

PONTIFICIA UNIVERSIDAD JAVERIANA

TRABAJO DE GRADO

INDICADOR EN VIVO SOBRE ESTADOS DE CONDUCCIÓN EN BUSES
DE TRANSMILENIO

Daniel Esteban González Zuluaga

Carlos José Medina Giner

Director:
Ing. Daniel Jaramillo Ramírez. PhD

Julio, 2020

Agradecimientos

Gracias a Dios por sus bendiciones a lo largo de este camino, al Ing Daniel Jaramillo por su apoyo y orientación profesional para obtener este logro. Agradecemos a todas aquellas personas que con su carisma y paciencia han puesto su granito de arena, principalmente a nuestras familias por ser un pilar fundamental y habernos apoyado incondicionalmente pese a las adversidades que se hayan presentado. Asimismo a nuestros profesores que siempre han estado transmitiéndonos sus conocimientos y experiencia para formar profesionales y personas de valor.

Índice general

1. Introducción	6
1.1. Descripción del Problema	6
1.2. Justificación	6
1.3. Organización	7
1.4. Objetivos	7
1.4.1. Objetivo general	7
1.4.2. Objetivos específicos	7
2. Marco teórico	8
2.1. Conceptos y principios fundamentales	8
2.1.1. IoT	8
2.1.2. Raspberry Pi	9
2.1.3. Firebase	11
2.1.4. Acelerómetro y aceleración	12
2.1.5. Acelerómetro (MPU6050)	13
2.1.6. GPRS	14
2.1.7. Aplicación Web	15
3. Desarrollo	17
3.1. Antecedentes	17
3.2. Montaje	17
3.3. Descripción general	19
3.4. Parámetros del sistema	20
3.4.1. Clasificación de riesgos	20
3.4.2. Costo de envío	21
3.5. Descripción de bloques e interfaces	23
3.5.1. Bloque de Adquisición de señal	24
3.5.2. Bloque de procesamiento de la señal	25
3.5.3. Bloque de Comunicaciones	26
3.5.4. Bloque Servidor	31
3.5.5. Bloque Presentación de la Información	32
4. Protocolo de pruebas	36
4.1. Bloque de adquisición de señales	36
4.2. Bloque procesamiento de datos	37
4.3. Bloque Comunicación	42
4.4. Presentación de la información	43
4.5. Condiciones de diseño, implementación y evaluación	46
4.5.1. Disponibilidad de espacio físico	46
4.5.2. Equipo	46
4.5.3. Software	46

4.6. Calibración del sistema	47
5. Funcionamiento del sistema	48
6. Conclusiones y recomendaciones	50
7. Bibliografia	52
8. Anexos	55

Índice de figuras

2.1.	Arquitectura por Niveles	9
2.2.	Raspberry Pi 3 B+	10
2.3.	Especificaciones Raspberry Pi 3B +.[11]	10
2.4.	GPIO Raspberry Pi 3 B+. [12]	11
2.5.	Sistema físico masa-resorte	12
2.6.	Acelerómetro MPU6050.	13
2.7.	Frame PPP[19]	15
2.8.	Aplicación Web [23]	16
2.9.	Diagrama Aplicación Web	16
3.1.	Diseño de caja.	18
3.2.	Diseño de caja 2.	18
3.3.	Montaje Final.	19
3.4.	Modelo Conceptual del Proyecto	20
3.5.	Correlación de muestras vs Veces.	21
3.6.	Almacenamiento en Firebase por Bytes.	22
3.7.	Cantidad de bytes enviados al final del trayecto	22
3.8.	Cantidad de bytes enviados durante el trayecto.	23
3.9.	Diagrama de bloques general.	23
3.10.	Protocolo I2C [30].	24
3.11.	Dirección bus I2c	25
3.12.	Comparación eje x con filtro 6 y sin filtro	26
3.13.	Modem Gprs Raspberry Pi	26
3.14.	Montaje Modulo GPRS Raspberry Pi	27
3.15.	Puertos Disponibles Raspberry Pi3	28
3.16.	Habilitación de la UART	28
3.17.	Configuración PPP	29
3.18.	Creación de la conexión y características de la red	30
3.19.	Registro de cantidad de riesgos por minuto.	31
3.20.	Registro con arreglo de muestras.	31
3.21.	Tiempo de muestreo de datos.	31
3.22.	Registro de muestras totales en el recorrido.	32
3.23.	Registro de tiempo toma de muestras.	32
3.24.	Registros de duración de riesgo.	32
3.25.	Gráfico Obtenido	34
4.1.	Validación eje x	36
4.2.	Validación eje y	36
4.3.	Validación eje z	37
4.4.	Riesgo medio y alto.	38
4.5.	Histogramas porcentaje Eje x.	38

4.6.	Histogramas porcentaje riesgo medio Eje x.	39
4.7.	Histogramas porcentaje riesgo alto Eje x.	39
4.8.	Histogramas porcentaje Eje y.	40
4.9.	Histogramas porcentaje riesgo medio Eje y.	40
4.10.	Histogramas porcentaje riesgo alto Eje y.	41
4.11.	Histogramas porcentaje Eje Z.	41
4.12.	Histogramas porcentaje riego medio Eje Z.	42
4.13.	Histogramas porcentaje riego alto Eje Z.	42
4.14.	Ping Servidor Web	43
4.15.	Datos en el Servidor Web	44
4.16.	Datos Extraídos del servidor	45
4.17.	Gráficos Obtenidos navegador Web	45
4.18.	Gráficos Obtenidos navegador Web	46
4.19.	Calibración del sistema.	47
5.1.	Dashboard Mostrando eje x	48
5.2.	Dashboard Clasificación	49

Capítulo 1

Introducción

1.1. Descripción del Problema

El sistema de transporte Transmilenio tiene como uno de sus objetivos principales, mejorar la calidad del servicio prestado a sus usuarios y en su momento buscar una solución de los problemas causados por la guerra del centavo, esta guerra se refería a una disputa permanente entre los buses que prestaban el servicio por llevar más pasajeros. Transmilenio está inspirado en referencias exitosas, como lo han sido las ciudades de Curitiba y Porto Alegre Brasil. La infraestructura del sistema utiliza los carriles centrales de las vías principales de la ciudad, tuvo mejoras con respecto a los anteriores buses en el aspecto ecológico y disminuyendo la frecuencia de accidentalidad. Esta es una muy buena alternativa al uso del auto individual, pero hay veces que los usuarios se ven afectados en comodidad y seguridad por la forma de conducción del mismo [1].

Los sistemas de transporte SITP y Transmilenio son un gran foco de críticas por la forma en la que se desempeñan dentro del medio urbano de la ciudad. Tales eventos físicos como la aceleración, desaceleración cambios abruptos de carril y negligencia a la hora de seguir las señalizaciones en la vía, hacen que en gran medida se incrementen los casos de accidentes siendo estos causantes de congestión vehicular e incomodidades en los pasajeros. Adicionalmente a lo anterior, las personas que viajan de pie ante una situación de estas son más sensibles a sufrir caídas y golpes, especialmente personas con alguna discapacidad o de la tercera edad.

Con el fin de ayudar a informar posibles casos de imprudencia mediante la adquisición de datos que determinen la manera de conducción. Se vio viable desarrollar un proyecto que permita identificar dinámicas de riesgo sobre la forma de conducir un vehículo integral de transporte público, donde se pueda analizar y reportar de forma rápida. Es decir, implementar IoT haciendo uso de un sistema embebido, transmitiendo señales y almacenando los datos recibidos en un servidor, finalmente se presentarán los datos en una página web de manera gráfica para poder observar mejor los detalles analizados.

1.2. Justificación

En el mundo mueren aproximadamente 1.3 millones de personas al año por accidentes de tránsito [2]. En los articulados del servicio integrado del transporte público (SITP) en un mes se recibieron 186 denuncias de mal servicio [3], se recibieron un total de 5329 quejas y reclamos en enero de 2015 [4], mientras que en septiembre de 2018 se presentaron 7282 [5]. Es decir que ha aumentado un 26% el sentimiento de inconformidad. La prudencia en la forma de conducir es importante si se espera llegar a disminuir el descontento en los usuarios. Según un reporte de El Tiempo se registran una hora o más de espera en las paradas y trancones, además de buses llenos [6]. De acuerdo con lo anterior el proyecto buscará publicar e indicar cifras, antecedentes y/o tendencias obtenidas acerca de cómo se conduce, procurando en un futuro poder adquirir un mayor volumen de información sobre la conducción en el casco urbano de la ciudad. El transporte público está pensado como una sucesión de paradas en la que los usuarios se suben y se

bajan de acuerdo a unas necesidades particulares. Dentro de cada uno de estos transportes o articulados de servicio público pueden viajar personas tanto sentadas como de pie, por lo que estas personas estarán sujetas a diferentes fuerzas físicas que en dada situación puedan afectar su integridad y la de las personas que van con ellos [7].

La calidad de servicio que se presta a los usuarios de transporte público se ve afectada en parte por la forma de conducir, pueden sufrir caídas por cambios bruscos de carril o frenado, aumentando sentimientos de incomodidad en los pasajeros, debido a esto se vio viable el desarrollo de un proyecto que permita informar en un corto intervalo de tiempo, cómo están conduciendo los vehículos del servicio público en Bogotá , es decir, analizar conductas que generen riesgos desde un marco de vista físico . Los datos serán trasmítidos de tal manera que permita indagar la forma cómo se sienten los pasajeros y cómo se conduce en ese momento. La información estará enfocada en aceleraciones, jalones y cambios de carril.

1.3. Organización

Capítulo 1: Introducción, se realiza una breve descripción de la problemática que se desea solucionar con el proyecto realizado y los objetivos planteados.

Capítulo 2: Marco teórico, se presentan los conceptos y principios fundamentales que permiten entender el proyecto desarrollado y sus antecedentes.

Capítulo 3: Desarrollo, se muestra la descripción general de la solución propuesta, incluyendo desarrollo hardware y software, además de la descripción del funcionamiento de los bloques.

Capítulo 4: Protocolo de pruebas, se analizaron los datos obtenidos teniendo en cuenta cada uno de los bloques concebidos y a su vez, se describió el proceso para calibrar inicialmente el sistema.

Capítulo 5: Funcionamiento del sistema, se evaluó el correcto funcionamiento del sistema

Capítulo 6: Conclusiones y recomendaciones, se detallaron los conocimientos adquiridos, los logros alcanzados y las dificultades encontradas a lo largo del desarrollo del proyecto.

Capítulo 7: Bibliografía.

Capítulo 8: Anexos, corresponde a un link donde se encuentra el repositorio con los códigos del proyecto.

1.4. Objetivos

1.4.1. Objetivo general

Adquirir, procesar y transmitir información en tiempo real, mediante el uso de un ordenador de tamaño reducido para analizar la calidad de conducción del transporte público.

1.4.2. Objetivos específicos

Los objetivos específicos que permiten el cumplimiento del objetivo general son:

1. Adquirir señales que nos permita determinar el comportamiento de la conducción.
2. Conversión de las señales adquiridas en datos, extrayendo de aquellas señales sus respectivas métricas.
3. Transmisión y almacenamiento de aquellas métricas que permiten la evaluación de la calidad de conducción.
4. Analizar y evaluar los datos obtenidos, presentando estos datos en diferentes representaciones gráficas que permitan concluir acerca de la clase de conducción.

Capítulo 2

Marco teórico

2.1. Conceptos y principios fundamentales

2.1.1. IoT

IoT (*Internet of Things*) se refiere a la red o conjunto de objetos que incluyen sensores integrados, software y otras tecnologías de comunicación en su estructura, y que tienen como objetivo conectarse e intercambiar datos con otros sistemas o dispositivos por medio de Internet, ya sea para análisis, recolección u monitoreo. Estos dispositivos actualmente se encuentran en la cotidianidad, desde objetos domésticos y de uso personal, hasta elementos industriales [8].

Características fundamentales IoT

Las características mas relevantes de IoT son las siguientes [9]:

- Interconectividad: premisa que enuncia que bajo el contexto de IoT dice que todo puede estar interconectado con la infraestructura mundial de la información y la comunicación.
- Servicios en relación con IoT: IoT es adecuado para suministrar servicios relacionados con los objetos dentro de las limitaciones de los mismos, como protección de la privacidad y coherencia entre el objeto físico y el objeto virtual
- Heterogeneidad: los dispositivos IoT son de carácter heterogéneos ya que se basan en diferentes plataformas de hardware y redes. Son capaces de interactuar con otros dispositivos o plataformas de servicios a través de una diversidad de redes.
- Cambios Dinámicos: el estado actual de los dispositivos es capaz de variar, es decir pasar del modo reposo al modo activo o conectado y/o desconectado esto aplica igual manera al objeto físico como la ubicación y su velocidad. Asimismo el numero de dispositivos o nodos pueden variar dinámicamente.
- Escala Enorme: la cantidad de dispositivos o nodos que ha de gestionarse y que se comunican entre si puede ser de hecho un orden de magnitud mayor que el numero de dispositivos conectados actualmente al internet. Para abordar un diseño debemos seguir un proceso por donde la información fluye de un medio físico a un medio virtual. A lo anterior se ilustra en el siguiente diagrama que muestra una arquitectura IoT por niveles sugerido por IBM [10]:

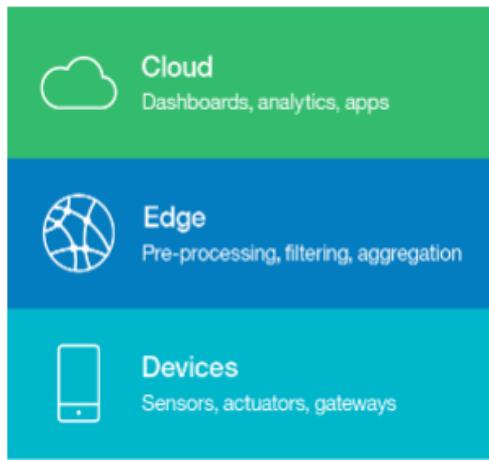


Figura 2.1: Arquitectura por Niveles.

- **Capa de dispositivos:** Los componentes pertenecientes a esta capa incluyen sensores físicos y actuadores que están conectados entre sí. Es importante tener en cuenta que los actuadores y los sensores no se consideran por sí mismos inteligentes muchas veces teniendo estos muy poca capacidad de procesamiento. Para esto los dispositivos gateway juegan un rol fundamental en el proceso de adquisición de datos.
- **Capa edge:** Este se relaciona con los servicios de analítica y de preprocesado que se ubican como lo dice su nombre al límite de la red. Este tipo de analítica ocurre en tiempo real o casi tiempo real al procesar el flujo de datos en el punto en que los datos se recopilan a medida que el flujo de datos llega al dispositivo que posee una capacidad de procesamiento mucho mayor que en la capa de dispositivos. Aquí también se realizan tareas como el filtrado de las señales adquiridas todo esto con el fin de lograr la representación más fiel proveniente del objeto físico.
- **Capa nube:** Despues de preparar los datos se envían en sentido ascendente como muestra el diagrama para procesarlos aún mas, esto quiere decir almacenarlos y utilizarlos dentro de aplicaciones de la nube. Por lo general las aplicaciones en la nube que realizan procesamiento de datos suelen ser complementadas con aplicaciones móviles y con aplicaciones de clientes basadas en la web presentando a los usuarios la información.

2.1.2. Raspberry Pi

La Raspberry Pi es básicamente una computadora de bajo costo que proporciona un conjunto de pines GPIO (Pines de entrada y salida de propósito general) que le permiten incorporar y controlar componentes electrónicos para ser usada en una amplia variedad de proyectos digitales, entre los cuales destacan las aplicaciones de IoT. El sistema operativo común y recomendado de esta serie de dispositivos es Raspbian, este se basa en una distribución de GNU/Linux llamada Debian que se encuentra optimizado específicamente para el hardware de las Raspberry Pi. Para el desarrollo del presente trabajo se utilizará el modelo de Raspberry Pi 3 B+ mostrado a continuación.



Figura 2.2: Raspberry Pi 3 B+.

En este orden de ideas, es posible hacer una tabla que contenga sus principales especificaciones obtenidas del fabricante.

Especificación	Raspberry Pi 3 Model b+
Procesador	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
Frecuencia de Reloj	1,4 GHz
GPU	1GB LPDDR2 SDRAM
Memoria	1GB LPDDR2 SDRAM
Conectividad Inalámbrica	2.4GHz / 5GHz IEEE 802.11.b/g/n/ac Bluetooth 4.2, BLE
Conectividad de Red	Gigabit Ethernet over USB 2.0 (300 Mbps de máximo teórico)
Puertos	GPIO 40 pines HDMI 4 x USB 2.0 CSI (cámara Raspberry Pi) DSI (pantalla táctil) Toma auriculares / vídeo compuesto Micro SD Micro USB (alimentación) Power-over-Ethernet (PoE)
Fecha de lanzamiento	14/3/2018
Precio	39.75 euros

Figura 2.3: Especificaciones Raspberry Pi 3B +.[11]

De acuerdo a lo anterior queda por definir la distribución de los puertos, más específicamente el GPIO compuesto de 40 pines, saber la distribución de estos es sumamente importante debido a que en ellos se conectarán el sensor, modem de comunicaciones y pulsadores que permitirán que el prototipo funcione correctamente como se puede ver a continuación en la siguiente figura donde se ven los pines de tierra, alimentación y pines como los que usa el protocolo I2C con el cual se comunican la mayoría de sensores digitales como el nuestro. También vemos los pines para la comunicación serial y la UART, sumamente importante para el uso y configuración del modem que se implementará. Por ejemplo, para la comunicación del sensor se usará el protocolo I2C el cual tiene sus correspondientes pines que son los pines 3 y 5 correspondientes al GPIO 2 Y GPIO 3 pines exclusivos para este tipo de comunicación como se detallará mas adelante.

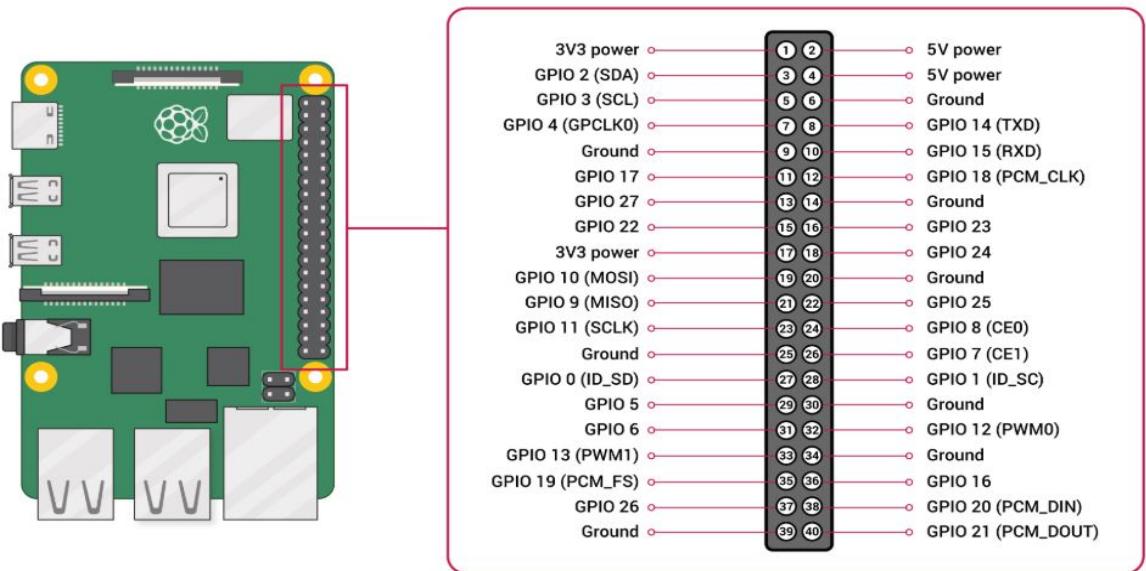


Figura 2.4: GPIO Raspberry Pi 3 B+. [12]

Es de tener en cuenta esta distribución de puertos ya que como veremos será útil para la programación y configuración de los dispositivos.

2.1.3. Firebase

Firebase de Google, es un conjunto de herramientas principalmente enfocadas al desarrollo de aplicaciones, entre los elementos, se ofrecen desde base de datos en tiempo real, hasta almacenamiento y servicios de hosting. Firebase es un servicio Backend, que permite la interacción amigable con el usuario, pues este no tiene que acceder directamente a los servidores para hacer uso de funciones como la base de datos en tiempo real [13].

Principales Características

Firebase cuenta con unas características que lo hacen ideal para proyectos como este, donde la integración de los sistemas es de suma importancia[14].

- **Agilidad:** Esta característica es vital sobre todo para nuestro objetivo orientado a la analítica y presentación de los datos obtenidos ya que nos permite el desarrollo en múltiples aplicaciones ya sea Android, IOS o la WEB, proporcionándonos además formas de programarlos en todos los diferentes lenguajes C++, JavaScript, Python, Java, HTML, entre otros. Esto mediante APIs integradas a SDK (Kit de desarrollo de software) individuales en estas aplicaciones móviles o web mencionadas y todo esto sin necesidad de salir de la propia plataforma.
- **Desarrollo gratuito:** Este aspecto es ideal para el desarrollo, ya que no pide pagar para la iniciación y uso de la plataforma, sin embargo, empieza a cobrar cuando el se sobrepasa una cantidad estipulada de almacenamiento y tráfico de información, no obstante estos límites como más de 100 conexiones simultáneas y mas de 1 Gb de almacenamiento no se excederán en lo que concierne al trabajo propuesto.
- **Permite adquirir dinero:** Esto lo permite a través del uso de Admob habilitando la posibilidad de ganar dinero mediante anuncios y publicidad. Vale recordar que este proyecto como trabajo de grado no esta contemplando algún valor monetario momentáneamente ya que esperamos seguir trabajando en el mismo posteriormente.

- **Servicios:** Una de los servicios más importantes que Firebase nos ofrece y del cual hacemos uso todo el tiempo es de la base de datos en tiempo real, vital para la realización del trabajo. También ofrece autenticación para la identificación de usuarios mediante email o redes sociales, así mismo nube de almacenamiento, Hosting, Test lab para probar la aplicación antes de una publicación y Crash Reporting para reportar errores que se puedan presentar en la aplicación.

2.1.4. Acelerómetro y aceleración

El desarrollo y análisis de este trabajo de grado está fundamentado en variables físicas que determinan la dinámica del movimiento, estas variables pueden ser entendidas mucho mejor en su relación matemática.

En este orden de ideas la aceleración es la variación de la velocidad por unidad de tiempo, en otros términos la razón de cambio en la velocidad con respecto al tiempo. Definida a continuación:

$$a = \frac{dv}{dt} \quad (2.1)$$

Además, si se observa detenidamente podemos deducir y definir la variable velocidad y distancia de la siguiente manera.

$$v = \int a \cdot dt \quad (2.2)$$

$$d = \int v \cdot dt \quad (2.3)$$

Así mismo, la segunda ley de Newton señala que un cuerpo con masa constante tiene una aceleración directamente proporcional a la fuerza que actúa sobre el mismo. Definiéndola matemáticamente se tiene:

$$F = m \cdot a \quad (2.4)$$

Esta ley de Newton es fundamental ya que es utilizado por los acelerómetros para medir la aceleración.

Los sensores de aceleración o más comúnmente llamados acelerómetros los cuales están pensados para la realización de una medida de vibración o aceleración. Estos realizan la medida de acuerdo a una señal eléctrica según la variación física que están midiendo, siendo la principal variación física la aceleración [15].

Los acelerómetros dentro de la mayoría de los integrados cuentan con un MEMS (Micro Electro Mechanical System) que se puede modelar de forma similar a un sistema físico masa-resorte de la siguiente manera:

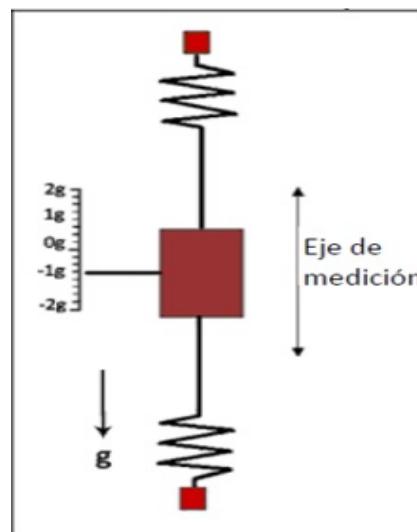


Figura 2.5: Sistema físico masa-resorte.

Existen diferentes tipos de acelerómetros que combinan diferentes tecnologías entre ellos los más comerciales y resaltables son del tipo piezoresistivos, piezoeléctricos y capacitivos. Según la tecnología que sea usada los rangos de medida pueden variar desde pocos g's hasta miles de g's. Esto ocurre de la misma forma con la frecuencia, donde según el tipo de tecnología que sea usada se podrá medir a muy bajas frecuencias desde los mili-Hz hasta los miles de Hz.

Una de los factores a tener en cuenta es el formato o empaque en el cual vienen ya que puede ser más o menos robusto lo cual involucra un mayor o menor peso, un factor esencial en la medida de la aceleración. Todo esto dependerá en gran medida de la aplicación.

2.1.5. Acelerómetro (MPU6050)

El módulo que se usará es el acelerómetro MPU6050 con 6 grados de libertad, ya que tiene un giroscopio de tres ejes con el cual es posible medir velocidad angular y un acelerómetro de tres ejes con el que se miden los componentes x,y,z de la aceleración respectivamente. Este acelerómetro tiene las siguientes características [16]:

- Voltaje de alimentación: 3V - 5V
- 6 Grados de libertad: 3 acelerómetro y 3 giroscopio
- Sensor de temperatura salida digital.
- Rango acelerómetro: medida ajustable ($\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$)
- Sensibilidad giroscopio: 131 LSBs/dps
- Rango giroscopio: medida ajustable (± 250 dps , ± 500 dps , ± 1000 dps y ± 2000 dps)
- Interfaz: Digital I2C
- Dimensiones: 19mm x 15mm
- ADC : Convertidor Analógico digital de 16 bits en todos los ejes.

La dirección de los ejes del modulo se pueden observar a continuación, es importante tenerlas en cuenta para el signo de las aceleraciones que pueden ser tanto positivas como negativas:

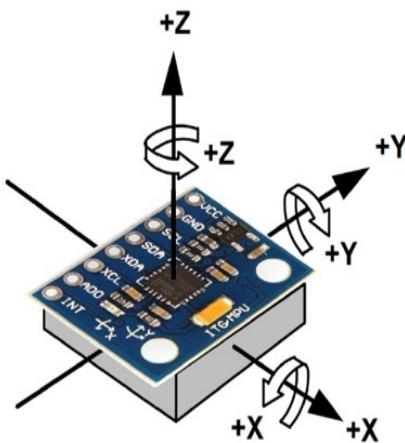


Figura 2.6: Acelerómetro MPU6050.

La comunicación del módulo es mediante el protocolo digital I2C, lo cual le permite trabajar con la mayoría de microcontroladores. Los pinos correspondientes a SCL y SDA tienen una resistencia pull-up para una conexión directa al microcontrolador. Adicionalmente el módulo tiene un regulador de voltaje de 3.3 voltios, esto permite alimentarlo con 5 voltios.

2.1.6. GPRS

GPRS, acrónimo de General Packet Radio Service, es una extensión de la tecnología de comunicaciones móviles GSM, diseñada para desarrollar las capacidades de transmisión de datos sobre la red, actualmente utilizada básicamente para la transmisión de voz, utilizando la técnica de conmutación de paquetes. Se la conoce como Generación 2,5 G de comunicaciones móviles que permite la conexión a internet y velocidades de transmisión teóricas de hasta 171 Kbps.

El protocolo de conexión a internet sobre GPRS que se utilizará es PPP *Point to point protocol* este es un protocolo de comunicación de la capa de enlace de datos que se utiliza para transmitir datos multi-protocolo en tramas entre dos computadoras conectadas directamente, es decir como su nombre lo dice, punto a punto, está orientado a bytes y se usa ampliamente en comunicaciones de banda ancha con cargas pesadas y altas velocidades [17].

COMPONENTES PPP

El protocolo punto a punto, consta de cuatro **componentes** específicos [18]:

- **Componente de encapsulación:** Encapsula el datagrama para que pueda transmitirse a través de la capa física especificada.
- **Protocolo de control de enlace (LCP) :** Es responsable de establecer, configurar, probar, mantener y terminar los enlaces para la transmisión. También imparte negociación para la configuración de opciones y el uso de características por los dos puntos finales de los enlaces.
- **Protocolos de autenticación (AP) :** Estos protocolos autentican los puntos finales para el uso de los servicios. Los dos protocolos de autenticación de PPP son:
 - Protocolo de autenticación de contraseña (PAP)
 - Protocolo de autenticación de desafío Handshake (CHAP)
- **Protocolos de control de red (NCP) :** estos protocolos se utilizan para negociar los parámetros e instalaciones para la capa de red. Por cada protocolo de capa superior compatible con PPP, hay un NCP, algunos de estos son:
 - Protocolo de control de protocolo de Internet (IPCP)
 - Protocolo de control de capa de red OSI (OSINLCP)
 - Protocolo de control IPv6 (IPV6CP)

TRAMA PPP

PPP es un protocolo orientado a bytes donde cada campo de la trama está compuesto por uno o más bytes. Los campos de una trama PPP son:

- **Indicador:** 1 byte que marca el principio y el final del marco. El patrón de bits de la bandera es 01111110.
- **Dirección:** 1 byte que se establece en 11111111 en caso de transmisión.
- **Control:** 1 byte establecido en un valor constante de 11000000.
- **Protocolo:** 1 o 2 bytes que definen el tipo de datos contenidos en el campo de carga útil.
- **Carga útil:** transporta los datos de la capa de red. La longitud máxima del campo de carga útil es de 1500 bytes. Sin embargo, esto puede negociarse entre los puntos finales de la comunicación.

- **FCS:** es una secuencia de verificación de trama de 2 bytes o 4 bytes para la detección de errores. El código estándar utilizado es CRC (código de redundancia cíclica)

En ocasiones se usa el relleno de bytes en la trama, por ejemplo en el campo de carga útil de PPP siempre que aparece la secuencia del indicador en el mensaje, de modo que el receptor no lo considere como el final de la trama. El byte de escape, 01111101, se rellena antes de cada byte que contiene el mismo dato que el byte indicador o el dato de escape. El receptor al recibir el mensaje elimina el byte de escape antes de pasarlo a la capa de red.

A continuación se muestra gráficamente el entrampado descrito.

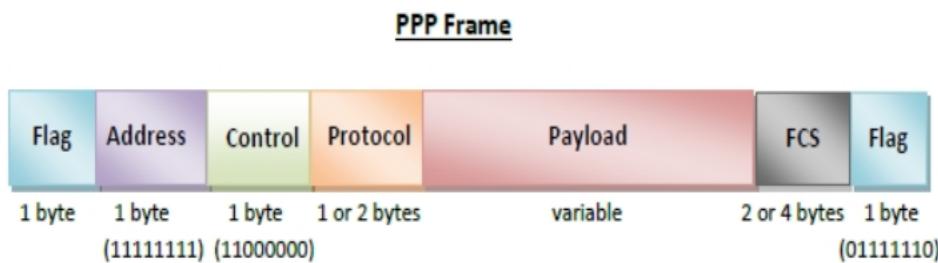


Figura 2.7: Frame PPP[19]

Servicios PPP

Entre los servicios prestados por el protocolo PPP se encuentran [20]:

- Definición del formato de trama de los datos a transmitir.
- Definir el procedimiento para establecer el vínculo entre dos puntos y el intercambio de datos.
- Indicar el método de encapsulación de los datos de la capa de red en el marco.
- Establecer reglas de autenticación de los dispositivos de comunicación.
- Proporcionar dirección para la comunicación de red.
- Proporcionar conexiones a través de múltiples enlaces.
- Admite una variedad de protocolos de capa de red al proporcionar una gama de servicios de os.

2.1.7. Aplicación Web

Cada vez, en estos tiempos, es más frecuente que cualquier tipo de entidad, ya sea empresa, asociación, institución, etc., requiere una solución informática apropiada para sus necesidades. Como respuesta a estas necesidades suele ser cada vez más habitual escuchar la palabra: aplicación web, antes de avanzar, es necesario saber qué es una aplicación web? esta se define como una herramienta informática accesible desde cualquier navegador, bien sea a través de internet (lo habitual) o a través de una red local. A través del navegador se puede acceder a toda la funcionalidad que la aplicación pueda proveer [21].

Servicio Web

Un servicio web se define como un sistema de software designado para dar soporte a la interacción de máquina a máquina interoperativa a través de una red [22]. Un servicio web se caracteriza por realizar un conjunto de tareas específicas. Este tipo de aplicaciones son implementaciones en diversas tecnologías y diversos lenguajes de codificación que hacen posible la integración de herramientas, proporcionando

al usuario una amplia gamma de todo tipo de servicios procurando en gran medida la protección de los datos del usuario.



Figura 2.8: Aplicación Web [23]

En este punto es conveniente definir mediante un diagrama como funciona una aplicación Web la cual se puede ver a continuación:

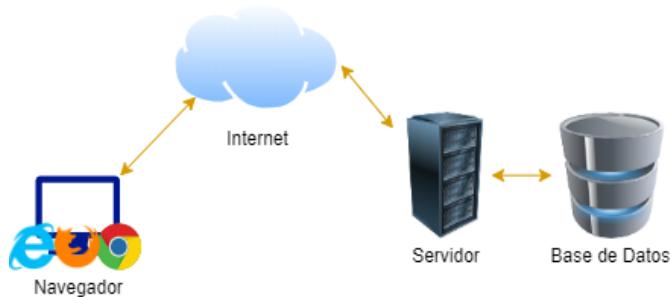


Figura 2.9: Diagrama Aplicación Web

Como se observa en el diagrama existen relaciones entre cada uno de los elementos vistos. Vale recordar que existen diferentes tecnologías para el desarrollo de una aplicación, no obstante, este esta compuesto principalmente por estos cuatro principales componentes.

- **Base de Datos :** Este componente encarga principalmente del almacenamiento de los datos bajo un mismo contexto para ser usados sistemáticamente posteriormente. Suelen ser usados en gran medida SQL, MYSQL y noSQL.
- **Servidor de la Aplicación:** Suele ser denominado como un dispositivo que provee servicios específicos de la aplicación a las computadoras de los clientes, entre los mas usados están ASP.NET o PHP.
- **Internet(Cloud Computing):** Al hablar de computo en la nube hablamos principalmente de un paradigma que provee servicios a través de una red usualmente internet.
- **Navegador:** Es una interfaz de usuario donde el cliente puede visualizar la información.

Es importante tener en cuenta que actualmente existen plataformas que integran estos elementos antes mencionados, como por ejemplo, Firebase permite tener una base de datos en tiempo real y además permite servicios de un servidor de aplicación como autenticación, Hosting para la página web y análisis tanto del almacenamiento como del uso de los clientes en la aplicación. Ahora se definen conceptos claves en el desarrollo de la pagina web, como lo son el lenguaje en el que esta escrita y son ampliamente usadas en cualquier otra aplicación web.

Capítulo 3

Desarrollo

3.1. Antecedentes

La manera de movilizarnos ha mejorado con respecto a uno de los primeros problemas del transporte público que se puede ver en el documental de la “guerra del centavo”[24], sin embargo aún existen demasiados problemas con estos tipos de transporte. En trabajos de grado antiguos se recogen ideas de lo más importante que se necesita saber para mejorar el servicio como: ¿Cuántas personas están tomando el servicio?, ¿se están cumpliendo los tiempos por parada y los dueños de este servicio saben ¿Cómo manejan sus conductores contratados? El trabajo se enfocará en la tercera pregunta para aumentar el conocimiento e información que se tiene acerca de los conductores que están laborando, nos guiamos en esta idea debido a una tesis hecha en la Universidad Javeriana donde se intentó separar los patrones de conducción e indicar si un conductor de transporte público seguía ciertos parámetros de seguridad, al final fue imposible catalogar ya que para clasificar la calidad de conducción se deben tener en cuenta muchas variables, haciendo complejo tener un resultado preciso [25].

A partir de lo anterior resulta factible ampliar el proyecto para hacer la trasmisión de datos desde el vehículo cada cierto intervalo de tiempo, recibiendo esos datos censados en un servidor, visualizando la frecuencia con la que se sobrepasa sobre pasa ciertos criterios físicos que se tomarán en cuenta como ciertas condiciones de seguridad. En un estudio realizado en Valencia y Zaragoza se determinó gracias a pruebas hechas en los autobuses que lo máximo que puede soportar un humano antes de ser susceptible a caídas o daños es de 0.15G de aceleración transversal y aceleración longitudinal [26].

3.2. Montaje

Los elementos necesarios que constituyen el montaje del dispositivo capaz de medir los indicadores de conducción son los siguientes:

- Acelerómetro MPU6050.
- Módulo GPRS SIM800.
- Raspberry Pi 3.
- Antena
- Pantalla LCD táctil.
- Caja contenedora

Debido a que el montaje debe ser instalado dentro de los buses del sistema integrado de transporte para la caja contenedora, se optó por un caja rectangular de madera como se muestra en la siguiente figura.

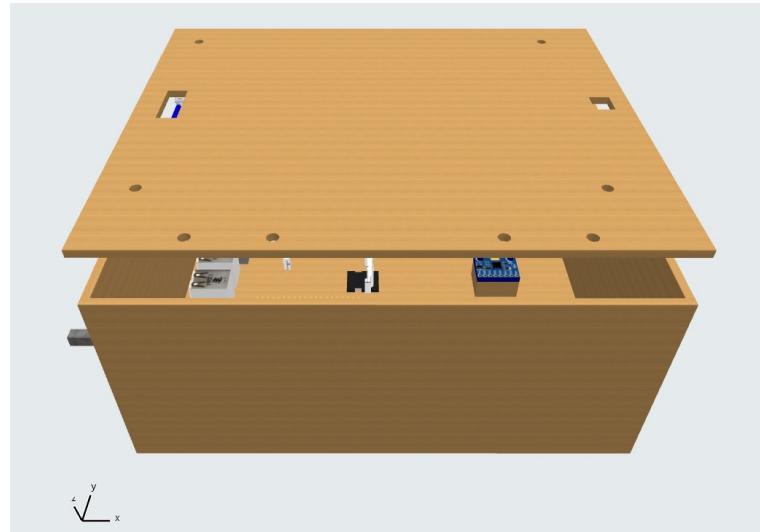


Figura 3.1: Diseño de caja.

Esta cuenta con todos los orificios y disposiciones necesarias para la fijación de componentes como la pantalla y el acelerómetro, y la salida de elementos como la antena del módulo GPRS. La pantalla se fijará en la parte interior de la tapa de la caja dispuesta hacia abajo, de manera que pueda visualizarse apenas esta se abra. El acelerómetro, por medio de una base pequeña también construida en madera, se fijará a la base total de la caja, esto con el fin de proporcionarle estabilidad y que las medidas reales no se vean afectadas.

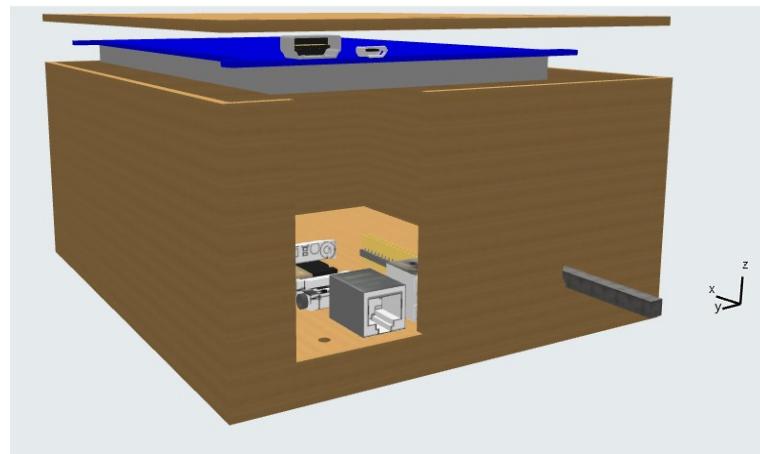


Figura 3.2: Diseño de caja 2.

Finalmente, el montaje realizado y dispuesto con todos sus componentes reales se muestra a continuación:

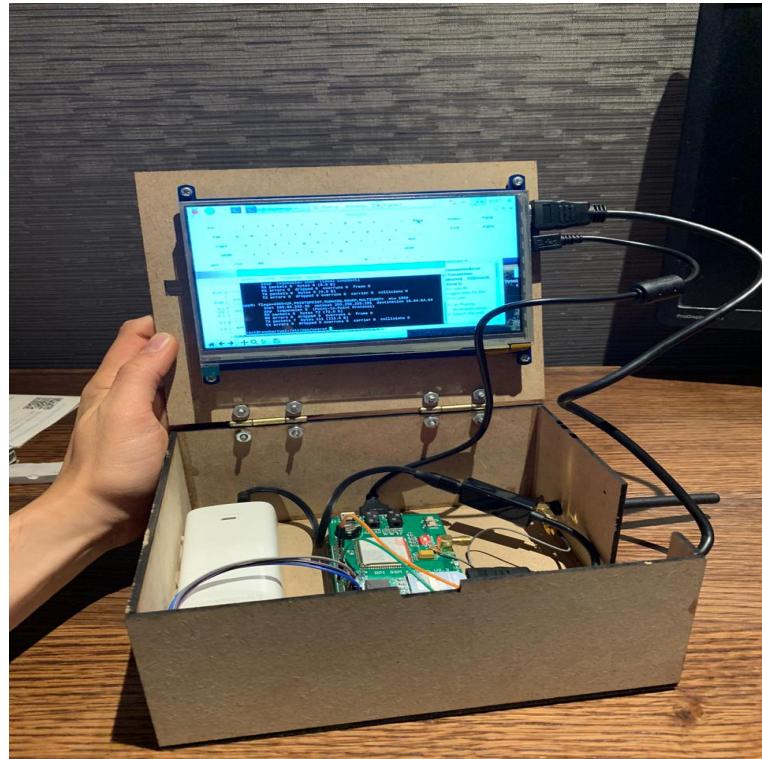


Figura 3.3: Montaje Final.

Este montaje se diseño de forma que fuera lo mas compacto y portátil posible, esto con el fin de permitir el análisis de conducción de varias rutas y asimismo facilitando la instalación del mismo. Es sin embargo, recomendable usar en la base del montaje plastilina para fijar el dispositivo ya sea al suelo o alguna superficie lisa sin inclinaciones. La plastilina se adhiere muy bien al suelo y además, permite retirar el dispositivo del bus sin llegar a violentarlo después de haber hecho el análisis.

3.3. Descripción general

El trabajo de grado tiene como propósito adquirir señales de aceleración en un bus de Transmilenio, una vez adquiridos procesarlos para poder enviar e indicar cuántas veces el conductor esta incurriendo en una situación de inconformidad o de riesgo para el pasajero. Para organizar las ideas que se tienen del proyecto se creó un mapa, indicando específicamente cuales son las tareas necesarias para poder cumplir con los objetivos propuestos.

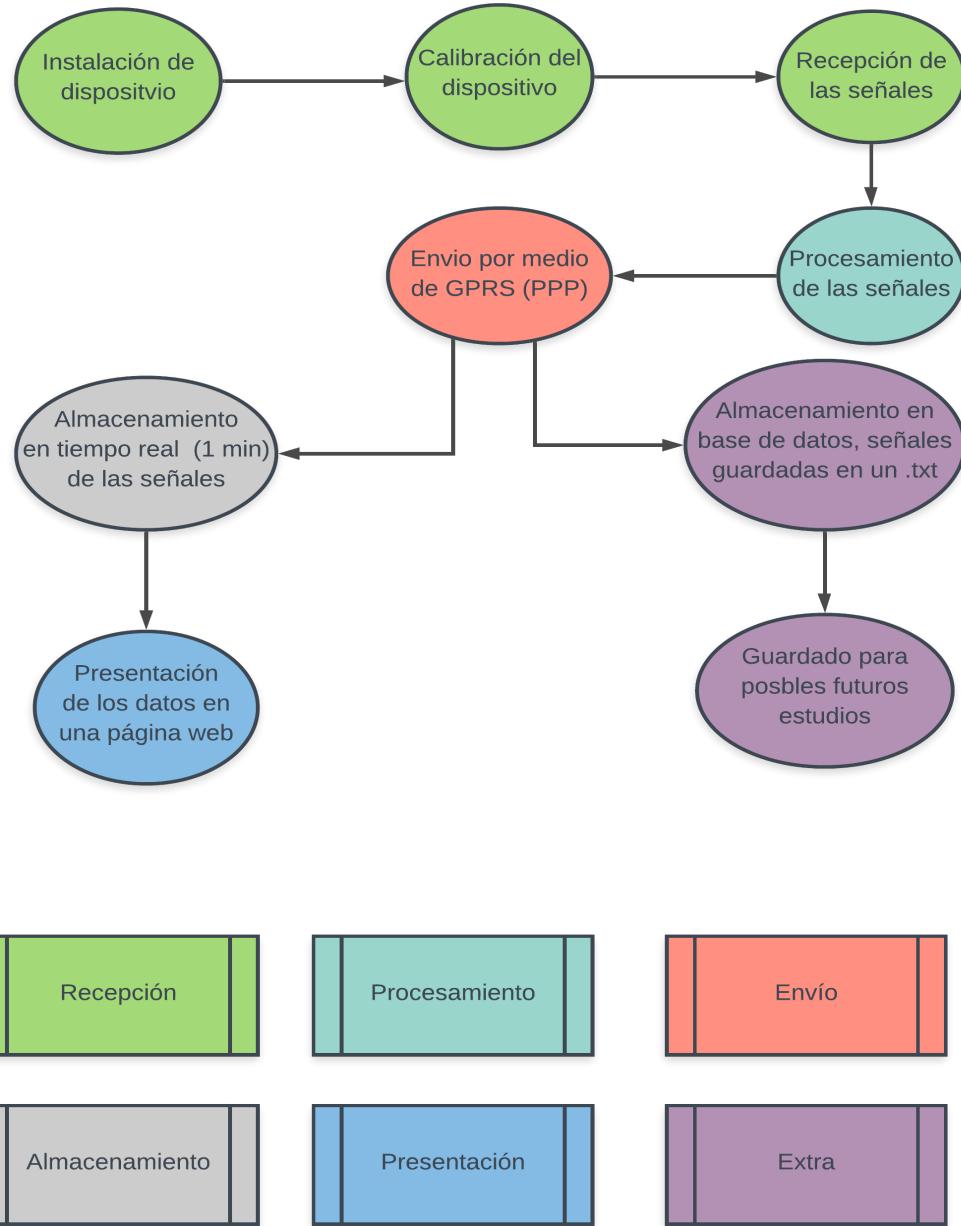


Figura 3.4: Modelo Conceptual del Proyecto

3.4. Parámetros del sistema

3.4.1. Clasificación de riesgos

Para determinar en las señales cuales van a ser los indicadores de riesgo, se toma como referencia un estudio en Valencia y Zaragoza que determinó gracias a pruebas hechas en los autobuses que lo máximo que puede soportar un humano antes de ser susceptible a caídas o daños es de 0.15G de aceleración transversal y aceleración longitudinal [27]. Otro estudio hecho en Pennsylvania arrojó como resultados que un humano promedio podía aguantar parado una aceleración de $0.93 \frac{m}{s^2}$ y que el tirón mas largo que puede soportar un humano sin perder su balance es de $0.6 \frac{m}{s^3}$ [28]. Estos datos nos dan alguna referencia para determinar los límites para considerar cuando el bus está realizando una maniobra de peligro o incomodidad para el pasajero.

Por lo tanto se decidió tomar como riesgo medio en el eje x y y las muestras de aceleración que

superen los 0.09G, por otro lado las muestras que estén por encima de los 0.15G se considerarán como riesgo alto, para terminar, las señales de riesgo del eje z que superen el 7.5 % del eje medio se consideran riesgo medio y las señales que superen el 15 % se considera riesgo alto. Para tener mejor definido que momento de la conducción afecta cada eje se aclara lo siguiente:

- 1. Los movimientos de carril afectan al eje x .
- 2. Los acelerones y frenadas afectan el eje y .
- 3. Los baches y huecos afectan el eje z .

Correlación:

Para el estudio de nuevos indicadores se intentó agregar una clasificación especial diferentes a las muestras que superan el riesgo de peligro, se analizaron el número de veces que se entraba en un indicador de riesgo, es decir, una vez es cuando una muestra supera el indicador hasta que deje de superar este riesgo, por ejemplo si hubo 4 muestras seguidas que sobrepasaron un riesgo quiere decir 1 vez, pero al contrario el número de muestras va a seguir siendo 4. Se planteó que a mayor duración de de un riesgo se iba a sentir mayor incomodidad. Al momento de definir si era apropiado agregarlo al trabajo, se llegó a la conclusión que mostrar estos dos indicadores por separado no tenía sentido ya que aumentaba el costo de envío y lo que se llama muestras y veces en el trabajo tiene una muy alta correlación, por lo tanto se decidió solo trabajar con las muestras, a continuación se muestra el grado de correlación:

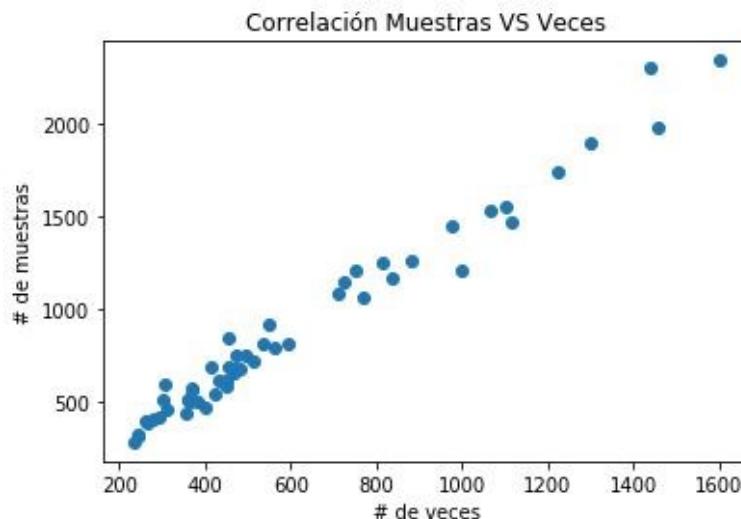


Figura 3.5: Correlación de muestras vs Veces.

3.4.2. Costo de envío

Para determinar que costo va a tener el envío de estos datos a la nube, se investigó cuántos bytes enviaba Firebase por petición de envío, se encontró la información sobre el calculo de tamaño de almacenamiento [29].

Tipo	Tamaño
Matriz	La suma de los tamaños de sus valores
Booleano	1 byte
Bytes	Longitud de bytes
Fecha y hora	8 bytes
Número de coma flotante	8 bytes
Punto geográfico	16 bytes
Entero	8 bytes
Mapa	El tamaño del mapa, calculado de la misma manera que el tamaño del documento
Nulo	1 byte
Referencia	El tamaño del nombre del documento
String de texto	Cantidad de bytes con codificación UTF-8 + 1

Figura 3.6: Almacenamiento en Firebase por Bytes.

Además se tiene en cuenta que por cada variable o arreglo enviado, por defecto se envían 16 Bytes, para calcular qué costo tiene hacer un envío a real time database se puso como condición que la duración del sistema en funcionamiento va a ser de 15 horas, ya que esto es el tiempo que se supone estará en funcionamiento el dispositivo al día.

Encabezado	Encabezado	Encabezado	Por defecto	Tamaño enviado	Total
MytestData	tiempo	transcurrido	16	8	57
MytestData	registrodatox	ejex	16	154286	154333
MytestData	registrodatoy	ejej	16	154286	154332
MytestData	registrodatoz	ejez	16	154286	154332
MytestData	reg_pel_xm		16	8	54
MytestData	reg_pel_ym		16	8	47
MytestData	reg_pel_zm		16	8	47
MytestData	reg_pel_xa		16	8	47
MytestData	reg_pel_ya		16	8	47
MytestData	reg_pel_za		16	8	47
				Envío por 15 horas	463343

Figura 3.7: Cantidad de bytes enviados al final del trayecto.

Encabezado	Encabezado	Encabezado	Por defecto	Tamaño enviado
MytestData	riesm_x			16 8
MytestData	riesm_y			16 8
MytestData	riesm_z			16 8
MytestData	riesa_x			16 8
MytestData	riesa_y			16 8
MytestData	riesa_z			16 8
MytestData	reg_x4	ejex	16	1376
MytestData	reg_y4	ejey	16	1376
MytestData	reg_z4	ejez	16	1376
Encabezado	Encabezado	Encabezado	Por defecto	Tamaño enviado
11	8		16	8
11	8		16	8
11	8		16	8
11	8		16	8
11	8		16	8
11	8		16	8
11	7	5	16	1376
11	7	5	16	1376
11	7	5	16	1376
			Envío por minuto	4176
			Envío a las 15 horas	3758400

Figura 3.8: Cantidad de bytes enviados durante el trayecto.

El número total de Bytes enviados por las 15 horas de uso del proyecto es de 4.2MB aproximadamente. El costo del uso se concluye a partir de los precios que tiene la empresa Tigo Colombia Móvil para planes prepago y se llega a la conclusión que el costo al día es de 43 pesos colombianos.

Por último en el proyecto se creó una base de datos en la que se guardan los trayectos analizados, para que posteriormente se puedan hacer análisis y avances sobre proyectos futuros, esto tiene un costo adicional al calculado, pero al no ser un objetivo del trabajo de grado no se analiza el costo de este.

3.5. Descripción de bloques e interfaces

En la figura 3.9 se puede observar el diagrama de bloques general utilizado para el desarrollo del proyecto.

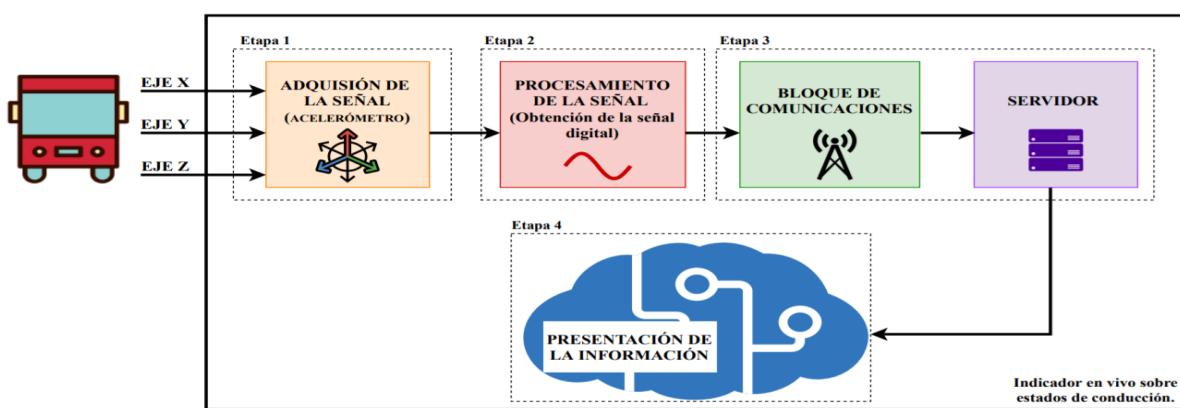


Figura 3.9: Diagrama de bloques general.

Cada uno de los bloques observados anteriormente, serán explicados con mayor detalle a continua-

ción.

3.5.1. Bloque de Adquisición de señal

Este bloque se encarga de la recolección de datos provenientes del acelerómetro, en este caso tres señales correspondientes a los tres ejes coordinados del espacio X,Y y Z.

El acelerómetro MPU6050, se comunicará con la Raspebrry Pi 3 por medio del protocolo I2C, en donde el acelerómetro funciona como esclavo únicamente enviando datos y la Raspberry Pi como maestro únicamente recibiendo según como se muestra en el siguiente diagrama.

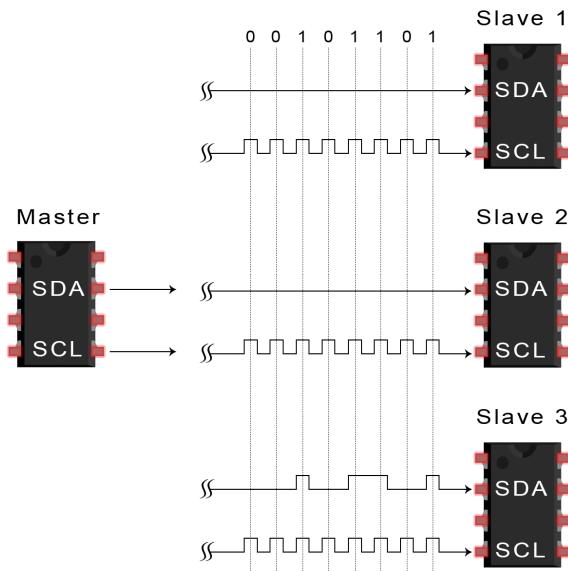


Figura 3.10: Protocolo I2C [30].

Los pasos de ejecución del protocolo I2C son los siguientes:

- 1. El maestro envía una condición de inicio a cada esclavo que este conectado en la vía SDA, seguidamente cambia el nivel de voltaje a bajo y deja vía SCL en estado alto.
- 2. El maestro envía la dirección de 7 a 10 bits a los esclavos con el fin de identificar el esclavo con el que se quiere comunicar.
- 3. Cada esclavo recibe la dirección y la compara con su propia dirección, si coincide el esclavo manda un bit de ACK seguidamente pone vía SDA en nivel bajo de voltaje, en dado caso que no sean la misma los esclavos no hacen nada y la vía SDA en el mismo nivel de voltaje alto.
- 4. El maestro envía o recibe los respectivos frames de información.
- 5. Luego de que cada frame de información es enviado al dispositivo que recibe, para este caso el maestro corresponde al Raspberry Pi y el esclavo o acelerómetro enviará un bit ACK al remitente, notificándole que la información se recibió exitosamente.
- 6. Para finalizar la transmisión de información el maestro envía al esclavo la condición de Stop con un nivel alto en la vía SDA, cuando cambia el estado SCL a alto [31].

Para activar el protocolo I2C en el Raspberry Pi 3, primero debe habilitarse el SPI(Serial Peripheral Interface) e I2C lo cual permitirá abrir la terminal, tras lo cual se escribirá el comando "sudo raspi-config", lo cual desplegará el menú que permite cambiar el estado del protocolo a ON. Luego es necesario reiniciar el dispositivo para que los cambios hechos tengan efecto. En seguida, se instala en la terminal las

librerías y herramientas necesarias para proceder a hacer la comunicación con un script, es decir, digitar el siguiente comando ”sudo apt-get install i2c-tools python-smbus”.

Asimismo es necesario determinar la dirección del bus I2C que se está utilizando para el módulo del sensor. Muchos de estos módulos tienen una dirección hexadecimal; para conocer esta dirección que se empleará en el script y así obtener la información del sensor, ejecutamos el siguiente comando por terminal

```
sudo i2cdetect -y 1
```

Al ejecutarlo se muestra la siguiente dirección hexadecimal del bus I2C.



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60:	-	-	-	-	-	-	-	-	-	-	68	-	-	-	-	-
70:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figura 3.11: Dirección bus I2C

Si se ha configurado y conectado bien el sensor obtendremos la dirección 0x68 como salida. También si hay mas módulos trabajando con este protocolo conectados las direcciones también serán desplegadas.

3.5.2. Bloque de procesamiento de la señal

- **Recepción de las señales:**

El MPU6050 tiene una resolución de 16 bits, se decide utilizarlo en la escala de $\pm 2G$, de esta forma los resultados son comparables a los indicadores referenciados de los documentos, ya que al leer los datos sin escalizar, se obtienen resultados desde 0 a 2^{16} . Para poder utilizar el acelerómetro correctamente se utiliza la siguiente ecuación.

$$\frac{2^{16}}{2 \cdot \text{escala}} \quad (3.1)$$

Como se mencionó la escala es de 2, pero también se puede escoger de 4, 8 o 16. El resultado de escalización para este proyecto es de 16384, por lo tanto todos las señales adquiridas por el acelerómetro del sensor se deben dividir por ese número. La calibración del sensor se hace automáticamente, es decir, el acelerómetro se va a calibrar en el momento que empieza el programa haciendo que el eje x sea 0, y sea 0 y z sea 0.98 inicialmente.

- **Filtrado de la señal:**

Al medir las señales de la aceleración del transmilenio, se observó que la señal tenía bastante ruido debido a las vibraciones del bus, por lo tanto, se decidió utilizar un filtro de media móvil para suavizar el ruido que interferiría en la señal. Este filtro es muy usado por su simplicidad, consiste en promediar cierto número de muestras y convertir este promedio en el resultado de una sola muestra. Hay que tener en cuenta que si se escoge un número muy grande de promedio, se perdería también la calidad de datos que se recolectan.

Por lo tanto escogemos una media móvil de 6 para que suavice la señal de tal forma que no modifique demasiado los datos, y pueda disminuir el ruido causado por el motor.

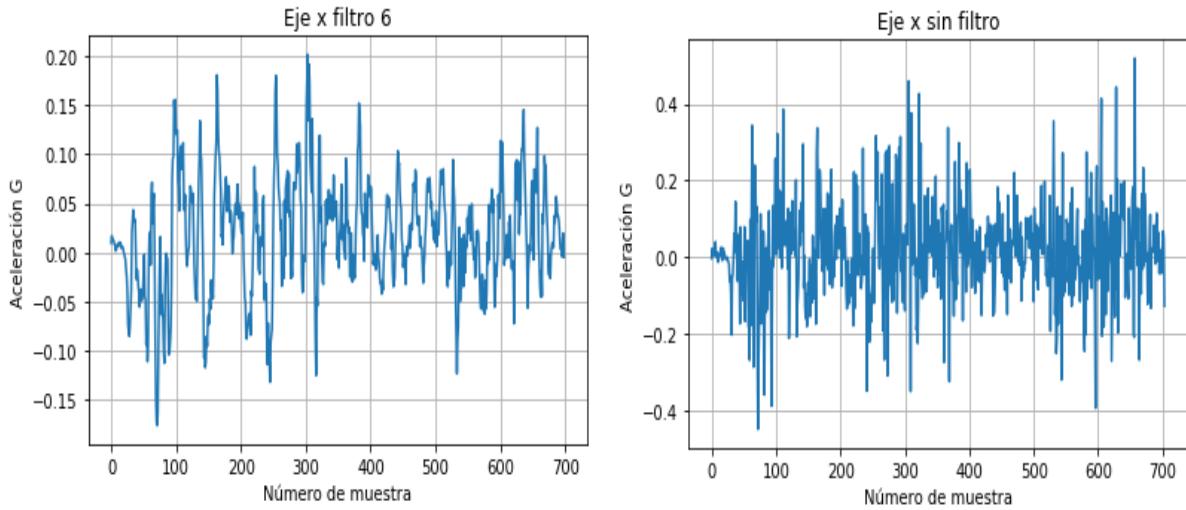


Figura 3.12: Comparación eje x con filtro 6 y sin filtro

3.5.3. Bloque de Comunicaciones

Este bloque se refiere al envío de los datos y valores del sistema cada cierto tiempo por medio del protocolo GPRS hacia el servidor en la nube. Para el desarrollo de la comunicación entre el Raspberry Pi y acceder a internet, usaremos un modem que mediante los puertos seriales se comunicará con el Raspberry Pi. El módulo adquirido se ilustra a continuación:

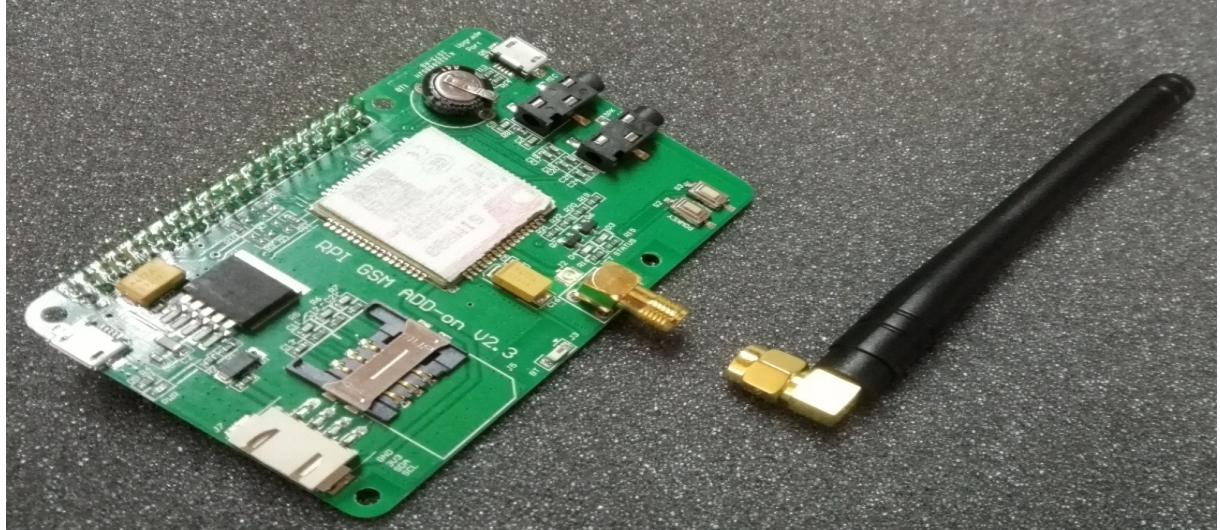


Figura 3.13: Modem Gprs Raspberry Pi

Esta tarjeta personalizada cuyo su desarrollo partió del módulo SIM800 de tecnología 2G GSM/GPRS/BT, el cual está homologado en el territorio colombiano para uso en cualquier operador celular; para su operación basta con saber los comandos AT enviados por el puerto serial desde el Raspberry Pi. Este módulo tiene disponibles funciones como las de contestar y dictar llamadas; recepción y envío de mensajes SMS y la de navegación entre redes, asimismo, soporta restablecimiento y apagado a través del software. Las características según el fabricante son las siguientes [32]:

- Quad Band 850/800/1800/1900 Mhz.

-
- GPRS multislot clase 12.
 - Temperatura de operación -40 °C hasta 85°C.
 - Interfase UART.
 - Baudios por defecto 9600.
 - Voltaje de alimentación 3.3 VDC.

El módulo posee un compartimiento para instalar una SIM; se recalca que la SIM debe tener una linea activada y con plan de datos para la navegación. Este módulo fue diseñado para acoplarse con los pines del raspberry y además permitir el uso de otros pines, bien sea para conectar el sensor o el pulsador. Una vez conectado a la alimentación el Raspberry Pi genera un voltaje de referencia por uno de sus pines. En este orden de ideas, la conexión del Raspberry Pi con el módulo se representa en la siguiente imagen:



Figura 3.14: Montaje Modulo GPRS Raspberry Pi

A continuación se programa el modem teniendo en cuenta el proveedor de servicios de internet (ISP) y se configura el puerto serial del Raspberry Pi y los parámetros necesarios para su correcto funcionamiento.

En primer lugar, es fundamental la configuración del puerto serial en el Raspberry Pi 3. Las versiones previas del Raspberry Pi tenían un puerto serial en el GPIO llamado UART0, luego el diseño fue actualizado al introducirse el Raspberry Pi 3 que cuenta con dos puertos seriales uno en el GPIO correspondiente y otro dedicado para el Bluetooth; por lo tanto es necesario habilitar el puerto serial correspondiente al GPIO que es donde está el modem GSM. Para cumplir con este propósito es necesario configurar el sistema operativo.

Es necesario tener en cuenta que el puerto correspondiente al GPIO se llama /dev/ttys0/, para establecer el nombre de este puerto y el del serial del bluetooth se debe abrir el terminal digitando el siguiente comando:

```
dmesg | grep tty
```

Este comando enlistará los puertos seriales disponibles, como se observa en la siguiente figura:

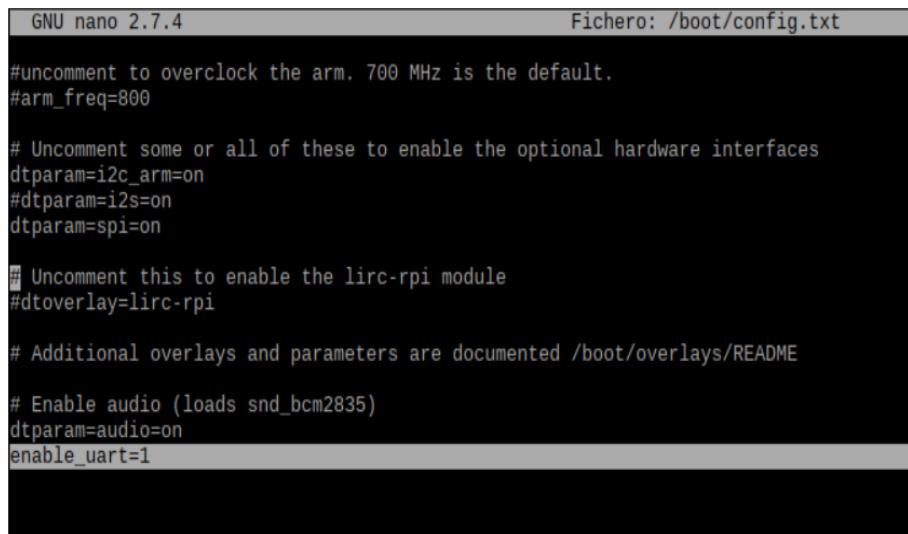
```
pi@raspberrypi:~ $ dmesg | grep tty
[    0.000300] console [tty0] enabled
[    0.744983] 3f201000.serial: ttyAMA0 at MMIO 0x3f201000 (irq = 81, base_baud = 0) is a PL011 rev2
[    0.746558] 3f215040.serial: ttyS0 at MMIO 0x0 (irq = 53, base_baud = 31250000) is a 16550
pi@raspberrypi:~ $
```

Figura 3.15: Puertos Disponibles Raspberry Pi3

Luego, desde el mismo terminal se abre un archivo de configuración con el que trabaja el microprocesador. Para habilitar este puerto serial se introduce el siguiente comando:

```
sudo nano /boot/config.txt
```

Linux tiene la funcionalidad de permitir editar documentos desde la terminal, concretamente se hace desde los comandos: "nano" para editar y "sudo" para dar permisos de administrador y abrir la ruta donde se encuentra el documento. Al ejecutar este comando se abrirá una ventana y se agregará una línea para habilitar el serial, allí se escribe: enableuart = 1 como se muestra a continuación:



```
GNU nano 2.7.4                                     Fichero: /boot/config.txt

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
dtparam=i2c_arm=on
#dtparam=i2s=on
dtparam=spi=on

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on
enable_uart=1
```

Figura 3.16: Habilitación de la UART

Es importante guardar y reiniciar el Raspberry Pi para que los cambios sean aplicados.

Configuración PPP

El PPP o protocolo punto a punto establece una comunicación nodo a nodo, usando una interfase serial. El nombre de punto de acceso (APN) corresponde al de una puerta de enlace entre una red GSM/GPRS e internet. Para esto es sustancial conocer el APN del proveedor de la red celular de la SIM insertada en el modem. Es de anotar que para este trabajo de grado se utilizó una SIM card del operador Tigo Colombia.

En el siguiente paso se procede a instalar el software PPP en la Raspberry Pi, para lo cual es necesario tener una conexión estable a internet. Abrimos la terminal Lx digitando los siguientes dos comandos :

```
sudo apt-get update
sudo apt-get install ppp screen elinks
```

Después de la instalación, debe crearse una nueva configuración de pares PPP; para esta operación se requiere iniciar sesión como root, el cual queda habilitado con acceso administrativo al sistema. Para este propósito debe ingresarse en la terminal sudo -i y navegar en el directorio de pares. La dirección del directorio es la siguiente:

```
cd /etc/ppp/peers/
```

Una vez en el directorio como root, se despliega un nuevo archivo que contendrá la configuración del modem, con el nombre rnet.^{el} cual contiene los siguientes parámetros para el correcto funcionamiento del modem GSM.

```
GNU nano 2.7.4 Fichero: rnet
# web.colombiamovil.com.co es la apn para Tigo colombia movil SIM usada.
connect "/usr/sbin/chat -v -f /etc/chatscripts/gprs -T web.colombiamovil.com.co"
# Para el Raspberry Pi3 usamos /dev/ttyS0 como el puerto de comunicacion:
/dev/ttyS0

# Baudrate
115200

# Asume que su direccion IP es asignada dinamicamente por el ISP.
noipdefault
# intenta obtener las direcciones del servidor de nombres del ISP.
usepeerdns

# Usa esta conexion como la ruta predeterminada a internet.
defaultroute

# "vuelve a marcar " cuando la conexion se pierde.
persist

# No pregunta por autenticacion.
noauth

# Sin control de flujo de hardware en el enlace serie con el modem GSM.
nocrtscts

# No hay lineas de control del modem con el modem GSM,
local

[ 28 líneas leídas ]
[G Ver ayuda   ^O Guardar   ^W Buscar   ^K Cortar txt   ^J Justificar   ^C Posición   ^Y Pág. ant.   M-] Pri. línea
^X Salir   ^R Leer fich.   ^L Reemplazar   ^U Pegar txt   ^T Ortografía   ^I Ir a línea   ^V Pág. sig.   M-/ Últ. línea
```

Figura 3.17: Configuración PPP

Del archivo anterior, se configuro de la siguiente manera.

- En el comando connect se ingresa el APN del operador que está brindando el servicio, para este caso es Tigo Colombia Móvil y su APN es web.colombiamovil.com.co
- En esta línea se configura el puerto serial de comunicación, puesto que se soporta en Raspberry Pi 3, la configuración de este puerto ya fue mencionada anteriormente.
- Se configura la velocidad en baudios del dispositivo tty y se selecciona la más alta posible debido a la aplicación.
- Este comando se configuró como ”noipdefault” que deshabilita el comportamiento predeterminado cuando no se especifica una dirección IP local, que es determinar (si es posible) la dirección IP local a partir del nombre de host. Con esta opción, el par deberá proporcionar la dirección IP local durante la negociación de IPCP (a menos que se especifique explícitamente en la línea de comando o en un archivo de opciones).
- El comando: ”usepeerdns”, solicita al par hasta dos direcciones del servidor DNS. Las direcciones proporcionadas se pasan al script /etc/ppp/ip-up/.
- Este comando llamado ruta por defecto, agrega una ruta predeterminada a las tablas de enrutamiento del sistema, que utiliza el igual como puerta de enlace, cuando la negociación de IPCP se complete con éxito.

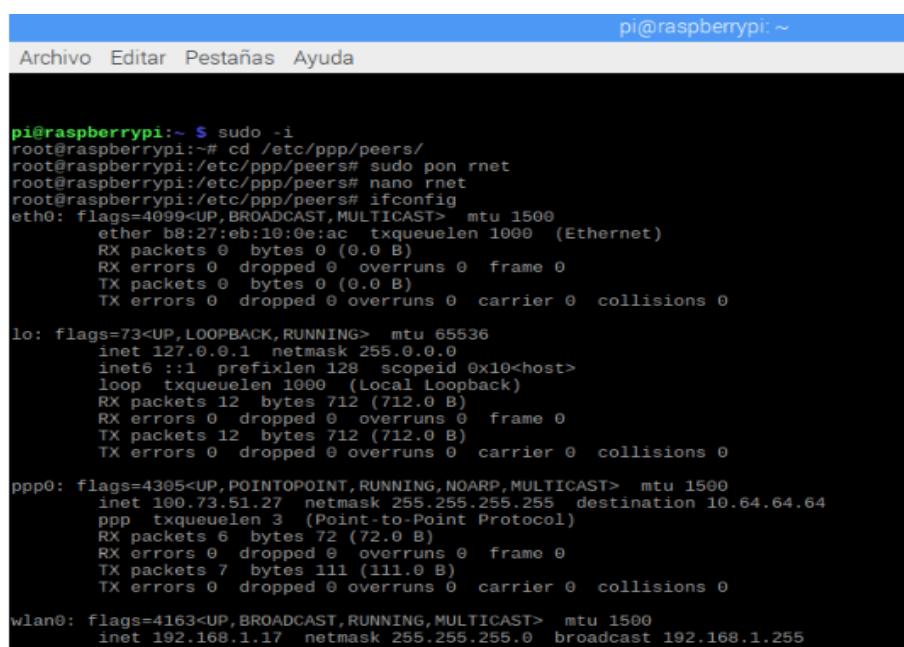
- El comando ”persistir”segun se anotó en el comentario dentro del archivo, reitera la marcacion si la conexión se ha perdido.
- En cuanto al comando ”noauth”no requiere que el par se autentique, esta opción es privilegiada.
- El comando ”nocrtscts”deshabilita el control de flujo de hardware (es decir, RTS / CTS), en el puerto serie.
- La opción ”local”, permite que la pppd ignore el estado de la señal CD (Detección de portadora)del módem y no cambiará el estado de la señal DTR (Listo para terminal de datos).

Estas definiciones son señaladas junto con ejemplos y toda una lista de comandos en la siguiente referencia [33].

Luego de haber creado este archivo llamado rnet en el directorio, se procede a guardar el archivo y se procede a crear la conexión digitando el siguiente comando en la terminal.

```
sudo pon rnet
```

Luego de crear la conexión podemos visualizarla escribiendo el comando ifconfig este mostrara en conjunto de redes que estemos usando como se puede ver a continuación.



A screenshot of a terminal window titled "pi@raspberrypi: ~". The window contains a command-line session. The user runs "sudo -i", changes directory to "/etc/ppp/peers", and executes "sudo pon rnet". Then, they run "ifconfig" to view network interfaces. The output shows four interfaces: eth0 (Ethernet), lo (loopback), ppp0 (Point-to-Point Protocol), and wlan0 (Wireless LAN). Each interface's flags, MTU, queueing discipline, and statistics (RX/TX packets, errors, overruns, frame, carrier, collisions) are listed.

```
pi@raspberrypi:~$ sudo -i
root@raspberrypi:~# cd /etc/ppp/peers/
root@raspberrypi:/etc/ppp/peers# sudo pon rnet
root@raspberrypi:/etc/ppp/peers# nano rnet
root@raspberrypi:/etc/ppp/peers# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether b8:27:eb:10:0e:ac txqueuelen 1000  (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
          inet6 ::1 prefixlen 128 scopeid 0x10<host>
              loop txqueuelen 1000  (Local Loopback)
              RX packets 12 bytes 712 (712.0 B)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 12 bytes 712 (712.0 B)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
      inet 100.73.51.27 netmask 255.255.255.255 destination 10.64.64.64
          ppp txqueuelen 3  (Point-to-Point Protocol)
          RX packets 6 bytes 72 (72.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 7 bytes 111 (111.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.17 netmask 255.255.255.0 broadcast 192.168.1.255
```

Figura 3.18: Creación de la conexión y características de la red

Como se observa se enlistan todos los protocolos en uso además, muestra la tasa de transferencia maxima (MTU) la dirección de la mascara, los paquetes transmitidos, las colisiones la longitud de cola de transmisión del dispositivo(txqueuelen),es util establecer esto en valores pequeños para dispositivos mas lentos con una latencia alta, para evitar que las transferencias masivas rápidas perturben demasiado el trafico interactivo. Por ultimo si se desea cerrar la conexión basta con escribir en el terminal.

```
sudo poff rnet
```

3.5.4. Bloque Servidor

Este bloque se refiere al almacenamiento de los datos en un servidor libre en la nube, en este caso Firebase de Google, el sistema se comunicará directamente con este servidor para el envío de los datos con el propósito de su almacenamiento y futura presentación, también se mantendrá un registro de base de datos con las muestras de aceleraciones y sus indicadores para que posteriormente se puedan hacer nuevos análisis en las trazas de las aceleraciones en Transmilenio. A continuación se mostrarán los datos enviados a Firebase

Reg_pel:

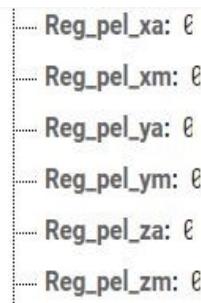


Figura 3.19: Registro de cantidad de riesgos por minuto.

Estos registros indican la cantidad total de veces en un minuto, que superan los riesgos en cada eje,
xm: x medio
xa: x alto
ym: y medio
ya: y alto
zm: z medio
za: z alto

Reg_4:



Figura 3.20: Registro con arreglo de muestras.

Cada uno de estos registros contiene un arreglo con todas las muestras de cada eje tomadas por un minuto.

Contador ty:



Figura 3.21: Tiempo de muestreo de datos.

Envía el tiempo de muestreo de los datos tomados, se envía por si en futuros cambios se desea disminuir o aumentar este tiempo.

Riesa:

```
  riesa_x: 0
  riesa_y: 0
  riesa_z: 0
  riesm_x: 0
  riesm_y: 0
  riesm_z: 0
```

Figura 3.22: Registro de muestras totales en el recorrido.

Estos registros indican las muestras totales en el recorrido en las que se infringió algún riesgo.

Tiempo:

```
+-- tiempo
```

Figura 3.23: Registro de tiempo toma de muestras.

Indica el tiempo que se utilizó para tomar las muestras.

Tiempox, tiempoy:

```
+-- tiempox
+-- tiempoy
```

Figura 3.24: Registros de duración de riesgo.

Estos dos registros son dos arreglos en los que cada uno contiene cual es la duración de cada riesgo.

3.5.5. Bloque Presentación de la Información

En este bloque finalmente, se realiza la visualización de los datos en gráficas y se desarrollan reportes cada ciertos intervalos de tiempo que informen sobre cómo se ha estado conduciendo.

Para la creación de las gráficas usaremos una librería especial en javaScript para generar diferentes tipos de gráficos.

Chart js

Chart js es una librería en JavaScript que nos permite a través de diferentes tipos de gráficos visualizar datos. Esta es una herramienta ideal para el desarrollo de una aplicación web, ya que se encuentra muy bien documentada y es completamente gratuita. Para su instalación y uso es posible descargar la ultima versión desde Github o usar el enlace CDN (Red de distribución de contenidos) para incluirla en cualquier proyecto HTML.

Creación de una Gráfica

Se presentará la creación de una gráfica de líneas la cual es necesaria para visualizar los datos del sensor. Primero se crea una variable en JavaScript que contendrá los datos de los ejes que se desean graficar. Para este ejemplo usaremos dos datasets con números aleatorios. Como se ve a continuación.

```
var chartData = {
    labels: ["Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado", "Domingo"],
    datasets: [
        {
            data: [58, 44, 48, 50, 68, 69, 63],
        },
        {
            data: [63, 46, 49, 47, 58, 63, 67],
        }
    ]
};
```

La variable chartData contiene datos aleatorios para los días de la semana, los días de la semana al ser el eje independiente irán en el eje x y los datos aleatorios irán separados de acuerdo a los valores que tomen ese día de la semana. Como se aprecia estos valores están contenidos en un arreglo. Para la configuración de los parámetros e iniciar la gráfica con JavaScript tenemos:

```
var chLine = document.getElementById("chLine");
if (chLine) {
    new Chart(chLine, {
        type: 'line',
        data: chartData,
        options: {
            scales: {
                yAxes: [
                    {
                        ticks: {
                            beginAtZero: false
                        }
                    }
                ],
                legend: {
                    display: true
                }
            }
        }
    });
}
```

Como se puede ver se genera una variable para luego ser llamada en el documento HTML, aquí en este condicional podemos programar a conveniencia los ejes y las características principales de la gráfica tales como mostrar la leyenda, agregar título y hasta los colores de las líneas.

Una vez configurado la gráfica procedemos a llamar el script el cual tiene por nombre chLine obteniendo la siguiente gráfica.

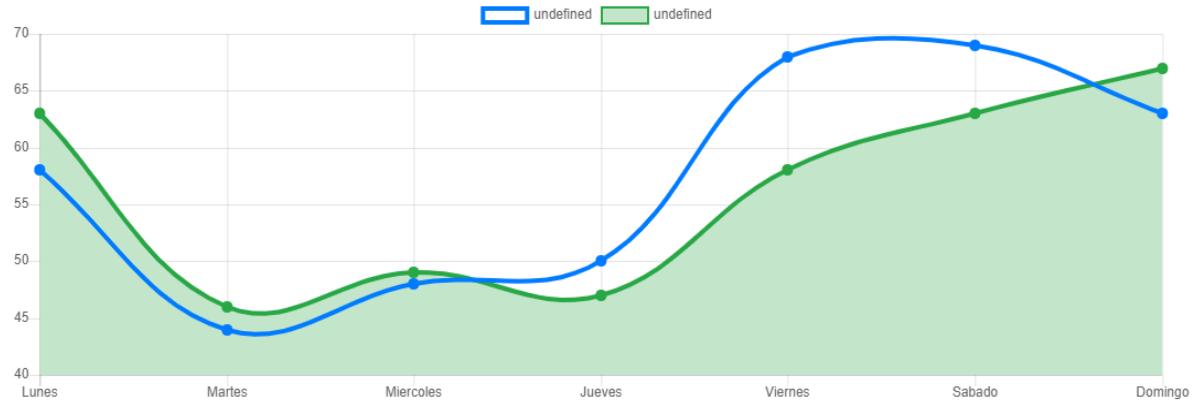


Figura 3.25: Gráfico Obtenido

Si se desea cambiar el tipo de gráfico o agregar mas datos de visualización bastaría con cambiar en el condicional 'line' por 'bar', por ejemplo, para obtener una gráfica con barras o pie, las cuales se usan en el presente proyecto. Toda la documentación de esta librería se encuentra en la pagina web "www.chartjs.org".

DashBoard

Un DashBoard se define como una representación gráfica de los principales indicadores que intervienen. Para fines de este proyecto la manera como se conduce un automotor de servicio público.

En este orden de ideas y a partir de las gráficas generadas por medio de la herramienta chart.js, es conveniente construir una Dashboard que permita una cómoda visualización de los datos.

Para la construcción del Dashboard se hace uso de la herramienta llamada Bootstrap, este framework permite diseñar las páginas mediante un sistema de columnas o rejillas que ayudan en la disposición del contenido y su adaptación a diferentes dispositivos. Dicho sistema consiste en dividir la pantalla en filas, donde cada fila puede dividirse en 12 columnas o espacios. De tal forma es posible, por ejemplo, crear un diseño con dos columnas, una que ocupe cinco espacios y otra que ocupe siete espacios.

Contenedores de Bootstrap

Este sistema se basa en contenedores flexibles que se escalan dependiendo del tamaño del dispositivo. Para ajustar los contenedores disponemos de diferentes clases. Tenemos un tipo ".container" donde el contenido aparece centrado y de un ancho fijo y el ".container-fluid" donde el contenido ocupa todo el ancho posible es decir el 100 %. Por lo tanto, el primer paso necesario para comenzar es definir un contenedor principal con cualquier tipo de clase. Dentro de este container vamos a crear filas row dentro de ellas columnas denominadas col". Seguidamente se crea un contenedor una fila que tiene tres columnas como se muestra a continuación.

```
<div class="container">
  <div class="row">
    <div class="col-4">
      Primera
    </div>
    <div class="col-4">
      Segunda
    </div>
    <div class="col-4">
      Tercera
    </div>
  </div>
</div>
```

```
</div>
</div>
</div>
```

Es importante recordar que esta estructura se usa dentro de la sección <body> del HTML y que al igual que la librería chartjs necesita que en el encabezado del archivo este el CDN correspondiente a Bootstrap el cual se encuentra junto con toda su documentación en la pagina principal del framework [34].

Capítulo 4

Protocolo de pruebas

A continuación, se presentan cada uno de los protocolos de prueba realizados para verificar el buen funcionamiento de cada uno de los bloques definidos en el capítulo anterior, además de la calibración y prueba del sistema general.

4.1. Bloque de adquisición de señales

Para verificar que las señales tomadas por nosotros sean coherentes deben tener las siguientes características: el eje x y y deben tener su centro muy cercano a 0, como se muestra en las siguientes muestras tomadas por nosotros, también se hace la comparación con una base de datos que se recolectó para un trabajo de grado anteriormente.

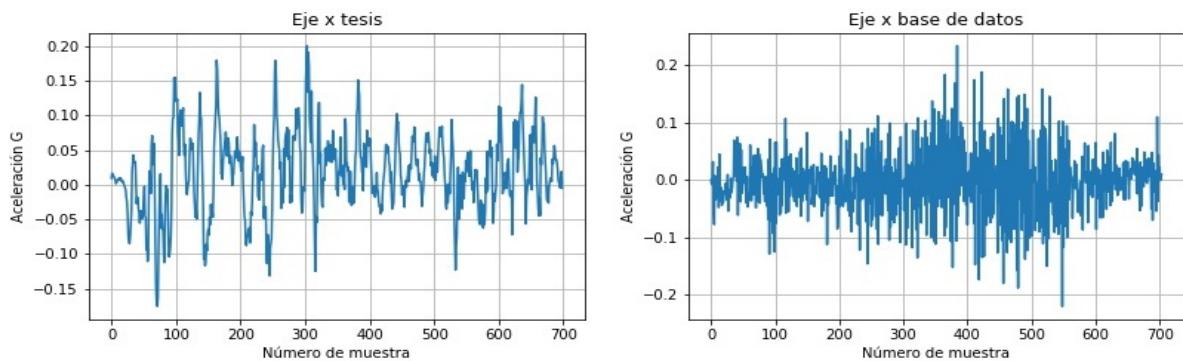


Figura 4.1: Validación eje x

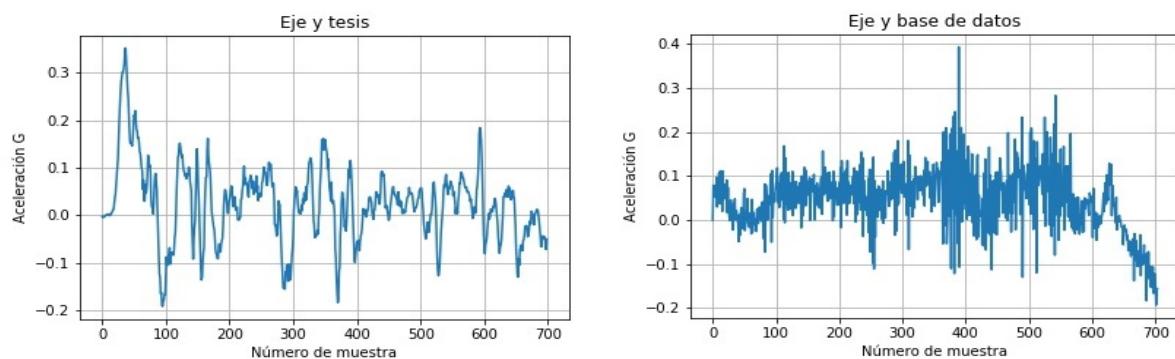


Figura 4.2: Validación eje y

Para el eje z debe tener una linea media en 0.098 tal como se muestra en la siguiente figura.

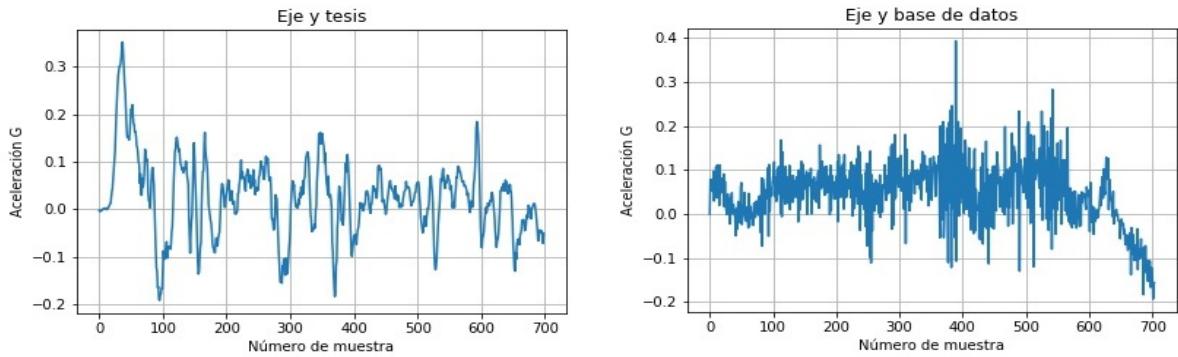


Figura 4.3: Validación eje z

Se aprecia que las señales están en una escala $\pm 2G$, debido a la programación en python, ya que esta escala puede variar en $\pm 4G, \pm 6G, \pm 8G$ y $\pm 16G$. Además, las señales son coherentes con una gráfica de 3 ejes de aceleración, ya que cumple que el eje x y y tienden a tener su centro en $0G$ y el eje z tiende a tener el centro en $0.98G$ correspondiente a la aceleración de la gravedad. También se observa que la frecuencia de muestreo es menor con respecto a la base de datos analizada ya que nuestra frecuencia es de 2.86Hz y la de la base de datos es de 30 Hz .

4.2. Bloque procesamiento de datos

Se plantea un sistema que recibe datos de aceleración, en los tres ejes coordenados, de un vehículo de transporte público en movimiento. Estos datos se refieren específicamente a cambios de carril, aceleraciones bruscas y saltos por baches, los valores numéricos, se compararán con ciertos indicadores que suponen aceleraciones de riesgo para el usuario del servicio de transporte, para luego de varias muestras, evaluar y analizar los estados y calidad de conducción prestada.

Para establecer a través de los indicadores un parámetro que permita clasificar la conducción como buena, mala o regular, se utilizó una base de datos con 51 archivos que contienen los registros de medición de aceleraciones de varios buses. Se empleó python para analizar la base de datos, posteriormente se analiza cada muestra tomada en los ejes x, y y z , se verifica el total de muestras que superaron los límites de riesgo, al dividir el número de muestras que superan un riesgo sobre el total de muestras, se obtiene el porcentaje de cuantas muestras superan un riesgo medio y un riesgo alto por cada archivo de la base de datos. Estos 51 porcentajes de la base de datos, se ordenan de menor a mayor en un arreglo para poder clasificar las futuras muestras. Si se encuentra por debajo del 50 % se considera bueno, entre el 50 % y 75 % se considera regular, por último, por encima del 75 % se considera malo. Hay que tener en cuenta que esta clasificación no asegura que el pasajero se sienta cómodo, es una comparación de manejo con respecto a otros conductores.

Primero se muestra que se considera como riesgo medio y alto para así después poder analizar la base de datos:

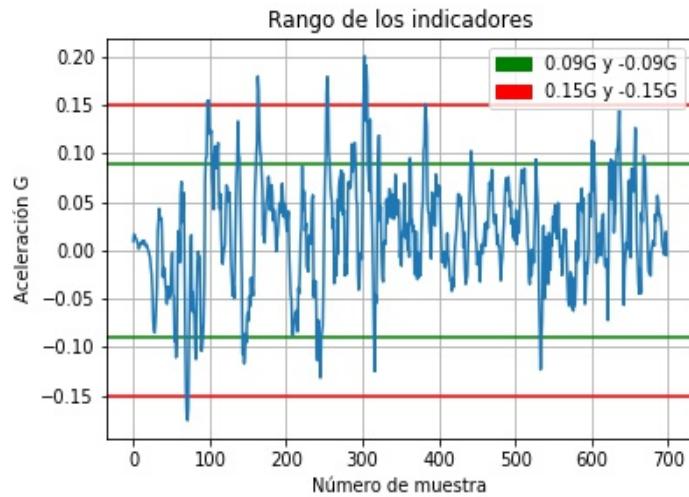


Figura 4.4: Riesgo medio y alto.

Lo que se encuentra dentro de los límites verdes no se considera como riesgo, dentro del límite verde y rojo se considera como riesgo medio y por último las muestra superiores al límite rojo es un riesgo alto.

Eje x

Para el eje x se obtuvieron los siguientes histogramas:

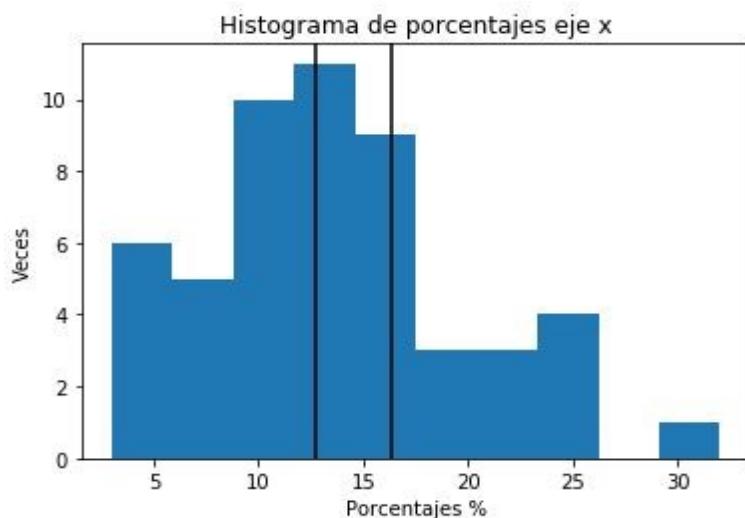


Figura 4.5: Histogramas porcentaje Eje x.

El porcentaje promedio de conducciones que superan cualquier tipo de riesgo en el eje x es de 13.4 %, se decide por medio de la gráfica:

Bueno: 0 % al 12.75 %.

Regular: 12.75 % al 16.31 %

Malo: Despu  s del 16.31 %

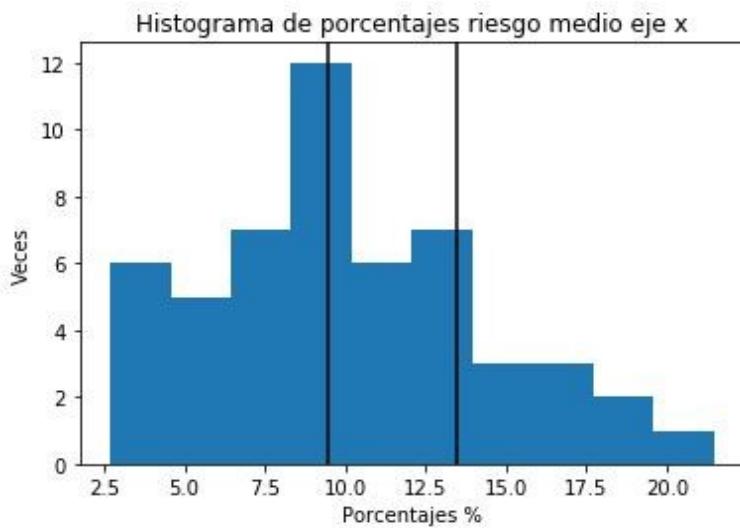


Figura 4.6: Histogramas porcentaje riesgo medio Eje x.

El porcentaje promedio de conducciones que superan el riesgo medio en el eje x es de 10 %, se decide por medio de la gráfica:

Bueno: 0 % al 9.48 %.

Regular: 9.48 % al 13.5 %

Malo: Despues del 13.5%

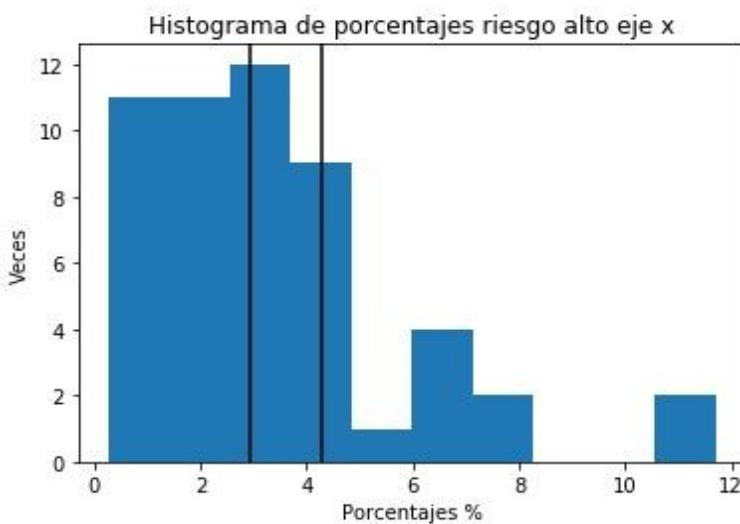


Figura 4.7: Histogramas porcentaje riesgo alto Eje x.

El porcentaje promedio de conducciones que superan el riesgo alto en el eje x es de 3.35 %, se clasifica de la siguiente manera.

Bueno: 0 % al 2.94 %

Regular: 2.94 % al 4.28 %

Malo: Despues del 4.28 %

Eje y

Para el eje y se obtuvieron los siguientes histogramas:

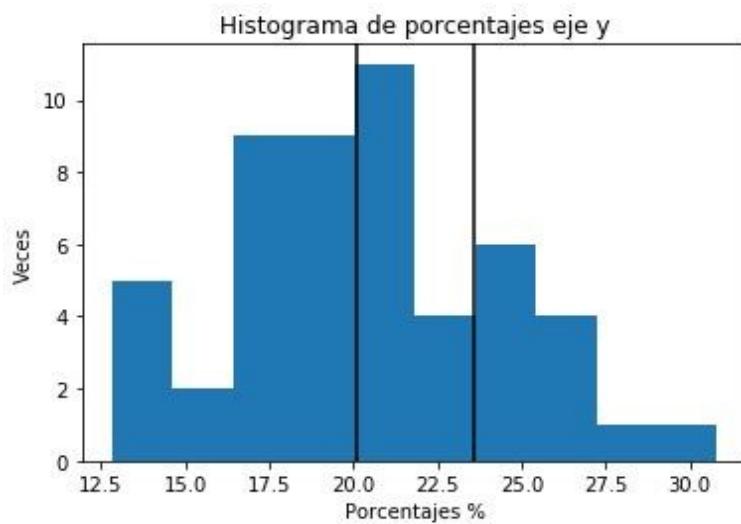


Figura 4.8: Histogramas porcentaje Eje y.

El porcentaje promedio de conducciones que superan cualquier tipo de riesgo en el eje y es de 20.34 %, se decide por medio de la gráfica:

Bueno: 0 % al 20.11 %.

Regular: 20.11 % al 23.56 %

Malo: Despues del 23.56 %

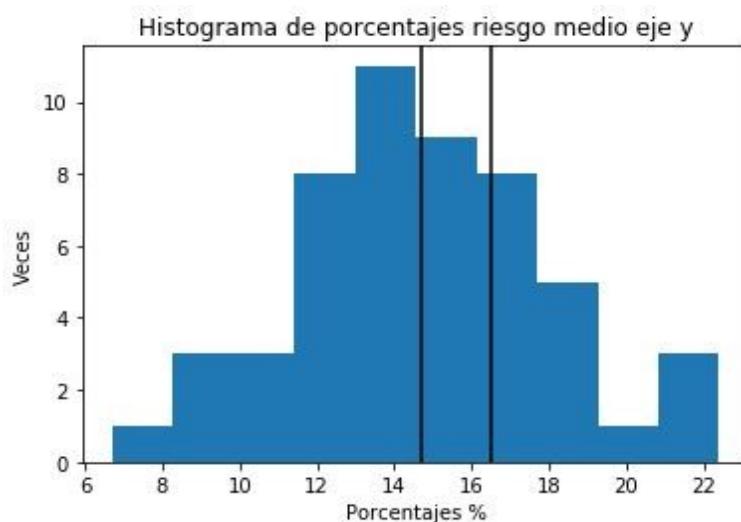


Figura 4.9: Histogramas porcentaje riesgo medio Eje y.

El porcentaje promedio de conducciones que superan el riesgo medio en el eje y es de 14.8 %, se decide por medio de la gráfica:

Bueno: 0 % al 14.69 %.

Regular: 14.69 % al 16.52 %

Malo: Despues del 16.52 %

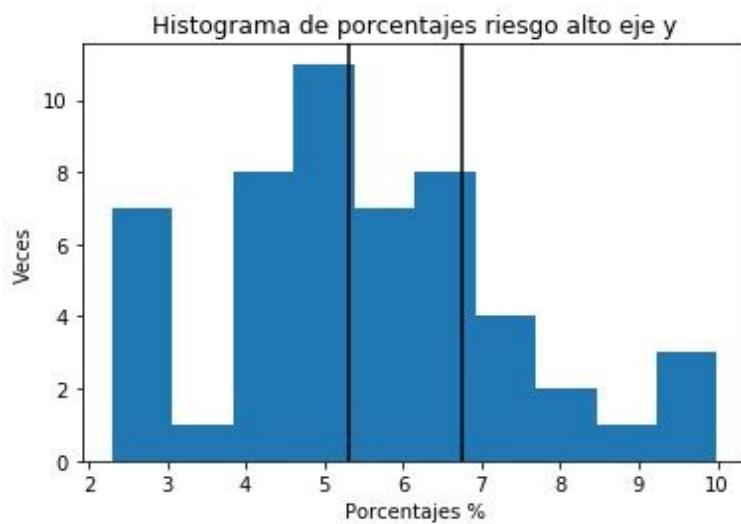


Figura 4.10: Histogramas porcentaje riesgo alto Eje y.

El porcentaje promedio de conducciones que superan el riesgo alto en el eje y es de 5.55 %, se decide por medio de la gráfica:

Bueno: 0 % al 5.3 %.

Regular: 5.3 % al 6.73 %

Malo: Despues del 6.73 %

Eje z

Para el eje z se obtuvieron los siguientes histogramas:

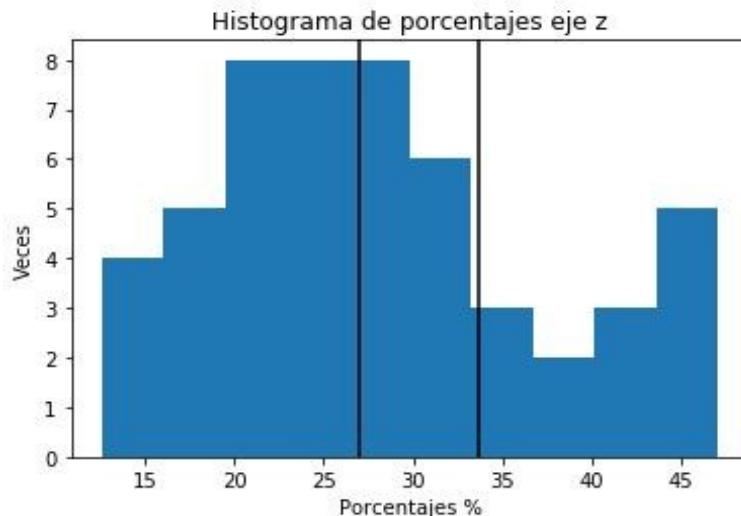


Figura 4.11: Histogramas porcentaje Eje Z.

El porcentaje promedio de conducciones que superan cualquier tipo de riesgo en el eje z es de 28.02 %, se decide por medio de la gráfica:

Bueno: 0 % al 26.99 %.

Regular: 26.99 % al 33.66 %

Malo: Despues del 33.66 %

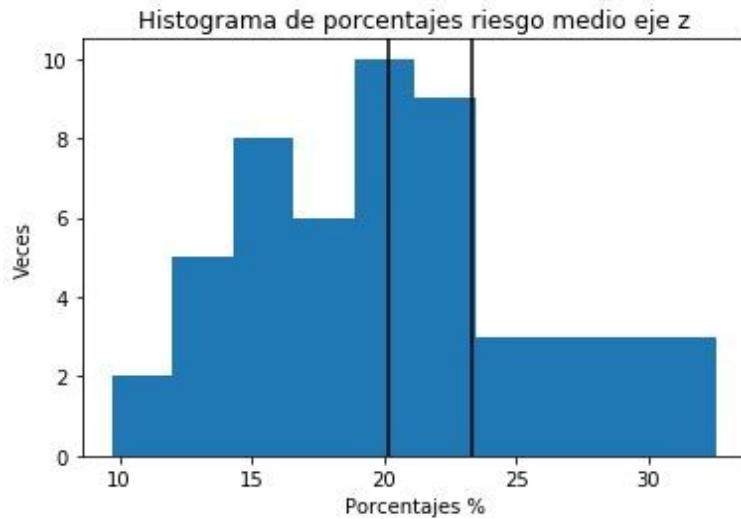


Figura 4.12: Histogramas porcentaje riego medio Eje Z.

El porcentaje promedio de conducciones que superan el riesgo medio en el eje z es de 20.37 %, se decide por medio de la gráfica:

Bueno: 0 % al 20.17 %.

Regular: 20.17 % al 23.31 %

Malo: Después del 23.31 %

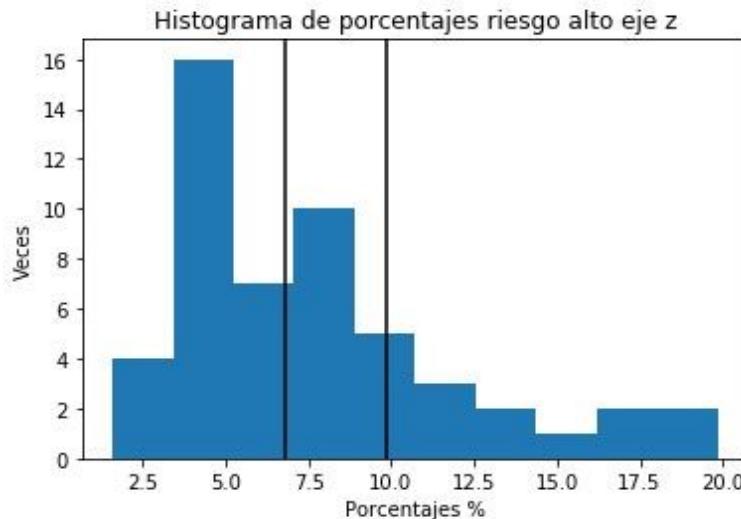


Figura 4.13: Histogramas porcentaje riego alto Eje Z.

El porcentaje promedio de conducciones que superan el riesgo alto en el eje z es de 7.65 %, se decide por medio de la gráfica:

Bueno: 0 % al 6.8 %.

Regular: 6.8 % al 9.88 %

Malo: Después del 9.88 %

4.3. Bloque Comunicación

Una vez se implementa la red, el administrador debe poder probar la conectividad de red para asegurarse de que esta funcione correctamente. Para verificar que los datos que estamos procesando están

enviándose correctamente al servidor, se crea la conexión ppp0 y se enviara un ping al servidor web donde se pretende que se almacene la información, la dirección del servidor web es conocida a priori y dada por la interfaz de la pagina:

```
https://iotdatos.firebaseio.com/
```

Al crear la conexión ppp0 y mandar el ping nos arroja los siguientes resultados.

```
pi@raspberrypi:~ $ sudo -i
root@raspberrypi:~# cd /etc/ppp/peers/
root@raspberrypi:/etc/ppp/peers# sudo pon rnet
root@raspberrypi:/etc/ppp/peers# ping www.firebaseio.com
PING www.firebaseio.com (35.201.97.85) 56(84) bytes of data.
64 bytes from 85.97.201.35.bc.googleusercontent.com (35.201.97.85): icmp_seq=1 ttl=116 time=200 ms
64 bytes from 85.97.201.35.bc.googleusercontent.com (35.201.97.85): icmp_seq=2 ttl=116 time=179 ms
64 bytes from 85.97.201.35.bc.googleusercontent.com (35.201.97.85): icmp_seq=3 ttl=116 time=497 ms
64 bytes from 85.97.201.35.bc.googleusercontent.com (35.201.97.85): icmp_seq=4 ttl=116 time=618 ms
64 bytes from 85.97.201.35.bc.googleusercontent.com (35.201.97.85): icmp_seq=5 ttl=116 time=555 ms
64 bytes from 85.97.201.35.bc.googleusercontent.com (35.201.97.85): icmp_seq=6 ttl=116 time=155 ms
...
--- www.firebaseio.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5002ms
rtt min/avg/max/mdev = 155.747/367.872/618.637/192.973 ms
root@raspberrypi:/etc/ppp/peers#
```

Figura 4.14: Ping Servidor Web

El ping confirma que se ha transmitido y recibido correctamente 6 paquetes de un tamaño de 64 bytes con cero perdidas en los paquetes y en un tiempo de 5002ms además, se observa el time to live (ttl) de cada paquete y su respectivo tiempo.

4.4. Presentación de la información

Como se mencionó en el capítulo anterior se nombro la manera de generar esquemas y como se construye un Dashboard, sin embargo, los datos con los que se graficaron fueron puestos de forma aleatoria, por lo que se debe tener a disposición los datos que se encuentran en el servidor web. Los insumos que recibe la gráfica es un vector que contiene los datos. Para obtener estos datos del servidor por medio de JavaScript mandaremos un "XMLHttpRequest" que proporciona una forma sencilla de obtener información de una URL sin tener que recargar la pagina completa. A pesar de su nombre XMLHttpRequest puede ser usado para recibir cualquier tipo de dato, no solo XML y HTTP sino que también admite formatos como file y ftp.

Como los datos almacenados en Firebase están contenidos por defecto en un formato denominado JSON(JavaScript Object Notation) un formato de texto que es completamente independiente del lenguaje, pero es ampliamente utilizado por los programadores de cualquier interprete de código permitiendo que este sea un lenguaje ideal para el intercambio de datos.

Al hacer este request a la URL que contiene los datos procesados provenientes del Raspberry Pi, datos almacenados en formato JSON, por lo que debemos poder escribirlos como vector en JavaScript, como solución se usa el método "JSON.parse" que analiza una cadena de texto como JSON transformándolo opcionalmente en el valor producido por el análisis. Esto se ilustra claramente en las siguientes lineas en JavaScript

```
var urlTxt="https://iotdatos.firebaseio.com/.json";
var jsonFile = new XMLHttpRequest();
jsonFile.open("GET",urlTxt,false);
```

```

jsonFile.send();
var data = JSON.parse(jsonFile.response);

```

Es práctico, la creación de variables para almacenar la URL del servidor web, la petición del tipo "GET" del archivo JSON , un comando booleano "false" para autorizar la extracción de los datos y la variable "datos" que contendrá finalmente el archivo transformado. Por ejemplo, para graficar el eje x,y,z se necesitan extraer 3 vectores de datos provenientes del acelerómetro, creando 3 variables con las direcciones donde se encuentra dicha información de la siguiente manera.

```

var ejeX_min = data.MyTestData.Reg_x4.ejex;
var ejeY_min = data.MyTestData.Reg_y4.ejey;
var ejeZ_min = data.MyTestData.Reg_z4.ejez;

```

En el anterior código se aprecia la variable "data" donde se encuentran los datos transformados al formato correspondiente y las demás direcciones son las rutas para obtener finalmente la información.

Para verificar que estamos obteniendo los resultados esperados del servidor web se compila el código y verificamos que la información sea idéntica y que además, se encuentre como un vector en el script, para tal fin Google Chrome permite inspeccionar por consola el valor de estas variables.

En el servidor web tenemos los siguientes datos.

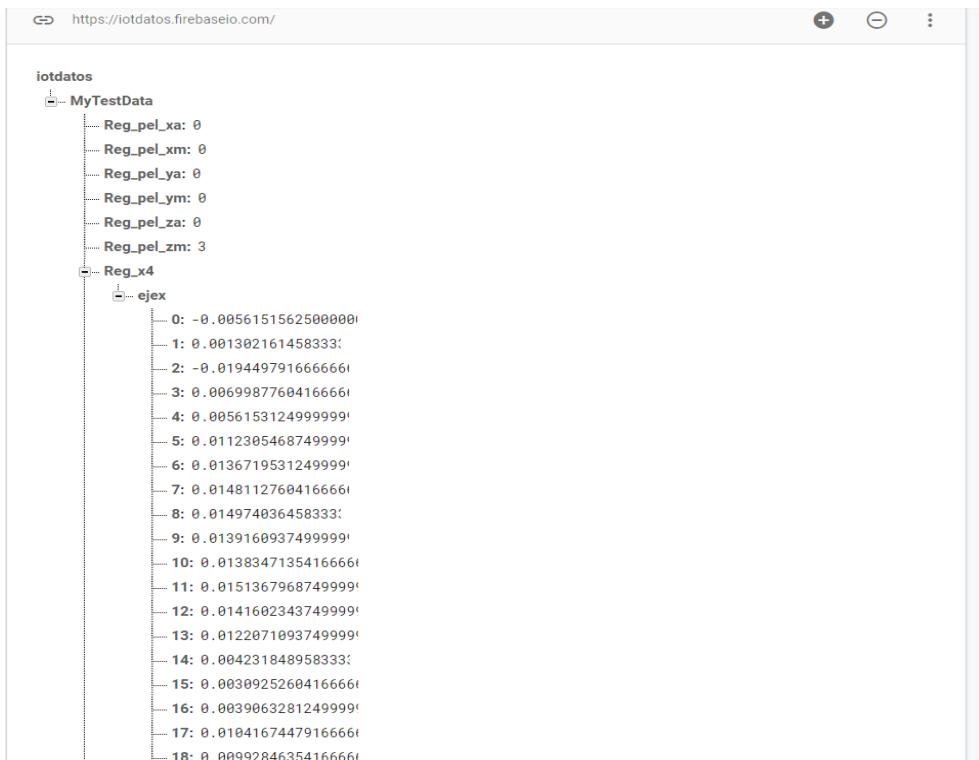


Figura 4.15: Datos en el Servidor Web

Al inspeccionar en el navegador para los datos del eje X correspondientes al acelerómetro durante 1 minuto.

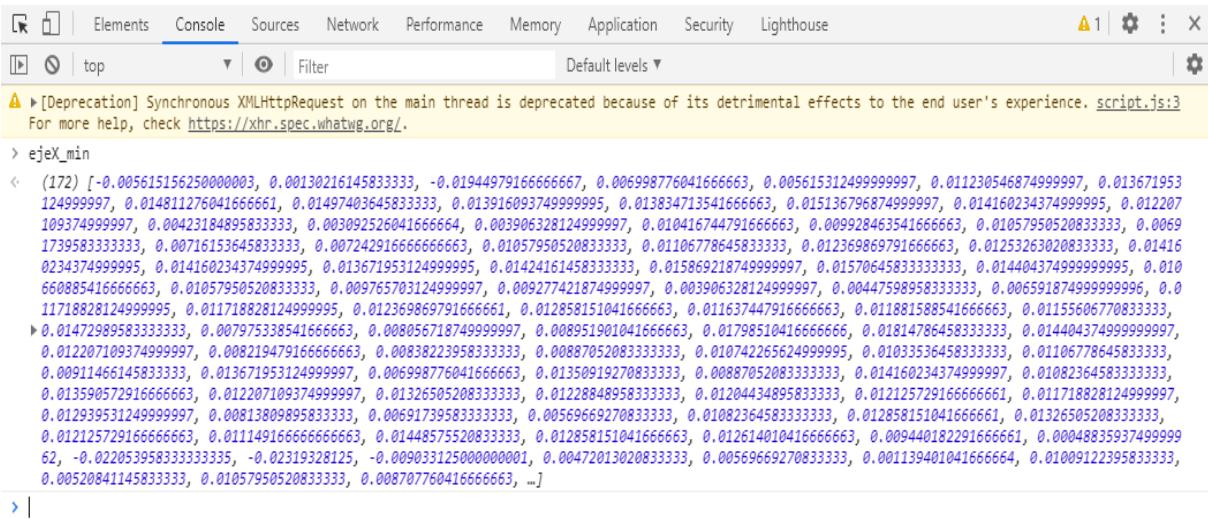


Figura 4.16: Datos Extraídos del servidor

Es evidente que los datos contenidos en el servidor son fieles al contenido de la pagina. No se colocaron los demás archivos que se sacaron debido al espacio del documento, sin embargo, en el repositorio anexo al trabajo de grado se puede ver cada variable extraída del servidor. Ya con esta información almacenada en nuestro script es posible graficar toda la información obtenida y asimismo procesar aún más la información, ya que mediante estructuras de control como lo son el condicional if se puede clasificar los datos y mostrarlos en diferentes esquemas. Estos condicionales obedecen a señales obtenidas de una base de datos hecha anteriormente en un trabajo de grado [35].

En este orden de ideas y haciendo los containers respectivos se tienen las siguientes gráficas que se analizarán mas detenidamente en la sección de resultados. Las gráficos que se obtuvieron son las siguientes.

DashBoard Indicadores

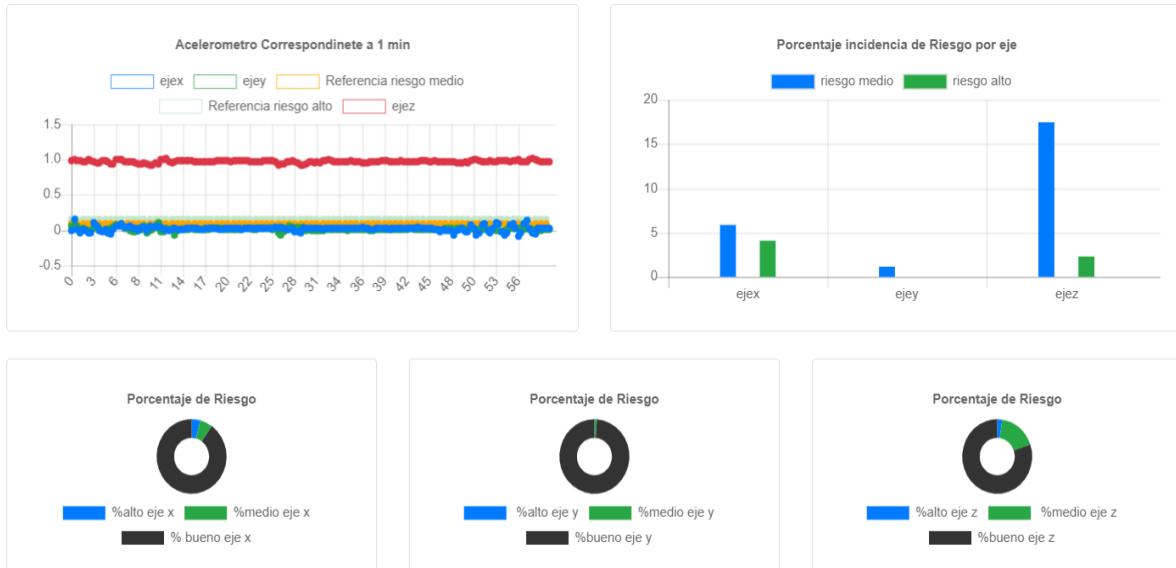


Figura 4.17: Gráficos Obtenidos navegador Web

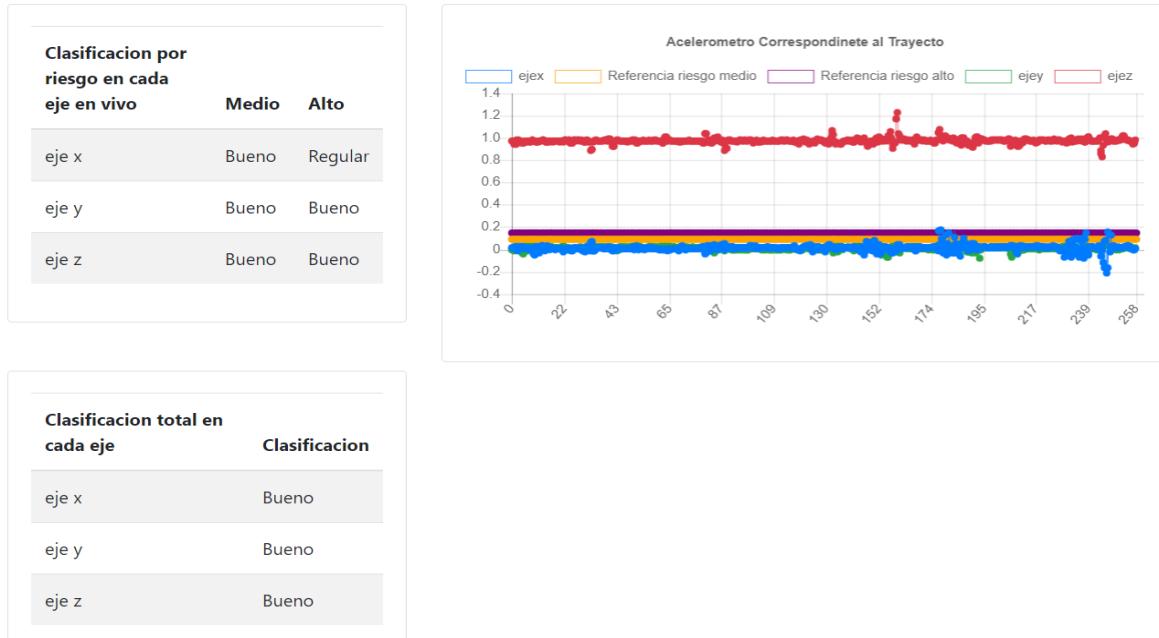


Figura 4.18: Gráficos Obtenidos navegador Web

Si se desea visualizar con mayor detenimiento, por ejemplo, los ejes obtenidos del acelerómetro, basta con ir a la leyenda y ir omitiendo los valores que se desean para visualizar mejor los datos. Ya en el siguiente capítulo se analizarán mas detenidamente estas trazas y gráficas que muestran ya un estudio tanto del recorrido general como pequeños intervalos de tiempo en vivo entorno a los indicadores de riesgo en perspectiva del conductor.

4.5. Condiciones de diseño, implementación y evaluación

Las condiciones mínimas con las que se contó para garantizar un proceso de diseño exitoso, serán numeradas a continuación.

4.5.1. Disponibilidad de espacio físico

1. Mesa
2. Silla

4.5.2. Equipo

1. Computador
2. Raspberry Pi 3 B+
3. Modulo GPRS/ GSM
4. pantalla LCD Táctil

4.5.3. Software

- Linux
- Python
- Windows 10

4.6. Calibración del sistema

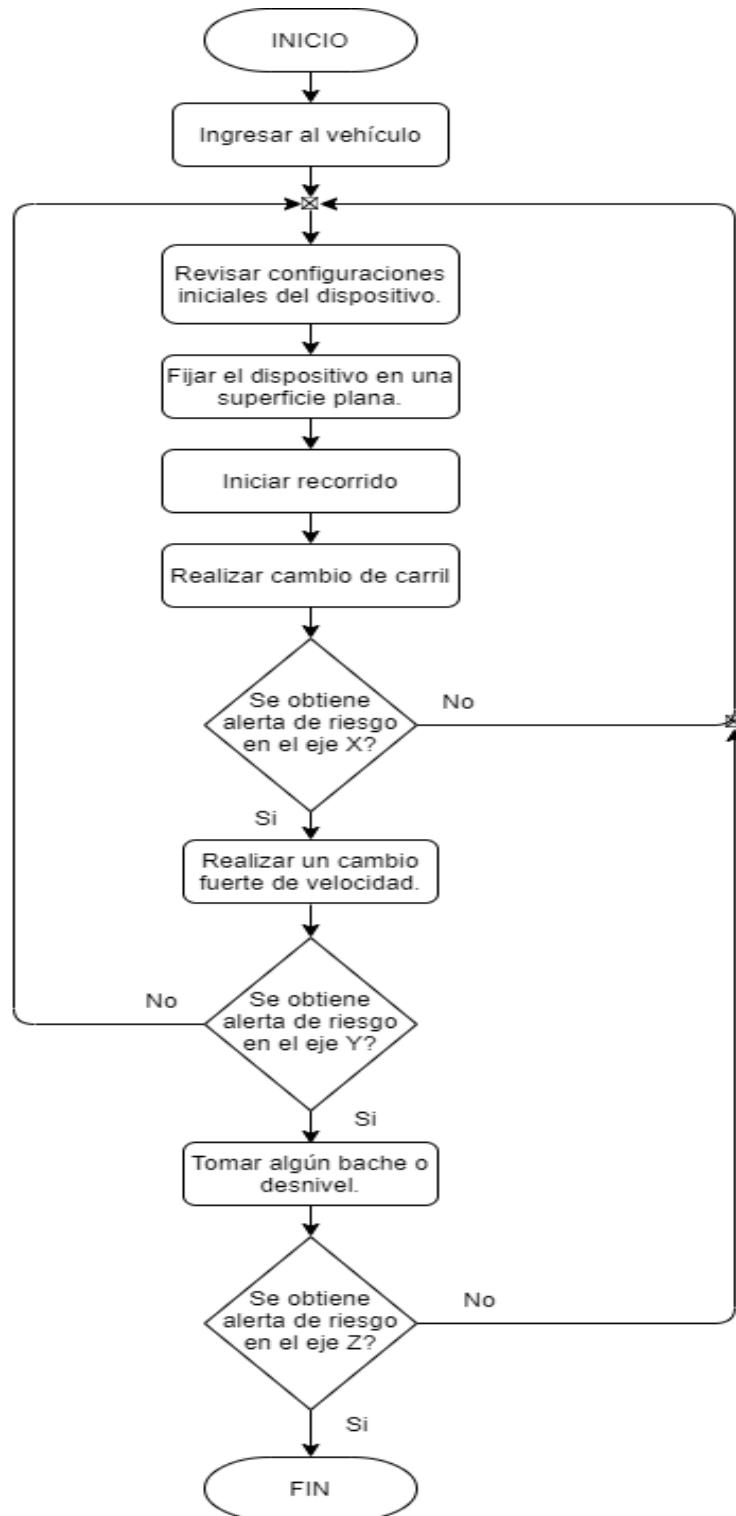


Figura 4.19: Calibración del sistema.

Capítulo 5

Funcionamiento del sistema

Una vez generado el Dashboard es importante explicar lo que se obtiene en cada gráfico. En primer lugar, como se explicó anteriormente se simplificará este proceso observando solo un eje, para constatar que la información obtenida es la correcta, como se observa en la siguiente imagen.

Dashboard Indicadores



Figura 5.1: Dashboard Mostrando eje x

A continuación analizaremos los resultados obtenidos del sensor en cada gráfica presentada.

- Gráfica de Lineas correspondiente a 1 min: En esta gráfica se observa el eje x junto con los indicadores correspondientes en ella se observa cuantas veces durante el intervalo de tiempo se supero el umbral de riesgo para ese eje. Esta primera gráfica esta mostrando el eje x, sin embargo, la gráfica permite ir añadiendo los otros ejes sin ningún problema, con el fin de una cómoda visualización.
- Gráfica de barras: Esta gráfica ilustra en el eje independiente los riesgos de cada eje durante un minuto de trayecto y asimismo lo compara con el porcentaje obtenido para ese intervalo de tiempo. Si se comparan ambas gráficas vemos que la frecuencia con la que incurre este eje en el riesgo medio es mucho mayor que la frecuencia en la que incurre en un riesgo alto congruente con la primera gráfica donde se ven como están posicionados los picos generadores de riesgo.
- Las gráficas correspondientes a los pie's o ruedas muestran el porcentaje de incidencia de riesgo por eje.

Siguiendo con las gráficas obtenidas se tiene otro grupo de esquemas que corresponden en gran medida al análisis de la base datos ya mencionada.



Figura 5.2: Dashboard Clasificación

Para la única gráfica de este grupo de esquemas del dashboard es equivalente que la primera mostrada con la diferencia que esta gráfica ilustra el trayecto completo y no solo durante un minuto, este es importante con el fin de mirar el comportamiento que tuvo el auto bus todo el tiempo, esta representación de igual forma permite visualizar todos los ejes con su respectiva referencia en caso de sobrepasar algún riesgo observando a través del gráfico un historial del eje seleccionado.

Por otro lado, al clasificar tanto el recorrido del auto bus como el intervalo de tiempo durante un minuto. Se clasifican los ejes durante cada minuto dependiendo de su riesgo alto o medio, esta clasificación se hizo de acuerdo con los porcentajes antes mencionados en relación con la base de datos.

Capítulo 6

Conclusiones y recomendaciones

A medida que avanzaba el proyecto fue notable el rendimiento que añadía al sistema la adaptación de las señales obtenidas por el acelerómetro en los ejes X,Y,Z, puesto que supone un pre-procesamiento del cual resultan en primera etapa los indicadores, si este procesamiento de datos y señales no se realizara antes de la comunicación, la información que se transmitiría consumiría más espacio, supondría mayor costo de red y no resultaría útil ni para el desarrollador del sistema, ni para el usuario.

Para obtener mayor cantidad de muestras y que el proyecto tenga más soporte estadístico, es recomendable acrecentar el numero de nodos y/o dispositivos empleados, lo cual proporcionará más información para compararla con otros conductores, generando una base de datos mas robusta para su análisis.

En cuanto al análisis producto de la comparación de los resultados obtenidos en la base de datos con los tomados en el bus, se observó la existencia de offsets en las señales del acelerómetro debido a los desniveles en el pavimento o las incidencias que estos sobresaltos puedan generar en el bus. Para mitigar esto, es recomendable conocer a priori la ruta objeto de análisis, para según las circunstancias, calibrar el sensor.

De otra parte, se evidenció que los alcances del proyecto apuntan hacia la integración de diversos protocolos de comunicación y que estos puedan ser modificados, ya sea para mejorar su velocidad de respuesta o para un mayor rango de seguridad al usuario.

Este proyecto no genera costos adicionales a los correspondientes al valor del uso de la red celular hacia internet. Lo cual es bien relevante, teniendo en cuenta que el software empleado es totalmente gratuito y ampliamente documentado.

Es recomendable para futuros trabajos implementar una tecnología superior a la 2G/GPRS, que aunque tiene una amplia cobertura a nivel nacional, no posee altas velocidades de transmisión como podría ofrecerla una red 3G y 4G,lo cual mejoraría los niveles de respuesta y seguridad del proyecto, esto debido a que en una ciudad como Bogotá estas redes no presentan problemas de cobertura.

Durante el desarrollo de la aplicación web fue vital tener en cuenta la cómoda visualización del usuario, por lo que el uso de frameworks como Bootstrap resultó muy útil, además, permite al usuario visualizar la página desde cualquier dispositivo, ya sea ordenador, celular, tablet o incluso televisor; de igual modo, el manejo de lenguajes de programación como JavaScript posibilita agregar diferentes dinámicas a los esquemas empleados como también la manipulación, clasificación etc., para darle un sentido mas profundo a los datos contenidos en el servidor.

Durante la sección del protocolo de pruebas se corroboró una alta correlación entre el número de muestras y el de veces que se entra en un indicador de riesgo, lo que es equivalente a enviar el número de muestras y el de veces en que se incurre en un indicador de riesgo.

El desarrollo del producto fue satisfactorio, ya que recopila y analiza las señales en los tiempos esperados y además, es congruente con el marco de referencia físico tomado en los ejes coordenados.

La falta de una base de datos propia, por la pandemia del COVID-19, impactó un poco la dinámica una base de datos propia; sin embargo, esta circunstancia se sorteó con una base de datos previamente hecha en un trabajo de grado titulado “Análisis de patrones de conducción de vehículos”, que permitió

concluir exitosamente el proyecto.

Restó escalar a una automatización más compleja del dispositivo, por ejemplo, que fuera capaz de ajustarse automáticamente en trayectos con puentes vehiculares, también encuestar a los usuarios y/o conductores acerca de su experiencia, procurando la mínima subjetividad y luego contrastarlos con los datos obtenidos.

En el futuro, podría complementarse tomando en cuenta el aforo de usuarios habituales midiendo en un periodo de tiempo su nivel de satisfacción, información valiosa para la mejora del servicio prestado por cualquier operador de Transmilenio.

Capítulo 7

Bibliografía

- [1] A. Martínez Olmedo, “Sistema de transporte urbano masivo de pasajeros de Bogotá Transmilenio”, Urbano, vol. 6, n.º 7, pp. 31-37, ene. 2003.
- [2] A. Loaiza, «www.sura.com,» 01 Enero 2015. [En línea]. Available: <http://www.sura.com/blogs/autos/accidentes-transito-colombia.aspx>. [Accessed:: 05 Septiembre 2015].
- [3] ”Las quejas más comunes de los bogotanos sobre el transporte público”, Publimetro Colombia, 2019. [En línea]. Available: <https://www.publimetro.co/co/loultimo/2015/07/20/quejas-mascomunes-bogotanos-transportepublico.html>. [Accessed: 08- Sep- 2019].
- [4] ”Visor PDF – transmilenio”, Transmilenio.gov.co, 2019.[Online].Available: <https://www.transmilenio.gov.co/loader.php?lServicio=Tools2&Tipo=descargasIFuncion=visorpdfid=13651&p=df=1>. [Accessed: 23- Sep- 2019].
- [5] “ANÁLISIS DE PETICIONES, QUEJAS, RECLAMOS Y SOLUCIONES.”, Bogotá: Transmilenio S.A, 2019, p. 2.Available:<https://www.transmilenio.gov.co/publicaciones/150504/publicaciones>[Accessed: 23- Sep- 2019].
- [6] C. Tiempo, ”Trancones y omisión de paradas, detrás de las demoras en rutas zonales”, El Tiempo, 2019. [Online]. Available: <https://www.eltiempo.com/bogota/trancones-y-omisiondeparadas-detrases-demoras-en-rutas-zonales-del-sitp-295800>. [Accessed: 22- Sep- 2019].
- [7] C. C. S.A., “La Seguridad Viaja en Autobus”, Instituto Biomedica de Valencia, 2012. [Online].Available: <https://riunet.upv.es/handle/10251/38499> [accessed; 23- Sep- 2019].
- [8] Oracle. ”¿Qué es Internet of Things (IoT)?”. [Online]. Available: <https://www.oracle.com/co/internet-of-things/what-is-iot.html>. [Accessed: 23- May- 2018]
- [9] UIT-T-REC-Y.2060, ”Descripción general de Internet de los objetos,”2012.
- [10] Gerber, A., 2020. Simplifique El Desarrollo De Sus Soluciones De Iot Con Arquitecturas De Iot.. [online] IBM Developer. Available at: <<https://developer.ibm.com/es/articles/iot-lp201-iot-architectures/>> [Accessed 9 May 2020].
- [11] Pastor, J., 2020. Raspberry Pi 3 Model B+, Análisis: Más Potencia Y Mejor Wifi Para Un Minipc Que Sigue Asombrando. [online] Xataka.com. Available at: <<https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando>>[Accessed 16

April 2020].

- [12] Raspberrypi.org. 2020. GPIO - Raspberry Pi Documentation. [online] Available at: <<https://www.raspberrypi.org/documentation/usage/gpio/>>[Accessed 13 March 2020].
- [13] Firebase. 2020. Firebase. [online] Available at: <<https://firebase.google.com/?hl=es-419>>[Accessed 17 February 2020].
- [14] Giraldo, V., 2019. ¿Qué Es Firebase Y Para Que Nos Sirve En El Desarrollo De Aplicaciones?. [online] Rock Content. Available at: <<https://rockcontent.com/es/blog/que-es-firebase/>>[Accessed 16 March 2020].
- [15] Naylampmechatronics.com. 2016. Tutorial MPU6050, Acelerómetro Y Giroscopio. [online] Available at: <https://naylampmechatronics.com/blog/45_Tutorial – MPU6050 – Aceler%C3%B3metro – Giroscopio.html> [Accessed 16 March 2020].
- [16] Módulo MPU6050, G., 2020. Módulo MPU6050, Acelerómetro, Giroscopio I2C. [online] Naylamp Mechatronics - Perú. Available at: <<https://naylampmechatronics.com/sensores-posicion-iniciales-gps/33-modulo-mpu6050-acelerometro-giroscopio-i2c.html>>[Accessed 28 June 2020].
- [17] Universidadviu.com. 2020. Sistemas GPRS, Concepto General Y Usos Principales | VIU. [online] Available at: <<https://www.universidadviu.com/sistemas-gprs-concepto-general-usos-principales/>> [Accessed 28 June 2020].
- [18] Ross, K., 1996. The PPP Protocol. [online] Net.t-labs.tu-berlin.de. Available at: <<https://www.net.t-labs.tu-berlin.de/teaching/computerNetworking/05,08.htm>> [Accessed 15 February 2020].
- [19] Rathor, R., 2018. Point-To-Point Protocol (PPP). [online] Tutorialspoint.com. Available at: <<https://www.tutorialspoint.com/point-to-point-protocol-ppp>>[Accessed 21 February 2020].
- [20] Rathor, R., 2018. Point-To-Point Protocol (PPP). [online] Tutorialspoint.com. Available at: <<https://www.tutorialspoint.com/point-to-point-protocol-ppp>>[Accessed 21 February 2020].
- [21] Neosoft.es. 2019. ¿Qué Es Una Aplicación Web? – Blog Neosoft Sistemas. [online] Available at: <<https://www.neosoft.es/blog/que-es-una-aplicacion-web/>>[Accessed 13 January 2020].
- [22] Ibm.com. 2020. IBM Knowledge Center. [online] Available at: <https://www.ibm.com/support/knowledgecenter/es/SSMKHH_9,0,0/com.ibm.etools.mft.doc/ac55710.htm> [Accessed 8 January 2020].
- [23] Design Code Solutions. 2020. DCS | Aplicaciones Y Servicio Web. [online] Available at: <<https://designcodesolutions.com/servicios/desarrollo-web/aplicaciones-y-servicio-web/>>[Accessed 18 April 2020].
- [24] C. Durán, "La guerra del centavo". Colombia: Ciro Durán, 1985.
- [25] M.A, Salamanca Ríos, J.F, Mora Martínez, .Análisis de patrones de conducción de vehículos”, proyecto de grado, Ing. electrónica., Universidad Javeriana, Bogotá, Cundinamarca, Colombia, 2016
- [26] Delton Martin, Dale Litwhiler, “AN INVESTIGATION OF ACCELERATION AND JERK PROFILES OF PUBLIC TRANSPORTATION VEHICLES”, Pennsylvania State University Berks,

Pensylvania, 2008.

- [27] C. C. S.A., “La Seguridad Viaja en Autobus”, Instituto Biomedica de Valencia, 2012.
[Online]. Available: <https://riunet.upv.es/handle/10251/38499> [accessed; 23- Sep- 2019]
- [28] Delton Martin, Dale Litwhiler, “AN INVESTIGATION OF ACCELERATION AND JERK PROFILES OF PUBLIC TRNASPORTATION VEHICLES”, Pennsylvania State UniversityBerks, Pensylvania, 2008.
- [29] Firebase. 2020. Cálculos De Tamaño De Almacenamiento | Firebase. [online] Available at: <<https://firebase.google.com/docs/firestore/storage-size?hl=es-419>>[Accessed 16 April 2020].
- [30] Circuit Basics. 2020. Basics Of The I2C Communication Protocol. [online] Available at: <<https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>>[Accessed 14 March 2020].
- [31] Robots-argentina.com.ar. 2018. Descripción Y Funcionamiento Del Bus I2C | Robots Didácticos. [online] Available at: <<http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-delbus-i2c>>[Accessed 23 April 2020].
- [32] Itead.cc. 2016. RPI SIM800 GSM/GPRS ADD-ON V2.0 - ITEAD Wiki. [online] Available at: <[https://www.itead.cc/wiki/RPISIM800GSM/GPRSADD-ONV 2,0](https://www.itead.cc/wiki/RPISIM800GSM/GPRSADD-ONV_2,0)>[Accessed 26 February 2020].
- [33] Linux.die.net. 2012. Pppd(8): Point-To-Point Protocol Daemon - Linux Man Page. [online] Available at: <<https://linux.die.net/man/8/pppd>>[Accessed 28 February 2020].
- [34] Otto, M., 2020. Overview. [online] Getbootstrap.com. Available at: <<https://getbootstrap.com/docs/4.4/layout/overview/>>[Accessed 23 May 2020].
- [35] M.A, Salamanca Ríos, J.F, Mora Martínez, .Análisis de patrones de conducción de vehículos”, proyecto de grado, Ing. electrónica., Universidad Javeriana, Bogotá, Cundinamarca, Colombia, 2016

Capítulo 8

Anexos

https://github.com/danizu001/TdG_DanielGonzalez_CarlosMedina