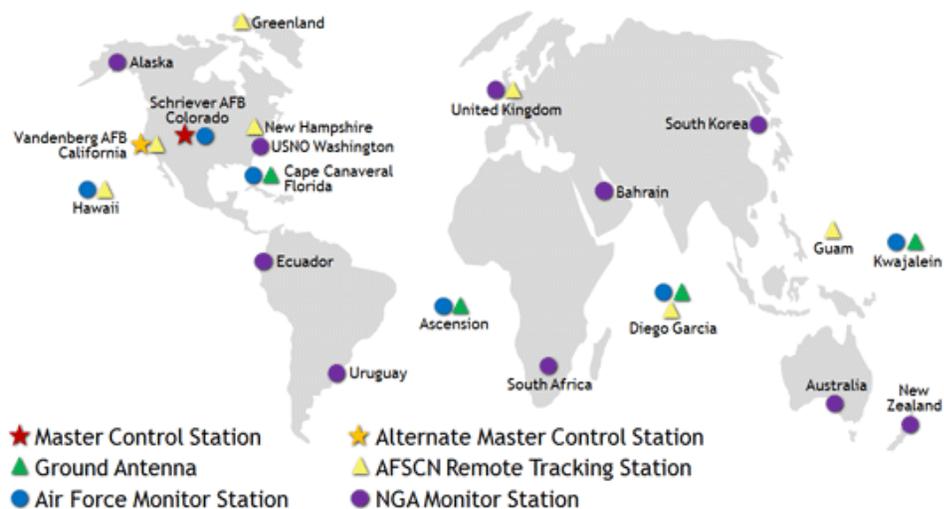


Figura 7 Estaciones de monitoreo .Elaborado por gps.gov [21]

## **Segmento de Usuario**

Conformado por los receptores GPS sus funciones son:

- Recibir las señales emitidas desde los satélites
- Decodificar mensajes de navegación
- Calcular el retardo de la señal
- Presentar la información de la posición en la que se encuentra [23]

## **Código C/A y señales GPS**

Cada uno de los satélites GPS transmite continuamente una señal utilizando dos frecuencias en la banda L: 1575.42 Mhz(L1) y 1227.6 Mhz (L2). L1 fue designada para uso sin límites del sector civil y es modulada por un código de ruido seudo aleatorio (PRN) denominado código de adquisición/grueso o estándar (C/A). Esta señal contiene un código único con longitud de 1023 bits que se repite cada milisegundo y 50 bits por segundo de información sobre aspectos de navegación. El código P está restringido a usuarios autorizados por el Departamento de Defensa de los Estados Unidos de América. Los usuarios civiles pueden utilizar el código P(Y) pero sin tener acceso al código de seguridad. [19]

## **Origen de error del GPS**

La información de los satélites que viaja dentro de una señal puede verse afectada por una variedad de fuentes de error como se puede observar en la figura 8:



Figura 8 Error GPS [24]

### Tiempo

Los relojes tanto de los satélites como de los receptores presentan variaciones al momento de la transmisión de radio, es decir que un microsegundo de desfase se convierte en un error de 300 metros en la medición de la distancia. [25]

### Ionósfera y Tropósfera

La señal de radio se transmite a la velocidad de la luz ( $3 \cdot 10^8$  m/s) siendo constante mientras se encuentra en el vacío, este parámetro varía al momento de ingresar a la atmósfera terrestre, debido a que debe pasar por una sección de partículas cargadas en la ionósfera y luego al entrar en contacto con el vapor de agua de la tropósfera disminuye la velocidad, mientras que el receptor asume que la velocidad sigue siendo constante esto introduce una fuente de error en el cálculo de la ubicación. [19]

### Efecto Multitrayectoria

Se presenta cuando las señales son reflejadas en objetos opacos antes de llegar al receptor son obstáculos como montañas grandes o edificios, incrementando el tiempo de envío de la información causando otra fuente de error. [26]

## **Red de comunicación móvil**

La evolución de las redes celulares es diversa y cambia con el tiempo. Por esta razón, la evolución de una tecnología a otra se conoce comúnmente como "generación de red celular". Cada generación tiene nuevos estándares, capacidades técnicas y características que la distinguen de la generación anterior.

### **Primera Generación**

NTT lanzó la primera red de comunicación móvil automatizada comercial en Japón en 1979, seguida de la introducción del sistema Nordic Mobile Phone (NMT) en Dinamarca, Finlandia, Noruega y Suecia, en 1981, utilizando el estándar AMPS (Advanced Mobile Phone System). Esta tecnología consumía un ancho de banda de RF de 30 kHz con una capacidad de 832 canales dúplex. Cabe señalar que la comunicación de voz es ineficaz ya que la llamada se reproduce en la torre de radio. [27]

La primera generación de red móvil utilizaba tecnología analógica alrededor del mundo, y ofrecía solo el servicio de voz para los usuarios, sin embargo la diferencia de sistemas y soluciones alrededor del mundo no permitían la continuidad de la conectividad internacional (roaming). [28]

### **Segunda Generación**

La transición de tecnología analógica a digital en la comunicación móvil comienza en los años 90. Servicios presentados son: Voz digital, SMS, roaming internacional transferencia y bloqueo de llamadas.

Varias tecnologías han dado forma al camino de esta generación:

- GSM (Sistema Global para Comunicaciones Móviles) en Europa

- PDC en Japón Personal Digital Celular, antes JDC en Japón
- IS-95(CDMA) utilizado para América y partes de Asia

En Sudamérica, especialmente en Ecuador, el advenimiento de la tecnología móvil se remonta a 1993, cuando dos empresas obtuvieron una licencia de uso de los servicios móviles de segunda generación Conecel S.A. y Otecel S.A. . La marca CLARO(Conecel S.A.)implementa GSM con tecnología TDMA y Movistar S.A. (Otecel) implementa IS-95 con tecnología CDMA. [29]

### Generación 2.5

Introducción al sistema Servicio General de Paquetes por Radio (GPRS) permite la transmisión de información en paquetes, los cuales son transmitidos, reunificados y presentados en una aplicación web o móvil. El GPRS logra esto utilizando la tecnología de ranuras múltiples. La ventaja es que solo se tiene que pagar por el contenido que se baja de la red y no por todo el tiempo que se está conectado a ella. [30]

GPRS está dirigida a la comunicación de paquetes. La diferencia principal entre la comunicación orientada a circuitos y una orientada a paquetes es la utilización de recursos de red; mientras en circuitos se ocupa el recurso durante toda la comunicación, en paquetes solo es mientras existe algo que enviar o recibir. [30] Esto optimiza los recursos y permite traficar no por tiempo de conexión, sino por volumen de datos conmutados.



Figura 9 Objetivo GPRS [30]

### **Tercera Generación**

La necesidad de integrar servicios de voz y datos con aplicaciones multimedia con acceso inalámbrico a internet y la alta transmisión de datos dieron origen a la tercera generación. [29] Esta generación es un estándar 3GPP global y se lo conoce como Sistema Universal de Telecomunicaciones Móviles (UMTS). [31]

La generación 3G presenta los siguientes servicios:

- Acceso a internet de alta velocidad para multimedia
- Correo electrónico
- Servicios de ubicación
- Navegación web
- Servicios locales para acceder a actualizaciones meteorológicas y de tráfico.

### **Cuarta Generación**

El sistema móvil 4G se basa completamente en el Protocolo de Internet (IP). Para 4G, la velocidad de transmisión es de 100 Mbit/s a 1 Gbit/s, aprovechando al máximo el protocolo IP. Las pruebas de funcionamiento de estas redes comenzaron en 2010 y se puede observar que las redes basadas en esta tecnología, todavía se están construyendo. [28] Para utilizar la red de comunicación móvil 4G, las estaciones de usuario deben poder identificar el sistema inalámbrico de destino.

La movilidad de los dispositivos es un elemento importante de las redes 4G para brindar servicios inalámbricos en cualquier momento y en cualquier lugar. [32] LTE opera en una arquitectura basada en dos protocolos: el protocolo de control de transmisión (TCP), que se encarga de la transmisión de datos, y el Protocolo de Internet (IP), que proporciona las direcciones de los dispositivos que se conectan a la red. Los sistemas LTE coexisten con los sistemas 3G y 2G (4G Américas). [31]

En la figura 10 se muestra el desarrollo de la infraestructura de cuarta generación con la tecnología del fabricante Nokia.

- Equipo de usuario (EU)
- Red de acceso terrestre Evolución UMTS(E-UTRAN)
- Red núcleo llamada Núcleo de Datos Evolucionado (EPC).

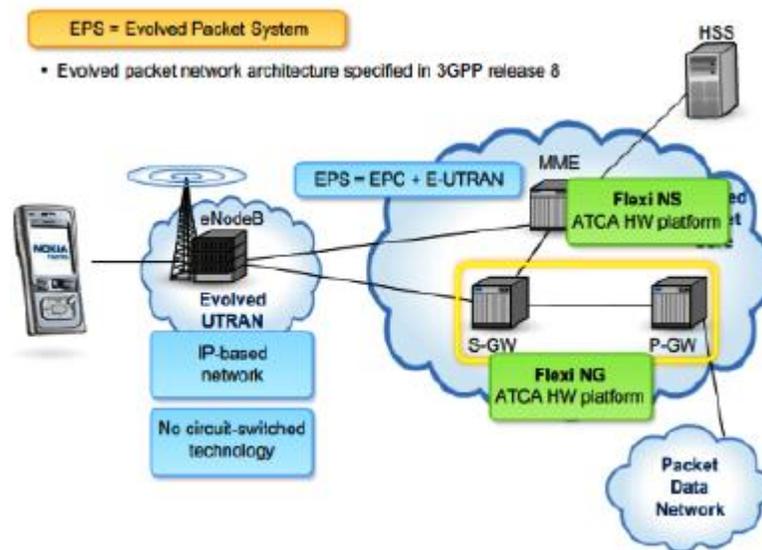


Figura 10 Arquitectura red LTE [33]

Las características más significativas de la red 4G son:

- Tasa de datos máxima de 1 Gbps para DL (Down link-forward channel) y 500 Mbit/s para UL (up link reverse channel) para usuarios fijos. Para usuarios móviles, 100 Mbit/s DL y 50 Mbit/s UL.
- Es necesario asegurar la movilidad hasta 350 km/h con IMT-avanzadas
- Bandas de frecuencia: - LTE cubre una gama de diferentes bandas en América del Sur cubren 700, 800, 900, 1800, 2600 MHz
- Estándares Long-Term Evolution Time-Division Duplex (LTE-TDD y LTE-FDD) estándar WiMAX móvil (802.16m estandarizado por el IEEE)
- Tecnologías de multiplexación / acceso - OFDM, MC-CDMA, CDMA y LAS-RedLMDS

- Ancho de Banda - 5-20 MHz, opcionalmente hasta 40 MHz. [34]

### 5ta Generación

Se espera que la quinta generación de tecnología proporcione una enorme capacidad de tráfico. Nuevo rango espectral para alta velocidad, uso eficiente del espectro y gran densidad, depende del usuario. Otra expectativa es que se puede esperar apoyo de misión crítica. Muy baja latencia, alta confiabilidad, disponibilidad y seguridad.

La capa de enlace físico y de datos define la tecnología inalámbrica 5G, lo que demuestra que se trata de una tecnología de arquitectura inalámbrica abierta (OWA). Para hacer esto, la capa de red se divide en dos capas. La capa de red superior de la estación móvil y la capa de red inferior de la interfaz. Aquí, todo el enrutamiento depende de la dirección IP. Esto es diferente para todas las redes IP del mundo. Con la tecnología 5G, la pérdida de energía se compensa con el Protocolo de transferencia abierta (OTF). [27]

En Ecuador en la actualidad no se dispone de una red móvil 5G , sin embargo, en el 2021 se dispuso de 3 áreas experimentales de la red de quinta generación las cuales se encuentran distribuidas dos en Guayaquil y una en Manta esto es gracias a la cooperación de las empresas CNT y la finlandesa Nokia.

**Tabla Comparativa Red móvil**

Generación	Estándar	Velocidad	Multiplexación	Servicios
1G	AMPS	1Kbps- 2,4Kbps	FDMA	Llamadas análogas
2G	D-AMPS	14Kbps- 64Kbps	TDMA-CDMA	Llamadas digitales y mensajería

2.5G	GPRS- EDGE	115 kbps(GPRS) 384Kbps (Edge)	GMSK(GPRS) EDGE (8PSK)	Soporte WAP,MMS, SMS, Juegos móviles
3G	UMTS HSPA HSPA+	384Kbps- 2Mbps	TD- SCDMA,WCDMA,TDD,FDD	Llamadas, Mensajería y Banda ancha
4G	LTE,LTE- A	100Mbps (en movimiento) 1Gbps (estático)	OFDMA, OFDM	Servicios IP

*Tabla 2 Tabla comparativa red móvil.*

*Elaborado por el investigador*

### **Tarjeta universal de Circuito Integrado (UICC)**

Esta tarjeta contiene toda la información del usuario y se utiliza para autenticar el procedimiento de registro de servicios telefónicos en la red del consumidor. En la 2.<sup>a</sup> generación se denomina Módulo de identidad del suscriptor (SIM) y en la 3.<sup>a</sup> y 4.<sup>a</sup> generación se denomina SIM universal (USIM). Desconectar el dispositivo de la tarjeta le permite al usuario autenticarse en la red usando otro dispositivo móvil. La UICC se obtiene del operador.

### **Operadoras Móviles**

Las empresas que cuentan con licencia o permiso del gobierno nacional para operar redes móviles son:

OPERADORAS	3G	4G
Claro	850MHz 1900 MHz	1700MHz 2100 MHz
Movistar	850MHz 1900 MHz	1900 MHz
Tuenti	850MHz 1900 MHz	1900 MHz
CNT	1900 MHz	700 MHz 1700 MHz 2100 MHz

*Tabla 3 Frecuencias para 3g , 4g en Ecuador*

*Elaborado por el investigador*

### **Modelo cliente servidor**

Es importante mencionar la arquitectura Cliente-Servidor dado que el proyecto de geolocalización necesita dar información a los usuarios finales en una interfaz transparente multiplataforma. En el modelo cliente/servidor, el cliente envía un mensaje solicitando un servicio específico a un servidor, y este envía uno o varios mensajes con la respuesta. [35]

### **Cliente**

Opera todas las funciones entre la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces graficas (GUI), conjuntamente accede a los servicios distribuido en cualquier parte de una red. [35]

Sus funciones principales son:

- Administras la interfaz de usuario.

- Interactuar con el usuario.
- Recibir resultados del servidor.
- Procesar la lógica de la aplicación y sustentar validaciones locales [35].

## **Servidor**

El objetivo del servidor es revisar a múltiples clientes que realizan peticiones de algún recurso administrado por él, las funciones que lleva a cabo se lo resumen a continuación:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Formatear bases de datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de base de datos. [35]

## **Hosting**

El hosting es un servicio de alojamiento en línea que le permite publicar sitios web o aplicaciones web en Internet. Cuando obtiene alojamiento, básicamente alquila espacio en el servidor para almacenar todos sus archivos y datos para que su sitio web funcione correctamente.

## **Lenguajes y entornos de programación**

### **Php**

PHP (acrónimo recursivo de PHP: preprocesador de hipertexto) es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo web, se centra en la secuencia de comandos del lado del servidor. [36]

## **phpMyAdmin**

Es un software de código libre basado en php, con la capacidad de manejar y gestionar la base de datos de MySQL, a través de una interfaz gráfica de usuario, con esta herramienta se puede crear y eliminar, bases de datos, tablas, campos , ejecutar sentencias SQL, modificar los privilegios , y exportar Datos en varios formatos. [37]

## **Kotlin**

Es un lenguaje de código abierto permite la interoperabilidad nativa con Java, incluso es posible desarrollar código para proyectos usando ambos lenguajes al mismo tiempo sin ningún problema, GOOGLE lo reconoció como oficial para dispositivos Android desde el año 2017, siendo el posible reemplazo de JAVA se ejecuta bajo la licencia de APACHE 2.0, su disminución en la repetición de códigos que representa el 40% frente a otros lenguajes , hace que sea más eficiente y facilita la localización de errores en caso de que se produzcan. [38]

## **Android studio**

Es un IDE de desarrollo basado en IntelliJ IDEA, es gratuito y oficial para la programación de aplicaciones Android, es compatible con lenguajes como Kotlin, NDK y C++. [38]

## **1.3 Objetivos**

### **1.3.1 Objetivo general**

Implementar un sistema inteligente de seguridad y alertas para bicicletas en el Ecuador basado en IoT

### **1.3.2 Objetivo específico**

- Analizar los dispositivos inteligentes de bicicletas que apoyan a la seguridad de los ciclistas en actividades recreativas o de transporte.
- Establecer la tecnología para el sistema inteligente para bicicletas dependiendo de las variables a monitorear.
- Diseñar un sistema IoT que cumpla con los requerimientos de un sistema de seguridad y alerta para bicicletas.

## CAPÍTULO II

### METODOLOGÍA

#### 2.1. Materiales

La información para el desarrollo e implementación del proyecto en cuestión se obtiene de artículos, tesis, libros, revistas, documentos web y cualquier tipo de información relacionada con el tema. Para la selección del material se seleccionó un análisis de los componentes del hardware, mientras que el software utilizado fue libre y específico para los dispositivos utilizados en el sistema. El hardware consta de varios dispositivos , cuyos componentes se muestran a continuación:

#### Hardware

- Placa IoT T-SIM7600
- Sensor de inclinación
- Sensor Biométrico
- Antena LTE
- Antena GPS
- Batería de Li-Ion
- Interruptor
- Cables

#### Software

- Ide Arduino
- Ide de Android
- Visual Basic Ide

- Hosting

## **2.2. Métodos**

### **Modalidad de la Investigación**

En el presente trabajo de investigación se aplicó la investigación bibliográfica fundamentando los conceptos teóricos necesarios sobre IoT, Redes móviles, y Modelo cliente/servidor para el diseño del dispositivo IoT de geolocalización y alarmas de seguridad para bicicletas basado en fuentes como libros y artículos científicos.

Se realizó la investigación aplicada empleando todos los conocimientos adquiridos a lo largo de la investigación bibliográfica, dando una solución tecnológica al hurto de bicicletas en el Ecuador y a la falta de alertas de seguridad ante accidentes en las vías.

Una vez realizado el diseño se aplicó la investigación experimental puesto que el objetivo principal es implementar dicho sistema, realizando pruebas de funcionamiento y error para obtener un dispositivo completamente funcional para el usuario.

### **Recolección de Información**

Para el desarrollo de este proyecto de investigación se utilizó información enfocada en sistemas inteligentes para vehículos, obtenida de libros, artículos científicos, y revistas científicas, simultáneamente se recolectó información sobre protocolos de comunicación y arquitectura basada en IoT disponibles en el repositorio digital de la Universidad Técnica de Ambato, así como guías prácticas y manuales técnicos para el uso de hardware y software por lo que se tomó en cuenta bases de datos confiables que permitieron el desarrollo del proyecto.

## **Procesamiento y Análisis de Datos**

El procesamiento y análisis de datos se realizará mediante los siguientes pasos:

- Revisión de la información recopilada.
- Extracción de los datos a puntos de interés clave específicos determinando las variables y alertas para monitorear.
- Identificación de las mejores alternativas para resolver el problema y desarrollar así una propuesta de solución.
- Verificación que el dispositivo implementado este completamente funcional y transmita la información correctamente.

## **Desarrollo del Proyecto**

El presente proyecto de investigación se desarrolló bajo las siguientes actividades:

- Análisis de la situación actual de los diferentes dispositivos inteligentes que se comercializan en el Ecuador para la seguridad de la bicicleta.
- Establecer los problemas que se generan al no utilizar ninguna medida de seguridad ya sea mecánica o electrónica.
- Análisis de variables que intervienen en monitoreo del estado de la bicicleta mediante el uso de sensores biométricos y geolocalización.
- Análisis de los sistemas de comunicación para la adquisición de la información de variables mediante IoT
- Selección de la plataforma IoT.
- Selección de sensores, y otros componentes de hardware para el dispositivo inteligente.
- Incorporación de módulos para la adquisición de datos.
- Creación de la aplicación servidor-cliente que permita la visualización de información sobre la bicicleta.

- Diseño de la interfaz multiplataforma para la visualización de datos por parte del usuario en la web o aplicación móvil.
- Implementación del sistema inteligente y alertas ante peligros de robos o accidentes.
- Corrección de errores mediante pruebas de funcionamiento del sistema inteligente en campo
- Elaboración de informe final del proyecto.

## **CAPÍTULO III**

### **RESULTADOS Y DISCUSIÓN**

#### **3.1 Análisis y discusión de los resultados.**

##### **Desarrollo de la propuesta**

Las bicicletas parecen ser una alternativa viable en el contexto de problemas en cuanto a la movilización en el país, debido a los altos costos que representa el combustible en los vehículos motorizados, el uso de la bicicleta se ha hecho más común en Ecuador desde la pandemia por el COVID en el 2019, por ende, el presente proyecto busca una solución a la inseguridad en cuanto al hurto de estas.

##### **Análisis de los dispositivos de seguridad**

El proyecto de investigación se encuentra destinado al resguardo de las bicicletas mediante el uso de la plataforma IoT, para el cual se desarrolla primero un análisis de dispositivos inteligentes en el Ecuador con prestaciones dirigidas a la seguridad del ciclista exponiendo sus características y tecnología en la siguiente tabla:

Marca	Características
<p data-bbox="496 342 724 373">GF07 - Mini GPS</p> 	<p data-bbox="919 289 1143 321">Red: GSM/GPRS</p> <p data-bbox="919 342 1409 373">Banda GSM: 850/900/1800/1900Mhz.</p> <p data-bbox="919 394 1414 485">Tiempo de localización GMS: 25 segundos</p> <p data-bbox="919 506 1414 596">Potencia de entrada: CA 110-220V 50/60Hz</p> <p data-bbox="919 617 1414 648">Potencia de salida: DC5V 300 - 500mA</p> <p data-bbox="919 669 1414 760">Capacidad de la batería: 3.7 V 400 mAh Li-ion Batería</p> <p data-bbox="919 781 1414 871">Voltaje de funcionamiento: 3,4 - 4,2 V CC.</p> <p data-bbox="919 892 1105 924">Precio: \$17.49</p>
<p data-bbox="532 1056 686 1087">IGPSPORT</p> 	<p data-bbox="919 1056 1203 1087">Tecnología: Bluetooth</p> <p data-bbox="919 1108 1341 1140">Batería duración: 30Horas aprox.</p> <p data-bbox="919 1161 1341 1192">GNSS: GPS + BEIDOU + QZSS</p> <p data-bbox="919 1213 1414 1367">Sensores: Monitor de frecuencia cardíaca, velocidad, cadencia, sensor combinado de cadencia y velocidad.</p> <p data-bbox="919 1388 1179 1419">Interfaz: Micro USB</p> <p data-bbox="919 1440 1162 1472">Incluye mapas: No</p> <p data-bbox="919 1545 1057 1577">Precio:\$50</p>

<p>Candado de Seguridad con Alarma Antirrobo para Bicis</p> 	<p>Sistema :auto alarma y bloqueo  Sirena : 110 decibeles  2 tonos de alarma  Reset automático  Sensor :choque y movimiento.  Resistente a la corrosión. e intemperie  Alarma suena en intervalos de 10 segundos  Resistente al abuso físico  Material: acero y aleación de zinc  Precio:8\$</p>
---	--

*Tabla 4 Comparación de dispositivos inteligentes para bicicletas en Ecuador*

*Elaborador por el investigador*

Tras analizar la tabla 4 se encuentra que en el mercado los dispositivos, ofrecen soluciones específicas para la bicicleta, como se observa el dispositivo Mini GPS GF07, no está diseñado para cumplir con las variables del prototipo que se propone en la investigación, sin embargo, se puede usar ya que presenta un GPS interno para la localización y usa tecnología GMS que está un poco atrasada con las tecnologías móviles actuales, para la geolocalización hace uso de un sitio web.

En el segundo dispositivo presentado IGPSPORT, el cual viene a ser un ciclo computador destinado para el ciclista, debido a que se encarga de almacenar los datos importantes de la actividad realizada como son: frecuencia cardiaca, potencia, velocidad y distancia. La desventaja principal es que a pesar de disponer de mapas y tener una pantalla integrada, su comunicación es vía bluetooth con el dispositivo celular para cargar los datos obtenidos dentro de su app.

Por último, se presenta un dispositivo dirigido a la seguridad de la bicicleta, es un candado fabricado en acero con aleación de zinc, su funcionamiento se basa en la manipulación del dispositivo, al pasar de 5 segundos este emite una alarma audible para alertar al dueño de una emergencia, presenta ineficiencia debido a que a una distancia considerable el sistema de alarma no servirá.

Entre los dispositivos analizados se observó que carecen de un monitoreo en tiempo real a larga distancia, por lo que no son factibles para considerar como dispositivos de seguridad en la actualidad.

### **Problema de seguridad**

La seguridad a nivel nacional ha disminuido y las bicicletas no están absueltas de esta problemática, partiendo del año 2020 donde el periódico “Últimas Noticias” publicó un artículo que describen el incremento de robos de bicicletas, 130 robos hasta el mes de noviembre de ese año y que las pesquisas apuntan a bandas delictivas organizadas que están dedicadas al robo de bicicletas de alta gama.

En el mercado legal cuestan entre \$3.000 y \$9.000, pero las redes criminales las revenden a la mitad, luego de ser puestas a la venta en cachinerías y redes sociales. Por ello, los agentes advierten que detrás de todos los hechos se esconde “organizaciones criminales especializadas”.

### **Requerimiento del sistema**

En vista de los dispositivos presentados anteriormente y las preocupaciones de seguridad actuales, se analizan las variables más importantes para establecer en un sistema inteligente basado en IoT.

Los parámetros más importantes que se debe tomar en cuenta para el diseño del dispositivo inteligente son: la ubicación en caso de robo, la capacidad de geolocalizar la bicicleta, el sistema de alerta o el modo de aparcamiento para el propietario en caso de que no esté cerca de la bicicleta dentro de este modo se necesitara de alguna forma verificar quien utilice el vehículo y de no ser el dueño el sistema realizara una llamada de emergencia a través de una red móvil al propietario.

El sistema desarrollado se muestra en la figura 11, consiste en una placa de desarrollo IoT instalado en el cuadro de la bicicleta, que permite el control entre sensores en la bicicleta, desde el mismo dispositivo se obtiene la ubicación mediante una antena GPS instalada, el IMEI del dispositivo que servirá como código de registro para los usuarios, estos datos se enviarán a un web Services para procesar la información mediante PHP y luego se almacenarán en la base de datos PhpmyAdmin, con los datos listos se puede enviar a una aplicación móvil, donde se encuentra la bicicleta, mediante las Apis de Google Maps.

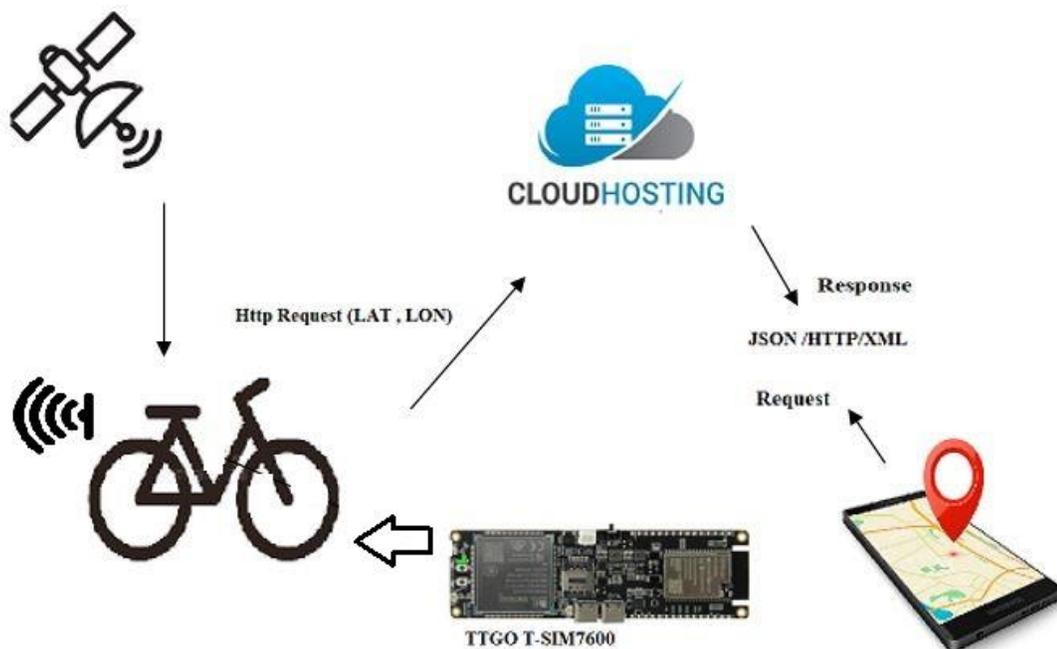


Figura 11 Sistema Inteligente de seguridad y alertas para bicicletas

*Elaborado por el investigador*

## Tecnologías de comunicación IoT

En términos de comunicación, hoy en día, la comunicación de fibra óptica de banda ancha con la llamada WLAN, así como la comunicación inalámbrica Wifi y los teléfonos móviles 4G son las tecnologías de comunicación dominantes.

Antes de seleccionar la plataforma IoT se establece el sistema de comunicación a utilizar en este proyecto sabiendo que se debe registrar la ubicación a distancias considerables, en el siguiente diagrama se visualiza las comunicaciones inalámbricas por sus distancias.

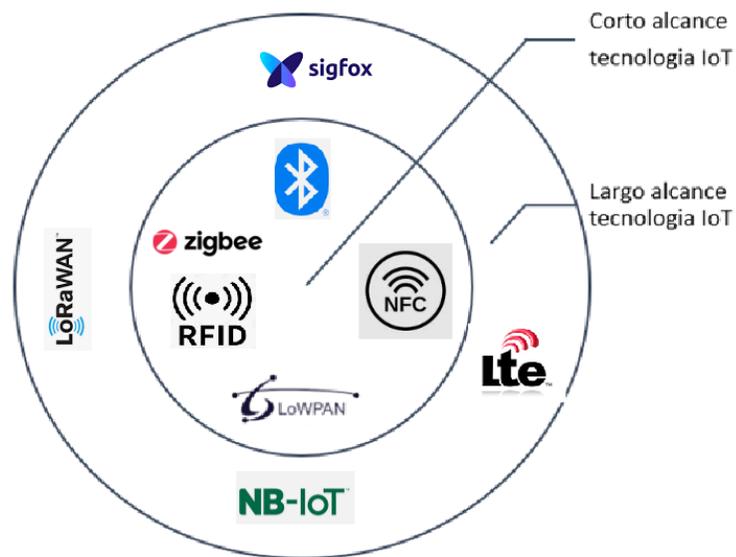


Figura 12 Tecnologías dedicadas al IoT

Dado que el objetivo de este proyecto es conseguir la detección de un dispositivo en zonas urbanas o rurales y también debe realizar llamadas de emergencia que servirán de alerta al dueño en caso de robo, se seleccionarán cuatro tecnologías de largo alcance para analizar sus características.

Características	LTE-M	NB-IoT	Sigfox	Lora WAN
Velocidad de transmisión de datos	1 Mbps	27.2-62.5 kbps (DL-UL)	10-1000bps	0.3-50 kbps
Alcance	1-50 Km	1-50 Km	30-50 Km (rural) 3-10Km (urbanos)	2-5 Km (urbano) 15 km (rural)
Frecuencia	Licencia 700-900 Mhz	Licencia 700-900 Mhz	Sin licencia 868 Mhz 915Mhz	Sin licencia 433 Mhz
Latencia	100 a 150 ms	1.5 – 10 s		

*Tabla 5 Cuadro comparativo de tecnologías IoT*

*Elaborado por el investigador*

Con los datos obtenidos en la tabla 5 se establece que LTE-M tiene una velocidad de transmisión superior a las demás, así como su alcance, lo que es fundamental para la selección de la plataforma IoT, una de las principales desventajas encontradas en las demás tecnologías es su disponibilidad en Ecuador, se tiene como ejemplo a Sigfox, que es una tecnología muy usada en Europa, Sigfox posee poca cobertura en Ecuador en cambio, Lora WAN la velocidad de transmisión es muy baja así como su alcance lo que no es útil para la aplicación deseada.

### **Etapas del sistema IoT**

Prototipando un sistema inteligente, se plantea el proceso de funcionamiento en tres etapas, la primera etapa es la captación de datos y procesamiento. La segunda etapa se basa en el envío de los datos procesados mediante comunicaciones móviles hacia el hosting donde se almacena la ubicación de la bicicleta, finalmente se procede a la última etapa, la aplicación, donde se visualiza el posicionamiento de la bicicleta.

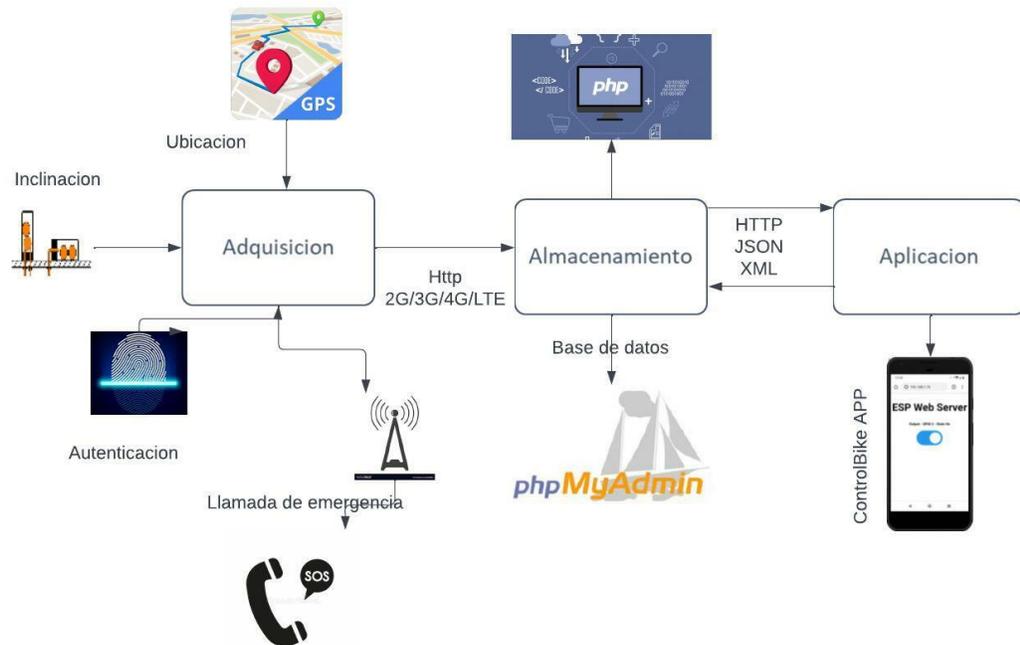


Figura 13 Estructura del sistema de seguridad a desarrollar

*Elaborado por el investigador*

### **Etapa de Adquisición**

En esta etapa se hace uso de la primera capa mencionada anteriormente, detección de objetos, donde se utilizó un GPS neo 6, incorporado en la placa de desarrollo seleccionada, esta puede conectarse a diferentes sistemas de navegación global por satélite como: Glonass, BeiDou, Galileo y GPS , su efectividad es mayor en zonas abiertas.

Un sensor lector de huellas capacitivo es colocado para autenticar al usuario en el modo parking caso contrario se hace uso del chip SIM 7600 G , que trabaja en 2g/3/4g y LTE el cual realiza la llamada de emergencia a un número determinado siendo esta la alerta para que el usuario revise la ubicación de su bicicleta en la app final.

## Diseño de la etapa de adquisición

Esta etapa tiene como principal objetivo el control de los elementos físicos y la obtención de la información, en la figura 14 se puede apreciar el diagrama de flujo que describe el funcionamiento del sistema en sus varios “bloques” con funciones específicas que se explican a continuación:

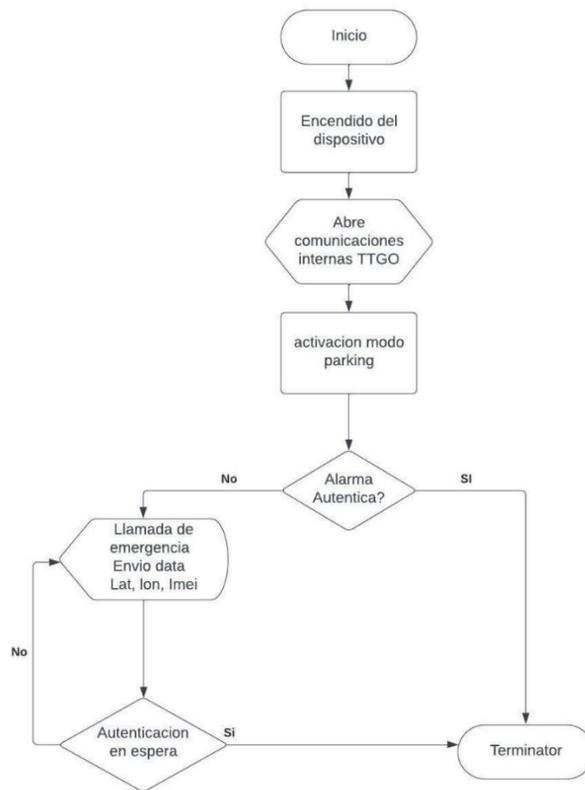


Figura 14 Algoritmo código t-sim7600

Elaborado por el investigador

## Encendido del dispositivo

Este bloque comprende la inicialización de la placa. La figura 14 muestra el algoritmo de programación de esta etapa, con la importación de librerías necesarias como “TinyGSMClient.h” que asiste en el acceso a funciones relacionadas a comunicaciones inalámbricas móviles como GPS o LTE o “HardwareSerial.h” que permite la declaración de varios canales de comunicación serial en la placa. A continuación, se definen variables

que ayudarán en el control de flujo del programa y funciones relacionadas a la comunicación.

```
1 #define TINY_GSM_MODEM_SIM7600
2 #define TINY_GSM_RX_BUFFER 1024 // Set RX buffer to 1Kb
3 #define IRQ 5
4 #define COLLECT_NUMBER 3
5
6 #include <HardwareSerial.h>
7 #include <DFRobot_ID809.h>
8 DFRobot_ID809 fingerprint;
9 HardwareSerial SerialAT(1);
10 HardwareSerial FPSerial(2);
11
12 define MODEM_TX          27
13 #define MODEM_RX          26
14 #define MODEM_PWRKEY     4
15 #define LED_PIN          12
16 #define MODEM_FLIGHT     25
17 #define MODEM_STATUS     34
18 #define SENSOR           23
19 #include <TinyGsmClient.h>
20 TinyGsm modem(SerialAT);
21 TinyGsmClient client(modem);
22 int pulsador = 0;           //almacena el estado del botón
23 int estado = 0;           //0=led apagado, 1=led encendido
24 int pulsadorAnt = 0;      //almacena el estado anterior del boton
```

*Figura 15 Variables del boceto definidas*

*Elaborado por el investigador*

## **Comunicaciones internas TTGO**

Tanto las funciones de módem como el sensor biométrico requieren de canales de comunicación serial para el intercambio de comandos e información, por lo que este bloque se refiere principalmente a la configuración de inicialización de los mismos. Tal como se puede ver en la figura 16, ambos canales seriales se configuran a una velocidad de 115200 baudios con pines específicos de cada canal UART.

```

void setup() {
  modem_on();
  pinMode(SENSOR, INPUT_PULLUP);
  pinMode(IRQ , INPUT);

  SerialAT.begin(115200, SERIAL_8N1, MODEM_RX, MODEM_TX);
  delay(1000);
  FPSerial.begin(115200, SERIAL_8N1, 13, 14);
  delay(1000);
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, HIGH);

  fingerprint.begin(FPSerial);

  while (fingerprint.isConnected() == false) {
    Serial.println("La comunicación con el dispositivo falló, verifique la conexión");
    delay(1000);
  }

  pinMode(MODEM_FLIGHT, OUTPUT);
  digitalWrite(MODEM_FLIGHT, HIGH);
}

```

*Figura 16 Comunicación Serial T-SIM7600*

*Elaborado por el investigador*

## Activación del modo Parking

Por defecto, el sistema acabará de iniciarse y pasará a un estado de espera mientras se usa normalmente la bicicleta. Una vez que el usuario desee estacionarse este fragmento de código deberá activarse con el toque del sensor biométrico. Inmediatamente la placa cambia a modo parking y si se detecta un movimiento anormal de la bicicleta se efectuará la llamada al siguiente bloque que detona con la función “comparar()”, como lo muestra la figura 17. Este estado se distingue por un titileo del sensor de huellas en color magenta.

```

void loop() {
  pulsador = digitalRead(IRQ); //lee si el botón está pulsado
  if ((pulsador == HIGH) && (pulsadorAnt == LOW)) { //si el boton es pulsado y antes no lo estaba
    estado = 1 - estado;
    delay(1000); //pausa
  }
  pulsadorAnt = pulsador; //actualiza el nuevo estado del boton
  if (estado == 1) { //si el estado es 1
    Serial.println("Modo parking"); //se enciende el led
    delay(2000);
    comparar();
    //gnss();
  }
  else { //si el estado es 0
    Serial.println("Alarma desactivada"); //se apaga el led
    delay(2000);
    fingerprint.ctrLEDD(/*LEDMode = */fingerprint.eFastBlink , /*LEDColor = */fingerprint.eLEDMagenta, /*blinkCount
  }
}

```

*Figura 17Control de encendido de alarma*

*Elaborado por el investigador*

## Autenticación y alarma

Este bloque abarca varias subfunciones. En caso de que la bicicleta experimente un cambio en su inclinación se espera que en el tiempo siguiente el usuario se identifique, de hacerlo, la placa sale del estado de parking y nuevamente vuelve a estar a la espera de su activación con el proceso mencionado en el paso anterior, como lo indica la figura 18, sin embargo si el usuario no ha desactivado la alarma comienza los procesa de alerta y ubicación.

```
void comparar() { // metodo de autenticacion

  fingerprint.ctrLED(/*LEMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */1);
  delay(2000);
  if (digitalRead(SENSOR) == LOW) { //detecta la manipulacion de la bicicleta
    uint8_t ret = 0;
    fingerprint.ctrLED(/*LEMode = */fingerprint.eBreathing, /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */2);
    Serial.println("Please press down your finger");
    while ( fingerprint.detectFinger() == 0) { // si no se detecta la huella una vez manipulada la
      //bicicleta se procede a llamar a un numero en especifico y enviar la ubicacion a la aplicacion
      Serial.println("No se detecto Huella");
      alerta();
      delay(4000);
      gsm();
      delay(2000);
      break;
    }
  }

  if ((fingerprint.collectionFingerprint(/*timeout=*/0)) != ERR_ID009) {
    Serial.println("Please release your fingers");
    fingerprint.ctrLED(/*LEMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDWhite, /*blinkCount = */1);

    if ( fingerprint.detectFinger() != 0) {
      ret = fingerprint.search();
      //Compare la huella digital capturada con una huella digital de identificación específica
      //Devuelve el ID de la huella digital (1-50) si tiene éxito, devuelve 0 cuando falla
      //
      Serial.print("Matching succeeds, ID=");
      Serial.println(ret);
      //Establezca el anillo LED de huella digital para que siempre esté ENCENDIDO en verde //
      fingerprint.ctrLED(/*LEMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDGreen, /*blinkCount = */3);
      Serial.println("Capturing succeeds");
      fin();
      delay(2000);
    } else {
      Serial.print("Matching succeeds, ID=");
      Serial.println(ret);
      //Set fingerprint LED ring to always ON in red//
      fingerprint.ctrLED(/*LEMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */3);
      Serial.println("Matching fails");
      alerta();
      gsm();
      delay(2000);
    }
  }
}
```

*Figura 18 Sistema de autenticación  
Elaborado por el investigador*

Si la bicicleta es alterada en su estado de reposo y no hay autenticación alguna, se empezará a enviar alertas llamando al método “alerta()”y registrar la actividad de ubicación desde ese mismo instante mediante el método “gnss()”. La parte de la alerta se realiza mediante llamadas al teléfono registrado del usuario mediante comandos AT, expuesto en la figura 19.

```
void alerta() {
    SerialAT.println("ATD+593998895642;");
    Serial.println("Llamando");
    delay(6000);
    SerialAT.println("AT+CVHU=0");
    delay(4000);
    Serial.println("Colgando");
    SerialAT.println("ATH");
    delay(4000);
}
```

*Figura 19 Método llamada de emergencia mediante comandos AT*

*Elaborado por el investigador*

En lo concerniente a la obtención de la ubicación, la función “gnss()” contiene a su vez subfunciones que asisten en la conectividad a la red móvil de la operadora, activación del módulo GPS, recibir los datos de localización, envío de la información y variables hacia la base de datos, y desactivación de las funciones de comunicación temporalmente para el ahorro de batería. La figura 20 muestra la obtención de los datos GPS y la figura 21 el armado del payload que se envía hacia el servidor web.

```
enableGPS();
String imei = modem.getIMEI();
float lat, lon;
while (1) {
    if (modem.getGPS(&lat, &lon)) {
        Serial.println("The location has been locked, the latitude and longitude are:");
        Serial.print("latitude:"); Serial.println(lat, 8);
        Serial.print("longititude:"); Serial.println(lon, 8);
        Serial.print("IMEI:"); Serial.println(imei);
        break;
    }
    digitalWrite(LED_PIN, !digitalRead(LED_PIN));
    delay(2000);
}
```

*Figura 20 Adquisición de datos a enviar latitud longitud e imei*

*Elaborado por el investigador*

```
String httpRequestData = "key=a@4K%3&latitude=" + String(lat, 8) + "&longitudo="
+ String(lon, 8) + "&imei=" + String(imei) + "" ;
client.print(String("POST ") + resource + " HTTP/1.1\r\n");
client.print(String("Host: ") + server + "\r\n");
client.println("Connection: close");
client.println("Content-Type: application/x-www-form-urlencoded");
client.print("Content-Length: ");
client.println(httpRequestData.length());
client.println();
client.println(httpRequestData);
timeout = millis();
while (client.connected() && millis() - timeout < 10000L) {
  while (client.available()) {
    char c = client.read();
    //Serial.print(c);
    timeout = millis();
  }
}
```

---

*Figura 21 Solicitud cliente mediante http para el envío de datos*

*Elaborado por el investigador*

No menos importante que los métodos expuestos para el envío de datos y manejo de sensores también se desarrolló métodos de activación de la placa de desarrollo “modoOn()”, método para activar y desactivar el GPS “enableGPS()”, “disableGPS”. Los cuales se encuentran expuestos en el ANEXO A, la activación y desactivación del GPS se lo realiza mediante comandos AT como se puede observar en la figura 22.

```

void enableGPS(void)//habilita gps{
  // Set SIM7000G GPIO4 LOW ,turn on GPS power
  // CMD:AT+SGPIO=0,4,1,1
  // Only in version 20200415 is there a function to control GPS power
  modem.sendAT("+SGPIO=0,4,1,1");
  if (modem.waitResponse(10000L) != 1) {
    DBG(" SGPIO=0,4,1,1 false ");
  }
  modem.enableGPS();
}

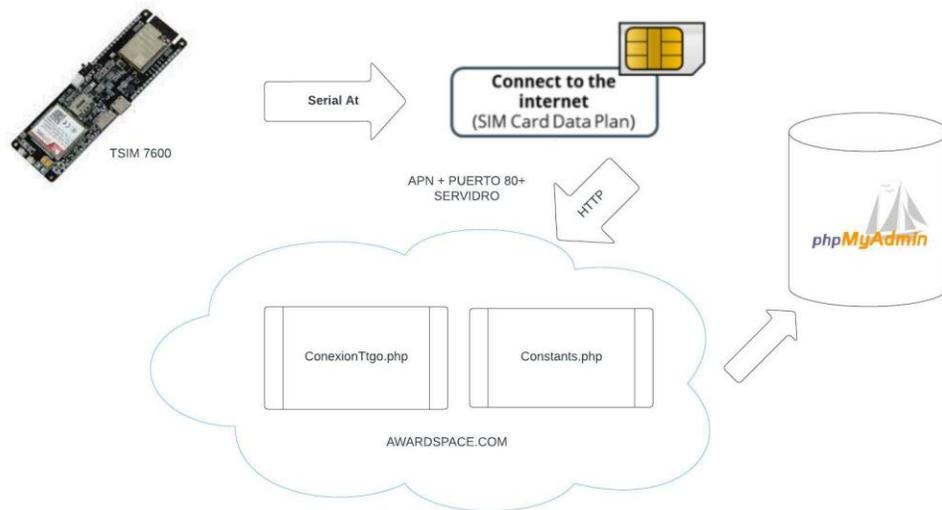
void disableGPS(void)//deshabilita gps{
  // Set SIM7000G GPIO4 LOW ,turn off GPS power
  // CMD:AT+SGPIO=0,4,1,0
  // Only in version 20200415 is there a function to control GPS power
  modem.sendAT("+SGPIO=0,4,1,0");
  if (modem.waitResponse(10000L) != 1) {
    DBG(" SGPIO=0,4,1,0 false ");
  }
  modem.disableGPS();
}

```

*Figura 22 Métodos del GPS on-off  
Elaborado por el investigador*

## **Etapa de Almacenamiento**

La integración de la información como se ilustra en la figura 23, es donde se procesa los datos correspondientes como a la ubicación del GPS y el IMEI del dispositivo , a su vez se realiza el control de login , registro de usuario y envió de ubicaciones hacia el dispositivo móvil, el proyecto requiere de una web Services.



*Figura 23 Estructura etapa de Almacenamiento  
Elaborado por el investigador*

## **Diseño de la etapa de almacenamiento**

Para desarrollar esta fase, es necesario el uso de un hosting donde se pueda utilizar tanto php para la conexión entre prototipo, servidor y aplicación como phpMyAdmin para el almacenamiento de datos dicho anteriormente.

### **Transmisión y conexión con el servidor**

Una vez enviado los datos al servidor se debe crear los archivos necesarios bajo el lenguaje Php, para ser almacenados en la base de datos phpMyAdmin, recordar que en el código de Arduino IDE se estableció los parámetros para la conexión LTE con la nube como se muestra en la figura 24, mediante la APN de Claro se conecta a internet de manera inalámbrica, se establece el directorio del hosting para establecer la transmisión y el puerto 80 que se refiere a Http.

```

const char apn[] = "internet.claro.com.ec";
const char gprsUser[] = "";
const char gprsPass[] = "";
const char server[] = "pruebaslaboratorio.atwebpages.com"; // servidor
const char resource[] = "/conexionTtgo.php"; //directorio
const int port = 80; //puerto
unsigned long timeout;|

```

*Figura 24 Direcciones del hosting y APN*

*Elaborado por el investigador*

De lado del servidor se crea un directorio donde se guardarán los archivos scripts.php que ejecutaran los comandos de conexión y envió de datos hacia las tablas de la base de datos, el script “conexionTtgo.php”, las credenciales (figura 25) serán las mismas tanto para el nodo como para la interfaz móvil.

```

<?php
define("DB_HOST", "pdb55.awardspace.net");
define("DB_USER", "4061622_bike");
define("DB_PASSWORD", "zdt.MP_1AS");
define("DB_DATABASE", "4061622_bike");
?>

```

*Figura 25 Credenciales para conexión placa y dispositivo*

*Elaborado por el investigador*

En la figura 26 se presenta la relación entre las tablas usuario, dispositivo y detalle dispositivo entonces, la tabla de usuarios va de uno a varios porque un usuario puede tener varios dispositivos, y un dispositivo puede tener varias marcaciones como se puede ver en la tbl\_detalle\_dispositivo. Las tablas de entidad relación se conectan mediante claves primarias. La clave primaria, ID del usuario, viaja hacia tabla dispositivo como foránea y la clave primaria de la tabla dispositivo. viaja como foránea hacia detalle dispositivo

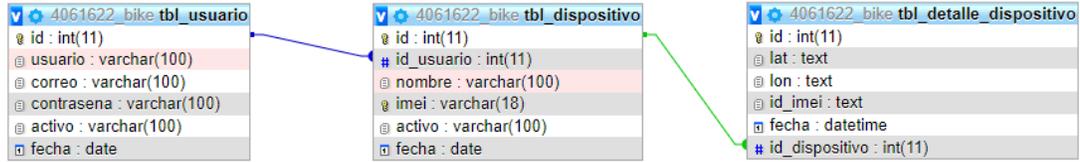


Figura 26 Tablas entidad relación  
Elaborado por el investigador

Continuando con el script “conexionTtgo.php”, luego de haber definido a que base se conectará se inicializa las variables a enviar y se procede a definir las variables figura 27 con las que se recibirá los datos enviados por la T-sim 7600.

```
include 'Constants.php';
// $key = "";
$longitude = "0";
$latitude = "0";
$imei = "0";
if (isset($_POST['key'])) $key = $_POST['key'];
if (isset($_POST['latitude'])) $latitude = $_POST['latitude'];
if (isset($_POST['longitude'])) $longitude = $_POST['longitude'];
if (isset($_POST['imei'])) $imei = $_POST['imei'];
```

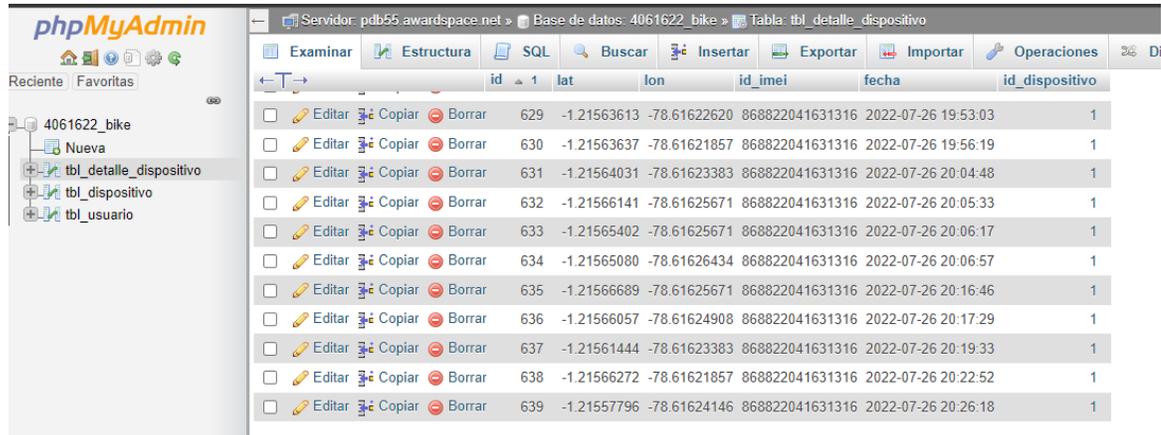
Figura 27 Definición variables a recibir en la base de datos phpMyAdmin  
Elaborado por el investigador

Una vez recibido las variables con los datos se procede a la conexión con la tabla “tbl\_detalle\_dispositivo” figura 28, donde se insertará los valores requeridos, se presenta de mejor manera el código completo en el anexo B8, recordar que en la cadena “httpRequestData” que envía el dispositivo se estableció una llave para poder comenzar la lectura de los enviados en el script del servidor.

```
$stmt = $conexion->prepare(" INSERT INTO tbl_detalle_dispositivo (id, lat, lon, id_imei, id_dispositivo).
VALUES (NULL, '$latitude', '$longitude', '$imei', (select id from tbl_dispositivo where imei='$imei') ); ");
// $stmt->bind_param("ss", $distancia, $date);
if ($stmt->execute()){
echo "Ejecutado con éxito";
}
```

Figura 28 Inserción datos en tablas phpMyAdmin  
Elaborado por el investigador

Debido a una conexión exitosa por la segunda capa de la arquitectura IoT, en la figura 29, se verifica la latitud como “lat”, longitud como “lon” e “id\_imei”. Se agregaron correctamente.



	id	lat	lon	id_imei	fecha	id_dispositivo
<input type="checkbox"/>	629	-1.21563613	-78.61622620	868822041631316	2022-07-26 19:53:03	1
<input type="checkbox"/>	630	-1.21563637	-78.61621857	868822041631316	2022-07-26 19:56:19	1
<input type="checkbox"/>	631	-1.21564031	-78.61623383	868822041631316	2022-07-26 20:04:48	1
<input type="checkbox"/>	632	-1.21566141	-78.61625671	868822041631316	2022-07-26 20:05:33	1
<input type="checkbox"/>	633	-1.21565402	-78.61625671	868822041631316	2022-07-26 20:06:17	1
<input type="checkbox"/>	634	-1.21565080	-78.61626434	868822041631316	2022-07-26 20:06:57	1
<input type="checkbox"/>	635	-1.21566689	-78.61625671	868822041631316	2022-07-26 20:16:46	1
<input type="checkbox"/>	636	-1.21566057	-78.61624908	868822041631316	2022-07-26 20:17:29	1
<input type="checkbox"/>	637	-1.21561444	-78.61623383	868822041631316	2022-07-26 20:19:33	1
<input type="checkbox"/>	638	-1.21566272	-78.61621857	868822041631316	2022-07-26 20:22:52	1
<input type="checkbox"/>	639	-1.21557796	-78.61624146	868822041631316	2022-07-26 20:26:18	1

*Figura 29 Datos t-sim en phpMyAdmin  
Elaborado por el investigador*

El “id\_dispositivo” sirve para activar el usuario creado en la app móvil con el prototipo, por seguridad solo el administrador a cargo de la base de datos podrá entrelazar a un dispositivo con un usuario, esto si en un futuro desean implementar para varias bicicletas, en vista que solo se tiene un dispositivo al momento sin embargo la plataforma queda con soporte.

## **Etapa de aplicación**

Por último, el resultado del sistema se presenta con la cuarta capa de la arquitectura IoT, servicio de aplicación donde intervienen los dispositivos conectados a la internet, dichos dispositivos se utilizan para la interpretación de toda la información en la nube, se puede presentar en plataformas web o móviles, para este proyecto interviene el uso de Google Apis, permitiendo usar los mapas en una aplicación móvil en Android studio IDE con lenguaje Kotlin.

### Funcionamiento tercera etapa

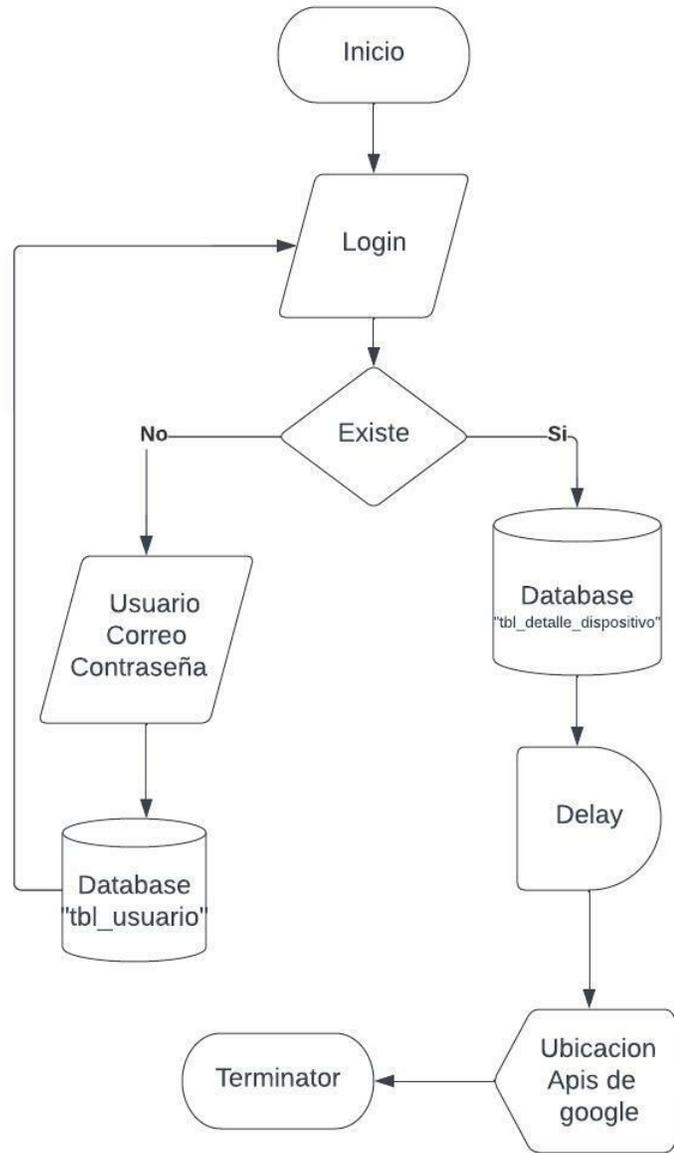


Figura 30 Algoritmo de la tercera etapa  
Elaborado por el investigador

En la primera activity “mainactivity.kt” de la interfaz, se estableció dos text input necesario para realizar el login (figura 31-a), si se encuentra el usuario registrado se abrirá el segundo activity “home.kt”(figura 31-c)



Figura 31 Interfaz móvil

*Elaborado por el investigador*

Si el usuario no se encuentra en la base de datos figura 31-b, no podrá ingresar y tendrá que dar clic en registrar , guardar los datos a enviar a la nube como “usuario”, “correo”, “contraseña” , estos datos llegan al script “conexionloginUser.php” en el anexo B4 muestra la codificación desarrollada para almacenar en la tabla “tbl\_usuario” figura 32 y posteriormente enviar alertas hacia la aplicación de haber sido exitoso o no.

	id	usuario	correo	contrasena	activo	fecha
<input type="checkbox"/> Editar Copiar Borrar	1	diego	Diego@gmail.com	2	1	2022-07-10
<input type="checkbox"/> Editar Copiar Borrar	3	gabriela	Gabriela@hotmail.com	gaby32	1	0000-00-00
<input type="checkbox"/> Editar Copiar Borrar	8	Carlos	carlos6@hotmail.com	567890	1	0000-00-00
<input type="checkbox"/> Editar Copiar Borrar	9	Diego	dekoopazmino@gmail.com	567890	0	0000-00-00
<input type="checkbox"/> Editar Copiar Borrar	14	8	9	890	0	0000-00-00
<input type="checkbox"/> Editar Copiar Borrar	15	7	7	890	0	0000-00-00

Figura 32 Usuarios registrados

Elaborado por el investigador

Un detalle que se debe mencionar, en la activity home se encuentra un button en la parte superior izquierda la cual me muestra el nombre de la bicicleta que está mostrando los datos de ubicación, una vez que se ingresó en activity home las ubicaciones se limpiarían y actualizaran a las ultimas 50 posiciones(figura 33) ejecutadas en el script “DbOperation.php”

```

public function getData($iduser, $iddevice){
    $sql = " SELECT distinct td.id_dispositivo, td.lat, td.lon, td.fecha ".
           " FROM tbl_detalle_dispositivo td ".
           " inner join tbl_dispositivo d on td.id_dispositivo=d.id ".
           " inner join tbl_usuario u on d.id_usuario=u.id ".
           " where u.id = ? ".
           " and id_dispositivo = ? ".
           " order by td.id desc limit 50 ";

    $stmt = $this->con->prepare($sql);
    $stmt->bind_param("ii",$iduser, $iddevice);
    $stmt->execute();
    $stmt->bind_result($Id, $Lat, $Lon, $Fecha);
    $datos = array();

    while($stmt->fetch()){
        $temp = array();
        $temp['id_dispositivo'] = $Id;
        $temp['lat'] = $Lat;
        $temp['lon'] = $Lon;
        $temp['fecha'] = $Fecha;
        array_push($datos, $temp);
    }
    return $datos;
}

```

Figura 33 Obtención ubicaciones cada 30s

Desarrollado por el investigador

La interfaz móvil trabaja conjuntamente con los scripts desarrollados y guardados en la web Services, como se puede observar en la figura 34 se solicita cada 30 segundos los datos.

```
override fun onMapReady(googleMap: GoogleMap) {
    map = googleMap

    //Obtener marcaciones del dispositivo
    //Actualización cada 30seg = 30000 malsegs
    val timer = Timer()

    timer.scheduleAtFixedRate( object : TimerTask() {
        override fun run() {
            getDatos(map, Seguridad.IdUsuario, Seguridad.IdDispositivo)

            if (Seguridad.IdDispositivo=="0")
            {
                timer.cancel()
            }
        }
    }, delay: 0, period: 30000)
}
```

*Figura 34 Actualización cada 30s de ubicaciones en la interfaz móvil*

*Elaborado por el investigador*

Para realizar las solicitudes de loguear, registrar y obtener datos de dispositivo y ubicación se lo realiza mediante la librería volley en Kotlin, como se puede observar en la figura 36, esta librería se ha utilizado con la misma lógica durante diferentes actividades por lo que sería muy reiterativo explicar el mismo proceso con diferentes solicitudes por ello se analizará la lógica del algoritmo para solicitar las ubicaciones ya que es lo más importante del proyecto.

```

private fun getDatos(google:GoogleMap, idUsuario:String, idDispositivo:String) {
    val url = "http://controlbike.atwebpages.com/getDatos.php"

    val stringRequest = object : StringRequest(
        Request.Method.POST, url,
        Response.Listener<String> { response ->
            try {
                val obj = JSONObject(response)
                val vl_correcto = obj.getBoolean( name: "correcto");

                if (vl_correcto) {
                    Toast.makeText( context: this, text: "Actualizando rutas...", Toast.LENGTH_SHORT).show()
                    google.clear()
                    //SERIALIZADO DE CAMPOS
                    var respuesta = Respuesta(obj.toString())
                    //GRAFICAR EN LA MARCA
                    if (respuesta.correcto.toBoolean())
                    {
                        respuesta.message?.forEach { msg ->
                            createMarcker(msg.lat, msg.lon, msg.fecha)
                        }
                    }
                }
            }
        }
    )
}

```

*Figura 35 función getDatos con RequestQueue*

*Elaborado por el investigador*

En la figura 35 primero se crea la función getDatos, luego se procede a definir la url del directorio en el hosting utilizado ,donde se solicitará los datos , para enviar una solicitud , simplemente se establece un RequestQueue como se muestra en la última línea de la figura 36 mediante el parámetro response se recibirá el objeto en formato Json. Dependiendo del “\$idUser”, “iddevice”, ilustrado en la figura 36, que se encuentren enlazados entre las tablas “tbl\_usuario” y la tabla “tbl\_dispositivo” , obtendrá los datos desde la tabla “tbl\_detalle\_dispositivo para devolver los datos solicitados por el RequestQueue, se muestra el código completo en el anexo C5.

```

        override fun getParams(): Map<String, String> {
            val params = HashMap<String, String>()
            params["idUser"] = idUsuario
            params["idDevice"] = idDispositivo
            return params
        }
    }
    //adding request to queue
    VolleySingleton.instance?.addToRequestQueue(stringRequest);

```

*Figura 36 Parámetros necesario para obtener los datos dependiendo el usuario*

*Elaborado por el investigador*

Se obtiene la respuesta en un JSONArray como se ilustra en la figura 37 y luego se transforma cada JSONObject de la class “Marcacion” en los parámetros.

```

class Respuesta(json: String) : JSONObject(json) {
    val correcto: String? = this.optString(name: "correcto")
    val message = this.optJSONArray(name: "message")
        ?.let { 0.until(it.length()).map { i -> it.optJSONObject(i) } } // returns an array of JSONObject
        ?.map { Marcacion(it.toString()) } // transforms each JSONObject of the array into Foo
}

class Marcacion(json: String) : JSONObject(json) {
    val id = this.optInt(name: "id")
    val lat: Double = this.optDouble(name: "lat")
    val lon: Double = this.optDouble(name: "lon")
    val fecha: String = this.optString(name: "fecha")
}

```

*Figura 37 Respuesta y conversion de datos del formato json*

*Elaborado por el investigador*

Con los datos listos se procede a graficar con un Marker dentro del mapa de Google Maps que se ha configurado previamente en la interfaz, esto se ilustra en la figura 38, se añade una animación para ampliar la ubicación exacta.

```

private fun createMarker(latitud: Double, longitud: Double, texto:String ){
    val coordinate =LatLng(latitud, longitud)
    val marker =MarkerOptions().position(coordinate).title(texto)
    map.addMarker(marker)
    map.animateCamera(CameraUpdateFactory.newLatLngZoom(coordinate, zoom: 18f), durationMs: 4000, callback: null)
}

```

*Figura 38 Diseño de Marker para ubicaciones en Google Maps Apis  
Elaborado por el investigador*

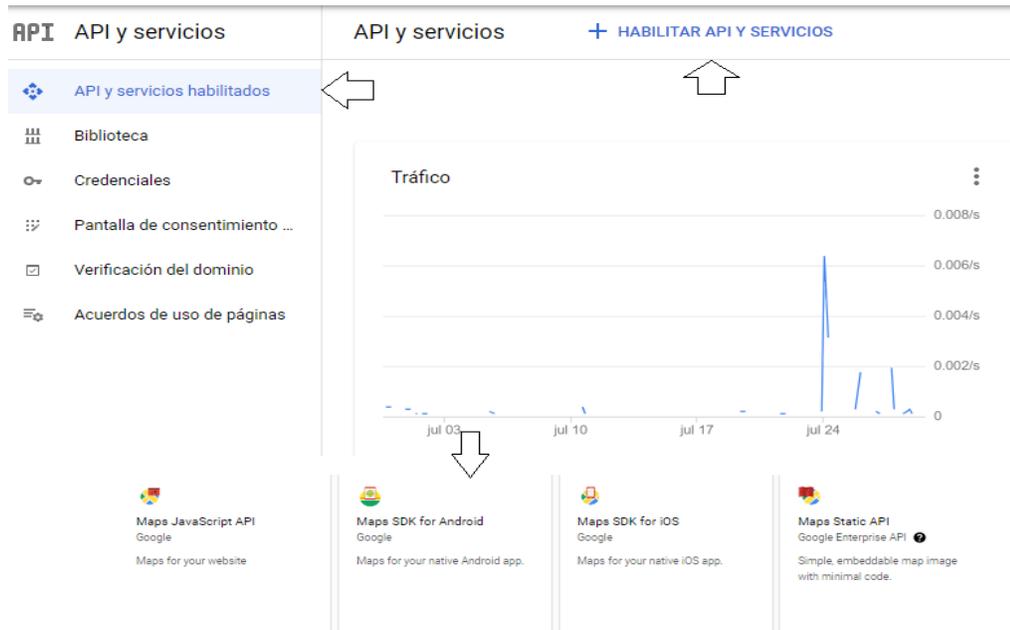
## Diseño de mapas en Android Studio con Kotlin

Se necesita un api de Google para la utilización de sus mapas en la aplicación ControlBike, por ello se accede mediante un navegador a la página web “ a console.google.com “donde se crea un nuevo proyecto como se ilustra en la figura 39.

The screenshot shows the Google Cloud 'Project new' interface. At the top, there's a blue header with the Google Cloud logo and a search bar. Below the header, the text 'Proyecto nuevo' is displayed. A warning box indicates that there are 24 projects remaining in the quota, with a link to 'Más información' and a 'MANAGE QUOTAS' button. The 'Nombre del proyecto' field is filled with 'My Project 86046'. Below it, the project ID 'causal-lattice-357816' is shown with an 'EDITAR' link. The 'Ubicación' dropdown is set to 'Sin organización', with an 'EXPLORAR' button. At the bottom, there are 'CREAR' and 'CANCELAR' buttons.

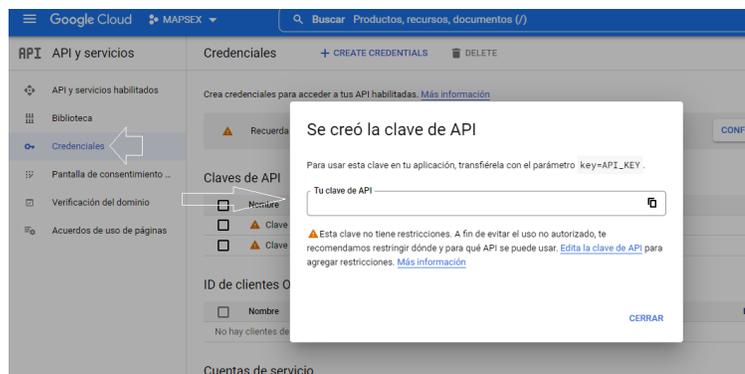
*Figura 39 Creacion proyecto APIs GOOGLE  
Elaborado por el investigador*

Una vez creado el proyecto se activa el API a utilizar como se visualiza en la figura 40, se selecciona API y servicios y “MAPS SDK FOR ANDROID”, hay varias Apis que Google ofrece sin embargo la mayor parte son de pago.



*Figura 40 Selección API a utilizar  
Elaborado por el investigador*

Ahora que se creó la API seleccionado , se procede a generar una clave ingresando en credenciales , luego en crear credenciales como se puede observar en la figura 41, se genera una clave la cual esta oculta por seguridad en la figura 41 esta clave API es estrictamente para la aplicación donde se desea trabajar con los mapas de Google.



*Figura 41Clave API Elaborado por el investigador*

Una vez generado la API, se debe crear un archivo “google\_maps\_api.xml” dentro de la carpeta values, en el entorno de Android Studio, donde se guarda la clave con el nombre “google\_maps\_key”, también se define sus propiedades porque no debe ser modificada nunca la API key, las propiedades son : templateMergeStrategy="preserve" translatable="false", va a quedar de la siguiente manera, esta oculto la clave por seguridad. Por último añadir la librería “com.google.android.gms:play-services-maps:18.0.2” junto con todas las librerías a utilizar como la ya mencionada volley en dependencias del archivo build.gradle revisar anexo C10 de ser necesario.

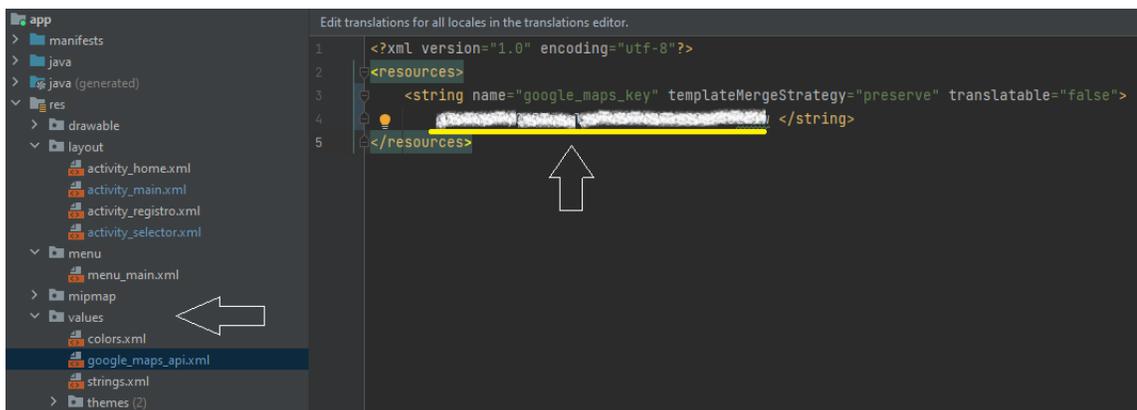


Figura 42 Clave api dentro de la aplicación ControlBike.

Elaborado por el investigador

Por último, se visualizará un activity como se ilustra en la figura 43, cabe recalcar que en la parte inferior derecha se encuentra dos botones los cuales cuando se da clic en cualquier marca de posición enviada de la bicicleta, esta se remite automáticamente a Google Maps la aplicación oficial, donde se pueden graficar rutas.

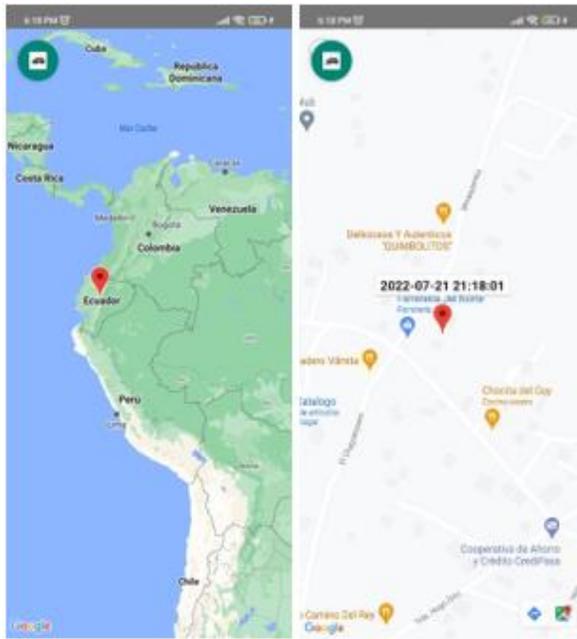


Figura 43 Marker de posición en mapas

Elaborado por el investigador

El sistema está diseñado para ejecutarse desde Android 5.0 y es compatible con el 98.6% de dispositivos con este sistema operativo, como se ilustra en la figura 44, oficial de su IDE ( Android Studio ).

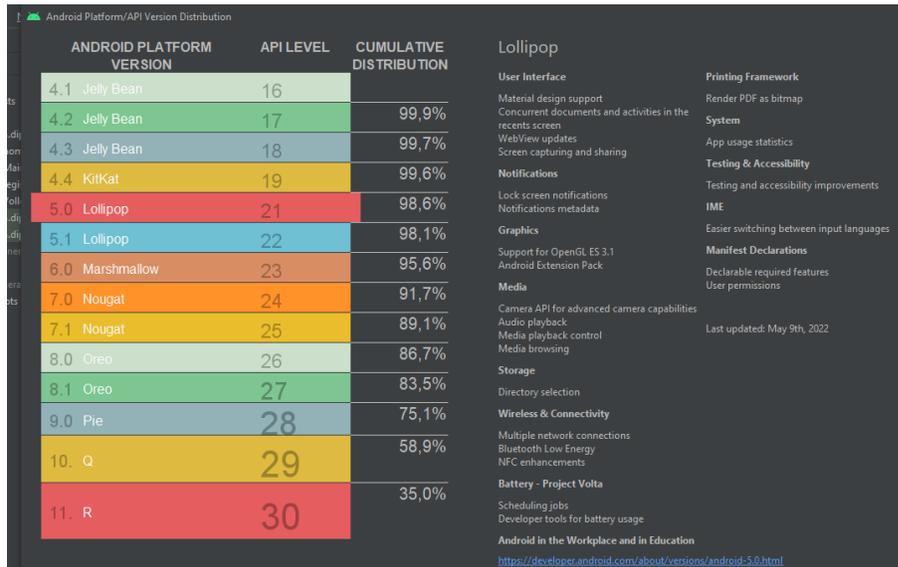


Figura 44 Compatibilidad con dispositivos Android

Elaborado por el investigador

## Hardware del prototipo

### Selección de la plataforma IoT

Las placas de desarrollo revolucionan la tecnología y la forma en que se interactúa con el mundo, estos dispositivos se caracterizan por sus capacidades de velocidad, transmisión y aplicaciones destinados al IoT.

Una vez seleccionado el sistema de comunicación a utilizar para el prototipo, se realiza una búsqueda de dispositivos para seleccionar la plataforma IoT, teniendo en cuenta su disponibilidad en el mercado, precio y características generales basado en los requerimientos del sistema. Para llegar a la elección se realiza un análisis comparativo en la tabla 6.

Placas de Desarrollo IoT			
Placa	Jetson Nano 4GB Sub Developer Kit	Raspberry pi 4 MODEL B	LILYGO T- SIM7600G-H
			
Procesador	ARM® Cortex® -A57 MPCore (Quad- Core) Processor	Quad core 64-bit ARM-Cortex A72 running at 1.5GHz	ESP32

Memoria RAM	4 GB	1 GB, 2 GB, 4 GB u 8 GB (compartidos con la GPU)	16 Mb(spram)
Fuente de Alimentación	5 V	5 V	3.4 -4.2 V
Sistemas operativos soportados	Android	GNU/Linux: Raspbian	Microsoft Windows 7/8/10, Linux, Android
Comunicaciones móviles	N/A	N/A	LTE-FDD,LTE-TDD,WCDMA,GSM Contiene modulo Sim7600G
Comandos AT	N/A	N/A	Disponible
Comunicaciones	USB, PCI Express, DSI, HDMI, DP, CSI/ VI,	Puerto RJ-45 10/100/1000Mbps vía hub USB 3.0, Wireless LAN: 802.11 b/g/n/ac Bluetooth 5.0/BLE	WI-FI: 802.11 b/g/n Bluetooth: v4.2 BR / EDR y BLE
GNSS	N/A	N/A	BeiDou, GPS Glonass Contiene una antena física (Neo 6)
Protocolos	SD y eMMC	IPv4 , IPv6, MQTT	TCP/IP/IPV4/IPV6/DNS/MultiPDP/FTP/FTPS/HTTP/HTTPS
SIM Card	N/A	N/A	1.8V / 3.0 V
Android RIL	Android 5.0	Android 5.0	Android 5.0/6.0/7.0/8.0/9.0
Dimensiones	16.31 x 10.79 x 4.19 cm	85mm x 53mm	30.0 * 30.0 * 2.9 mm

Consumo corriente	NA	NA	20mA
Precio	US\$298.99	US\$61.88	US\$65.99

*Tabla 6 Comparativa placas de desarrollo IoT*

*Elaborado por el investigador*

Con los datos técnicos obtenidos en la tabla 6, se puede determinar que Jetson y Raspberry tienen capacidades de muy alta velocidad en sus microcontroladores, sin embargo, no cumplen con los requisitos técnicos para comunicarse con el servicio web, debido a que este proyecto tiene por objetivo implementar un sistema inteligente de seguridad en bicicletas, y las comunicaciones inalámbricas que manejan dichas placas como wifi, bluetooth; no son la mejor opción, por ser de corto alcance. Por otro lado, el costo del proyecto también aumentará debido a que se debería colocar módulos LTE para poder conectar de manera inalámbrica y abarcar mayor cobertura, y también un módulo para la antena GPS, lo cual implica mayores costos, el prototipo tendría dimensiones que no son soportables en un cuadro de bicicleta, se debe disminuir lo más posible la estética final del prototipo, por lo que se ha seleccionado como mejor opción para implementar el proyecto con la placa de desarrollo de IoT (LILYGO TSIM7600GH).

### **Descripción de la placa de desarrollo IoT**

TSIM7600 es un módulo de red inalámbrica ilustrada en la figura 45 con GPS WIFI/Bluetooth/4G LTE CAT4, integra un soporte de batería 18650 con una capacidad de 15000 mAh a 3.7v y agrega una interfaz de carga solar. Para facilitar el uso de diferentes escenarios de aplicación, se reservan dos interfaces tipo c en la placa de desarrollo. Una se utiliza para descargar el programa al chip de control principal. Y el otro es la interfaz de tarjeta de red industrial USB dongle. Puede conectarse a una computadora a través de un cable USB y configurarse como una tarjeta de red inalámbrica.

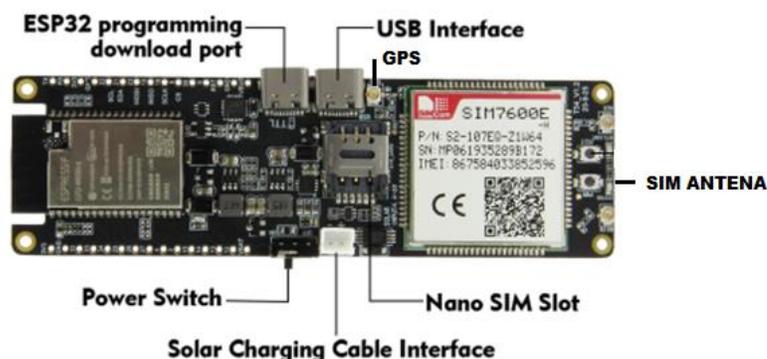


Figura 45 Placa de desarrollo (IOT) T-SIM 7600 G-H [36]

## Selección de sensores

### Sensor Biométrico

La verificación del dueño es esencial en el proyecto para monitorear el estado de la bicicleta ya que, sin este, solo se podría utilizar seguridad mecánica sin alertas telefónicas ni ubicación. Para ello se presenta un lector de huellas ubicado en el timón, que pueden ser instalados y configurados en la placa de desarrollo montada en el cuadro de la bicicleta.

Los sensores seleccionados para la verificación del usuario son entre el sensor óptico AS608 y el sensor capacitivo DFRobot SEN0348, dos sensores útiles para la comparación de huellas dactilares.

Parámetros técnicos	AS608 	DFRobot SEN0348 
Tipo	Óptico	Capacitivo

Comunicación	Serial/UART TTL	Serial/UART TTL
Tiempo de verificación	<1ms	300~400ms
Voltaje de operación	3.6 a 6.0 VDC	3.3V
Consumo de funcionamiento	<120mA	<60 mA
Almacenamiento	162 huellas dactilares	80 huellas dactilares
Temperatura de trabajo	-20°C a 50°C	40-60 °C
Dimensiones	(23.3 * 20.3 * 48.1) mm	8.0mm x 8.0mm
Precio	18\$	30\$

*Tabla 7 Comparación de sensores biométricos*

*Elaborado por el investigador*

Después de realizar la tabla 7 con las características de cada sensor se analiza el consumo de corriente y el tipo de sensor, por una parte, el sensor óptico tiene mayor tiempo de vida útil ya que es resistente a arañazos y la placa de vidrio es inquebrantable, mientras que la superficie del sensor óptico sufre de un desgaste notorio, sin embargo el primer sensor es vulnerable en seguridad pueden ser engañados por impresiones latentes, cuando la luz atraviese la superficie con una huella latente puede ser falsificada, y el capacitivo podría ser inseguro con dedos artificiales blandos lo cual va ser más difícil para superar su seguridad. En cuanto al consumo de corriente el sensor capacitivo consume un 50% menos que el sensor óptico y sus dimensiones son aceptables para ser implementadas en el timón de la bicicleta. Por ello se ha decidido utilizar el sensor capacitivo de la marca DFRobot.

### **Sensor de inclinación**

En cuanto a la alarma para un sistema de seguridad, que sería una sirena alertando al dueño de algún inmueble que quiera cuidar, este proyecto de investigación busca una solución IoT para la inseguridad, por ello se realiza una comparación de dos sensores silenciosos en la tabla 8, se analizan sensores de bajo costo y dimensiones compactas.

Características	MPU 6050	SW-520D
		
Voltaje	3.3V a 5V	5 – 12V
Consumo Corriente	50uA	5 mA
Temperatura de funcionamiento máxima	-40°C ~ 85°C (TA)	70 °C
Comunicación	I2c	Digital
Rango	Giroscopio: +/-250, +/- 500, +/-1000 y +/- 2000° /s(dps) Acelerómetro: +/-2 , +/-4 , +/-8 y +/-16g	+/- 90 °

*Tabla 8 Comparación sensores inclinación*

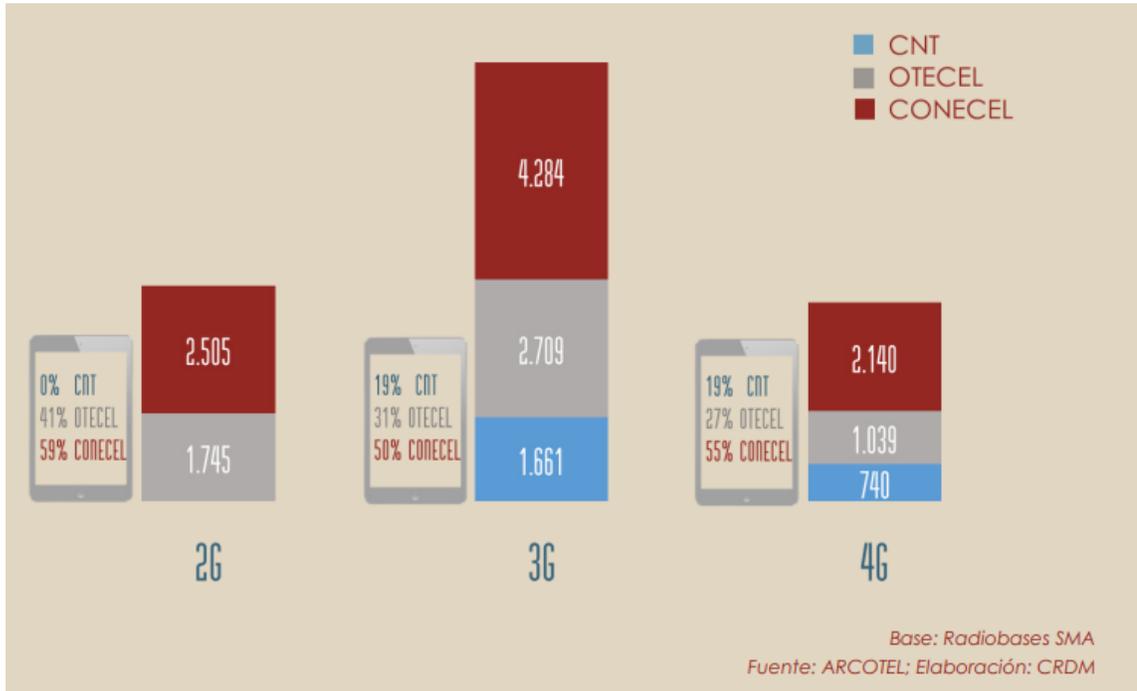
*Elaborado por el investigador*

Se ha seleccionado el sensor de inclinación SW-520D por su precio, optimización al momento de programar debido a que el MPU 6050 utiliza comunicación I2C o SPI , reduciendo la cantidad de pines específicos de la placa a utilizar y generando errores al momento de obtener resultados de un IMU(Unidad de Medición de Inercia), y por otra parte es irrelevante obtener el grado de inclinación de la bicicleta, mientras que el sensor de SW520D tipo interruptor genera una señal digital si su inclinación supera un umbral, esto simplifica tiempos dentro del código, y sirve como alerta rápida para el momento de enviar una alerta.

### **Selección de la operadora móvil**

El prototipo debe contar con una SIM y un plan de datos, esto depende del cliente debido a que las operadoras del país ofrecen diferentes paquetes desde diarios hasta anuales, esto

presenta una gran ventaja para el dueño ya que el dispositivo funciona con cualquier SIM, sin embargo se ha elegido para el momento de pruebas la operadora de claro debido a la gran cantidad de radio bases que tiene instaladas en el país como se muestra en la figura 46.



*Figura 46 Radio bases SMA  
Elaborado por ARCOTEL*

Según el boletín del IV trimestre del 2018 entregado por el ARCOTEL, CONECEL (Claro), tiene el 50% de las radios bases en el país en 2G/3G y 4G, lo cual demuestra su cobertura en el país.

## Esquema de conexión del prototipo

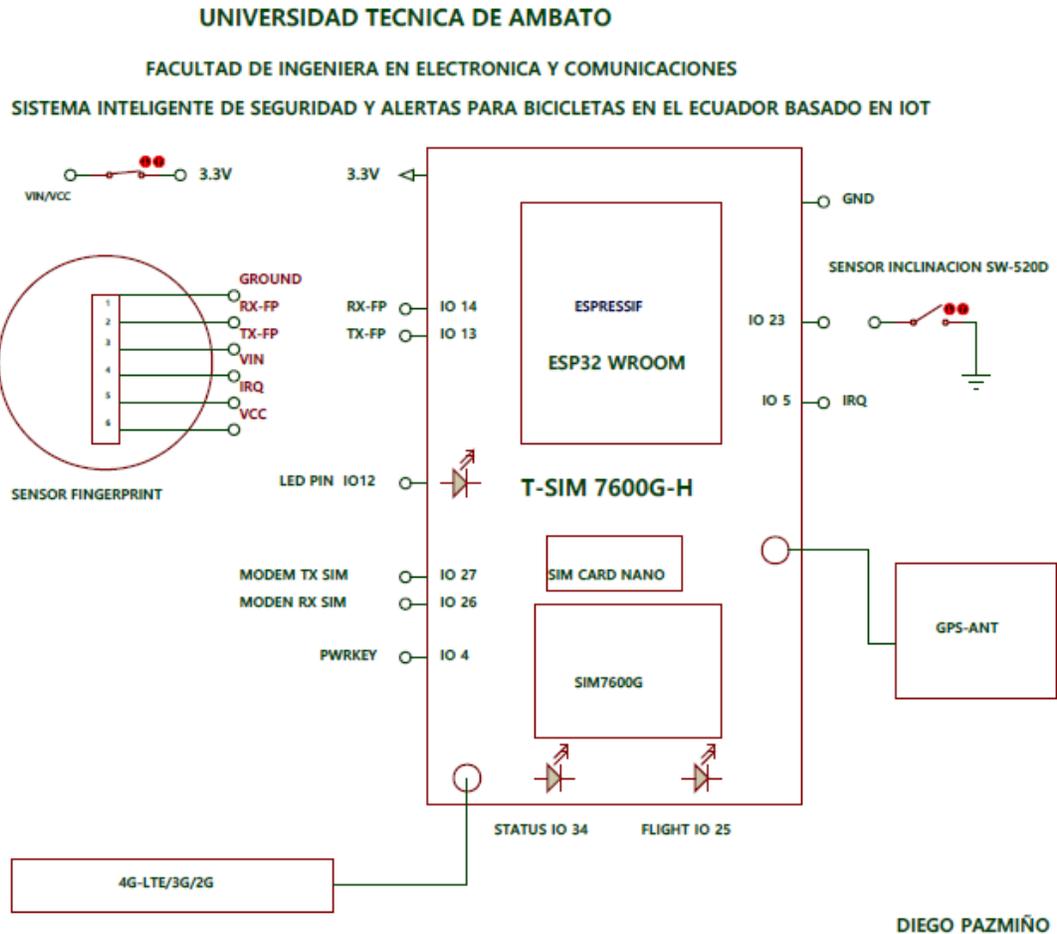


Figura 47 Diagrama de conexión del sistema IoT

Elaborado por el investigador

En el esquema de la figura 47 se ilustra la conexión de la placa T-SIM76000G con los sensores y las antenas, la placa es poco conocida por lo que no tiene librerías dentro del software de proteus para poder diseñarlo de mejor manera, se lo representa mediante bloques, se ha reducido lo más posible los circuitos para poder montar en el cuadro de la bicicleta el sistema teniendo como resultado en la siguiente figura 48.



*Figura 48 Montaje del sistema en la bicicleta Marin  
Elaborado por el investigador*

Los sensores se montaron en carcasas diseñadas en el Anexo D1, el sensor de inclinación y la placa junto con una batería de litio se encuentra dentro de la ilustración 1 del anexo mencionado, y está montado en el cuadro de la bicicleta , en cuanto al sensor biométrico se encuentra sobre el manubrio con el diseño de la ilustración 3 del mismo anexo.

### **Software del sistema**

#### **Selección Hosting**

Características	<p>Hostinger</p> 	<p>Awardspace</p> 	<p>Infinity Free</p> 
-----------------	--	--	--

Espacio almacenamiento	100 Gb	Ilimitado	Ilimitado
Ancho de banda	Ilimitada	Ilimitado	250Gb
Asistencia Técnica	24/7/365	24/7	24/7
Bases de datos	Ilimitada	2	400
Php	7.4	8.1.8	8.0
MySQL	Si	5.7	5.7
Sitios Webs	Ilimitada	Ilimitado	N/A
Correos	Ilimitada	1000	100
Cuentas Ftp	Ilimitada	Ilimitada	100
Seguridad SSL	si	si	si
Dominios	8.99\$ cada dominio	2 dominios 5 Subdominios	20
Precio	6.99 \$c/m (plan popular)	\$2.99 anual (plan básico)	3.99\$ c/m ( plan super premium)

*Tabla 9 Comparación Hosting*

*Elaborado por el investigador*

El hosting elegido para el proyecto es Awardspace, su precio es asequible en comparación con otro hosting y solo se necesita una base de datos para alojar la información, además tiene la versión de Php más actualizada entre los sitios de alojamiento.

### **Pruebas de funcionamiento**

Antes de las pruebas en campo se realizó un testeó de las dos antenas a utilizar para evaluar si las bandas establecidas por el fabricante eran globales en el Ecuador y sería exitosa, debido a que diferentes centros comerciales electrónicos del país, durante la investigación dieron a conocer que placas de desarrollo IoT para 3g o 4g de procedencia China, no funcionaban y por esta razón no importaban dichas placas al país, debido a estas circunstancias se presentó los análisis previos en tabla comparativa 6 de un dispositivo

que abarque la mayor cantidad de tecnologías móviles posibles para poder comprobarlas una vez importada al país.



*Figura 49 Comprobacion de la placa IoT con la PC  
Elaborado por el investigador*

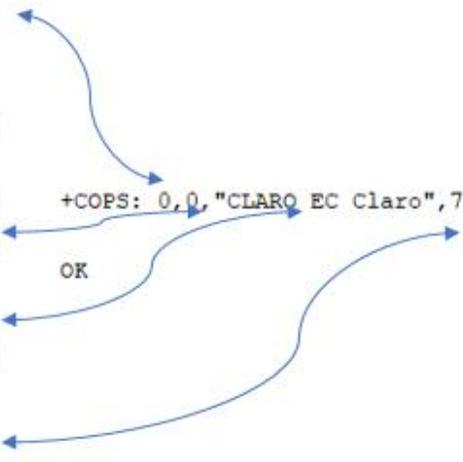
### **Testeo de funcionamiento Red Móvil**

La comprobación del funcionamiento de la conexión a la red LTE / 3G o 2G se la realiza con la comunicación uart del dispositivo conectada al pc como se puede visualizar en la figura 49, luego se procede a utilizar los códigos AT para recibir los datos de la operadora:

AT+COPS? : El comando de prueba devuelve una lista de 4 valores, cada uno de los cuales representa a un operador presente en la red (modo, formato, operadora y el acceso a la tecnología) como se puede ver en la figura siguiente:

**Defined values**

<b>&lt;mode&gt;</b>
0 - automatic
1 - manual
2 - force deregister
3 - set only <format>
4 - manual/automatic
5 - manual, but do not modify the network selection mode (e.g. GSM, WCDMA) after module resets.
<small>NOTE: if &lt;mode&gt; is set to 1, 4, 5 in write command, the &lt;oper&gt; is needed.</small>
<b>&lt;format&gt;</b>
0 - long format alphanumeric <oper>
1 - short format alphanumeric <oper>
2 - numeric <oper>
<b>&lt;oper&gt;</b>
string type, <format> indicates if the format is alphanumeric or numeric.
<b>&lt;stat&gt;</b>
0 - unknown
1 - available
2 - current
3 - forbidden
<b>&lt;AcT&gt;</b>
Access technology selected
0 - GSM
1 - GSM Compact
2 - UTRAN
7 - EUTRAN
8 - CDMA/HDR



*Figura 50 Comando AT +COPS  
Elaborado por el investigador*

AT+CNMP=? : Este comando se utiliza para seleccionar o establecer el estado de la preferencia de modo de la red móvil a utilizar, se comprobó mediante el monitor serial que las tecnologías 2G ,3G y 4G funcionan correctamente

	OK
	+CNMP: 2
	OK
	OK
	+CNMP: 13
	OK
	OK
	+CNMP: 14
	OK
	OK
	+CNMP: 38
	OK

2- Automático  
13-GSM Only  
14- WCDMA Only  
38-LTE Only

*Figura 51 Comando AT+CNMP. Elaborado por el investigador*

AT+CNAOP? : Este comando se utiliza para restablecer el estado de preferencia de orden de adquisiciones de la red, indica en orden jerárquico el modo de selección según la operada lo establezca para sus abonados.

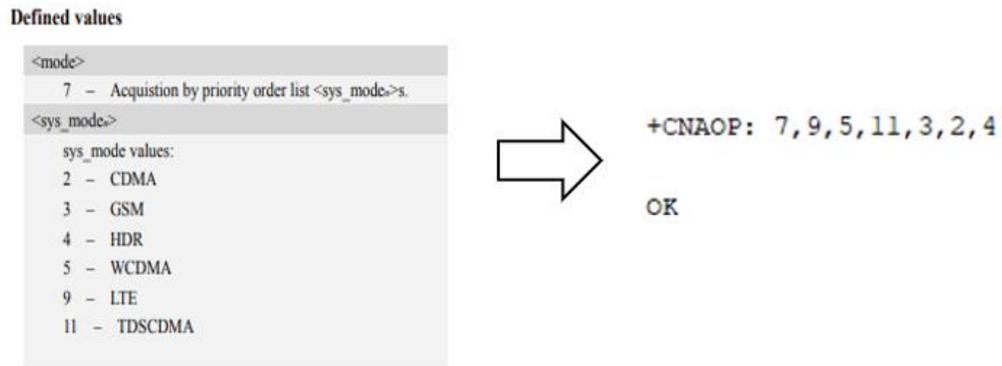


Figura 52 Comando AT+CNAOP. Elaborado por el investigador

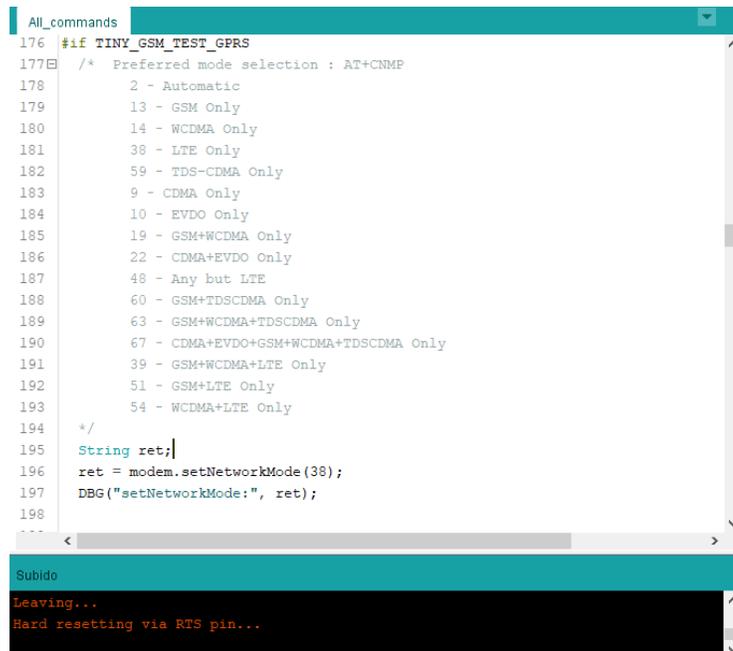
En la figura 52 se ilustra del lado derecho la respuesta en el monitor serial se ha recortado la imagen para poder visualizar e interpretar de mejor manera como selecciona la tarjeta la mejor cobertura para enviar datos, teniendo así el primer valor 7 que indica a la tarjeta establecer la conexión para trabajar , y luego imprime el orden siguiente :LTE , WCDMA, TDSCDMA,GSM,CDMA Y HDR



Figura 53 Cobertura red 4G de Claro. Elaborado por el investigador

Analizando la distribución de red móvil en la figura 53 se visualiza que Ecuador, tiene una cobertura que varía entre 3G y 4G, por este motivo se decide no establecer un modo de red específico en el código del anexo A.

Otra forma en la que se comprobó el estado de la red, fue utilizando el código del fabricante para el testeo, donde se obtuvo las siguientes respuestas: el modo de la red “setNetworkMode” donde devuelve 1 confirmando que el dispositivo se ha conectado a la red (figura 55), en el caso de la figura 54 se configuró para tener conexión solamente con LTE, en la figura 55 muestra el 99% la calidad de la señal, esto va a depender de la operadora contratada y la zona donde se encuentre dentro del país.



```
All_commands
176 #if TINY_GSM_TEST_GPRS
177 /* Preferred mode selection : AT+CNMP
178     2 - Automatic
179     13 - GSM Only
180     14 - WCDMA Only
181     38 - LTE Only
182     59 - TDS-CDMA Only
183     9 - CDMA Only
184     10 - EVDO Only
185     19 - GSM+WCDMA Only
186     22 - CDMA+EVDO Only
187     48 - Any but LTE
188     60 - GSM+TDSCDMA Only
189     63 - GSM+WCDMA+TDSCDMA Only
190     67 - CDMA+EVDO+GSM+WCDMA+TDSCDMA Only
191     39 - GSM+WCDMA+LTE Only
192     51 - GSM+LTE Only
193     54 - WCDMA+LTE Only
194 */
195 String ret;|
196 ret = modem.setNetworkMode(38);
197 DBG("setNetworkMode:", ret);
198
Subido
Leaving...
Hard resetting via RTS pin...
```

*Figura 54 Directiva preprocesada de la red móvil. Elaborado por el investigador*

En el monitor serial se obtiene los datos de la placa que se está utilizando, para abrir la comunicación con claro se estableció previamente la APN “internet.claro.com.ec” con ella se puede tener conexión al hosting, por último devuelve el imei, ip y la operada como se ilustra en la figura 55.

```

load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4
SDCard MOUNT FAIL
[1771] Initializing modem...
[1771] ### TinyGSM Version: 0.11.5
[1771] ### TinyGSM Compiled Module: TinyGsmClientSIM7600
[1820] ### Modem: SIMCOM SIM7600G-H
[1820] ### Modem: SIMCOM SIM7600G-H
[1874] setNetworkMode: 1
[1889] GNSM Mode: 1
[2908] ### Modem: SIMCOM SIM7600G-H
[2908] Modem Name: SIMCOM SIM7600G-H
[2936] Modem Info: SMS DONE Manufacturer: SIMCOM INCORPORATED Model: SIMCOM_SIM7600G-H Revision: SIM7600M22_V2.0 IMEI: 868822041631316 +GCAP: +CGSM
[2950] Waiting for network...
[4012] Network connected
[4012] Connecting to internet.claro.com.ec
[4170] GPRS status: connected
[4185] CCID: 8959301000893366737
[4197] IMEI: 868822041631316
[4207] IMSI: 740010189336673
[4220] Operator: CLARO EC Claro
[4232] Local IP: 100.103.164.180
[4241] Signal quality: 99
[10255] ### Unhandled: +NETOPEN: 0
OK
[10255] GPRS disconnected
End of tests. Enable deep sleep , Will wake up in 30 seconds

```

Tarjeta que se está utilizando

Conexión exitosa con la red LTE

Conexión a internet de claro

IMEI y operadora

Porcentaje de la señal

Figura 55 Datos de la red móvil por parte de la placa IoT

.Elaborado por el investigador

### Testeo funcionamiento GPS

Para la segunda antena se habilita la directiva preprocesada del fabricante(figura 56) para obtener los datos ilustrados en la figura 57 con una precisión de 2.70 metros, se copia la latitud y longitud del monitor serial y se lo define en GOOGLE MAPS para verificar la posición visible en la figura 56, siendo la ubicación precisa de la zona en donde se encuentra la bicicleta al momento de la investigación.

```

// #define TINY_GSM_TEST_GPRS true
// #define TINY_GSM_TEST_TCP true
// #define TINY_GSM_TEST_CALL true
// #define TINY_GSM_TEST_SMS true
// #define TINY_GSM_TEST USSD true
// #define TINY_GSM_TEST TEMPERATURE true
// #define TINY_GSM TEST TIME true
// #define TINY_GSM TEST GPS true
// powerdown modem after tests
// #define TINY_GSM_POWERDOWN true
// #define TEST_RING_RI_PIN true

DBG("Requesting current GPS/GNSS/GLONASS location");
for (;;) {
    digitalWrite(LED_PIN, !digitalRead(LED_PIN));
    if (modem.getGPS(&lat2, &lon2, &speed2, &alt2, &vsat2, &usat2, &accuracy2,
        &year2, &month2, &day2, &hour2, &min2, &sec2)) {
        DBG("Latitude:", String(lat2, 8), "Longitude:", String(lon2, 8));
        DBG("Speed:", speed2, "Altitude:", alt2);
        DBG("Visible Satellites:", vsat2, "Used Satellites:", usat2);
        DBG("Accuracy:", accuracy2);
        DBG("Year:", year2, "Month:", month2, "Day:", day2);
        DBG("Hour:", hour2, "Minute:", min2, "Second:", sec2);
        break;
    } else {
        light_sleep(2);
    }
}
DBG("Retrieving GPS/GNSS/GLONASS location again as a string");
String gps_raw = modem.getGPSraw();
DBG("GPS/GNSS Based Location String:", gps_raw);
DBG("Disabling GPS");
modem.disableGPS();
#endif

```

Figura 56 Código de la directiva preprocesada del GPS. Elaborado por el investigador

```

[2893] Modem Name: SIMCOM SIM7600G-H
[2920] Modem Info: Manufacturer: SIMCOM INCORPORATED Model: SIMCOM_SIM7600G-H Revision: SIM7600M22_V2.0 IMEI: 868822041631
[2934] Waiting for network...
+DCard MOUNT FAIL
[1771] Initializing modem...
[1771] ### TinyGSM Version: 0.11.5
[1771] ### TinyGSM Compiled Module: TinyGsmClientSIM7600
[1827] ### Modem: SIMCOM SIM7600G-H
[1828] ### Modem: SIMCOM SIM7600G-H
[1871] Connecting to vsh.pp.ua
[1918] ... failed
[1918] Enabling GPS/GNSS/GLONASS
[3920] Requesting current GPS/GNSS/GLONASS location
[94521] Latitude: -1.21575141 Longitude: -78.61633301
[94521] Speed: 0.00 Altitude: 2601.50
[94521] Visible Satellites: 0 Used Satellites: 0
[94522] Accuracy: 2.70
[94524] Year: 2022 Month: 7 Day: 31
[94528] Hour: 5 Minute: 32 Second: 2
[94531] Retrieving GPS/GNSS/GLONASS location again as a string
[94556] GPS/GNSS Based Location String: 2,04,01,00,0112.945120,S,07836.980276,W,310722,053206.0,2601.6,0.0,,2.9,2.7,1.0
[94556] Disabling GPS

```



*Figura 57 Datos del GPS que se obtiene por parte de la placa IoT, con la representación de la ubicación  
Elaborado por el investigador*

## **Prueba de funcionamiento del Sistema**

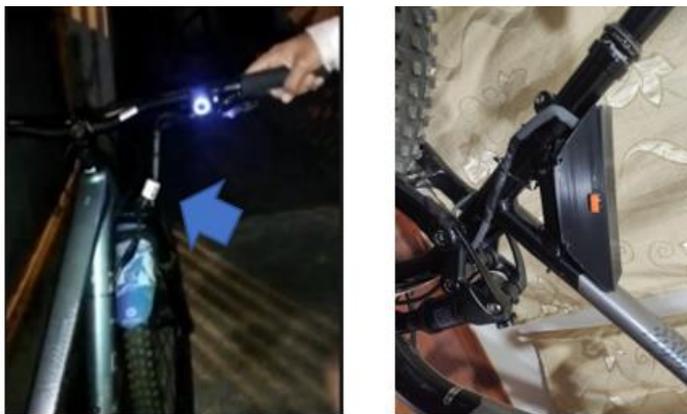
Luego de haber realizado las pruebas de módulos, compilado los códigos, cargado los archivos en la nube para el enlace con la base de datos y la aplicación “CONTROLBIKE” se procede a campo, el sistema que se expone a continuación no dispone de una pantalla para poder seleccionar menús ya que es un sistema de alertas para evitar la pérdida de la

bicicleta en caso de robo, por ello se decide utilizar el led que tiene el lector de huella integrado para demostrar los procesos de llamada, envío de la ubicación y activación de alarmas y verificación del usuario con los diferentes colores como se ilustra en la figura 58.



*Figura 58 Sensor Biométrico. Elaborado por el investigador*

El encendido del sistema consta de dos partes, se debe primero encender la placa T-SIM7600 con el switch propio, este se ve de color naranja por fuera de la carcasa, la placa IoT se mantendrá en espera de activación, para ello se procederá a encender el sensor biométrico con el interruptor que se encuentra debajo del manubrio como se visualiza en la figura 59, esto ayuda a disminuir el consumo de energía en un 79.8%, aumentando el tiempo de vida de la batería a utilizar ya que solo permite el funcionamiento del microcontrolador dejando en stand by el sistema.



*Figura 59 Encendido del prototipo. Elaborado por el investigador*

Después de encender el sistema, la alarma desactivada se muestra en color magenta como se visualiza en la figura 60, entonces el dueño puede dejar estacionada su bicicleta inclinada, se debe dar un pulso para activar al sensor de inclinación, sin este no se detectará manipulación del vehículo.



*Figura 60 Alarma desactivada. Elaborado por el investigador*

En la figura 60 lado izquierdo se ilustra el encendido del sistema en magenta y el derecho la activación del sensor para monitorear el estado del vehículo. Una vez en color rojo, si la bicicleta se manipula pasa a color azul por breves instantes entonces se realizará la llamada a un número de teléfono preestablecido



*Figura 61 Sistema de alerta activado. Elaborado por el investigador*

Cuando se encuentre en verde, se determina que se ha enviado las coordenadas de la bicicleta, este ciclo se repetirá y enviara las ubicaciones con las llamadas hacia el teléfono

y la aplicación ,mientras no se verifique el dueño en el momento del parpadeo de color verde (figura 61) se repetirá el ciclo de alerta y rastreo, asimismo si otra persona es la que se quiere verificar en el momento del parpadeo se encenderá una luz roja que indica un usuario incorrecto y no se desactivara la alarma.

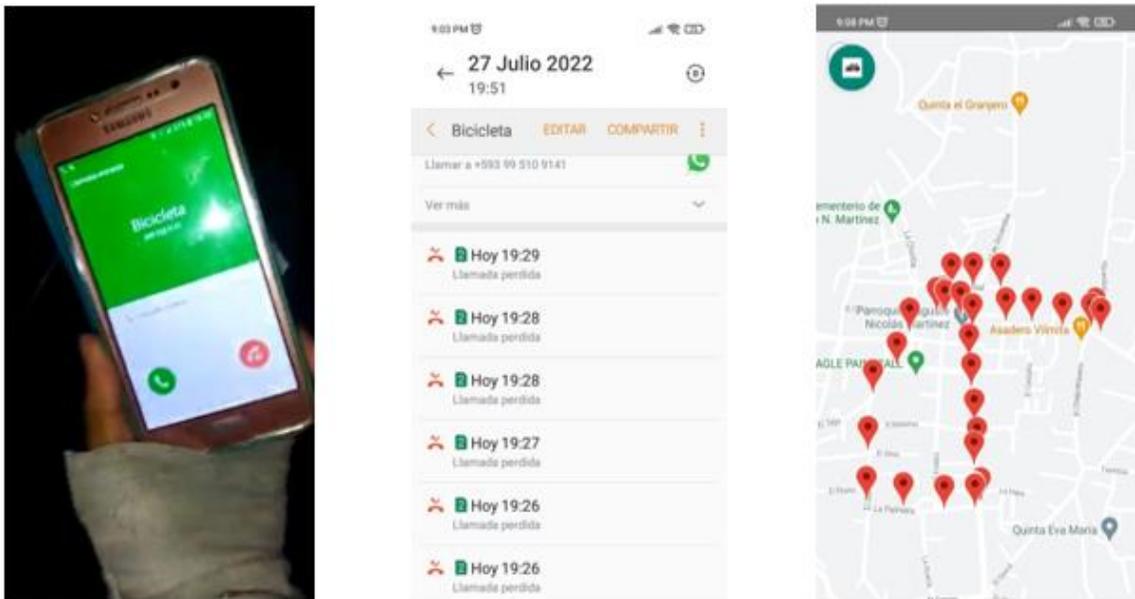


Figura 62 Notificaciones del sistema activado. Elaborado por el investigador.

Se puede verificar el proceso de alerta en la figura 62, se ha capturado el momento de la llamada perdida, se utiliza este método como notificación ya que el teléfono celular interrumpe toda actividad cuando hay una llamada entrante con ello logramos que el dueño, preste atención a lo sucedido el podrá ir a ver su vehículo y al mismo tiempo seguir verificando si se está trasladando de un lugar hacia otro, como se puede ver en el mapa anterior se visualizan los últimos 50 Marker que envía desde el servidor, estos se irán actualizando cada 30 segundos en la aplicación y limpiando los anteriores para poder visualizar una ruta.

La latencia del sistema en cuanto al envío de datos hacia la base, esta entre 30 a 42 segundos en movimiento, esto se verifica en la figura 63 columna amarilla, un error que no se pudo solucionar fue la hora debido a que el hosting no lo permite, se sobrescribió la

zona horaria en el archivo php.ini del hosting, sin tener resultado alguno en la base de datos de phpMyAdmin, el hosting entrega la zona horaria UTC y para poder interpretar en los mapas de la aplicación se debe tomar en cuenta 5 horas de diferencia.

	id	lat	lon	id_imei	fecha	id_dispositivo
<input type="checkbox"/>	678	-1.22180951	-78.61710358	868822041631316	2022-07-28 00:21:15	1
<input type="checkbox"/>	679	-1.22181404	-78.61708069	868822041631316	2022-07-28 00:21:46	1
<input type="checkbox"/>	680	-1.22158837	-78.61702728	868822041631316	2022-07-28 00:23:57	1
<input type="checkbox"/>	681	-1.22080493	-78.61764526	868822041631316	2022-07-28 00:24:38	1
<input type="checkbox"/>	682	-1.22041571	-78.61775970	868822041631316	2022-07-28 00:25:18	1
<input type="checkbox"/>	683	-1.21976697	-78.61819458	868822041631316	2022-07-28 00:26:00	1
<input type="checkbox"/>	684	-1.21894300	-78.61872101	868822041631316	2022-07-28 00:26:42	1
<input type="checkbox"/>	685	-1.21826696	-78.61915588	868822041631316	2022-07-28 00:27:22	1
<input type="checkbox"/>	686	-1.21748817	-78.61946106	868822041631316	2022-07-28 00:28:02	1
<input type="checkbox"/>	687	-1.21734548	-78.61991119	868822041631316	2022-07-28 00:28:43	1
<input type="checkbox"/>	688	-1.21743834	-78.62038422	868822041631316	2022-07-28 00:29:23	1
<input type="checkbox"/>	689	-1.21749532	-78.62031555	868822041631316	2022-07-28 00:29:58	1

Figura 63 Datos recibidos del prototipo en la BD. Elaborado por el investigador

### Consumo de energía del prototipo

Después de presentar las pruebas en campo se evaluará el tiempo de descarga de la batería 18650 Li-ion la cual tiene una capacidad de 1500mAh a , que se utiliza en el prototipo para ello se ha utilizado la fórmula matemática establecida por diferentes fabricantes en la ecuación (1).

$$\text{Horas estimadas} = \frac{\text{Capacidad de la batería(mAh)}}{\text{Corriente de carga (mA)}} * 0.7$$

La capacidad de la batería utilizada es de 1500mAh, en cuanto a la corriente de carga se refiere al consumo total de energía del prototipo por lo que se realiza la sumatoria del consumo de energía de cada módulo, estos datos se encuentran en las características técnicas como se muestra en la siguiente tabla:

Módulos	Consumo
ESP 32	20 mA
GPS NEO 6	10 mA
SIM7600G	2.8 mA
Sensor biométrico	60 mA
Sensor inclinación	5mA

*Tabla 10 Consumo de energías de los módulos utilizados  
Elaborado por el investigador*

La sumatoria de los módulos y sensores da como resultado un consumo de 97.8mA, se reemplaza los datos en la ecuación uno y se multiplica por un factor externo que es igual a 0.7

$$\text{Horas estimadas} = \frac{1500(\text{mAh})}{97.8 (\text{mA})} * 0.7 \quad \text{Ecuacion 1}$$

$$\text{Horas estimadas} = 10.73 \text{ h}$$

Con la capacidad de corriente de operación de 1500mAh que opera el prototipo , se obtiene la duración igual a 10.73 h, este es un valor teórico. Luego de haber echo pruebas en campo su duración fue cercana a las 9 horas lo cual permite el cálculo de eficiencia del sistema en la ecuación 2 dando como resultado un sistema 83% .

$$\text{Eficiencia del sistema} = \frac{\text{Horas practicas}}{\text{Horas estimadas}} * 100 \quad \text{Ecuacion 2}$$

$$\text{Eficiencia del sistema} = \frac{9}{10.73} * 100 = 83\%$$

## Presupuesto del proyecto

El costo total de construcción del proyecto se detalla en la siguiente tabla

Presupuesto				
#	Detalle	Cantidad	Valor Unitario	Valor total
1	T-sim 7600G con importación	1	\$120	\$120
2	Sensor biométrico DFRobot	1	\$35	\$35
3	Sensor inclinación KY-020	1	\$0.90	\$0.90
4	Carcasas del prototipo	2	\$12.5	\$25
5	Cables	Varios	\$5	\$5
6	Batería Li-ion 1500 mAh 3.7V	1	\$3	\$3
7	Interruptor	1	\$0.25	\$0.25
8	Hosting	1 año	\$2.99	\$2.99
			Total	\$192.14

Para el diseño del presupuesto se tiene en cuenta el valor del salario mínimo según el ministerio de trabajo desde enero del 2022 para profesionales con tercer grado es de \$467.19 mensuales. A partir del salario mensual se debe obtener el salario diario mediante la ecuación 3

$$\text{Salario Diario} = \frac{\text{Salario mensual}}{\text{Días laborables}} \quad \text{Ecuacion 3}$$

$$\text{Salario Diario} = \frac{\$467.19}{21} = 22.24$$

Además, se tiene en cuenta un promedio de 21 días hábiles al mes. De acuerdo con la ecuación 3 se calcula el valor del salario por día. Por último se calculó el salario por horas con la ecuación 4, el día de trabajo es de 8 horas.

$$\text{Salario horas} = \frac{\text{Salario diario}}{\text{Horas}} \quad \text{Ecuacion 4}$$

$$\text{Salario horas} = \frac{\$22.24}{8} = \$2.78c/h$$

Después de haber calculado el salario por hora se obtiene el costo del diseño del proyecto respecto a las horas empleadas son 60 horas.

$$\textit{Costo del diseño} = \textit{Salario horas} * \textit{horas empleadas} \quad \textit{Ecuacion 4}$$

$$\textit{Costo del diseño} = \$2.78 * 60 = \$166.80$$

Finalmente se obtiene el presupuesto total del proyecto con la sumatoria del costo total de construcción y el costo del diseño dando como resultado:

$$\textit{Presupuesto total} = \textit{costo del diseño} + \textit{costo de construccion} \quad \textit{Ecuacion 5}$$

$$\textit{Presupuesto total} = \$166.80 + \$192.14$$

$$\textit{Presupuesto total} = \$358.94$$

## CAPITULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

- Tras el análisis de los dispositivos inteligentes para bicicletas de uso recreativo o de transporte se evidencio que la mayor parte de dispositivos están destinados al deporte semiprofesional , es decir que estos dispositivos se encargan solo del almacenamiento de las estadísticas deportivas que se ha realizado, como tiempos, distancias o rutas en cuanto a otros dispositivos dan seguridad local es decir con alarmas sonoras ineficientes a largas distancias, lo expuesto anteriormente dio paso al desarrollo de este proyecto dando como resultado el desarrollo de un prototipo IoT que controla quien utiliza la bicicleta , monitorea a la distancia con la aplicación “CONTROLBIKE” y da alertas de llamadas.
- De acuerdo con las tecnologías inalámbricas utilizadas en dispositivos ya mencionados anteriormente, se utilizaron redes PAN (Personal Área Network) en algunos casos , y las variables eran determinadas para evaluar la condición física de los ciclistas mas no se diseñaron para cuidar sus vehículos en caso de robo, por ello se estableció que se diseñe un sistema que abarque mayor cobertura y al mismo tiempo pueda alertar al dueño, la tecnología más viable es LTE ya que se puede desarrollar en un entorno IoT , con la tecnología resuelta se selecciona la placa de desarrollo IoT T-SIM7600G,la cual incorpora antenas LTE y GPS disminuyendo el uso de módulos externos, para la transferencia de datos con el Hosting se requiere de una tarjeta Nano Sim de cualquier operada del país con una conexión de datos, dando facilidad al usuario de contratar paquetes solo cuando requiera del servicio.

- La investigación culmina con el diseño de un prototipo con una autonomía de 9 horas, con respecto a la precisión, se ha realizado un test del GPS en el cual se obtuvo un margen de error de 2.70 m, la cual demuestra que los datos obtenidos por parte del sistema son confiables al momento de rastrear su ubicación mediante la aplicación que se puede instalar en todo dispositivo que ejecute el sistema operativo de Android desde la versión 5.0 en adelante, por otra parte la latencia de los datos generados por el prototipo, obtuvo un tiempo entre 30 a 40s en ser recibido por el servidor, el tiempo de transmisión y recepción se analizó en la base de datos. El uso del sensor biométrico e inclinación permitieron el control y alerta, realizando llamadas de emergencia cuando el vehículo sea manipulado, mejorando así la seguridad total de la bicicleta.

#### **4.2 Recomendaciones**

- Se recomienda utilizar un batería de mayor capacidad, si es posible importar uno no genérico debido a que la mayor parte de baterías económicas no presentan las características establecidas , esto ayudara a tener una mayor autonomía en el sistema, con ello el diseño del prototipo podrá ser modificado y ensamblado en otros tipos de vehículos que requieran servicios de rastreo a bajo costo.
- Se desarrollo un web Services con el soporte para que un cliente pueda tener varios dispositivos , sin embargo por costos de importación solo se trabajó con uno, se recomienda para futuras investigaciones aumentar las características como un perfil en la aplicación móvil si el proyecto se quiere vender como una idea de negocio.

## C. MATERIALES DE REFERENCIA

### Bibliografía

- [1] C. ShinyeJoo, «sciencedirect,» 22 Abril 2015. [En línea]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X15001539>.
- [2] F. B., «Why Cycling Matters for Smart Cities. Internet of Bicycles for Intelligent Transport.,» *ELSEVIER*, pp. 157-164, 2016.
- [3] O. BUSINESS, «Sherlock connects GPS-based anti-theft device for bicycles to global IoT network from Orange Business Services,» *ORANGE BUSINESS*, Paris, 2016.
- [4] J. V. D. JACOME, «UNIVERSIDAD TECNICA DEL ´ NORTE,» 12 2018. [En línea]. Available: <http://repositorio.utn.edu.ec/bitstream/123456789/8825/1/04%20MEC%20245%20TRA%20BAJO%20DE%20GRADO.pdf>.
- [5] B. J. M. Xavier, «Sistema autonomo de monitoreo de señales fisiologicas con gestion de emergencias para seguridad vial de ciclistas amateur,» repositorio.uta.edu.ec, Ambato, 2019.
- [6] M. T. Rodó, «Transformando la bicicleta en laboratorios móviles. El caso de un self-tracking ciudadano para la ciudad de Santiago,» Pontificia Universidad Católica de Chile, Santiago de Chile , 2016.
- [7] G. a. K. N. Norman, «Using Smartphones For Cycle Planning,» 2015.
- [8] Tim Schwanen, «Beyond instrument: smartphone app and sustainable mobility,» University of Oxford, UK.
- [9] C. Kiefer, «Smart E-Bike Monitoring System:Realtime Open-Source and Open Hardware GPS, Assistance and Sensor Data,» *Researchgate*, 2015.
- [10] labicicletainfo, «labicicletainfo,» 2017. [En línea]. Available: <https://labicicleta.info/>.
- [11] JORDI SALAZAR ,SANTIAGO SILVESTRE, «INTERNET DE LAS COSAS,» ERASMUS+, Czech Republic.
- [12] A. L. Perez Colome , . C. A. Calderón y T. Delgado Fernández, «Procedimiento para la implementación,» RIELAC, 2021.
- [13] S. I. Diaz Guerrero , J. A. Cordoba Carrero y O. A. Corzo Gomez, «Análisis del Paradigma Fog y Edge Computing,» Facultad de Fisicomecánicas, Ingeniería de Sistemas, Universidad Industrial de Bucaramanga.

- [14] M. Iorda y N. Goren, «NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY U.S. DEPARTAMENT OF COMMERCE,» 03 2018. [En línea]. Available: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-325.pdf>.
- [15] N. Mohan y J. Kangasharju, «Edge-Fog Cloud: A Distributed Cloud for Internet of Things Computations",» 2017. [En línea]. Available: DOI:10.1109/CIOT.2016.7872914.
- [16] S. E. L. C. Karen Rose, «Internet Society (ISOC),» [En línea]. Available: <https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf>.
- [17] A. M. B. OBANDO, «ESTUDIO DE PARÁMETROS Y CARACTERÍSTICAS PARA EL DESARROLLO DE APLICACIONES,» Bogota, 2018.
- [18] N. G. V. R. C. R. Jose Berne Valero, GNSS:GPS,GALILEO,GLONASS,BEIDOU FUNDAMENTOS Y METODOS DE POSICIONAMIENTO, Valencia: Universidad Politecnica de Valencia, 2019.
- [19] J. Fallas, «Sistema de Posicionamiento Global,» Universidad Nacional Heredia, 2002.
- [20] L. C. -. E. S., «Sistema de Navegacion para personas no videntes,» repositorio.espe.edu.ec.
- [21] U. government, «GPS.GOV,» 2021. [En línea]. Available: <https://www.gps.gov/systems/gps/control/>.
- [22] E. Huerta , A. Mangiaterra y G. Noguera, «GPS Posicionamiento Satelital,» UNR, 2005.
- [23] I. J. A. Capdevila, «El sistema de Posicionamiento Global,» [En línea]. Available: <https://ansenuza.unc.edu.ar/comunidades/bitstream/handle/11086.1/1258/EI%20Sistema%20de%20Posicionamiento%20Global.pdf?sequence=1&isAllowed=y>.
- [24] D. D. C. Flores y D. D. Àvila, «researchgate,» 2021. [En línea]. Available: [https://www.researchgate.net/publication/354247970\\_Sobre\\_los\\_Sistemas\\_Satelitales\\_de\\_Navegacion\\_Global\\_guia\\_basica\\_para\\_ecologos\\_On\\_Global\\_Navigation\\_Satellite\\_Systems\\_a\\_primer\\_for\\_ecologists](https://www.researchgate.net/publication/354247970_Sobre_los_Sistemas_Satelitales_de_Navegacion_Global_guia_basica_para_ecologos_On_Global_Navigation_Satellite_Systems_a_primer_for_ecologists).
- [25] P. Gonzales Alcaraz , «Levantamiento mediante GPS de una red de puntos establecidos para correlacionar los distintos espaciones de la universidad en el msimo sistema de coordenadas,» Universidad Politecnica de Cartagena, Cartagena .
- [26] F. Orozco Iguasni, R. Villa López, . L. Guzmán y . W. Chango, «ANÁLISIS PARA LA CORRECCIÓN DE ERRORES EN LA SEÑAL GPS,» Mikarimin. Revista Científica Multidisciplinaria ISSN 2528-7842, Santo Domingo. Ecuador, 2020.
- [27] U. I. d. Valencia, «Evolución de la red de comunicación móvil, del 1G al 5G,» 21 03 2018. [En línea]. Available: <https://www.universidadviu.com/int/actualidad/nuestros-expertos/evolucion-de-la-red-de-comunicacion-movil-del-1g-al-5g>.

- [28] J. d. J. Arias Franco, «Entorno regulatorio del precio del espectro radioelectrico y sus efectos en la introduccion de tecnologias moviles de quinta generacion en Mexico,» Infotec Posgrados, Ciudad de Mexico, 2019.
- [29] D. G. Barreno Naranjo, D. P. Carrión Buenaño y v. Darwin Paúl, «Evolución de la tecnología móvil. Camino a 5G” Revista Contribuciones a las Ciencias Sociales,» eumed, Guaranda Ecuador, 2016.
- [30] J. A. Sánchez, «Análisis y Estudio de Redes Gprs,» Universidad Austral de Chile, Valdivia.
- [31] L. E. Peñafiel Yanza y F. Chenche Viliacís, «La convivencia de la tecnología 4G con 3G y 2G en Ecuador.,» *Research Review*, nº 6, pp. 15-37, 2021.
- [32] Equipo de Expertos en Ciencia y Tecnología, «Universidad Internacional de Valencia,» 21 Marzo 2018. [En línea]. Available: <https://www.universidadviu.com/int/actualidad/nuestros-expertos/evolucion-de-la-red-de-comunicacion-movil-del-1g-al-5g>.
- [33] M. Jaramillo, «Implementación de Red Móvil con Tecnología 4G LTE,» Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil-Ecuador.
- [34] Arcotel , «www.arcotel.gob.ec,» 2018. [En línea]. Available: [https://www.arcotel.gob.ec/wp-content/uploads/2015/01/BOLETIN-ESTADISTICO-Diciembre-2018-v4\\_4.pdf](https://www.arcotel.gob.ec/wp-content/uploads/2015/01/BOLETIN-ESTADISTICO-Diciembre-2018-v4_4.pdf).
- [35] Repositorio Universidad de las Americas, «Servidor/Cliente, Capitulo 5,» 2020.
- [36] PHP, «PHP.net,» 2021. [En línea]. Available: <https://www.php.net/manual/es/intro-whatis.php>.
- [37] phpmyadmin, «phpMyAdin,» 2022, [En línea]. Available: <https://www.phpmyadmin.net/>.
- [38] GOOGLE DEVELOPERS, «developer.android,» [En línea]. Available: [https://developer.android.com/kotlin?hl=es-419&gclid=CjwKCAjwrZOXBhACEiwA0EoRD4BkaqYI0TUCNDAvfeISWE7BYWguYvKb703fWJ1P9L688EaxNGPekhoC8ZUQAuD\\_BwE&gclsrc=aw.ds](https://developer.android.com/kotlin?hl=es-419&gclid=CjwKCAjwrZOXBhACEiwA0EoRD4BkaqYI0TUCNDAvfeISWE7BYWguYvKb703fWJ1P9L688EaxNGPekhoC8ZUQAuD_BwE&gclsrc=aw.ds).
- [39] Lilygo, «Lilygo,» [En línea]. Available: [http://www.lilygo.cn/prod\\_view.aspx?TypeId=50063&Id=1246](http://www.lilygo.cn/prod_view.aspx?TypeId=50063&Id=1246).
- [40] J. R. M. Fernandez, «ResearchGate,» 01 216. [En línea]. Available: [https://www.researchgate.net/figure/Figura-3-Conmutacion-de-Paquetes-y-Conmutacion-de-Circuitos-Representada-a-la-derecha\\_fig3\\_308878882](https://www.researchgate.net/figure/Figura-3-Conmutacion-de-Paquetes-y-Conmutacion-de-Circuitos-Representada-a-la-derecha_fig3_308878882).
- [41] INEC, «A PEDALEAR,» 2016.
- [42] OPS, «Organización Mundial de La Salud,» 03 06 2020. [En línea]. Available: <https://www.paho.org/es/noticias/3-6-2020-bicicleta-movilizacion-segura-saludable-tiempos-covid-19-3-junio-dia-mundial>.

- [43] F. F. D. A. Nataly Pinto Alvaro, «La situación de la bicicleta en Ecuador,» *FRIEDRICH EBERT STIFTUNG*, p. 19, 2015.
- [44] «bikehouse,» [En línea]. Available: <https://bikehouse.ec/tienda/ecommerce.buscador.php>.
- [45] [eltelegrafo.com.ec](http://eltelegrafo.com.ec), *42 ciclistas murieron arrollados desde 2017*, 05 06 2020.
- [46] Cisco, «Cisco Annual Internet Report (2018–2023) White Paper,» 9 Marzo 2020. [En línea]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
- [47] M. A. O. Gonzales, «Método diferencial GPS en tiempo real aplicado a una red geodésica vía GPRS,» Elibro, Santiago de Chile, 2007.

## Anexos

### Anexo A Código T sim 7600

```
#define TINY_GSM_MODEM_SIM7600 //placa a utilizar
#define TINY_GSM_RX_BUFFER 1024 // Set RX buffer to 1Kb

#define COLLECT_NUMBER 3
#include <HardwareSerial.h>
#include <DFRobot_ID809.h>

DFRobot_ID809 fingerprint;
HardwareSerial SerialAT(1);
HardwareSerial FPSerial(2);

#if !defined(TINY_GSM_RX_BUFFER)
#define TINY_GSM_RX_BUFFER 650
#endif
#define TINY_GSM_YIELD() { delay(2); }
const char apn[] = "internet.claro.com.ec"; // Change this to your Provider details
const char gprsUser[] = "";
const char gprsPass[] = "";
const char server[] = "pruebaslaboratorio.atwebpages.com"; // Change this to your
selection
const char resource[] = "/conexionTtgo.php";
const int port = 80;
unsigned long timeout;

#define MODEM_TX          27 //tx sim
#define MODEM_RX          26 //rx sim
#define MODEM_PWRKEY      4 //on placa IoT
#define LED_PIN           12 //indicador
#define MODEM_FLIGHT      25 // led indicado de sistema
#define MODEM_STATUS      34 // led de estado
#define SENSOR            23 // alarma de inclinacion
#define IRQ                5 // retroalimentacion lector de huellas
#include <TinyGsmClient.h>
TinyGsm modem(SerialAT);
TinyGsmClient client(modem);

int pulsador = 0; //almacena el estado del botón
int estado = 0; //0=biometrico sin pulsar, 1=biometrico pulsado
int pulsadorAnt = 0; //almacena el estado anterior del boton

void setup() {
  modem_on();
  pinMode(SENSOR, INPUT_PULLUP);
  pinMode(IRQ , INPUT);

  // descomentar si necesita visualizar
  Serial.begin(115200); // Set console baud rate
```

```

delay(50);
//comunicacion serial entre T-SIM y Esp32
SerialAT.begin(115200, SERIAL_8N1, MODEM_RX, MODEM_TX);
delay(1000);
//comunicacion serial entre lector de huellas y Esp32
FPSerial.begin(115200, SERIAL_8N1, 13, 14); //az am
delay(1000);
//indicacion de sistema listo
pinMode(LED_PIN, OUTPUT);
digitalWrite(LED_PIN, HIGH);

fingerprint.begin(FPSerial);

while (fingerprint.isConnected() == false) {
    Serial.println("La comunicación con el dispositivo falló, verifique la conexión");
    delay(1000);
}

pinMode(MODEM_FLIGHT, OUTPUT);
digitalWrite(MODEM_FLIGHT, HIGH);
}

void loop() {
    pulsador = digitalRead(IRQ); //lee si el botón está pulsado
    if ((pulsador == HIGH) && (pulsadorAnt == LOW)) { //si el boton es pulsado y antes
lo estaba
        estado = 1 - estado;
        delay(1000); //pausa
    }
    pulsadorAnt = pulsador; //actualiza el nuevo estado del boton
    if (estado == 1) { //si el estado es 1
        Serial.println("Modo parking"); //se enciende el led
        delay(2000);
        comparar();
        //gnss();
    }
    else { //si el estado es 0
        Serial.println("Alarma desactivada"); //se apaga el led
        delay(1000);
        //Serial.println("Ultrasonico");
        fingerprint.ctrlLED(/*LEDMode = */fingerprint.eFastBlink , /*LEDColor = */fingerp
eLEDMagenta, /*blinkCount = */2);
    }
}
}

void modem_on() { // encendido placa

```

```

    pinMode(MODEM_PWRKEY, OUTPUT);
    digitalWrite(MODEM_PWRKEY, LOW);
    delay(1500); //Need delay
    digitalWrite(MODEM_PWRKEY, HIGH);
}

void alerta() { // llamada de emergencia
    SerialAT.println("ATD+593998895642;");
    Serial.println("Llamando");
    delay(6000);
    SerialAT.println("AT+CVHU=0");
    delay(4000);
    Serial.println("Colgando");
    SerialAT.println("ATH");
    delay(4000);
}

void comparar() { // metodo de autenticacion

    fingerprint.ctrLED(/*LEDMode = */fingerprint.eFastBlink , /*LEDColor = */fingerprint.eLEDRed, /*blinkCount = */1);
    delay(2000);
    if (digitalRead(SENSOR) == LOW) { //detecta la manipulacion de la bicicleta
        uint8_t ret = 0;
        fingerprint.ctrLED(/*LEDMode = */fingerprint.eBreathing , /*LEDColor = */fingerprint.eLEDBlue, /*blinkCount = */2);
        Serial.println("Please press down your finger");
        while ( fingerprint.detectFinger() == 0) { // si no se detecta la huella una vez manipulada la
            //bicicleta se procede a llamar a un numero en especifico y enviar la ubicacion
            la aplicacion
            Serial.println("No se detecto Huella");
            alerta();
            delay(4000);
            gnss();
            delay(2000);
            break;
        }

        if ((fingerprint.collectionFingerprint(/*timeout=*/0)) != ERR_ID809) {
            Serial.println("Please release your fingers");
            fingerprint.ctrLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor = */fingerprint.eLEDWhite, /*blinkCount = */1);

            if ( fingerprint.detectFinger() != 0) {
                ret = fingerprint.search();
                /*Compare la huella digital capturada con una huella digital de identificación específica
                Devuelve el ID de la huella digital (1-80) si tiene éxito, devuelve 0 cuando

```

```

falla
    */
    Serial.print("Matching succeeds,ID=");
    Serial.println(ret);
    /*Establezca el anillo LED de huella digital para que siempre esté ENCENDIDO
verde */
    fingerprint.ctrLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor =
*/fingerprint.eLEDGreen, /*blinkCount = */3);
    Serial.println("Capturing succeeds");
    fin();
    delay(2000);

    } else {
        Serial.print("Matching succeeds,ID=");
        Serial.println(ret);
        /*Set fingerprint LED ring to always ON in red*/
        fingerprint.ctrLED(/*LEDMode = */fingerprint.eFastBlink, /*LEDColor =
*/fingerprint.eLEDRed, /*blinkCount = */3);
        Serial.println("Matching fails");
        alerta();
        gnss();
        delay(2000);
    }
}
Serial.println("-----");
}
}

void enableGPS(void) { //habilita gps
    // Set SIM7000G GPIO4 LOW ,turn on GPS power
    // CMD:AT+SGPIO=0,4,1,1
    // Only in version 20200415 is there a function to control GPS power
    modem.sendAT("+SGPIO=0,4,1,1");
    if (modem.waitResponse(10000L) != 1) {
        DBG(" SGPIO=0,4,1,1 false ");
    }
    modem.enableGPS();
}

void disableGPS(void) { //deshabilita gps
    // Set SIM7000G GPIO4 LOW ,turn off GPS power
    // CMD:AT+SGPIO=0,4,1,0
    // Only in version 20200415 is there a function to control GPS power
    modem.sendAT("+SGPIO=0,4,1,0");
    if (modem.waitResponse(10000L) != 1) {
        DBG(" SGPIO=0,4,1,0 false ");
    }
}

```

```

    modem.disableGPS();
}

void gnss() { //metodo conexion a internet de claro y envio de ubicacion
    Serial.print("Waiting for network...");//c
    if (!modem.waitForNetwork()) {
        // Serial.println(" fail");
        delay(1000);
        return;
    }
    Serial.println(" success");//c
    if (modem.isNetworkConnected()) {
        Serial.println("Network connected");//c
    }

    Serial.print(F("Connecting to "));//c
    Serial.print(apn);
    if (!modem.gprsConnect(apn, gprsUser, gprsPass)) {
        Serial.println(" fail");//c
        delay(1000);
        return;
    }
    Serial.println(" success");//c

    if (modem.isGprsConnected()) {
        Serial.println("GPRS connected"); //c
    }

    if (!client.connect(server, port)) {
        Serial.println(" fail");//c
    }

    Serial.println("Realizando solicitud HTTP POST...");

    //INGRESAR LOS DATOS DE LAT LON
    Serial.println("Empezar a posicionar. Asegúrese de ubicar al aire libre.");
    Serial.println("La luz indicadora azul parpadea para indicar el posicionamiento.");

    enableGPS();
    String imei = modem.getIMEI();// funcion para obtener el imei del dispositivo
    float lat, lon;
    while (1) {
        if (modem.getGPS(&lat, &lon)) { // funcion para obtener latitud y longitud
            Serial.println("The location has been locked, the latitude and longitude are:");
            Serial.print("latitude:"); Serial.println(lat, 8);
            Serial.print("longitude:"); Serial.println(lon, 8);
            Serial.print("IMEI:"); Serial.println(imei);
            break;
        }
    }
}

```

```

    }
    digitalWrite(LED_PIN, !digitalRead(LED_PIN));
    delay(2000);
}
//peticion envio datas
String httpRequestData = "key=a@4K%3&latitude=" + String(lat, 8) + "&longitude=" +
String(lon, 8) + "&imei=" + String(imei) + "" ;
client.print(String("POST ") + resource + " HTTP/1.1\r\n");
client.print(String("Host: ") + server + "\r\n");
client.println("Connection: close");
client.println("Content-Type: application/x-www-form-urlencoded");
client.print("Content-Length: ");
client.println(httpRequestData.length());
client.println();
client.println(httpRequestData);
timeout = millis();
while (client.connected() && millis() - timeout < 10000L) {
    while (client.available()) {
        char c = client.read();
        Serial.print(c);
        timeout = millis();
    }
}
Serial.println();
client.stop();
Serial.println(F("Server disconnected"));
modem.gprsDisconnect();
Serial.println(F("GPRS disconnected"));
modem.disableGPS();
Serial.println(F("GPS disconnected"));

fingerprint.ctrlLED(/*LEDMode = */fingerprint.eSlowBlink , /*LEDColor = */fingerpri
eLEDGreen, /*blinkCount = */3);
}

void fin() { // reinicio del sistema
    estado = 0;
    delay(100);
    Serial.println("Alarma desactivada");
}

```

## Anexo B1

### Constants.php

```
1 <?php
2     define("DB_HOST", "pdb55.awardspace.net");
3     define("DB_USER", "4061622_bike");
4     define("DB_PASSWORD", "");//se oculta por seguridad
5     define("DB_DATABASE", "");//se oculta por seguridad |
6 ?>
```

## Anexo B2

### DbConnect.php

```
1 <?php
2     class DbConnect
3     {
4         //Variable para almacenar el enlace de la base de datos
5         private $con;
6
7         //Class constructor
8         function __construct()
9         {
10
11         }
12
13         //This method will connect to the database
14         function connect()
15         {
16             //Incluyendo el archivo constants.php para obtener las constantes de la base de datos
17             //include_once dirname(__FILE__) . '/Constants.php';
18             include ('Constants.php');
19             //conecatando con mysql database
20             $this->con = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_DATABASE);//conexion a la BD
21
22             //Comprobando si ocurrió algún error al conectar
23             if (mysqli_connect_errno()) {
24                 echo "Failed to connect to MySQL: " . mysqli_connect_error();
25                 return null;
26             }
27
28             //finalmente devolviendo el enlace de conexión|
29             return $this->con;
30         }
31     }
32 ?>
```

## Anexo B3

### DbOperation.php

```
1  <?php
2  class DbOperation
3  {
4      private $con;
5      function __construct()
6      {
7          //require_once dirname(__FILE__) . '/DbConnect.php';
8          include ('DbConnect.php');
9          $db = new DbConnect();
10         $this->con = $db->connect();
11     }
12     //añade usuarios a la base bdBike
13     public function insertUser($usuario, $correo, $imei, $contrasena){
14         $stmt = $this->con->prepare(" INSERT INTO tbl_usuario (id,usuario, correo,
15         contrasena, activo) VALUES (NULL,?,?,?,1); ");
16         $stmt->bind_param("sss",$usuario, $correo, $contrasena);
17         if($stmt->execute())
18             return true;
19         return false;
20     }
21     public function getData($iduser, $iddevice){
22         $sql = " SELECT distinct td.id_dispositivo, td.lat, td.lon, td.fecha "
23         .
24         " FROM tbl_detalle_dispositivo td ".
25         " inner join tbl_dispositivo d on
26         td.id_dispositivo=d.id ".
27         " inner join tbl_usuario u on
28         d.id_usuario=u.id ".
29         " where u.id = ? ".
30         " and id_dispositivo = ? ".
31         " order by td.id desc limit 50 ";
32         $stmt = $this->con->prepare($sql);
33         $stmt->bind_param("ii",$iduser, $iddevice);
34         $stmt->execute();
35         $stmt->bind_result($Id, $Lat, $Lon, $Fecha);
36         $datos = array();
37         while($stmt->fetch()){
38             $temp = array();
39             $temp['id_dispositivo'] = $Id;
40             $temp['lat'] = $Lat;
41             $temp['lon'] = $Lon;
42             $temp['fecha'] = $Fecha;
43             array_push($datos, $temp);
44         }
45         return $datos;
46     }
47     public function getDispositivos($iduser){
48         $sql = " SELECT d.id, d.nombre, d.imei ".
49         " FROM tbl_dispositivo d ".
50         " inner join tbl_usuario u on
51         d.id_usuario=u.id ".
52         " where d.activo=1 and u.id = ? ";
53         $stmt = $this->con->prepare($sql);
54         $stmt->bind_param("i",$iduser);
55         $stmt->execute();
56         $stmt->bind_result($Id, $Nombre, $Imei);
57         $datos = array();
58         while($stmt->fetch()){
59             $temp = array();
60             $temp['id'] = $Id;
61             $temp['nombre'] = $Nombre;
62             $temp['imei'] = $Imei;
63             array_push($datos, $temp);
64         }
65         return $datos;
66     }
67 }
```

## Anexo B4

### Index.php Registro de usuarios desde Android

```
1 <?php
2
3     //adding dboperation file
4     //require_once '../DbOperation.php';
5     include ('DbOperation.php');
6     //response array
7     $response = array();
8
9     if (empty($_POST['usuario']) && empty($_POST['correo']) && empty($_POST['contrasena'])) {
10    $response['correcto'] = false;
11    $response['message'] = 'Campos Vacios al enviar los datos';
12
13 } else {
14     if (isset($_POST['usuario']) && isset($_POST['correo']) && isset($_POST['contrasena'])) {
15         $db = new DbOperation();
16         if($db->insertUser($_POST['usuario'], $_POST['correo'], $_POST['contrasena'])){
17             $response['correcto'] = true;
18             $response['message'] = 'Usuario agregado correctamente';
19         }else{
20             $response['correcto'] = false;
21             $response['message'] = 'No se pudo agregar el usuario';
22         }
23     }
24     }else{
25         $response['correcto'] = false;
26         $response['message'] = 'Invalid Request';
27     }
28 }
29
30 //displaying the data in json
31 echo json_encode($response);
32
33 ?>
```

## Anexo B5

### getDevices.php

```
1 <?php
2     include ('DbOperation.php');
3
4     $response = array();
5
6     if (empty($_POST['idUser'])) {
7         $response['correcto'] = false;
8         $response['message'] = 'Campos Vacios al enviar los datos';
9
10    } else {
11
12        //Obtener registros marcados
13        $db = new DbOperation();
14        if(isset($_POST['idUser'])) {
15            $ListComment = $db->getDispositivos($_POST['idUser']);
16            if(count($ListComment)<=0){
17                $response['correcto'] = false;
18                $response['message'] = 'No se encontró nada en la base de datos.';
19            }else{
20                $response['correcto'] = true;
21                $response['message'] = $ListComment;
22            }
23        }else{
24            $response['correcto'] = false;
25            $response['message'] = 'Invalid Request';
26        }
27    }
28    //displaying the data in json
29    echo json_encode($response);
30 ?>
```

## Anexo B6

### getDatos.php

```
1 <?php
2 include ('DbOperation.php');
3
4 $response = array();
5
6 if (empty($_POST['idUser']) && empty($_POST['idDevice'])) {
7     $response['correcto'] = false;
8     $response['message'] = 'Campos Vacios al enviar los datos';
9
10 } else {
11
12     //Obtener registros marcados
13     $db = new DbOperation();
14     if(isset($_POST['idUser']) && isset($_POST['idDevice'])) {
15         $listComment = $db->getData($_POST['idUser'],$_POST['idDevice']);
16         if(count($listComment)<=0){
17             $response['correcto'] = false;
18             $response['message'] = 'No se encontró nada en la base de datos.';
19         }else{
20             $response['correcto'] = true;
21             $response['message'] = $listComment;
22         }
23     }else{
24         $response['correcto'] = false;
25         $response['message'] = 'Invalid Request';
26     }
27 }
28 //displaying the data in json
29 echo json_encode($response);
30 ?>
```

## Anexo B7

### conexionloginUser.php

```
1 <?php
2
3 include 'conexionLogin.php';
4 $usuario=$_POST['usuario'];
5 $contrasena=$_POST['password'];
6
7 //$usuario="diego";
8 //$contrasena="asdfgh";
9
10 $sentencia=$conexion->prepare(" SELECT u.id idUsuario,d.id idDispositivo ".
11     " FROM tbl_usuario u ".
12     " left join tbl_dispositivo d on u.id=d.id_usuario ".
13     " WHERE u.usuario=? AND u.contrasena=? AND u.activo=1 ");
14 $sentencia->bind_param('ss',$usuario,$contrasena);
15 $sentencia->execute();
16
17
18 $resultado = $sentencia->get_result();
19 if ($fila = $resultado->fetch_assoc()) {
20     echo json_encode($fila,JSON_UNESCAPED_UNICODE);
21 }
22 $sentencia->close();
23 $conexion->close();
24
25
26 ?>
```

## Anexo B8

### conexionTtgo.php

```
1
2 <?php
3 date_default_timezone_set('America/Guayaquil');
4 $script_tz = date_default_timezone_get();
5 include 'Constants.php';
6 // $key = "";
7 $longitude = "0";
8 $latitude = "0";
9 $imei = "0";
10 if (isset($_POST['key'])) $key = $_POST['key'];
11 if (isset($_POST['latitude'])) $latitude = $_POST['latitude'];
12 if (isset($_POST['longitude'])) $longitude = $_POST['longitude'];
13 if (isset($_POST['imei'])) $imei = $_POST['imei'];
14
15 if ($key = "a@4K%3") {
16     $date = $script_tz ;
17     // $intdistance = intval($distance);
18     // $lat = floatval($latitude);
19     // $lon = floatval($longitude);
20     // $id_imei = intval($imei);
21
22     $conexion=new mysqli(DB_HOST,DB_USER,DB_PASSWORD,DB_DATABASE);
23     if($conexion->connect_errno){
24         echo "El sitio web está experimentado problemas";
25     }
26
27     $stmt = $conexion->prepare(" INSERT INTO tbl_detalle_dispositivo (id, lat, lon, id_imei, id_dispositivo)
28     \VALUES (NULL, '$latitude', '$longitude', '$imei', (select id from tbl_dispositivo where imei='$imei') ); ");
29     // $stmt->bind_param("ss", $distancia, $date);
30     if ($stmt->execute()){
31         echo "Ejecutado con exito";
32     }
33
34 }
35 }
36 ?>
```

## Anexo C1

### Código MainActivity.kt

```
1 package com.dipp.cntrolbike
2
3 import android.content.Context
4 import android.content.Intent
5 import android.content.pm.ActivityInfo
6 import android.net.ConnectivityManager
7 import android.os.Bundle
8 import android.widget.Toast
9 import androidx.appcompat.app.AppCompatActivity
10 import com.android.volley.AuthFailureError
11 import com.android.volley.Request
12 import com.android.volley.Response
13 import com.android.volley.toolbox.StringRequest
14 import com.android.volley.toolbox.Volley
15 import com.dipp.cntrolbike.databinding.
    ActivityMainBinding
16 import org.json.JSONObject
17
18 //import kotlinx.android.synthetic.main.activity_main
    .*
19
20
21 class MainActivity : AppCompatActivity() {
22     private lateinit var binding: ActivityMainBinding
23
24     override fun onCreate(savedInstanceState: Bundle
    ?) {
25         super.onCreate(savedInstanceState)
26
27         binding = ActivityMainBinding.inflate(
    layoutInflater)
28         setContentView(binding.root)
29         //setContentView(R.layout.activity_main)
30         requestedOrientation = ActivityInfo.
    SCREEN_ORIENTATION_UNSPECIFIED
31
32         val conexion = getSystemService(Context.
    CONNECTIVITY_SERVICE) as ConnectivityManager
33         val status = conexion.activeNetworkInfo
34         Seguridad.IdDispositivo="0"
35
```

```

36
37         binding.btnregistrar.setOnClickListener {
38             val intento1 = Intent(this, registro::
class.java)
39                 startActivity(intento1)
40                 finish()
41         }
42
43         binding.btnLogin.setOnClickListener() {
44             if(binding.editUsuario.text.trim().
isNotEmpty() || binding.editPassword.text.trim().
isNotEmpty() ){
45
46                 if (status != null && status.isConnected
) {
47                     validUsuario()
48
49                 }else{
50                     Toast.makeText(this, "Revise su conexion a
internet", Toast.LENGTH_SHORT).show()
51                 }else{
52                     Toast.makeText(this, "Credenciales
requeridas", Toast.LENGTH_SHORT).show()
53                 }
54
55             }
56         }
57     }
58
59     private fun validUsuario() {
60
61         val logUsuario = binding.editUsuario.text.
toString()
62         val logPass = binding.editPassword.text.
toString()
63         val url = "http://controlbikec.atwebpages.com
/loginUser.php"
64
65         val stringRequest = object : StringRequest(
66             Request.Method.POST, url,
67             Response.Listener<String> { response ->

```

```

68
69         if(response.isNotEmpty()){
70             val obj = JSONObject(response)
71             val user = obj.getString("
idUsuario");
72             val device = obj.getString("
idDispositivo");
73
74             Seguridad.IdUsuario=user
75
76             if (device=="null")
77             {
78                 device="0"
79             }
80             Seguridad.IdDispositivo=device
81
82             val intent =Intent(this,home::
class.java)
83             startActivity(intent)
84             finish()
85             //Toast.makeText(this, " "+
binding.editUsuario.text, Toast.LENGTH_SHORT).show()
86             }else{
87                 Toast.makeText(this,"Usuario/
contraseña invalida ó Usuario inactivo",Toast.
LENGTH_SHORT).show()
88             }
89         },
90
91         Response.ErrorListener {
92             Toast.makeText(this,"Fatal error",
Toast.LENGTH_SHORT).show()
93         })
94
95     {
96         override fun getParams(): Map<String,
String> {
97             val params = HashMap<String, String
>()
98             params["usuario"] = logUsuario
99             params["password"] = logPass

```

```

100             return params
101         }}
102         //adding request to queue
103         VolleySingleton.instance?.addToRequestQueue(
stringRequest);
104         // val requestQueue = Volley.newRequestQueue(
this)
105         //requestQueue.add(stringRequest)
106     }
107 }
108
109 class Seguridad{
110     companion object {
111         var IdUsuario = "0"
112         var IdDispositivo = "0"
113     }
114 }

```

## Anexo C2

### Código activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/
4     android"
5     xmlns:app="http://schemas.android.com/apk/res-
6     auto"
7     xmlns:tools="http://schemas.android.com/tools"
8     android:layout_width="match_parent"
9     android:layout_height="wrap_content"
10    tools:context=".MainActivity">
11
12    <ImageView
13        android:id="@+id/imageView4"
14        android:layout_width="333dp"
15        android:layout_height="281dp"
16        android:layout_marginStart="8dp"
17        android:layout_marginTop="36dp"
18        android:layout_marginEnd="8dp"
19        android:background="@drawable/button_re"
20        app:layout_constraintEnd_toEndOf="parent"
21        app:layout_constraintStart_toStartOf="parent"
22        app:layout_constraintTop_toTopOf="parent"
23        app:srcCompat="@drawable/logooficial" />
24
25    <EditText
26        android:id="@+id/editUsuario"
27        android:layout_width="match_parent"
28        android:layout_height="wrap_content"
29        android:layout_marginStart="32dp"
30        android:layout_marginTop="56dp"
31        android:layout_marginEnd="32dp"
32        android:ems="10"
33        android:hint="Email"
34        android:inputType="textPersonName"
35        app:layout_constraintEnd_toEndOf="parent"
36        app:layout_constraintHorizontal_bias="0.0"
37        app:layout_constraintStart_toStartOf="parent"
38        app:layout_constraintTop_toBottomOf="@+id/
39        imageView4" />
```

```

38
39     <EditText
40         android:id="@+id/editPassword"
41         android:layout_width="match_parent"
42         android:layout_height="wrap_content"
43         android:layout_marginStart="32dp"
44         android:layout_marginTop="28dp"
45         android:layout_marginEnd="32dp"
46         android:ems="10"
47         android:hint="Password"
48         android:inputType="textPassword"
49         app:layout_constraintEnd_toEndOf="parent"
50         app:layout_constraintHorizontal_bias="0.0"
51         app:layout_constraintStart_toStartOf="parent"
52         app:layout_constraintTop_toBottomOf="@+id/
editUsuario" />
53
54     <Button
55         android:id="@+id/btnLogin"
56         android:layout_width="match_parent"
57         android:layout_height="wrap_content"
58         android:layout_marginStart="32dp"
59         android:layout_marginTop="44dp"
60         android:layout_marginEnd="32dp"
61         android:background="@drawable/button_re_mod2"
62         android:text="Ingresar"
63         app:layout_constraintEnd_toEndOf="parent"
64         app:layout_constraintHorizontal_bias="0.0"
65         app:layout_constraintStart_toStartOf="parent"
66         app:layout_constraintTop_toBottomOf="@+id/
editPassword"
67         tools:visibility="visible" />
68
69     <Button
70         android:id="@+id/btnregistrar"
71         android:layout_width="match_parent"
72         android:layout_height="wrap_content"
73         android:layout_marginStart="32dp"
74         android:layout_marginTop="8dp"
75         android:layout_marginEnd="32dp"
76         android:background="@drawable/button_re_mod2"

```

```
77         android:text="Registrar"
78         app:layout_constraintBottom_toBottomOf="
parent"
79         app:layout_constraintEnd_toEndOf="parent"
80         app:layout_constraintHorizontal_bias="0.0"
81         app:layout_constraintStart_toStartOf="parent
"
82         app:layout_constraintTop_toBottomOf="@+id/
btnLogin"
83         app:layout_constraintVertical_bias="0.117"
84         tools:ignore="UsingOnClickInXml" />
85
86
87 </androidx.constraintlayout.widget.ConstraintLayout>
```

## Anexo C3

### Register.kt

```
1 @file:Suppress("ReplacePutWithAssignment")
2
3 package com.dipp.cntrolbike
4
5 import android.content.Context
6 import android.content.Intent
7 import android.content.pm.ActivityInfo
8 import android.net.ConnectivityManager
9 import android.os.Bundle
10 import android.widget.EditText
11 import android.widget.Toast
12 import androidx.appcompat.app.AppCompatActivity
13 import com.android.volley.AuthFailureError
14 import com.android.volley.Request
15 import com.android.volley.Request.Method.*
16 import com.android.volley.Response
17 import com.android.volley.toolbox.StringRequest
18 import com.dipp.cntrolbike.databinding.
    ActivityRegistroBinding
19 import kotlinx.android.synthetic.main.
    activity_registro.*
20 import org.json.JSONException
21 import org.json.JSONObject
22
23 class registro : AppCompatActivity() {
24
25     override fun onCreate(savedInstanceState: Bundle
    ?) {
26         super.onCreate(savedInstanceState)
27         //binding = ActivityRegistroBinding.inflate(
    layoutInflater)
28         //setContentView(binding.root)
29         setContentView(R.layout.activity_registro)
30
31
32         requestedOrientation = ActivityInfo.
    SCREEN_ORIENTATION_UNSPECIFIED
33         val conexion = getSystemService(Context.
    CONNECTIVITY_SERVICE) as ConnectivityManager
34         val status = conexion.activeNetworkInfo
35
```

```

36         btnregistrar.setOnClick1ClickListener() {
37             if (status != null && status.isConnected
) {
38                 addUser()
39             } else {
40                 Toast.makeText(this, "Revise su
41                 conexion a internet", Toast.LENGTH_LONG).show()
42             }
43         }
44     }
45
46     //adding a new record to database
47     private fun addUser() {
48         //getting the record values
49         val lUser = baseUser?.text.toString()
50         val lEmail = baseEmail?.text.toString()
51         val lPass = basePass?.text.toString()
52
53         val url = "http://controlbike.atwebpages.com/
54         index.php"
55         //creating volley string request
56         val stringRequest = object : StringRequest(
Request.Method.POST, url,
57         Response.Listener<String> { response ->
58             try {
59                 val obj = JSONObject(response)
60                 val vl_correcto = obj.getBoolean(
61                 "correcto");
62                 if (vl_correcto) {
63                     //Para exitoso
64                     Toast.makeText(
applicationContext, obj.getString("message"), Toast.
LENGTH_LONG).show();
65                     val intent = Intent(this,
MainActivity:: class.java)
66                     startActivity(intent)
67                     finish()
//Enviar a la página de
inicio

```

```

68
69         }
70         else{
71             //En caso de error
72
73         }
74
75         } catch (e: JSONException) {
76             Toast.makeText(
applicationContext, e.printStackTrace().toString(),
Toast.LENGTH_LONG).show()
77             e.printStackTrace()
78         }
79     },
80     Response.ErrorListener { volleyError ->
81         Toast.makeText(applicationContext,
volleyError.message, Toast.LENGTH_LONG).show()
82     }) {
83
84         @Throws(AuthFailureError::class)
85         override fun getParams(): Map<String,
String> {
86             val params = HashMap<String, String
>()
87             params.put("usuario", lUser)
88             params.put("correo", lEmail)
89             params.put("contrasena", lPass)
90             return params
91         }
92     }
93     //adding request to queue
94     VolleySingleton.instance?.addToRequestQueue(
stringRequest);
95 }
96 }
97
98
99
100
101
102

```

## Anexo C4

### Registo.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.appcompat.widget.LinearLayoutCompat xmlns:
  android="http://schemas.android.com/apk/res/android"
3
4     xmlns:app="http://schemas.android.com/apk/res-
  auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     app:divider="@drawable/fondobike"
9     tools:context=".registro">
10
11     <FrameLayout
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14     >
15
16         <androidx.cardview.widget.CardView
17             android:layout_width="match_parent"
18             android:layout_height="wrap_content">
19
20             <ImageView
21
22                 android:layout_width="match_parent"
23                 android:layout_height="800dp"
24                 android:scaleType="centerCrop"
25                 android:src="@drawable/fondobike" />
26
27             <LinearLayout
28                 android:layout_width="match_parent"
29                 android:layout_height="321dp"
30                 android:layout_marginTop="250dp"
31                 android:orientation="vertical">
32
33                 <EditText
34                     android:id="@+id/baseUser"
35                     android:layout_width="
36 match_parent"
37
38                     android:layout_height="48dp"
39                     android:layout_marginStart="5dp"
```

```

39         android:layout_marginEnd="5dp"
40         android:background="@drawable/
fndotxt1"
41         android:ems="10"
42         android:hint=" Usuario"
43         android:inputType="textPersonName
" />
44
45     <EditText
46         android:id="@+id/baseEmail"
47         android:layout_width="
match_parent"
48         android:layout_height="48dp"
49         android:layout_marginStart="5dp"
50         android:layout_marginTop="20dp"
51         android:layout_marginEnd="5dp"
52         android:background="@drawable/
fndotxt1"
53         android:ems="10"
54         android:hint=" Correo"
55         android:inputType="textPersonName
" />
56
57
58     <EditText
59         android:id="@+id/basePass"
60         android:layout_width="
match_parent"
61         android:layout_height="48dp"
62         android:layout_marginStart="5dp"
63         android:layout_marginTop="20dp"
64         android:layout_marginEnd="5dp"
65         android:background="@drawable/
fndotxt1"
66         android:ems="10"
67         android:hint="Contraseña"
68         android:inputType="textPassword"
/>
69
70     <Button
71         android:id="@+id/btnregistrar"

```

```

72         android:layout_width="
match_parent"
73         android:layout_height="48dp"
74         android:layout_marginStart="5dp"
75         android:layout_marginTop="20dp"
76         android:layout_marginEnd="5dp"
77         android:background="@drawable/
fndotxt1"
78         android:text="Guardar" />
79
80
81     </LinearLayout>
82
83
84     </androidx.cardview.widget.CardView>
85 </FrameLayout>
86
87 </androidx.appcompat.widget.LinearLayoutCompat>
88
89

```

## Anexo C5

```
home.kt
1 package com.dipp.cntrolbike
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.os.Handler
6 import android.os.Looper
7 import android.view.Menu
8 import android.widget.Toast
9 import androidx.appcompat.app.AppCompatActivity
10 import com.android.volley.Request
11 import com.android.volley.Response
12 import com.android.volley.toolbox.StringRequest
13 import com.google.android.gms.maps.
    CameraUpdateFactory
14 import com.google.android.gms.maps.GoogleMap
15 import com.google.android.gms.maps.OnMapReadyCallback
16 import com.google.android.gms.maps.SupportMapFragment
17 import com.google.android.gms.maps.model.LatLng
18 import com.google.android.gms.maps.model.
    MarkerOptions
19 import kotlinx.android.synthetic.main.activity_home.*
20 import org.json.JSONException
21 import org.json.JSONObject
22 import java.util.*
23
24 class home : AppCompatActivity(), OnMapReadyCallback {
25     private lateinit var map: GoogleMap
26     val handler = Handler(Looper.getMainLooper())
27
28
29     override fun onCreate(savedInstanceState: Bundle
    ?) {
30         super.onCreate(savedInstanceState)
31         setContentView(R.layout.activity_home)
32         createFragment()
33
34         btnDispositivos.setOnClickListener() {
35             val intent = Intent(this, selector:: class
                .java)
36                 startActivity(intent)
37                 finish()
```

```

38         }
39
40
41     }
42     private fun createFragment(){
43         val mapFragment :SupportMapFragment=
supportFragmentManager.findFragmentById(R.id.map) as
SupportMapFragment
44         mapFragment.getMapAsync(this)
45     }
46
47
48
49     override fun onCreateOptionsMenu(menu: Menu):
Boolean {
50         // Inflate the menu; this adds items to the
action bar if it is present.
51         menuInflater.inflate(R.menu.menu_main, menu)
52         return true
53     }
54
55     override fun onMapReady(googleMap: GoogleMap) {
56         map =googleMap
57
58         //Obtener marcaciones del dispositivo
59         //Actualizacion cada 30seg = 30000 ms
60         val timer = Timer()
61
62
63         timer.scheduleAtFixedRate( object : TimerTask
() {
64             override fun run() {
65                 getDatos(map,Seguridad.IdUsuario,
Seguridad.IdDispositivo)
66
67                 if (Seguridad.IdDispositivo=="0")
68                 {
69                     timer.cancel()
70                 }
71             }
72         }, 0, 30000)

```

```

73
74     }
75     private fun createMarcker(latitud: Double,
76     longitud: Double, texto:String ){
76         val coordinate =LatLng(latitud, longitud)
77         val marker =MarkerOptions().position(
78         coordinate).title(texto)
78         map.addMarker(marker)
79         map.animateCamera(CameraUpdateFactory.
79         newLatLngZoom(coordinate,18f),4000,null)
80     }
81
82
83
84     private fun getDatos(google:GoogleMap,
85     idUsuario:String, idDispositivo:String) {
85         val url = "http://controlbike.atwebpages.com
86         /getDatos.php"
87         val stringRequest = object : StringRequest(
88         Request.Method.POST, url,
89         Response.Listener<String> { response ->
90         try {
91         val obj = JSONObject(response)
92         val vl_correcto = obj.getBoolean
93         ("correcto");
94         if (vl_correcto) {
95         Toast.makeText(this,"
96         Actualizando rutas...",Toast.LENGTH_SHORT).show()
97         google.clear()
98         //SERIALIZADO DE CAMPOS
98         var respuesta = Respuesta(
99         obj.toString())
100         //GRAFICAR EN LA MARCA
100         if (respuesta.correcto.
101         toBoolean())
102         {
102         //Iteracion de los
103         registros marcados para graficar en el mapa
103         respuesta.message?.

```

```

103 forEach { msg ->
104                                     createMarcker(msg.
lat, msg.lon, msg.fecha)
105                                     }
106                                     }
107                                     }
108                                     else{
109                                     Toast.makeText(this, obj.
getString("message"), Toast.LENGTH_SHORT).show()
110                                     }
111
112                                     } catch (e: JSONException) {
113                                     Toast.makeText(
applicationContext, e.printStackTrace().toString(),
Toast.LENGTH_LONG).show()
114                                     e.printStackTrace()
115                                     }
116                                     },
117
118                                     Response.ErrorListener {
119                                     Toast.makeText(this, "Fatal error",
Toast.LENGTH_SHORT).show()
120                                     })
121
122                                     {
123                                     override fun getParams(): Map<String,
String> {
124                                     val params = HashMap<String, String
>()
125                                     params["idUser"] = idUsuario
126                                     params["idDevice"] = idDispositive
127                                     return params
128                                     }}
129                                     //adding request to queue
130                                     VolleySingleton.instance?.addToRequestQueue(
stringRequest);
131
132                                     }
133 }
134
135 class Respuesta(json: String) : JSONObject(json) {

```

```

136     val correcto: String? = this.optString("correcto
137     ")
137     val message = this.optJSONArray("message")
138     ?.let { 0.until(it.length()).map { i -> it.
optJSONObject(i) } } // returns an array of
JSONObject
139     ?.map { Marcacion(it.toString()) } //
transforms each JSONObject of the array into call
140 }
141
142 class Marcacion(json: String) : JSONObject(json) {
143     val id = this.optInt("id")
144     val lat: Double = this.optDouble("lat")
145     val lon: Double = this.optDouble("lon")
146     val fecha: String = this.optString("fecha")
147 }

```

## Anexo C6

### Código activity\_home.xml

File - E:\proyectotes\cntrlbike\app\src\main\res\layout\activity\_home.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <RelativeLayout xmlns:android="http://schemas.android
4     .com/apk/res/android"
5     xmlns:app="http://schemas.android.com/apk/res-
6     auto"
7     xmlns:tools="http://schemas.android.com/tools"
8     android:layout_width="match_parent"
9     android:layout_height="match_parent"
10    tools:context=".home">
11
12    <androidx.fragment.app.FragmentContainerView
13        android:id="@+id/map"
14        android:name="com.google.android.gms.maps.
15        SupportMapFragment"
16        android:layout_width="match_parent"
17        android:layout_height="match_parent"
18        tools:context=".home">
19
20    </androidx.fragment.app.FragmentContainerView>
21
22    <com.google.android.material.floatingactionbutton
23        .FloatingActionButton
24        android:id="@+id/btnDispositivos"
25        android:layout_width="wrap_content"
26        android:layout_height="wrap_content"
27        android:layout_gravity="end|bottom"
28        android:layout_marginStart="16dp"
29        android:layout_marginTop="16dp"
30        android:layout_marginEnd="16dp"
31        android:layout_marginBottom="16dp"
32        android:clickable="true"
33        app:layout_constraintBottom_toBottomOf="
34        parent"
35        app:layout_constraintEnd_toEndOf="parent"
36        app:srcCompat="@drawable/bike" />
37
38 </RelativeLayout>
```

## Anexo C7

### Código selector.kt

```
1 package com.dipp.cntrolbike
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.view.View
7 import android.widget.AdapterView
8 import android.widget.AdapterView.OnItemClickListener
9 import android.widget.AdapterView.OnItemClickListener
10 import android.widget.AdapterView.OnItemClickListener
11 import android.widget.AdapterView.OnItemClickListener
12 import android.widget.AdapterView.OnItemClickListener
13 import android.widget.AdapterView.OnItemClickListener
14 import android.widget.AdapterView.OnItemClickListener
15 import android.widget.AdapterView.OnItemClickListener
16 import android.widget.AdapterView.OnItemClickListener
17 import android.widget.AdapterView.OnItemClickListener
18 import android.widget.AdapterView.OnItemClickListener
19 import android.widget.AdapterView.OnItemClickListener
20 import android.widget.AdapterView.OnItemClickListener
21
22 //Obtiene los dispositivos registrados de un usuario
23 // (list)
24
25 class selector : AppCompatActivity() {
26     override fun onCreate(savedInstanceState: Bundle
27     ?) {
28         super.onCreate(savedInstanceState)
29         setContentView(R.layout.activity_selector)
30         getDispositivos()
31         btnMapa.setOnClickListener() {
32             val intent =Intent(this,home:: class.java
33             )
34             startActivity(intent)
35             finish()
36         }
37     }
38     private fun getDispositivos() {
39         var mListview = findViewById<ListView>(R.id.
40         lstDevices)
```

```

37         val idUsuario = Seguridad.IdUsuario
38         val url = "http://controlbike.atwebpages.com/
getDevices.php"
39         var arrayAdapter: ArrayAdapter<*>
40
41         if (Seguridad.IdDispositivo!="0")
42         {
43             val stringRequest = object :
StringRequest(
44                 Request.Method.POST, url,
45                 Response.Listener<String> { response
->
46                     try {
47                         val obj = JSONObject(response
)
48                         val vl_correcto = obj.
getBoolean("correcto");
49                         if (vl_correcto) {
50                             Toast.makeText(this, "
Obteniendo dispositivos...", Toast.LENGTH_SHORT).show
()
51                             val typeToken = object :
TypeToken<List<*>>() {}.type
52                             val registros = Gson().
fromJson<List<*>>(obj.getString("message"), typeToken
)
53                             val device =
mutableListOf("");
54                             registros?.forEach { msg
->
55                                 device.add(msg.
toString());
56                             }
57                             arrayAdapter =
ArrayAdapter(this,
58                                 android.R.layout.
simple_list_item_1, device);
59                             arrayAdapter
mListView.adapter =
60                                 //Evento click item de la
lista

```

```

61         mListView.
setOnItemClickListener(){parent, view,
62
        position, id->
63             try{
64                 val trama =
JSONObject(parent.getItemAtPosition(position).
toString());
65                 val
idDispositivo = trama.optString("id");
66                 val
nombreDispositivo = trama.optString("nombre");
67
68                 Seguridad.
IdDispositivo=idDispositivo
69
70                 Toast.makeText(
applicationContext,
71                 "Dispositivo
seleccionado: " + nombreDispositivo, Toast.
LENGTH_LONG)
72                 .show()
73             } catch (e:
Exception) {
74
75                 }
76             }
77         }
78         else{
79             Toast.makeText(this, obj.
getString("message"), Toast.LENGTH_SHORT).show()
80         }
81
82     } catch (e: JSONException) {
83         Toast.makeText(
applicationContext, e.printStackTrace().toString(),
Toast.LENGTH_LONG).show()
84         e.printStackTrace()
85     }
86 },
87

```

```

88             Response.ErrorListener {
89                 Toast.makeText(this, "Fatal error
90             ", Toast.LENGTH_SHORT).show()
91             })
92         {
93             override fun getParams(): Map<String
94             , String> {
95                 val params = HashMap<String,
96                 String>()
97                 params["idUser"] = idUsuario
98                 return params
99             }}
100             //adding request to queue
101             VolleySingleton.instance?.
102             addToRequestQueue(stringRequest);
103             // val requestQueue = Volley.
104             newRequestQueue(this)
105             //requestQueue.add(stringRequest)
106         }
107     else
108     {
109         Toast.makeText(this, "No se encontraron
110         dispositivos para el usuario ingresado!", Toast.
111         LENGTH_SHORT).show()
112     }
113 }

```

## Anexo C8

### Selector.xml

File - E:\proyectotes\cntrolbike\app\src\main\res\layout\activity\_selector.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android
   .com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-
   auto"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:background="#EEE">
7
8
9     <ListView
10        android:id="@+id/lstDevices"
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content" >
13    </ListView>
14
15    <com.google.android.material.floatingactionbutton
   .FloatingActionButton
16        android:id="@+id/btnMapa"
17        android:layout_width="wrap_content"
18        android:layout_height="wrap_content"
19        android:layout_gravity="end|bottom"
20        android:layout_marginStart="270dp"
21        android:layout_marginTop="406dp"
22        android:layout_marginEnd="16dp"
23        android:layout_marginBottom="16dp"
24        android:clickable="true"
25        app:layout_constraintBottom_toBottomOf="
   parent"
26        app:layout_constraintEnd_toEndOf="parent"
27        app:srcCompat="@drawable/mapa" />
28
29
30
31 </RelativeLayout>
```

## Anexo C9

### AndroidManifest.xml

File - E:\proyectotes\cntrolbike\app\src\main\AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/
  apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     package="com.dipp.cntrolbike">
5
6     <uses-permission android:name="android.permission
  .INTERNET" />
7     <uses-permission android:name="android.permission
  .ACCESS_NETWORK_STATE" />
8
9     <application
10        android:name=".VolleySingleton"
11        android:allowBackup="true"
12        android:icon="@mipmap/ic_launcher"
13        android:label="@string/app_name"
14        android:roundIcon="@mipmap/ic_launcher_round"
15        android:supportsRtl="true"
16        android:theme="@style/Theme.AppCompat.
  DayNight.NoActionBar"
17        android:usesCleartextTraffic="true"
18        tools:targetApi="m">
19        <activity
20            android:name=".selector"
21            android:exported="false" />
22        <activity
23            android:name=".registro"
24            android:exported="false" />
25
26        <meta-data
27            android:name="com.google.android.geo.
  API_KEY"
28            android:value="@string/google_maps_key"
29        />
30
31        <activity
32            android:name=".home"
33            android:exported="false" />
34        <activity
35            android:name=".MainActivity"
36            android:exported="true">
```

Page 1 of 2

```
36         <intent-filter>
37             <action android:name="android.intent.
38                 action.MAIN" />
39             <category android:name="android.
40                 intent.category.LAUNCHER" />
41         </intent-filter>
42     </activity>
43 </application>
44 </manifest>
```

## Anexo C10

### Build.gradle

File - E:\proyectotes\cntrolbike\app\build.gradle

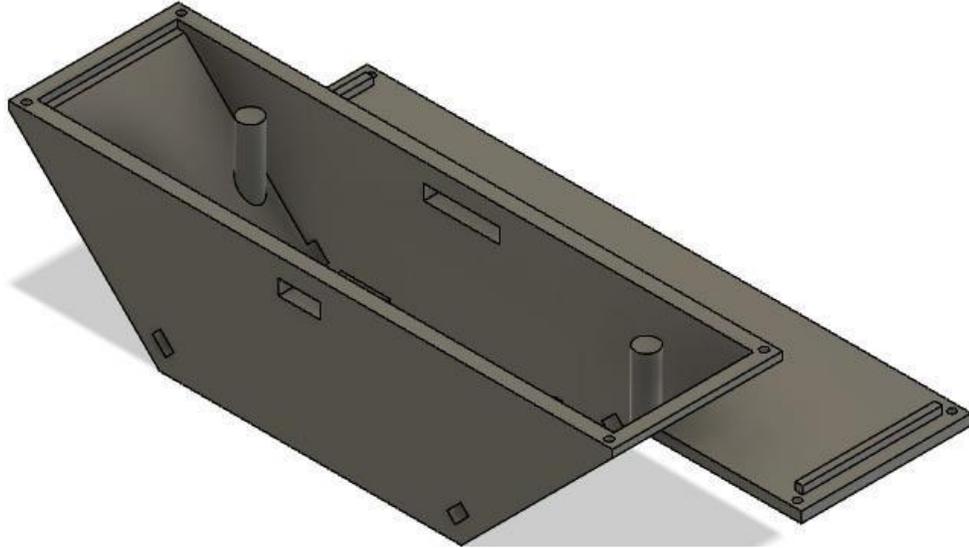
```
1 plugins {
2     id 'com.android.application'
3     id 'org.jetbrains.kotlin.android'
4     id 'kotlin-android-extensions'
5 }
6
7 android {
8     compileSdk 32
9
10    defaultConfig {
11        applicationId "com.dipp.cntrolbike"
12        minSdk 21
13        targetSdk 32
14        versionCode 1
15        versionName "1.0"
16
17        testInstrumentationRunner "androidx.test.
runner.AndroidJUnitRunner"
18    }
19
20    buildTypes {
21        release {
22            minifyEnabled false
23            proguardFiles getDefaultProguardFile('
proguard-android-optimize.txt'), 'proguard-rules.pro'
24        }
25    }
26    compileOptions {
27        sourceCompatibility JavaVersion.VERSION_1_8
28        targetCompatibility JavaVersion.VERSION_1_8
29    }
30
31    buildFeatures{
32        viewBinding = true
33    }
34
35
36    kotlinOptions {
37        jvmTarget = '1.8'
38    }
39
```

Page 1 of 2

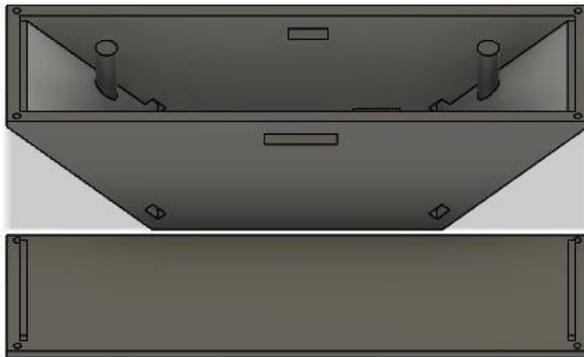
```
40 }
41
42 dependencies {
43
44     implementation 'androidx.core:core-ktx:1.8.0'
45     implementation 'androidx.appcompat:appcompat:1.4.
46     2'
47     implementation 'com.google.android.material:
48     material:1.6.1'
49     implementation 'androidx.constraintlayout:
50     constraintlayout:2.1.4'
51     implementation 'com.google.android.gms:play-
52     services-maps:18.0.2'
53     implementation 'com.android.volley:volley:1.2.1'
54     testImplementation 'junit:junit:4.13.2'
55     androidTestImplementation 'androidx.test.ext:
56     junit:1.1.3'
57     androidTestImplementation 'androidx.test.espresso
58     :espresso-core:3.4.0'
59 }
```

## Anexo D1

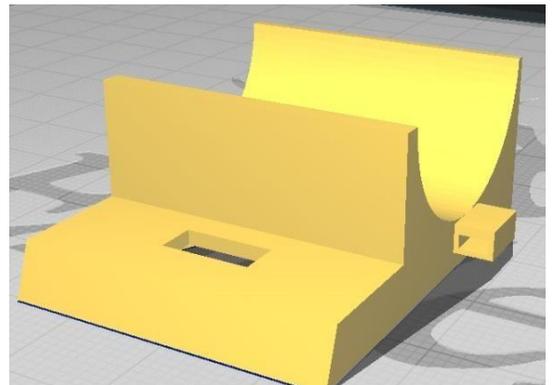
Diseño de la carcasa para el prototipo



*Ilustración 1*

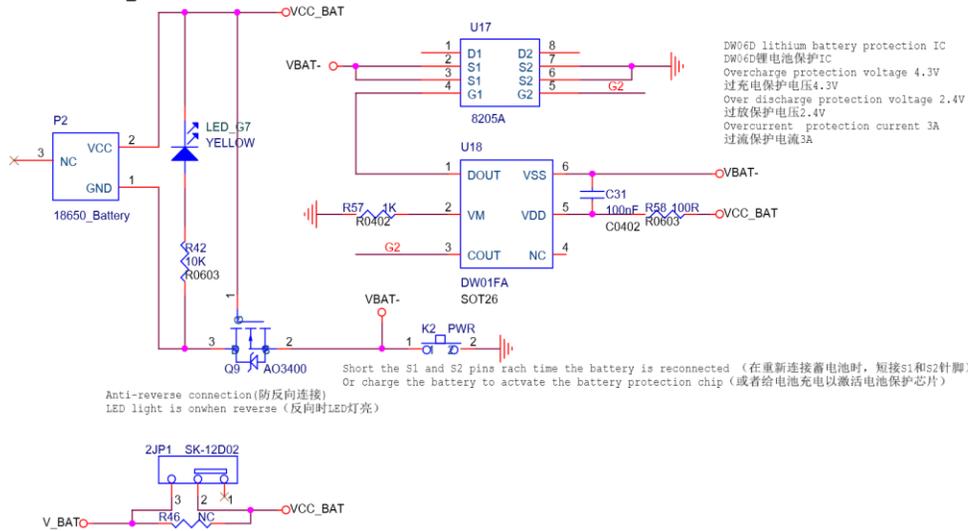


*Ilustración 2*

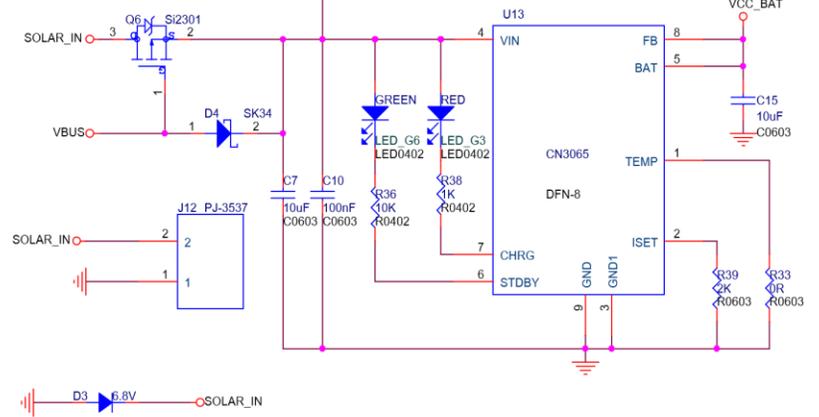


*Ilustración 3*

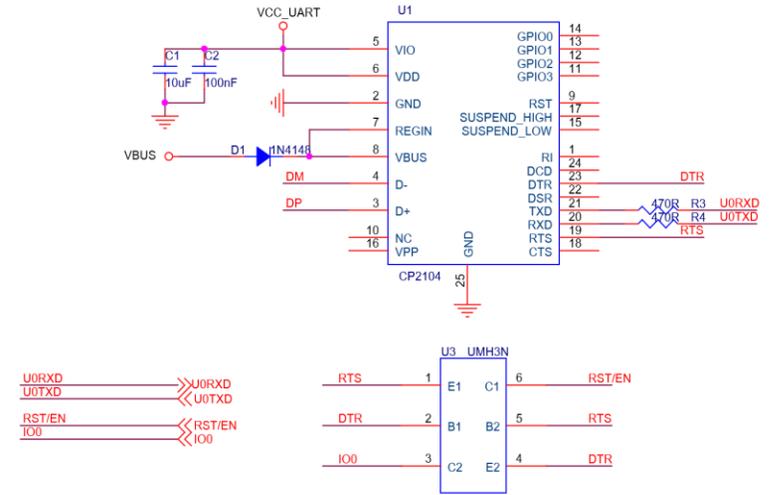
### Battery Interface



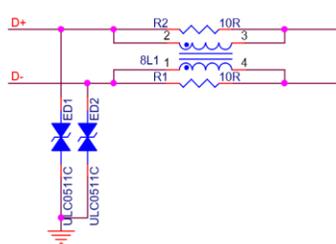
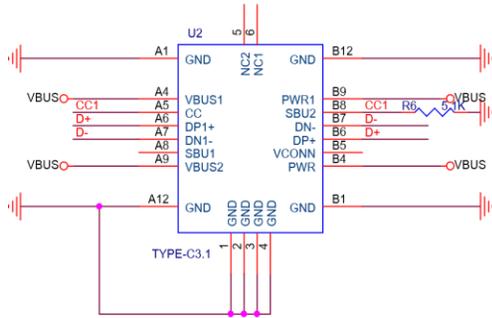
### 输入电压 4.4-6V



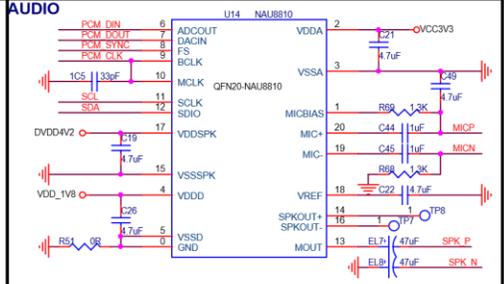
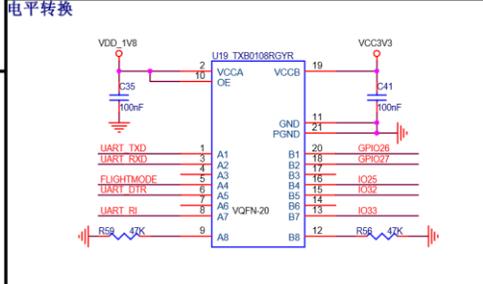
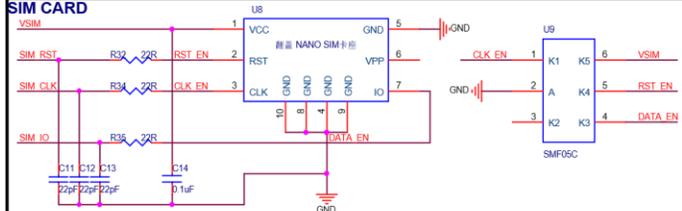
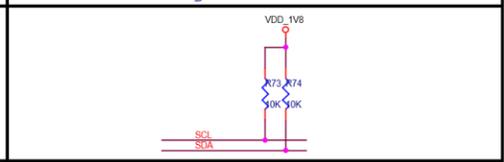
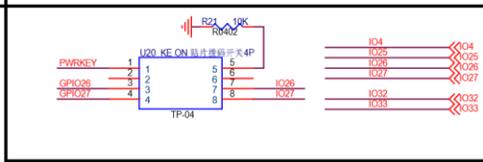
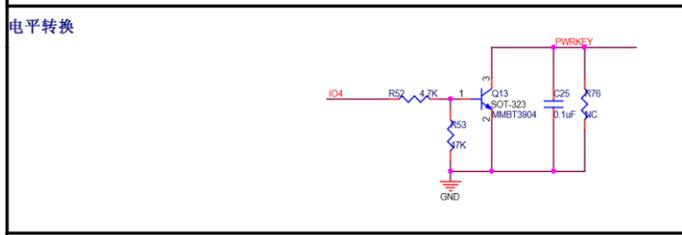
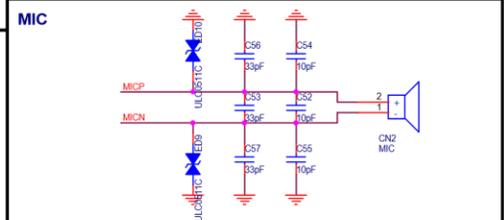
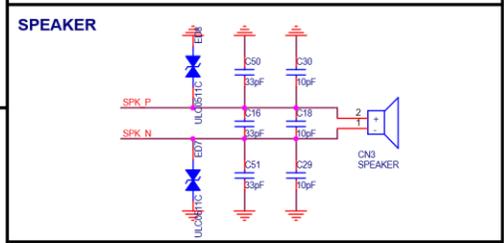
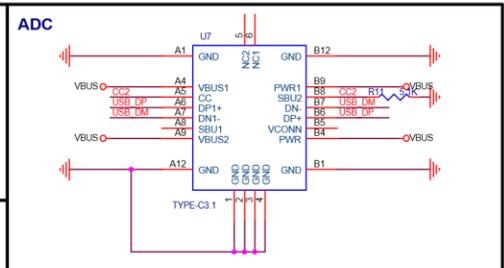
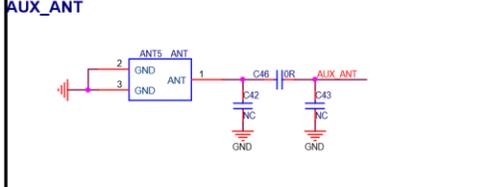
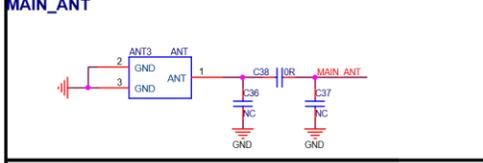
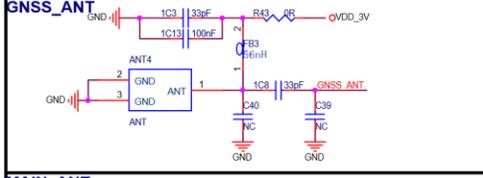
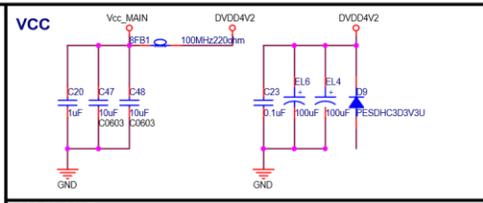
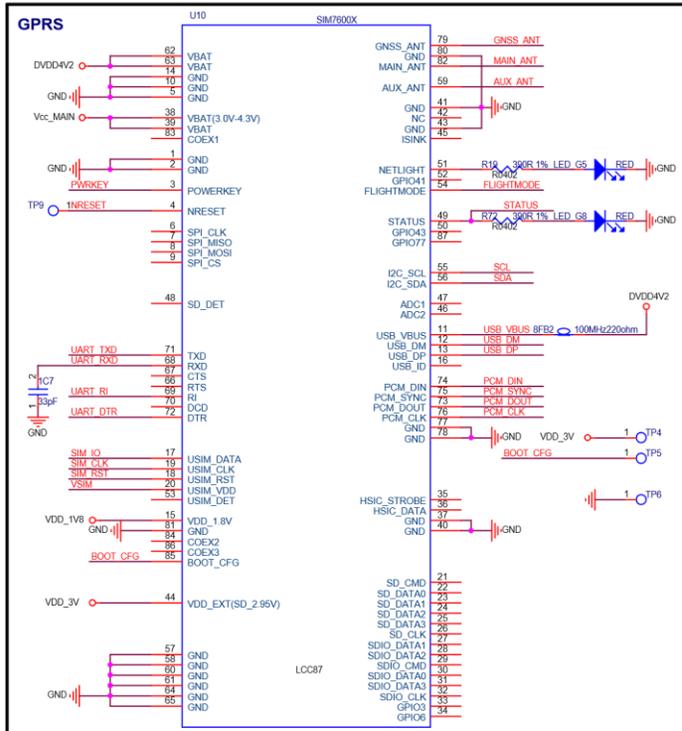
### UART



### USB-TYPE-C 3.0

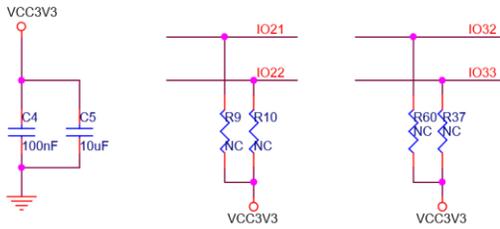
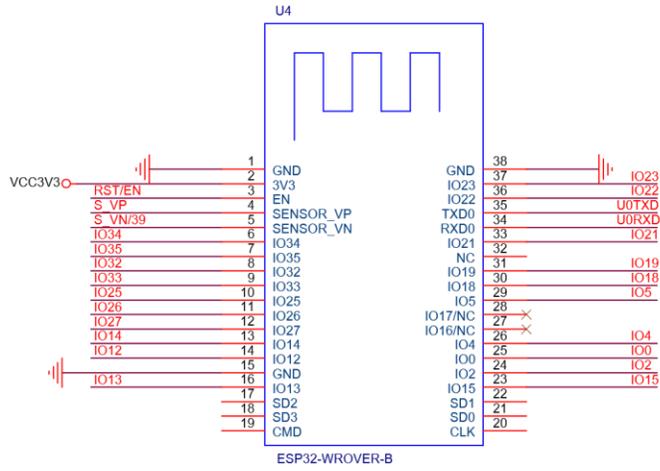


Title		
T54_V1.2		
Size	Document Number	Rev
B	POWER-INPUT	V1.2
Date:	Monday, October 12, 2020	Sheet 1 of 4

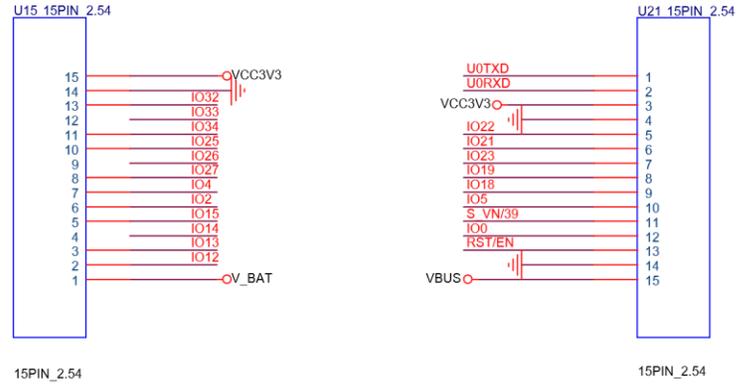


File	T54_V1.2		
Size	Document Number		Rev
C	<Doc>		V1.2
Date	Monday, October 12, 2020	Sheet	1 of 1

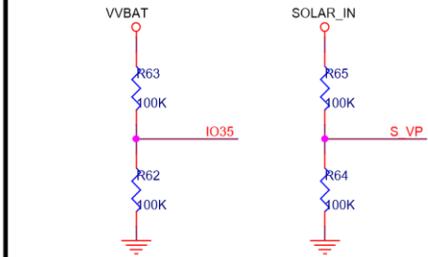
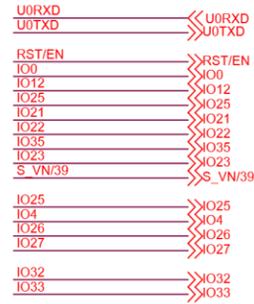
### Core



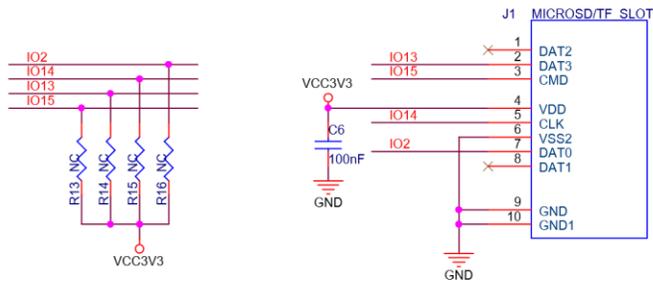
### Connector



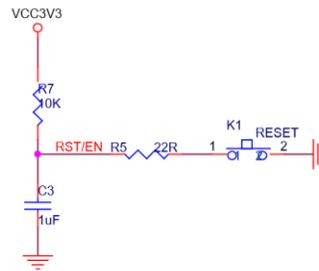
### TEST



### SD CARD



### RST-KEY

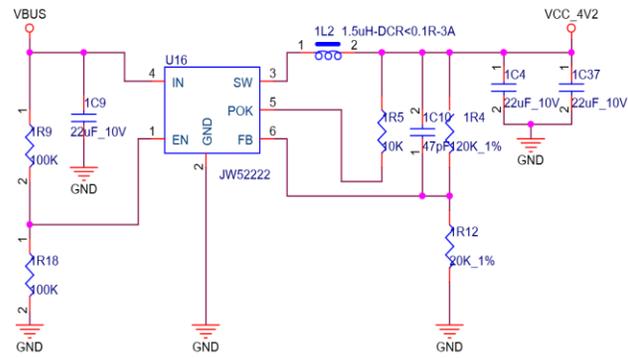


### LED

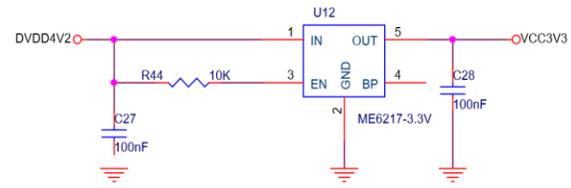


Title		T54_V1.2	
Size	B	Document Number	ESP32
Date:	Monday, October 12, 2020	Sheet	2 of 1

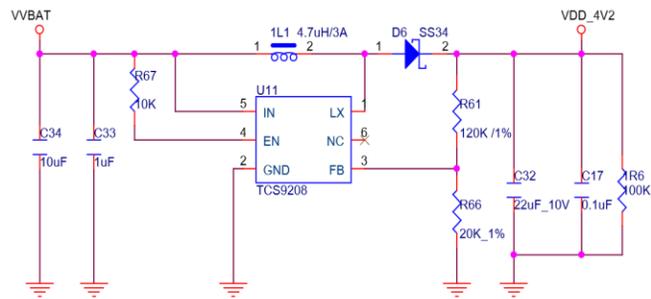
### VBUS-VCC4.2V



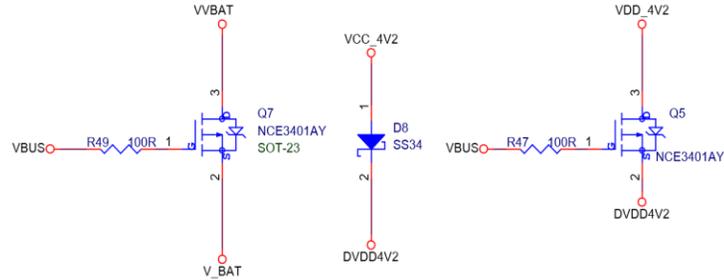
### 4.2V-3.3V



### VBAT-VDD4.2V



### VDD4.2V/VCC\_4V2---DVDD4V2



Title		
T54_V1.2		
Size B	Document Number power	Rev V1.2
Date:	Monday, October 12, 2020	Sheet 4 of 4