



UNIVERSIDAD TÉCNICA DE BABAHOYO

FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA

PROCESO DE TITULACIÓN

MAYO – OCTUBRE 2018

PROPUESTA TECNOLÓGICA DE GRADO O DE FIN DE CARRERA

PROPUESTA TECNOLÓGICA

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

TEMA:

**SISTEMA DE INFORMACIÓN PARA MONITOREO Y CONTROL DE PARADAS DE
BUSES EN LA CIUDAD DE BABAHOYO**

EGRESADO:

JOSÉ GREGORIO CHERO MALAGÓN

TUTOR:

ING. ANGEL ESPAÑA LEON

AÑO 2018

ÍNDICE GENERAL

INTRODUCCIÓN	1
CAPITULO I.....	3
DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS	3
1. Ámbito de aplicación: Descripción del contexto y ámbito de interés.....	3
2. Establecimiento de requerimientos	4
2.1 Propósito.....	4
2.2 Alcance.....	4
2.3 Metodología.....	5
2.4 Arquitectura.....	5
2.5 Restricciones.....	6
2.6 Requisitos Funcionales.....	6
3. Justificación del requerimiento a satisfacer	8
CAPITULO 2	9
DESARROLLO DEL PROTOTIPO	9
1. Definición del prototipo tecnológico.....	9
2. Fundamentación teórica del prototipo.....	10
2.1 Metodología.....	10
3. Objetivos del prototipo	21
3.1 Objetivo general.....	21
3.2 Objetivos específicos.....	21
4. Diseño del prototipo	22
4.1. Fase de análisis.....	22
4.2. Fase de diseño.....	22
Diagramas de Caso de uso del sistema:	23
Diagrama de Clases.....	24
Diagrama de Actividades.....	25
Diagrama de secuencia.....	26
Diagrama de Despliegue	27
Diagrama de Conexión.....	28

Esquema de Conexión.....	29
4.3. Fase de desarrollo	30
5. Ejecución y/o ensamblaje	32
Capítulo 3.....	45
1. Plan de Evaluación.....	45
1.1. Funcionalidad y facilidad de uso.....	45
1.2. Estabilidad.	46
1.3. Compatibilidad.....	47
1.4. Interoperabilidad.	48
2. Resultados de evaluación.....	49
2.1. Análisis de resultados.	50
3. Conclusiones y Recomendaciones.....	51
3.1. Conclusiones.....	51
3.2. Recomendaciones.....	52
Bibliografía	53
Anexos:.....	55

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Placa Arduino Mega	15
Ilustración 2: Módulo Sim 808.....	16
Ilustración 3: Protoboard.....	17
Ilustración 4: Resistencias	17
Ilustración 5: Leds	18
Ilustración 6: Sensor Ultrasónico	19
Ilustración 7: Micro Servo SG90	20
Ilustración 9: Diagrama de caso de uso de acción de usuario.....	23
Ilustración 10: Diagrama de Clase	24
Ilustración 11: Diagrama de actividades	25
Ilustración 12: Diagrama de Secuencia.....	26
Ilustración 13: Diagrama de Despliegue	27
Ilustración 14: Diagrama de conexión	28
Ilustración 15: Esquema de Conexión	29
Ilustración 16: Interfaz de MIT App Inventor.....	30
Ilustración 17: Interfaz de Firebase	31
Ilustración 18: Interfaz de Arduino	31
Ilustración 19: Prueba de Sensor Ultrasónico.....	32
Ilustración 20: Conexión sensor ultrasónico y micro servo	32
Ilustración 21: Conexión Modulo Bluetooth	33
Ilustración 22: Interfaz inicio de sesión	33
Ilustración 23: Interfaz inicio	34
Ilustración 24: Consultar longitud latitud.....	34
Ilustración 25: Conexión por bluetooth	35
Ilustración 26: Interfaz Información aplicación	35

ÍNDICES DE TABLAS

Tabla 1: Rol Programador y Analista	5
Tabla 2: Funcionalidad y facilidad de uso.....	45
Tabla 3: Estabilidad.....	46
Tabla 4: Compatibilidad.....	47
Tabla 5: Interoperabilidad.....	48
Tabla 6: Resultados de evaluación	49

INTRODUCCIÓN

En la Ciudad de Babahoyo el año pasado se creó una Empresa Pública Municipal Terminal Terrestre, Tránsito y Seguridad Vial (Transvial), para realizar tareas asignadas a la administración y operación de tránsito de la ciudad, para mejorar el control de vehículos y reducir los accidentes de tránsito en esta ciudad. (Hora, 2017).

La tecnología cada día evoluciona y está a cada vez más a nuestro alcance para facilitarnos la vida, el avance de la tecnología nos ayuda a adaptarnos a nuestro entorno, tanto que prácticamente puede llegar a ser hasta indispensable, la tecnología móvil ofrece a los usuarios diferentes experiencias y nos ha facilitado la comunicación, son fáciles de usar y son pocas las personas que no poseen un dispositivo móvil, la tecnología móvil tiene varios usos dependiendo también de su capacidad y aplicaciones que pueden ser de sencillas hasta robustas.

La Cooperativa de Transportes Urbano de pasajeros Santa Rita fue fundada en el año 1996 el presidente de la Cooperativa Jaime Barragán, afirmó que tuvieron muchos tropiezos en todo el trayecto desde su fundación hasta la actualidad e inclusive problemas con los usuarios, pero han logrado salir adelante, también afirmó que desean seguir mejorando su servicio para beneficio de la colectividad. (Hora, LaHora, 2018)

La presente propuesta tiene como finalidad ayudar a controlar y monitorear a los buses de transporte público de la ciudad de Babahoyo, mediante el diseño de un sistema de información, que permita evitar que los buses se estacionen y abran sus puertas en paradas no establecidas, para ello el sistema se diseñará usando la plataforma Arduino y para monitorear se usará el Sistema de Posicionamiento Satelital (GPS), para verificar y observar el lugar del bus monitoreado a través de un mapa, todos estos datos serán observados a través de una aplicación móvil.

Esto contribuirá a mejorar el tránsito vehicular en las horas pico, evitando molestias del congestionamiento causado por los buses de transporte público, ya que los vehículos solo abrirán sus puertas en paradas establecidas, y los pasajeros deberán esperarlos solo en dichas paradas.

En este proyecto se utilizará la línea de investigación desarrollo de sistemas de la información, comunicación y emprendimientos empresariales y tecnológicos de la Facultad de Administración, Finanzas e Informática (FAFI), también se hará uso de la sublínea de investigación de desarrollo de sistemas informáticos de la Carrera Ingeniería en Sistemas

CAPITULO I

DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS

1. **Ámbito de aplicación: Descripción del contexto y ámbito de interés.**

En la ciudad de Guayaquil los ingenieros electrónicos mencionados sistemas industriales Pedro Alzamora Ramírez y Alex Bautista Ramírez propusieron controlar y monitorear el recorrido de los buses de transporte público mediante GPS y GSM, con el motivo de controlar el recorrido de los buses mediante coordenadas emitidas por el GPS y visualizadas en una página web, en esta propuesta utilizaron el módulo GSM para transmitir información mediante este servicio enviando mensajes de texto los datos referentes a la longitud, latitud y estado del bus a un receptor para que se encargue de introducirla de forma serial al computador base donde se aloja la aplicación web.

La Autoridad de Tránsito Municipal (ATM) de la ciudad de Guayaquil controlará los buses urbanos implementando dispositivos GPS en las unidades de las Cooperativas de Transporte Público, e inclusive las Cooperativas de Transporte Comercial podrán incorporar este servicio, la ATM pretende monitorear estos buses para regular el cumplimiento de los recorridos y para los infractores serán sancionados con una multa, la ATM también pretende que los vehículos sujetos a este control, detectar y sancionar a los choferes que mantengan las puertas abiertas cuando el automotor este en marcha. (Universo, 2017).

En la ciudad de Babahoyo circulan 2 empresas de transporte públicos, las cuales son la Cooperativa de Transportes Santa Rita y la Cooperativa de Transportes Fluminense, tanto estas cooperativas como los señores usuarios de esta, hacen caso omiso a los paraderos establecidos, se estacionan donde desean con el único fin de recoger pasajeros, esto genera molestias a otros conductores sin importarles si están o no bloqueando intersecciones, generando discusiones entre conductores, llegando a insultos e inclusive agredirse físicamente.

Cuando dejan un pasajero en media calle, se bajan corriendo o como dicen al vuelo, sin darle importancia al peligro de ser atropellado. De la misma manera es al coger el pasajero si este está a media cuadra se estacionan donde más les place, y esperan a que el pasajero llegue corriendo y se suba. Al momento de marcar en una maquina con reloj, lo hacen apurados, diciéndoles a los encargados como comúnmente se los conoce como controladores o tarjeteros que se apuren, o en el caso de ir con buen tiempo tienden a estacionarse a conversar con el controlador.

Muchos usuarios se molestan porque al recoger pasajeros en cualquier sitio, puede ser peligroso para ellos ya que muchos delincuentes se aprovechan de eso, para cometer asaltos haciéndose pasar como pasajeros.

2. Establecimiento de requerimientos

2.1 Propósito

Este documento tiene como propósito, definir los requerimientos necesarios para el diseño de un sistema geo referenciado para monitoreo y control de puertas de buses en las paradas establecidas en la ciudad de Babahoyo, que les permitirá controlar a los buses de transporte publico de esta ciudad, y evitar congestionamiento vehicular.

2.2 Alcance

En la ciudad de Babahoyo, Provincia de los Ríos este proyecto propone automatizar las puertas de los buses sean a través de un sensor magnético, un sensor emisor estará ubicado en la puerta del bus y otro receptor estará ubicado en las paradas establecidas, en cada parada se abrirán las puertas.

Para el monitoreo usaremos la tecnología GPS, para poder obtener la localización o ubicación específica del bus.

El módulo Wifi servirá para poder enviar los datos a al servidor de Base de datos con las coordenadas específicas de cada parada.

Personal involucrado

Nombre	José Chero
Rol	Programador, Diseñador y Analista de requerimientos.
Categoría profesional	Ingeniero en Sistemas.
Responsabilidades	Análisis y especificación de requerimientos, Diseño de base datos.

Tabla 1: Rol Programador y Analista

Desarrollado por: José Chero

Luego de haber realizado un análisis se logró determinar que la aplicación móvil necesitara como mínimo las siguientes características:

- Se debe monitorear a los buses en tiempo real mediante GPS.
- Generar reportes referentes a la hora que las puertas se abren y constatar mediante GPS.
- El sistema debe poseer una interfaz amigable y de fácil manejo para el o los usuarios.

El usuario Administrador se encargará de monitorear los buses y generar reportes.

2.3 Metodología

Para el desarrollo de la aplicación móvil, será necesario usar la metodología ágil Mobile-D ya que es una metodología ágil y está basada en metodologías como: Crystal Methodologies y Rational Unified process, este modelo permitirá llevar un correcto seguimiento de desarrollo gracias a sus fases para poder entregar una aplicación publicable y entregable al cliente en poco tiempo.

2.4 Arquitectura

Se ha decidido hacer uso de una arquitectura Cliente – Servidor, puesto que la aplicación móvil consumirá los recursos de su servidor de base de datos. Este patrón de diseño es el más utilizado, además se debe implementar una base de datos para llevar el registro de la hora, fecha, posicionamiento, según las peticiones que haga el cliente, estas serán recopiladas en una ventana, para ser extraídas o poder almacenarlas en la base de datos.

2.5 Restricciones

La aplicación se desarrollará mediante el software MIT App Inventor 2, ya que es fácil de usar y permite desarrollar aplicaciones de forma rápida y este software posee un servidor de base datos llamado FirebaseDB, es de código abierto por lo que no es necesario en gastar dinero para licenciamiento de estos programas a excepción de FirebaseDB que nos permite probar una versión de prueba que nos da un gran espacio de almacenamiento pero si nos excedemos será necesario pagar para tener más espacio.

Para poder usar la aplicación se debe autenticar con el respectivo usuario y contraseña para evitar el acceso no autorizado a este, el sistema deberá validar el usuario, campos nulos, etc.

2.6 Requisitos Funcionales

- Autenticación de usuarios, para interactuar con el sistema será, necesario proporcionar usuario y contraseña.
- Configuración de usuario en caso de no acordarse la clave el usuario deberá restaurar la clave.
- Visualización de mapa.
- Generar reportes.

Requerimientos de software:

- Sistema Operativo preferencia de la plataforma de Windows 7 - 8.1 o Windows 10.
- MIT App Inventor 2
- Firebase

- Arduino versión 1.8.3

Para poder ejecutar la aplicación es necesario que el dispositivo cuente con los siguientes requerimientos mínimos:

- Debe tener un sistema operativo 4.4.2
- Un procesador Quad-Core 1.2 GHZ
- 512 MB de RAM

Requerimientos de hardware:

- Procesador Intel Core i7 3.40 Ghz.
- Mínimo de 6 a 8 Gb de Ram.
- Disco duro mínimo de 500 Gb de almacenamiento.

Para desarrollar el circuito en Arduino es necesario:

- Placa de Arduino Mega
- Sensor Ultrasónico
- Cables de conexión
- Modulo sim 808 + GPS
- Micro Servo
- Modulo Bluetooth

3. Justificación del requerimiento a satisfacer

Es indispensable el transporte público en la población, ya que todos los ciudadanos dependen de ello para movilizarse de un lugar a otro y debido a las constantes molestias que se generan en las horas pico, ya sea por la salida de estudiantes tanto de las escuelas, colegios y de universidades es difícil poder movilizarse de un lugar a otro debido a la alta demanda de usuarios, esto genera congestionamiento vehicular en el momento de estacionarse a recoger pasajeros haciendo caso omiso a los paraderos establecidos, en la zona céntrica de la ciudad algunas calles son doble vía, y cada lado de la vía solo suelen tener dos carriles, y si un vehículo se estaciona solo quedaría un carril para rebasar y uno para libre circulación, de la misma manera es con los buses, ya que la mayoría de los conductores recogen y dejan pasajeros donde el usuario desea, obstaculizando el paso a los otros vehículos que circulan, esto hace que los conductores de los otros vehículos hagan uso de sus bocinas, este ruido puede provocar molestias a los transeúntes y personas que viven por la zona.

Dada esta necesidad de mejorar tanto el servicio, como el correcto control de los buses, se ha planteado realizar un sistema de información para monitoreo y control de paradas de buses en la ciudad de Babahoyo, Con esta propuesta tecnológica los conductores evitaran recoger o dejar pasajeros en paradas no establecidas, de modo que van a ser monitoreados a través de GPS, que estarán instalados en cada bus, este sistema brindará muchas ventajas a usuarios ya que no al realizar una para no establecida podrían poner en peligro a los pasajeros e inclusive el mismo conductor ya que los delincuentes cometen sus delitos y se bajan donde ellos desean con amenazas al conductor, con la aplicación de esta medida el conductor no necesitara presionar ningún botón para abrir y cerrar las puertas.

CAPITULO 2

DESARROLLO DEL PROTOTIPO

1. Definición del prototipo tecnológico

El Sistema de información arrojará como resultado un prototipo de una aplicación que va a facilitar el monitoreo en tiempo real mediante una aplicación móvil, se observará el recorrido de los buses mediante tecnología GPS, y controlar que los conductores no se estacionen a dejar ni recoger pasajeros en lugares no establecidos.

Para esto, el sistema informático aparte de tener una interfaz amigable para el o los usuarios, este proyecto será creado en dos fases, una que es la parte de programación móvil y la segunda que es programación en Arduino, usando las diferentes herramientas y métodos que permita un eficiente funcionamiento.

El módulo móvil del sistema será desarrollado usando la aplicación web MIT App Inventor ya que esta herramienta permite realizar aplicaciones Android enlazando bloques para crearlas, este sistema es gratuito y para acceder se debe sincronizar una cuenta de Google.

En la segunda parte es necesario usar Arduino tanto hardware como software para la programación y de los diferentes dispositivos uso de los circuitos electrónicos para el correcto funcionamiento del GPS.

2. Fundamentación teórica del prototipo

La tecnología móvil cada día avanza brindándonos grandes funcionalidades, hasta llegar al punto de querer considerarlos sus ordenadores portátiles, con la integración de sensores, y creación de aplicaciones que ayudan en la vida cotidiana, esto ayuda a reducir la brecha digital debido a que son pocas las personas que hoy en día no hacen uso de smartphones, en la actualidad hasta los niños de temprana edad usan smartphones. (Hernández, 2015)

2.1 Metodología

Una metodología ágil para el desarrollo de aplicaciones móviles es la metodología Mobile-D, esta fue creada en base a otras soluciones tales como Xtreme Programing (XP), Cystal methodologies y Rational Unified Process (RUP), con la finalidad de desarrollar en cortos ciclos en dispositivos pequeños. Esta metodología está apoyada en un desarrollo basado en pruebas hasta que la aplicación final cumpla con los requerimientos estimados. (Cruz, 2014)

El principal objetivo de esta metodología es que los ciclos para el desarrollo sean cortos y obtener en poco tiempo un producto para el dispositivo móvil. Esta metodología consta de 5 fases:

- **Fase de Exploración**

En esta fase se va a planificar y deducir los requisitos del proyecto, para obtener una perspectiva de alcance del proyecto y describir las funcionalidades de la aplicación móvil.

- **Fase de inicialización**

En esta etapa se prepararan los recursos necesarios tales como recursos físicos, recursos tecnológicos y de comunicación, se planificara las actividades para las otras fases, esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, planificación inicial, día de pruebas y día de salida.

- **Fase de producción**

En esta fase se usará el desarrollo dirigido por pruebas (TDD), en sí se pondrá a prueba y se verificará si está o no correcto el funcionamiento de los desarrollos, es decir se repetirá la planificación para especificar los requisitos y tareas a realizar.

- **Fase de estabilización**

Esta fase es muy importante por tal motivo se verificará el completo funcionamiento del sistema, se realizarán tareas parecidas a las de la fase de producción, aunque el esfuerzo está dirigido a la integración del sistema.

- **Fase de pruebas**

Una vez terminada la aplicación se realizará un testeo para verificar si es estable, en caso de encontrarse con un error se procederá a su arreglo pero no se realizará desarrollo nuevo porque se rompe el ciclo. (Guerrero, 2015)

2.2 Tecnología

Android

Es un sistema operativo empleado para dispositivos móviles tales como teléfonos móviles, tablets, la mayoría de pantalla táctil y en la actualidad existen televisores, relojes e inclusive vehículos utilizan Android, fue creado por Android INC, compañía perteneciente a Google, está basado en Unix es de código abierto, está destacado por su seguridad, según expertos las vulnerabilidades encontradas en su estructura son pocas. (Julián Pérez Porto, 2015)

App Inventor 2

Es una plataforma de desarrollo de aplicaciones para dispositivos móviles que usan el sistema operativo Android, este Framework fue creado inicialmente por Google Labs y actualmente está a cargo del Instituto Tecnológico de Massachusetts, este entorno es fácil de usar, incluso aquellas personas que no tienen conocimientos en programación podrán desarrollar sin dificultad sus aplicaciones. (Alberto Gomez, 2016)

Características:

- Hace uso de la librería Open Books de Java permitiendo crear un lenguaje visual, a través de bloques.
- Permite crear aplicaciones sencillas para smartphones y no requieren de mucha memoria Ram.
- Es fácil de usar, es ideal para principiantes que buscas sacar provecho de su dispositivo Android. (Coronel, 2017).

Ventajas:

- Permite crear aplicaciones mediante bloques, sin la necesidad de tener conocimientos sobre códigos de programación.
- Se puede acceder en cualquier lugar y en todo momento, mientras se tenga conexión a internet para tener conexión con los proyectos que se realizan.
- Nos podemos conectar a través de código Qr, usando un emulador e inclusive mediante conexión USB, para poder emular nuestros proyectos.

Desventajas:

- No se pueden desarrollar aplicaciones robustas, este entorno es para desarrollar aplicaciones sencillas.
- Las aplicaciones que se desarrollan son solo para el SO Android.

Firestore

Es una plataforma creada por Google, con el propósito de desarrollar y crear aplicaciones de manera rápida, esta plataforma se encuentra en la nube y está disponible para diversas plataformas tales como: IOS, Android y Web, permite hacer un test y detectar los errores para poder brindar una aplicación de calidad, Firestore tiene 3 planes para desarrolladores en las que varía el precio según el espacio de almacenamiento y conexiones simultáneas, el primer plan es gratuito. (Cardona, 2016)

Razones para usar Firestore:

- Rapidez: posee funciones que ayudan al programador a optimizar tanto la aplicación y el tiempo de desarrollo, nos permite detectar errores, todo se almacena en la nube, y se puede configurar de manera remota.

- Infraestructura

Con firestore nos evitamos los dolores de cabeza con respecto a la complejidad de la infraestructura de la aplicación, la implementación de firestore es fácil y rápido.

- Analítica

Gracias a la función Firestore Analytics nos va a ayudar a tomar decisiones basadas en datos, es decir puedes controlarlo todo desde obtener información sobre los usuarios hasta los fallos de la aplicación.

- Multiplataforma

Posee APIs integradas a SDK individuales para Android, iOS y Java Script.

- Notificaciones

Permite gestionar la programación y enviar notificaciones a los usuarios de nuestra aplicación.

- Inicio gratuito y proyecto escalable

Casi todas las funciones de Firebase son gratuitas, en caso de que la aplicación sea exitosa, y lleguemos al límite del plan gratuito podremos escoger un plan con más beneficios.

- Soporte gratuito

Google nos ofrece soporte gratuito por correo electrónico, los expertos en desarrollo de Google y de Firebase participan en comunidades en línea como Stak Overflow y GitHub. (Ramirez, 2017)

Arduino

Es una plataforma de programación electrónica de código abierto para todo tipo de usuario, es fácil de usar para los desarrolladores debido a su código es accesible para quienes deseen usarlo y modificarlo, sin tener que gastar dinero en una tarjeta pre-fabricada, también ofrece una plataforma de Entorno de Desarrollo Integrado (IDE), este proyecto se originó en el 2003, por varios estudiantes del Instituto de Diseño Interactivo de Ivrea, Italia, con el propósito de facilitar el uso de dispositivos electrónicos y programarlos. (González, 2016)

¿Que se puede crear con arduino?

Se pueden crear un sin numero de proyectos, desde un sistema de alarma hasta un robot gracias a instrucciones que dan es

Ventajas:

- Es multiplataforma, a diferencia de otros microcontroladores que estan limitados a windows.
- Referente al precio es mucho mas barato comparado a otros microcontroladores.
- Es facil de usar y facilita la programacion de dispositivos electronicos, es facil obetener codigos de arduino y modificarlo.
- Es de codigo abierto y para los programadores experimentados se puede ampliar el lenguaje gracias a las librerias de C++.



Ilustración 1: Placa Arduino Mega

Desarrollado por: José Chero

Módulo SIM 808 GSM/GPRS Y GPS

Este módulo esta basado en el chip sim 808 y permite enviar y recibir datos mediante GMS/GPRS de telefonos móviles 2g, viene integrado tambien la tecnología GPS, gracias a esto

sera posible enviar y recibir tanto llamadas como mensajes y obtener las coordenadas de un objeto. (Prometec, 2017)

Caracteristicas:

- Posee módulo SIMCom 808.
- Controla comandos AT.
- Servicio de mensjeria.
- Posee slot para Sim y conector para antena GPS.
- Soporta hasta 26 voltios, 2 Ammperios.
- Soporta carga de bateria Li-lion
- Dimension: 50mm x 7mm x 16 mm



Ilustración 2: Módulo Sim 808

Fuente: (maxelectronica, 2018)

Protoboard

Tambien conocida como tabla de pruebas, los orificios que posee esta conectados ya sea vertical o horizontalmente, posee 2 rieles a los lados usadas como lineas positiva y negativa,

es comunmente usado para realizar pruebas de funcionamiento de dispositivos y circuitos electronicos que creamos. (Reyes, 2015).



Ilustración 3: Protoboard

Fuente (Cruz L. d., 2015)

Las lines de color Azul y Rojo es la zona de alimentación, las partes centrales son para la conexión de dispositivos.

Resistencias

También conocidas como resistores son aquellos elementos que interrumpen o se oponen al paso de electrones en un circuito electrico, provocando que el voltaje disminuya, protege a dispositivos electronicos evitando que se quemen, estan fabricados por carbón y sus valores varian en valores según Ohmios, Kilohmios, Megaohmios. (Perez & Gardey, 2014).



Ilustración 4: Resistencias

Fuente: (telesaonline, 2018)

Led

También conocido como Diodo emisor de luz, es un semiconductor capaz de emitir radiación electromagnética en forma de luz, los primeros led solo emitían luz de baja intensidad, pero en la actualidad tienen la capacidad de transmitir señales infrarrojas, ultravioletas, espectros y led ultra brillante. (Reyes L. A., 2015).

La patita más larga del led es el ánodo (positivo), y la patita pequeña es el cátodo (Negativo).



Ilustración 5: Leds

Fuente: (Reyes L. A., 2015)

Sensores

Son dispositivos que permiten percibir el entorno para poder determinar el estado del proceso para el que se lo emplea, transforman las variaciones medidas en una señal para que el destinatario la perciba, cuando estos tienen salidas digitales son llamados interruptores y cuando tienen salidas analógicas son llamados transmisores.

Existen una gran variedad de sensores dependiendo de lo que se desea medir, hay sensores de temperatura, velocidad, peso, voltaje, humedad, radiación, humedad, de impacto, magnético, entre otros. (Real, 2015).

Características de los sensores:

- Dinamicos: la cantidad a medir varía en cualquier momento y es necesario conocer el comportamiento dinamico del instrumento cuando varie.
- Estaticos: son todos los que miden cantidades estables o no sufren de una variacion de magnitud brusca. (Mecafenix, 2017)

Sensor Ultrasonico

Usa ondas ultrasonicas para poder detectar obstaculos y poder medir distancias, es facil de usar y muy sensible este sensor envia una señal ultrasonica y entrega el tiempo que tomo en ir y venir hasta el obstaculo que detecto, tiene cuatro pines Vcc que es para alimentarlo, GND que es el negativo, el Trigger que es el sensor para enviar la señal y el pin hecho para recibirla. (Veloso, 2016)



Ilustración 6: Sensor Ultrasónico

Desarrollado por: José Chero

MicroServo

El Servo SG90 es un servomotor pequeño con buena calidad y es barato comparado con un Servomotor, funciona con casi todas las tarjetas electrónicas de control de microcontroladores, es ideal para practicas porque no necesitan de mucha energia posee un conector universal tipo S el cable rojo es el de alimentación, el cable café es negativo y el de señal es el cable de color naranja. (ElectroniLab, 2016)

Caracteristicas:

- Velocidad de 0.10 sec/60°.
- Su torque es de 1.8 kg-cm .
- Su ángulo de rotación es de 180°.
- Requiere de 3 a 7.2 voltios para funcionar.

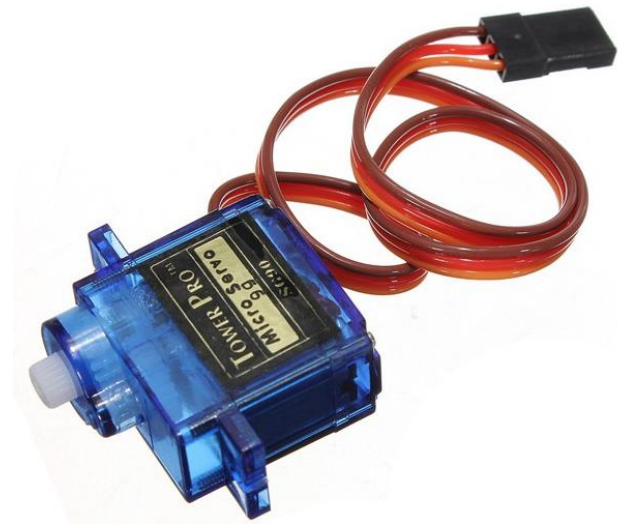


Ilustración 7: Micro Servo SG90

Desarrollado por: José Chero

3. Objetivos del prototipo

3.1 Objetivo general

Diseñar y desarrollar un sistema para controlar y monitorear a través de tecnología GPS las paradas que realizan los buses de transporte urbano de la ciudad de Babahoyo.

3.2 Objetivos específicos

- Automatizar las puertas de los buses para que solo se abran en paraderos establecidos mediante el sensor magnético.
- Desarrollar una aplicación móvil que permita controlar y monitorear a través de tecnología GPS las paradas que realizan los buses de transporte urbano de la ciudad de Babahoyo.
- Monitorear los buses de transporte urbano en tiempo real con el uso de tecnología GPS y visualizarlo mediante un mapa en un dispositivo móvil.
- Ayudar a reducir el congestionamiento vehicular que se forma cuando los buses realizan paradas no establecidas para recoger o dejar pasajeros.

4. Diseño del prototipo

Las Cooperativas de Transporte Público tienden a tener problemas con los conductores de otros vehículos e inclusive con las otras líneas de buses en el momento de estacionarse donde no deben se aglomeran y dado el caso si un bus desea continuar su recorrido no puede porque el bus que esta adelante o atrás le bloquea la salida, y todo por no usar los respectivos paraderos por ese motivo se intenta solucionar este problema por medio de la geolocalización, fomentando el uso de la tecnología móvil de manera efectiva, obteniendo beneficios como monitoreo en tiempo real de los buses, bloquear las puertas y abrirlas solo en las paradas establecidas.

4.1. Fase de análisis

Según el análisis realizado en las Cooperativas de Transporte Públicos, se pudo determinar que, la única forma de evitar que los buses se estacionen en paradas no establecidas es mediante la automatización de las puertas de los buses, y con esto ayudar a contribuir en la regulación de tránsito vehicular en la ciudad, por ese motivo se ha decidido realizar este proyecto para mejorar el control y monitoreo de las paradas.

4.2. Fase de diseño

En esta etapa en base al análisis se detallará los procesos que realizara cada usuario de la aplicación, de acuerdo con los distintos accesos dependiendo del rol que tienen asignado cada uno.

Diagramas de Caso de uso del sistema:

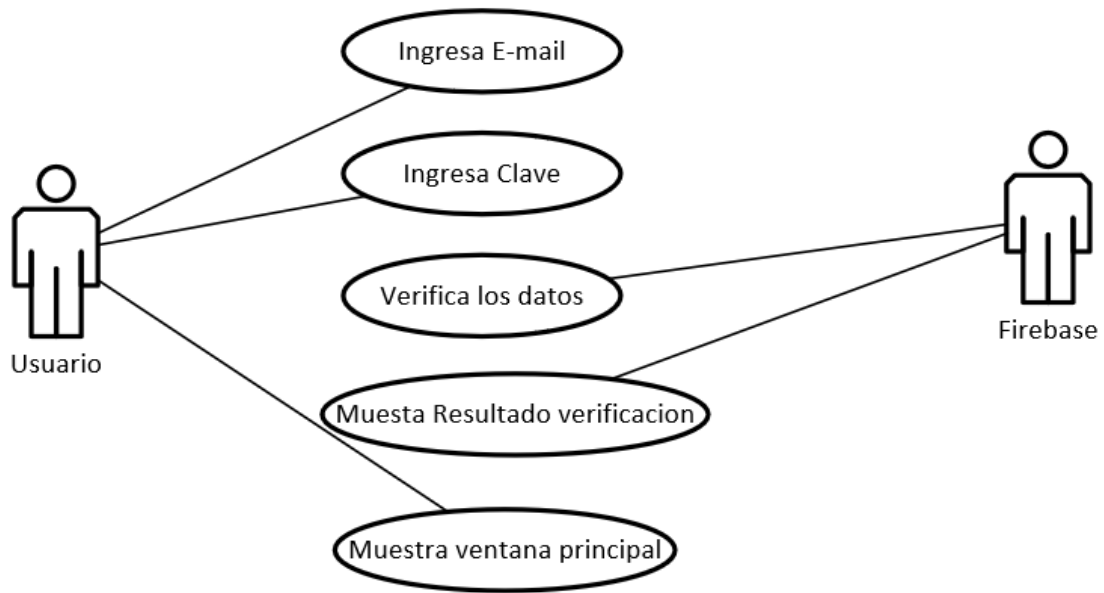


Ilustración 8: Diagrama de caso de uso Inicio de Sesión

Desarrollado por: José Chero

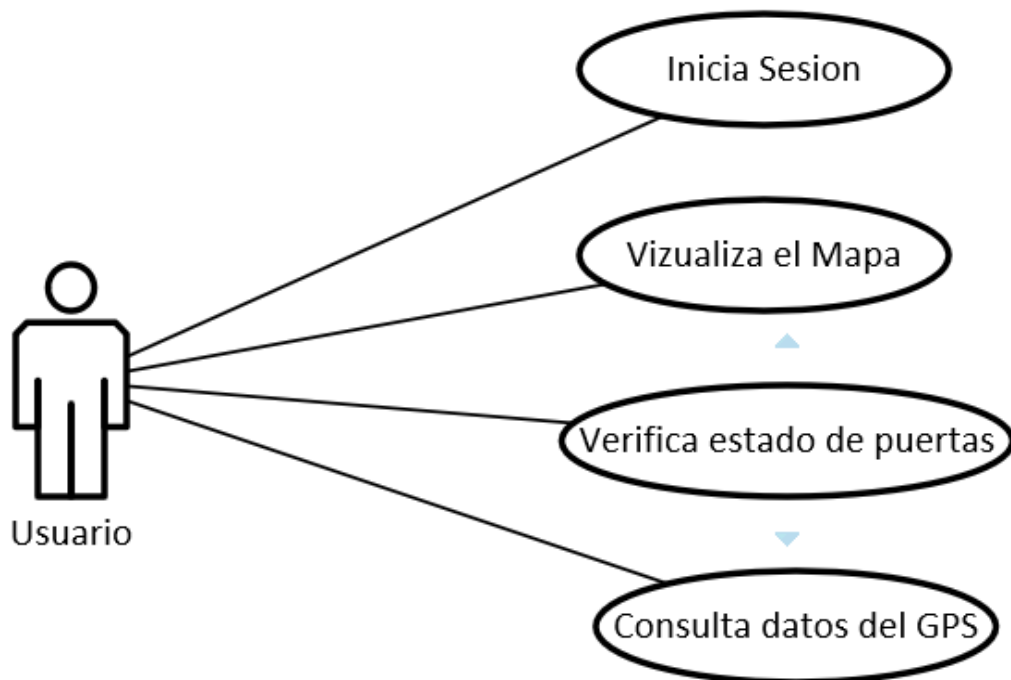


Ilustración 9: Diagrama de caso de uso de acción de usuario

Desarrollado por: José Chero

Diagrama de Clases

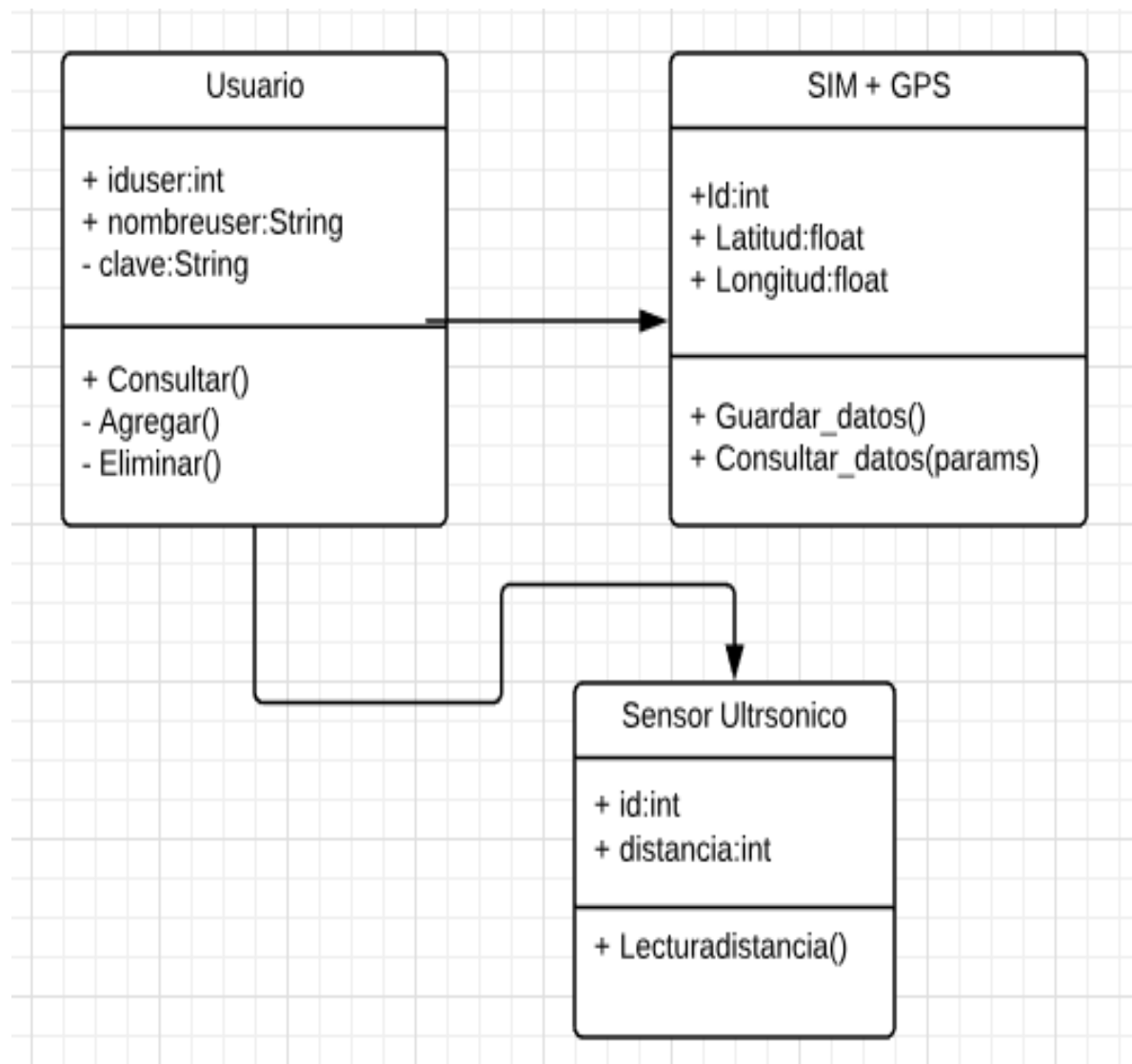


Ilustración 10: Diagrama de Clase

Desarrollado por: José Chero

Diagrama de Actividades

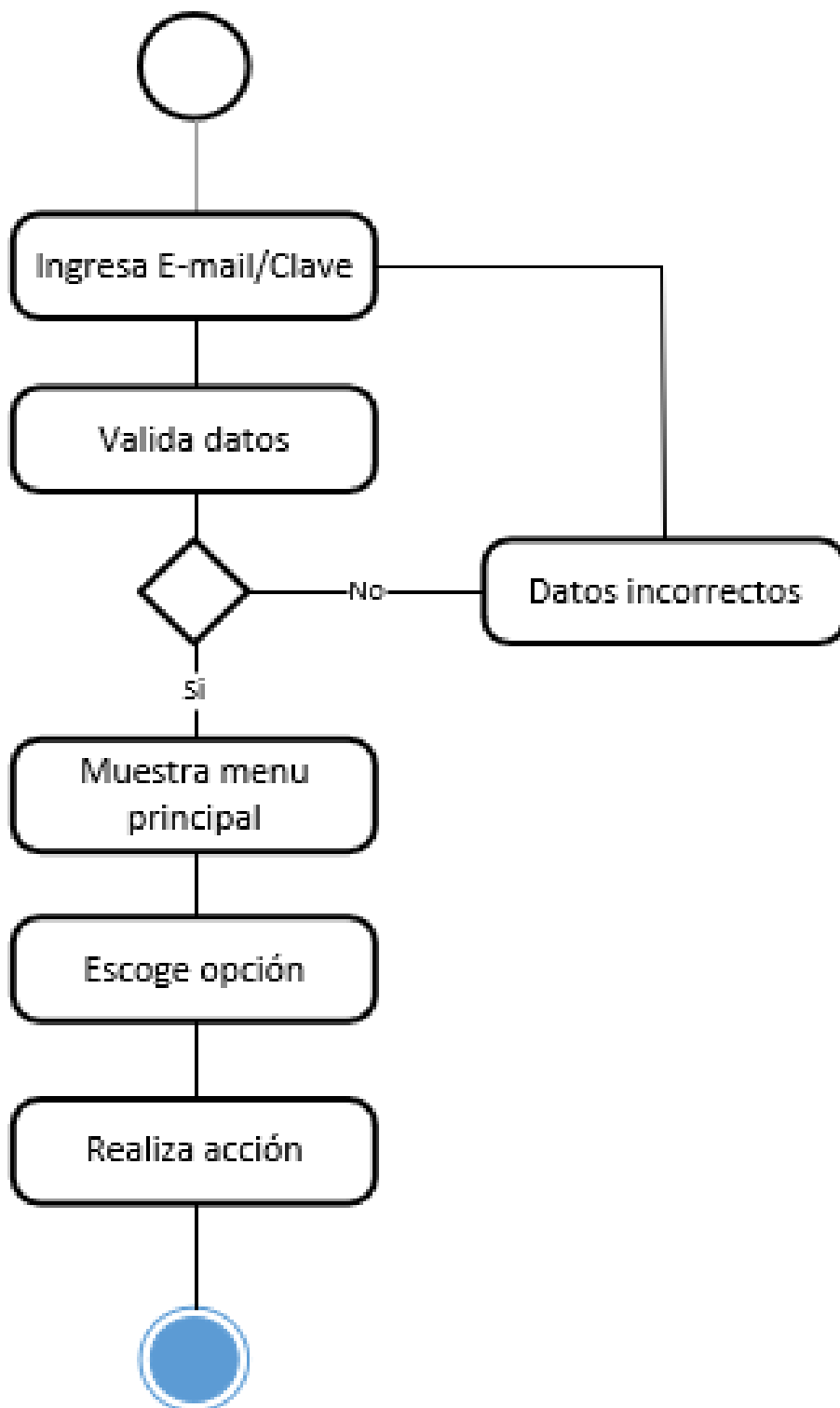


Ilustración 11: Diagrama de actividades

Desarrollado por: José Chero

Diagrama de secuencia

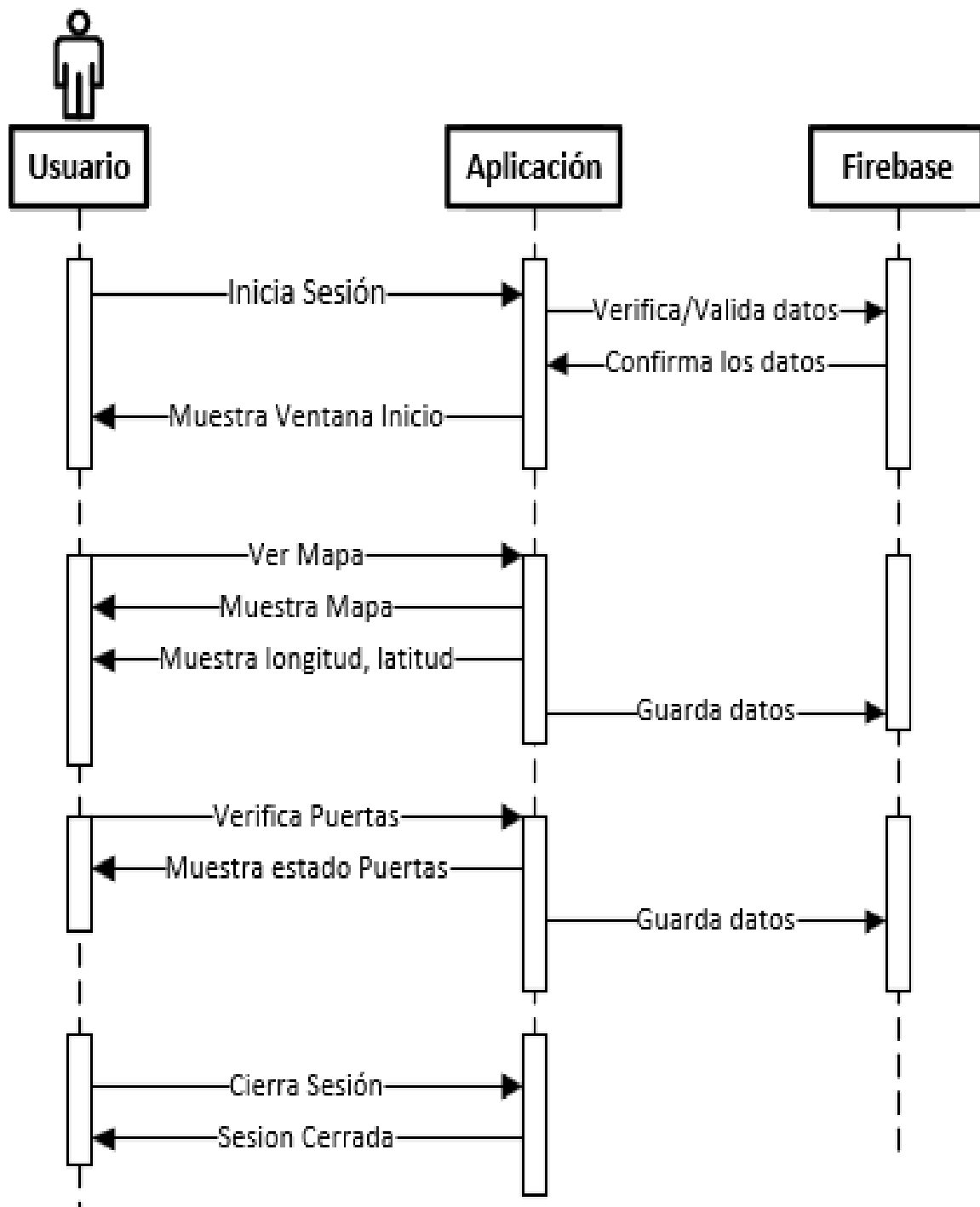


Ilustración 12: Diagrama de Secuencia
Desarrollado por José Chero

Diagrama de Despliegue

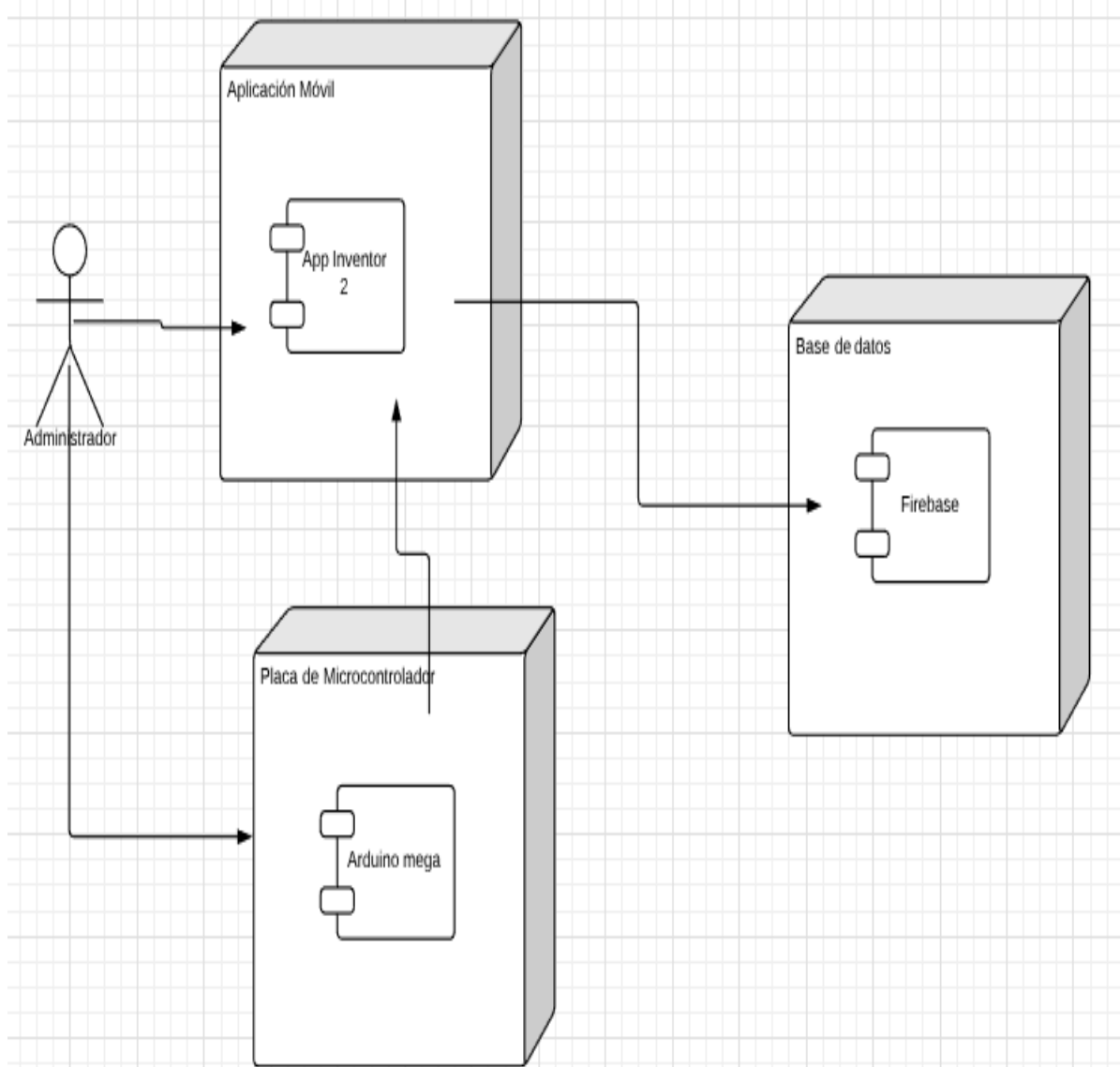


Ilustración 13: Diagrama de Despliegue

Desarrollado por José Chero

Diagrama de Conexión

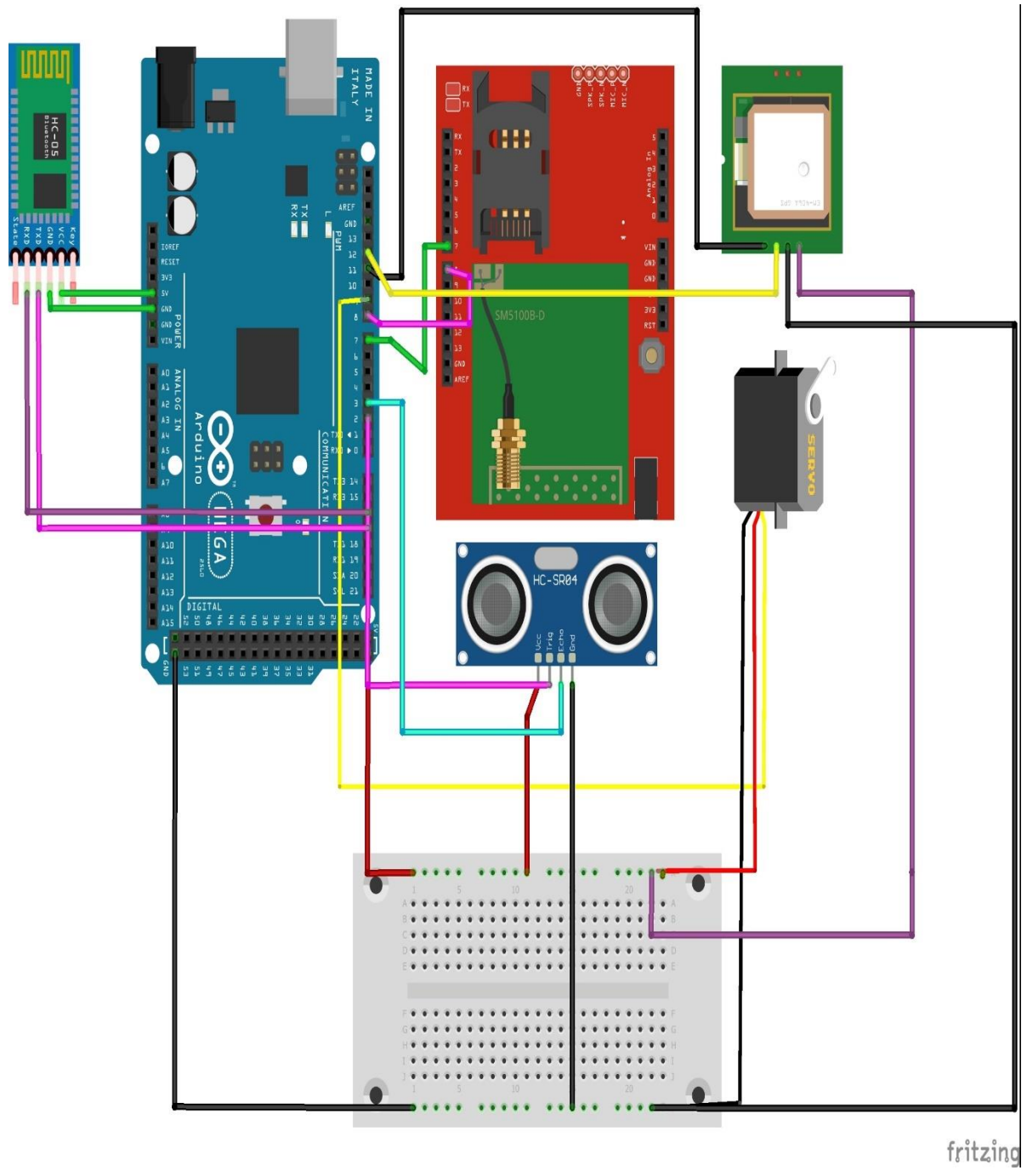


Ilustración 14: Diagrama de conexión

Desarrollado por José Chero

Esquema de Conexión

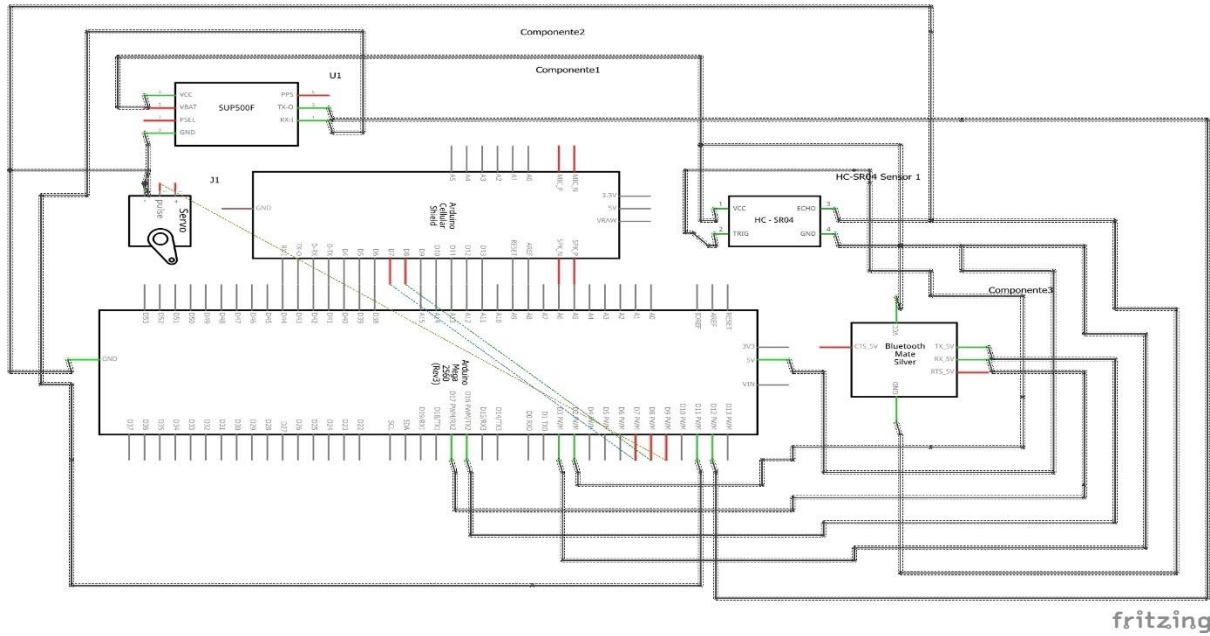


Ilustración 15: Esquema de Conexión

Desarrollado por: José Chero

4.3. Fase de desarrollo

En esta fase se indicará una pequeña descripción de las herramientas utilizadas para el desarrollo del prototipo.

MIT App Inventor 2

Es una herramienta web para el desarrollo rápido de aplicaciones para Android, a través de bloques.

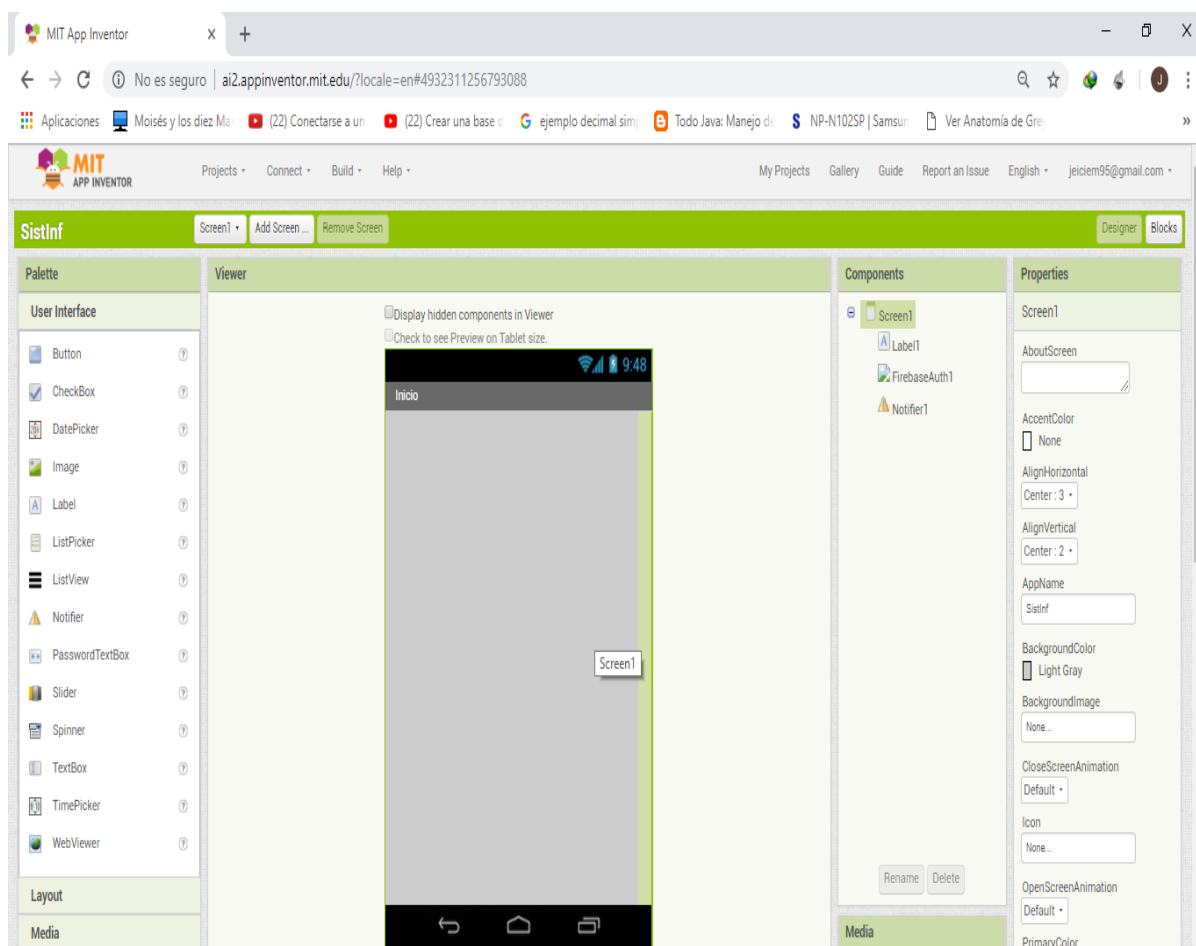


Ilustración 16: Interfaz de MIT App Inventor

Elaborado por: José Chero

Firebase

Plataforma en la nube que permitira administrar nuestra base de datos, identificar los errores de nuestra aplicación, etc.

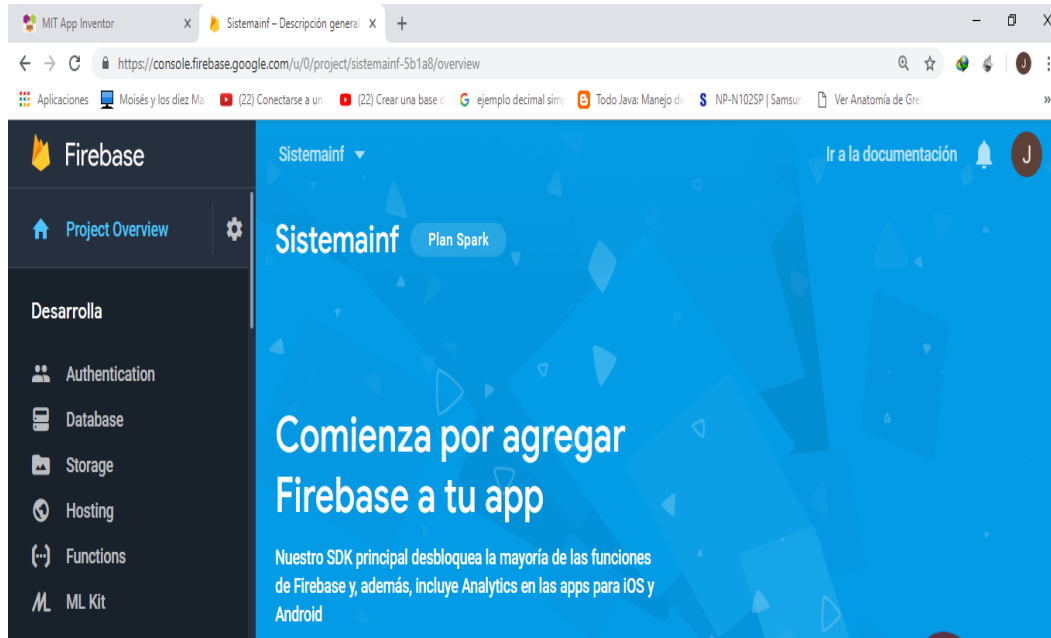


Ilustración 17: Interfaz de Firebase

Desarrollado por: José Chero

Arduino

Plataforma para compilar y subir programas en las placas de Arduino.

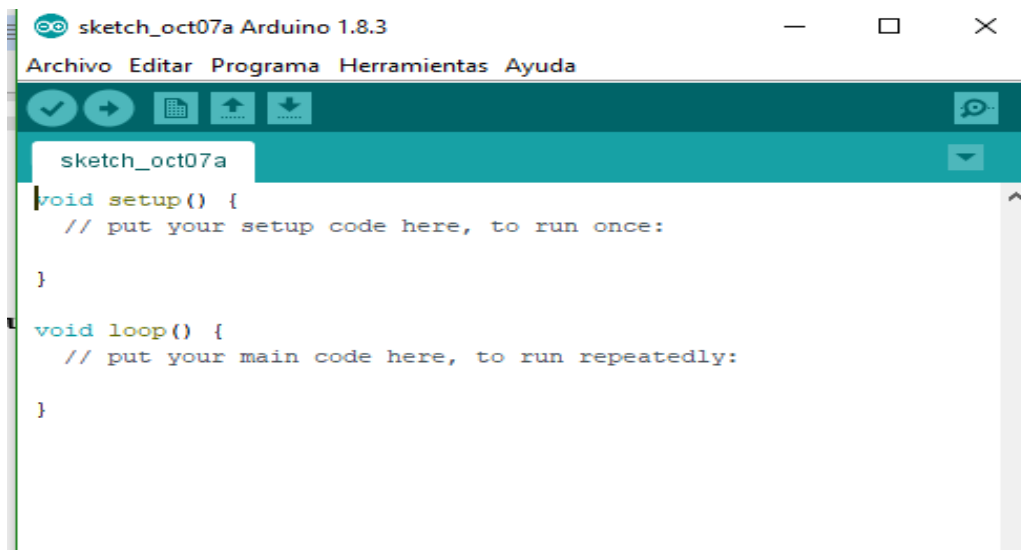


Ilustración 18: Interfaz de Arduino

Desarrollado por: José Chero

5. Ejecución y/o ensamblaje

Sistema Arduino

Se probó la conexión del sensor ultrasónico para verificar si detectaba obstáculos

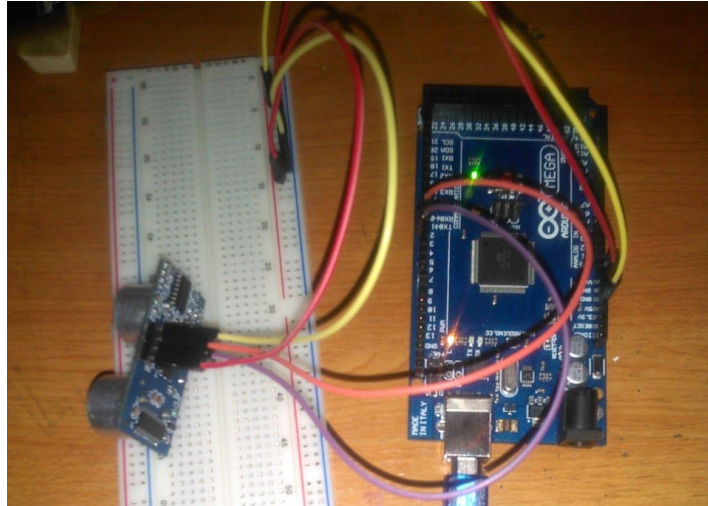


Ilustración 19: Prueba de Sensor Ultrasónico

Desarrollado por: José Chero

Se probó la conexión del Micro servo para comprobar si giraba su ángulo si el sensor ultrasónico detectaba algún obstáculo a una distancia dada.

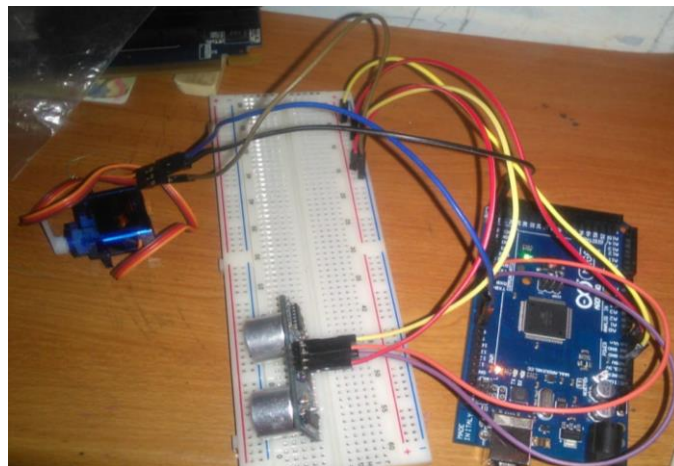


Ilustración 20: Conexión sensor ultrasónico y micro servo

Desarrollado por: José Chero

Se probó el módulo Bluetooth para constatar si no hay fallos de fábrica en el dispositivo y emparejarlo con el dispositivo móvil.

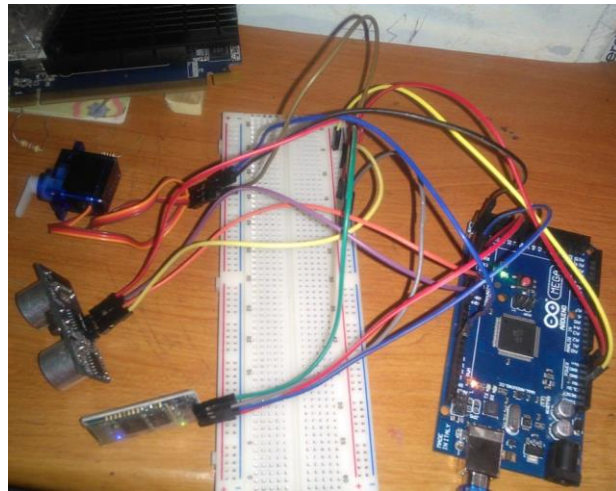


Ilustración 21: Conexión Modulo Bluetooth

Al ejecutar la aplicación automáticamente nos aparecerá la ventana de inicio sesión el usuario ingresará sus datos, para poder ingresar al sistema.



Ilustración 22: Interfaz inicio de sesión

Desarrollado por: José Chero.

Luego de haber ingresado los datos de usuario se va a mostrar la ventana de inicio o de menú donde el usuario podrá escoger una opción.



Ilustración 23: Interfaz inicio

Desarrollado por: José Chero

La ventana Mapa se podra vizualizar la ubicación del bus.

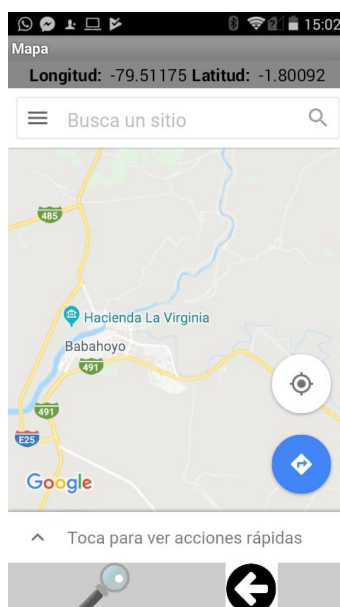


Ilustración 24: Consultar longitud latitud

Desarrollado por: José Chero

El usuario primero debe conectarse por vía Bluetooth para poder recibir desde Arduino la longitud y latitud del GPS a la aplicación.



Ilustración 25: Conexión por Bluetooth

Desarrollado por: José Chero

La ventana Información muestra datos relacionado a el autor y nombre del sistema.



Ilustración 26: Interfaz Información aplicación

Desarrollado por: José Chero

Código Arduino

```
//Se declaran las Librerias necesarias

#include <SoftwareSerial.h>

#include <Servo.h>

//Se declaran variables

Servo servoMotor;

float rango;

char dato_ultrasonico;

//Variables del sim 808

SoftwareSerial sim808(7,8);

#define DEBUG true

String state,timegps,latitude,longitude;

char phone_no[] = "0982407726"; // número del chip insertado en el SIM

int puerta =10;

char datos_serial;

char enviar_dato ;

char buffer[20];

//Variables para ultrasonico #1 de distancia

#define pEcho 2 //Conecta pin echo

#define pTrig 3 //Conecta el trigger

int duracion; //Captura el pulso que emite el echo
```

```

int ultrasonico=0; //Captura la distancia

//Variable para control del ciclo

int c1=0,c2=0,c3=0;

void setup()

{

//Se inicializan las variables

sim808.begin(9600);

sim808.println("AT+CPIN=\\"1111\\""); //Comando AT para introducir el PIN de la tarjeta

    delay(25000);

sim808.print("AT+CSMP=17,167,0,0"); // set this parameter if empty SMS received

delay(100);

sim808.print("AT+CMGF=1\r");

delay(400);

sendData("AT+CGNSPWR=1",1000,DEBUG);

delay(50);

sendData("AT+CGNSSEQ=RMC",1000,DEBUG);

delay(150);

/////////configuracion de velocidad

servoMotor.attach(8);

Serial.begin(9600); // Inicia la velocidad en 9600 baudios

Serial2.begin(9600); // Inicia la velocidad en 9600 baudios

```

```

pinMode(pEcho, INPUT); // Defino echo como entrada

delayMicroseconds(2); // duracion que demora en hacerse cada accion

pinMode(pTrig, OUTPUT); // Defino trigger como salida

delayMicroseconds(2);

pinMode(puerta, OUTPUT);

Serial.println("INICIANDO PROPUESTA TECNOLOGICA");

delay(3000);

}

char letra;

int cont=0;

String data[5];

void loop()

{

sendTabData("AT+CGNSINF",1000,DEBUG);

if (state !=0) {

Serial.println("State :"+state);

Serial.println("Time :"+timegps);

Serial.println("Latitude :"+latitude);

Serial.println("Longitude :"+longitude);

Serial.println("Realizando llamada...");

sim808.println("ATD0993476176;"); //Comando AT para realizar una llamada

```

```
delay(30000); // Espera 30 segundos mientras realiza la llamada

sim808.println("ATH"); // Cuelga la llamada

delay(1000);

Serial.println("Llamada finalizada");

sim808.print("AT+CMGS=\"");

sim808.print(phone_no);

sim808.println("\");

delay(300);

sim808.print("http://maps.google.com/maps?q=loc:");

sim808.print(latitude);

sim808.print(",");

sim808.print (longitude);

delay(200);

sim808.println((char)26); // End AT command with a ^Z, ASCII code 26

delay(200);

sim808.println();

delay(20000);

sim808.flush();

} else {

    Serial.println("GPS Initialising...");

}
```

```

//Codigo para calcular la distancia

int dist = lectura_ultrasonico();//almacena los datos enviados desde las funciones

//este es el orden que debe ir para recibirse en la aplicacion

//sprintf(buffer, "%d",dist);

Serial.print("Dist del sensor uno = ");

Serial.println(dist);

Serial.println(buffer);

delay(500);

}

//Función para la lectura del sensor ultrasonico

float lectura_ultrasonico()

{

    digitalWrite(pTrig, LOW);    //Envio un pulso bajo al trigger

    delayMicroseconds(2);        //Retardo de 2us

    digitalWrite(pTrig, HIGH);   //Genera el pulso de trigger

    delayMicroseconds(10);       //Retardo de 10us

    digitalWrite(pTrig, LOW);    //Vuelve a estado bajo trigger

    duracion=pulseIn(pEcho, HIGH); //Recibo el pulso con echo y lo asigno a duracion

    ultrasonico=(duracion/2)/29; //Asigno a distancia la duracion del pulso y lo convierto a

centímetros

//Condición para establecer la distancia

```

```

if ((ultrasonico <= 5)&&(ultrasonico >= 0) ) {

c1++;

if(c1==1){

Serial.println("PUERTA ABIERTA");//desabilitar este mensaje

Serial2.println("PUERTA ABIERTA");

servoMotor.write(90);

digitalWrite(puerta,HIGH);

delay(1000);

digitalWrite(puerta,LOW);

delay(1000);

}

}

else {

c1=0;

Serial.println("PUERTA CERRADA");//desabilitar este mensaje

Serial2.println("PUERTA CERRADA");

servoMotor.write(0);

}

return ultrasonico;

}

void sendTabData(String command , const int timeout , boolean debug){

```

```

String data[5]={ "", "", "", "", "" };

state = data[1];

timegps = data[2];

latitude = data[3];

longitude =data[4];

sim808.println(command);

long int time = millis();

int i = 0;

while((time+timeout) > millis()){

while(sim808.available()){

char c = sim808.read();

if (c != ',') {

data[i]+=c;

delay(100);

} else {

i++;

}

if (i == 5) {

delay(100);

goto exitL;

}

```

```

    }

}exitL:

if (debug) {

    state = data[1];

    timegps = data[2];

    latitude = data[3];

    longitude =data[4];

}

}

String sendData (String command , const int timeout ,boolean debug){

String response = "";

sim808.println(command);

long int time = millis();

int i = 0;

while ( (time+timeout ) > millis()){

while (sim808.available()){

char c = sim808.read();

response +=c;

}

}

if (debug) {

```



```
Serial.print(response);  
  
}  
  
return response;  
  
}
```

Capítulo 3

1. Plan de Evaluación

1.1. Funcionalidad y facilidad de uso.

Destinario.	Decente. Guía.	Fecha:	Actividades. Desarrolladas.	Observaciones.	Cambios en el sistema.
Usuario.	Ing. Ángel España León	03-09- 2018	Prueba de navegación de vistas	Algunos botones no enviaban a la siguiente ventana	Cambiar llamado de ventanas en los botones
		04-09- 2018	Pruebas de validación de usuario y campos vacíos	Ninguna	Ninguna
		05-09- 2018	Pruebas de registro de usuario	Ninguna	Ninguna
		06-09- 2018	Pruebas de visualización de ubicación de GPS	El intervalo de temporizador era muy lento.	Se redujo el tiempo de consulta de coordenadas
		07-09- 2018	Pruebas de funcionalidad	Corre en versiones de Android 4.4.2 en adelante	Ninguna

Tabla 2: Funcionalidad y facilidad de uso

Desarrollado por: José Chero

1.2. Estabilidad.

Destinario.	Docente Guía	Fecha:	Actividades desarrolladas	Observaciones	Cambios en el sistema.
base de datos	Ing. Ángel España León	09-09-2018	Prueba de envío y consulta de datos	No hubo perdida de datos	Ninguno
Sistema		10-09-2018	Prueba de carga de recursos	Funciono como se esperaba.	Ninguno
Aplicación Móvil		12-09-2018	Prueba de velocidad	Inicia rápido ya que no consume muchos recursos del dispositivo	Ninguno
Arduino		13-09-2018	Verificación de conexiones/estado sensor	Los Sensores y las conexiones funcionaron según lo esperado	Ninguno

Tabla 3: Estabilidad

Desarrollado por: José Chero

1.3. Compatibilidad.

Destinario.	Docente Guía	Fecha:	Actividades desarrolladas	Observaciones	Cambios en el sistema.
Sistema	Ing. Ángel España León	15-09- 2018	Pruebas de funcionalidad	Corre en versiones de Android 4.4.2 en adelante	Ninguno
Arduino		16-09- 2018	Prueba de dispositivos	El GPS no enviabas los valores correctos	Se verificaron y se corrigieron los errores.
Sistema		17-09- 2018	Prueba de APIs	En versiones inferiores a la versión 4.4.2 se distorsionan las imágenes, botones, etc. No es aconsejable instalar en estas versiones	Ninguno

Tabla 4: Compatibilidad

Desarrollado por: José Chero

1.4. Interoperabilidad.

Destinario.	Docente Guía	Fecha:	Actividades desarrolladas	Observaciones	Cambios en el sistema.
Sistema	Ing. Ángel España León	19-09- 2018	Test de intercambios de datos entre la aplicación móvil y la base de datos	El tiempo de intercambio de datos es aceptable	Ninguno
Arduino		20-09- 2018	Test de envío de datos del módulo GPS y el sistema	El módulo GPS envió los datos dentro del tiempo estipulado	Ninguno

Tabla 5: Interoperabilidad

Desarrollado por: José Chero

2. Resultados de evaluación.

PLAN DE EVALUACION	ACEPTACION	RECHAZO
FUNCIONALIDAD Y FACILIDADES DE USO	80%	20%
ESTABILIDAD	95%	5%
COMPATIBILIDAD	90%	10%
INTEROPERABILIDAD	100%	0%
RESULTADO DE LA EVALUACION	91%	9%

Tabla 6: Resultados de evaluación

Desarrollado por: José Chero

2.1. Análisis de resultados.

La información obtenida referente al resultado de las pruebas en la evaluación del proyecto, esta información es muy notable y trascendental para el desarrollo del prototipo. En la fase de funcionalidad y facilidades de uso se realizaron cambios en el manejo de ventanas porque unos botones enviaban a una ventana que no debían, también se realizó cambios en el tiempo de consulta de la longitud y latitud porque si el bus está en movimiento las coordenadas cambian constantemente y es necesario usar el menor tiempo para tener un monitoreo exacto, obteniendo un porcentaje de aceptación del 80%.

Con respecto a la estabilidad no se realizaron cambios en el sistema y se obtuvo un porcentaje de aceptación del 95%, en la compatibilidad para no que la aplicación funcione correctamente es necesario tener la versión KitKat 4.4.2 de Android como mínimo, el GPS no enviaba la longitud y latitud exacta se verifico la conexión del mismo y luego se revisó el código para realizar los respectivos cambios obteniendo una aceptación del 90% y por último la fase de interoperabilidad se realizaron varias pruebas de intercambios de información entre la aplicación móvil y la base de datos en tiempo real para constatar que los datos ingresados y consultados sean los correctos, también se realizó pruebas con respecto al envío de datos de Arduino a la aplicación móvil obteniendo una aceptación del 100%.

El resultado de la evaluación del prototipo es del 91% de aceptación y 9% de rechazo con este resultado el prototipo alcanzo los requerimientos del usuario.

3. Conclusiones y Recomendaciones.

3.1. Conclusiones.

- El GPS es de gran utilidad y ayuda a las Cooperativas de Transporte porque permite obtener datos reales referentes a la ubicación del vehículo permitiendo un monitoreo fácil y efectivo.
- Los usuarios esperaran los buses en las paradas establecidas, ya que las puertas son automáticas y se abren solo en dichas paradas.
- Ya no existirá bloqueos en las puertas de entrada y salida de la Universidad Técnica de Babahoyo por parte de los buses.

3.2. Recomendaciones.

- Se recomienda verificar y consultar las especificaciones del módulo GPS a adquirir, existen muchos modelos y según el uso que se le vaya a dar.
- Es importante utilizar Google Maps para el monitoreo de vehículos, esto facilitara la ubicación del vehículo en cualquier lugar que este se encuentre.
- Se recomienda que el lugar que se encuentre el sistema tenga una fuente de alimentación externa en caso de emergencia.
- Se recomienda realizar actualizaciones de sistema de forma semestral.
- El ancho de banda utilizado en el servidor debe ser óptimo para un mejor desempeño de este.

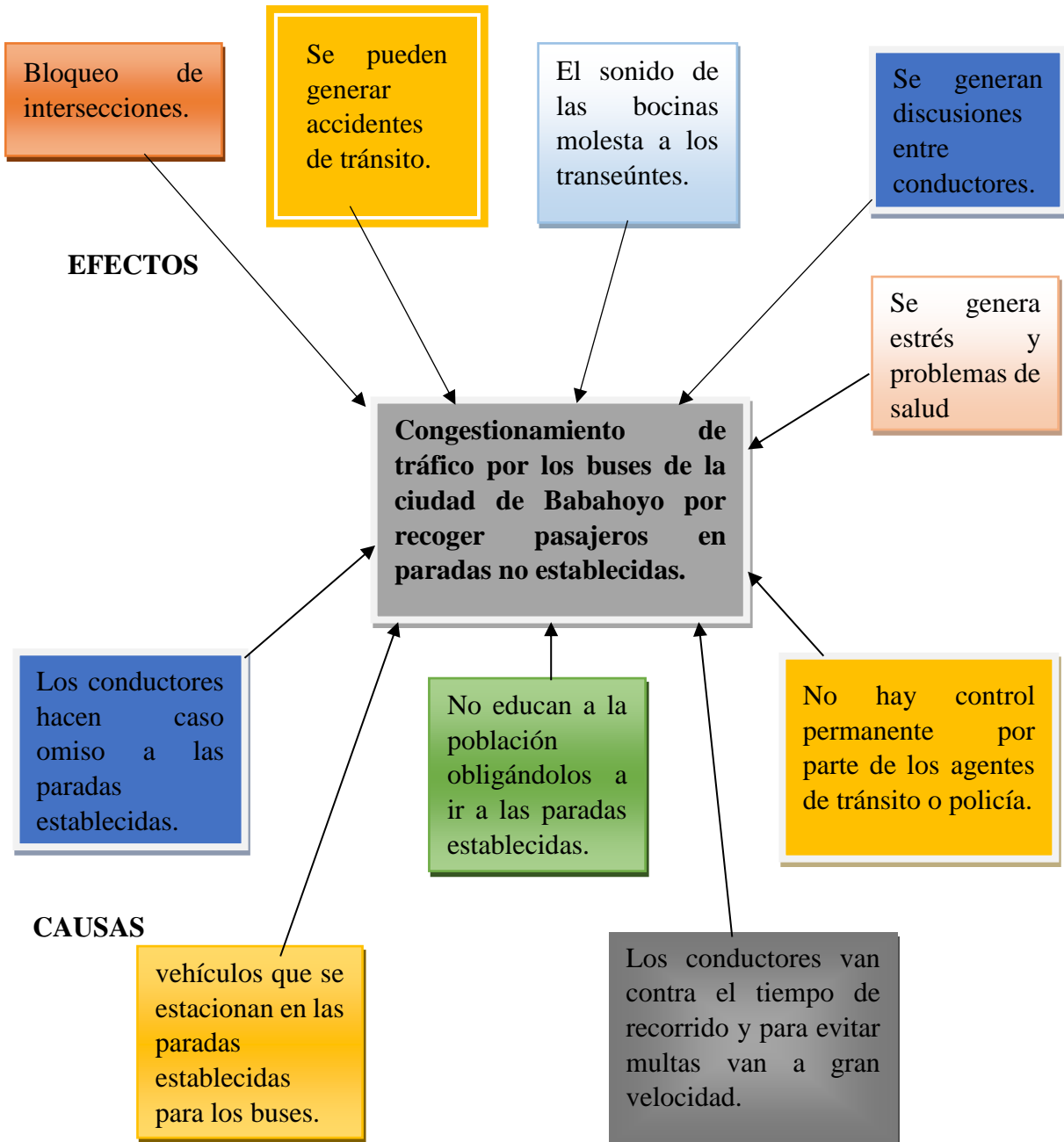
Bibliografía

- Alberto Gomez, C. L. (4 de Febrero de 2016). *codigo21.educacion.navarra.es*. Obtenido de <http://codigo21.educacion.navarra.es/2016/02/04/app-inventor-2-conceptos-basicos-configuracion-descripcion-de-los-bloques-integrados-y-otros-recursos/>
- Cardona, M. P. (14 de Octubre de 2016). *iebschool*. Obtenido de Firebase, qué es y para qué sirve la plataforma de Google: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>
- Coronel, J. C. (25 de Marzo de 2017). *Dispositivos Mviles (Telefonia Movil)*. Obtenido de blogspot.com: <http://1jx87xw329skl2ar8.blogspot.com/2017/03/caracteristicas-de-app-inventor.html>
- Cruz, L. d. (2015). *ingenieriaelectronica*. Obtenido de <https://ingenieriaelectronica.org/definicion-de-protoboard-y-como-utilizarlo/>
- Cruz, R. M. (2014). *Metodología*. Obtenido de <http://pegasus.javeriana.edu.co/~PA133-05-PMovVidaAutomotor/Metodologia.html>
- Electronics, M. (2018). *madnesselectronics*. Obtenido de <http://www.madnesselectronics.com/producto/modulo-gps-neo-6m-v2/>
- ElectroniLab. (2016). *electronilab.co*. Obtenido de <https://electronilab.co/tienda/micro-servo-9g-towerpro/>
- Factory, G. (2018). *geekfactory*. Obtenido de <https://www.geekfactory.mx/tienda/sensores/sensor-magnetico-para-ventanas-y-puertas/>
- FM, Y. (21 de Julio de 2018). *xataka*. Obtenido de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- González, G. (26 de Abril de 2016). *blogthinkbig*. Obtenido de <https://blogthinkbig.com/por-que-es-arduino-util-y-que-se-puede-crear-con-el>
- Guerrero, M. (2015). <http://manuelguerrero.blogspot.es>. Obtenido de <http://manuelguerrero.blogspot.es/1446543763/metodologia-mobile-d-para-desarrollos-de-aplicaciones-moviles/>
- Hernández, K. V. (2015). *TECNOLOGIA MOVIL*. Obtenido de DEFINICIÓN (tecnología móvil): <http://tecno109.blogspot.com/2015/10/definiciontecnologia-movil.html>
- Hora, L. (06 de Febrero de 2017). *lahora*. Obtenido de <https://lahora.com.ec/noticia/1102027765/competencia-de-trc3a1nsito-para-nueva-empresa-municipal>
- Hora, L. (13 de Agosto de 2018). *LaHora*. Obtenido de <https://lahora.com.ec/losrios/noticia/1102178049/la-santa-rita-festejo-sus-22-anos->
- Julián Pérez Porto, M. M. (2015). *definicion.de*. Obtenido de <https://definicion.de/android/>
- maxelectronica. (2018). *maxelectronica*. Obtenido de <http://www.maxelectronica.cl/wireless/290-modulo-de-comunicaciones-cuadribanda-gsmgprs-y-gps-modelo-sim808.html>
- Mecafenix, F. (11 de Abril de 2017). *ingmecafenix*. Obtenido de <http://www.ingmecafenix.com/automatizacion/sensores/>
- Perez, J., & Gardey, A. (2014). *definicion.de*. Obtenido de <https://definicion.de/resistencia-electrica/>

- Prometec. (2017). *prometec*. Obtenido de <https://www.prometec.net/sim808/>
- Ramirez, V. (13 de Febrero de 2017). *marketing4ecommerce*. Obtenido de <https://marketing4ecommerce.net/que-es-firebase-de-google/>
- Real, C. (18 de Agosto de 2015). *controlreal.com*. Obtenido de <https://controlreal.com/es/sensores-definicion-y-caracteristicas/>
- Reyes, L. A. (2015). *Ingenieriaelectronica*. Obtenido de <https://ingenieriaelectronica.org/definicion-y-caracteristicas-de-led-diodo-emisor-de-luz/>
- Reyes, L. d. (8 de Agosto de 2015). *ingenieriaelectronica*. Obtenido de <https://ingenieriaelectronica.org/definicion-de-protoboard-y-como-utilizarlo/>
- telesaonline. (2018). Obtenido de <http://telesaonline.com/componentes-electronicos/componentes-electronicos-componentes-pasivos/componentes-electronicos-componentes-pasivos-resistencias/resistencia-carbon-1-2w-1k-ohms>
- Universo, E. (29 de Enero de 2017). *ElUniverso*. Obtenido de <https://www.eluniverso.com/noticias/2017/01/29/nota/6020314/atm-controlara-buses-traves-gps>
- Veloso, C. (2016). *Electrontools*. Obtenido de <http://www.electrontools.com/Home/WP/2016/04/01/como-funciona-el-sensor-ultrasonico-hc-sr04/>

Anexos:

ARBOL DE PROBLEMAS



F.O.D.A

FORTALEZA	OPORTUNIDADES
<ul style="list-style-type: none">• Facilita la libre circulación de vehículos.• La ciudadanía se verá obligada a esperar los buses en los paraderos establecidos.• Se reduciría el congestionamiento vehicular, más que todo en el centro.	<ul style="list-style-type: none">• Una ciudad más organizada en lo que es el transporte.• Se reduce el tiempo de llegar al destino.• Habrá más control en el transporte.
DEBILIDADES	AMENAZAS
<ul style="list-style-type: none">• Son pocas las personas que esperan los buses en los paraderos establecidos.• El poco tiempo de recorrido que han de tener los conductores.	<ul style="list-style-type: none">• Se pueden producir accidentes de tránsito.• Puede generar molestias tanto a los transeúntes como a los dueños de casa o locales cercanos con el ruido de las bocinas.

ENTREVISTA

1.- ¿Cómo considera usted el servicio de los transportes urbanos?

Malo: Regular: Bueno: Excelente:

2.- ¿Considera usted seguro los buses de transporte urbano?

Si No

3.- ¿Los buses respetan las paradas establecidas?

Nunca A veces Regularmente Siempre

4.- ¿Le gustaría que los buses solo recojan y dejen pasajeros en las paradas establecidas?

Si No

5.- ¿Qué le parece la idea de un sistema para controlar y monitorear los buses?

Malo Bueno Excelente

Análisis de la entrevista realizada

Al realizar la entrevista se tuvo constancia de las irregularidades e inconformidades que presenta la ciudadanía con respecto al servicio de transporte urbano, no hay un control fijo para verificar que los buses realicen solo paradas establecidas, dado este problema surge la propuesta tecnológica de un sistema de información para monitorear y controlar las paradas de los buses en la ciudad de Babahoyo.

Con esta propuesta se espera evitar que los buses se estacionen en el lugar que desean sin importar bloquear intersecciones y monitorearlos con tecnología GPS.

Validación de un Experto

FECHA	USUARIO	DESCRIPCIÓN	ACEPTACIÓN	FIRMA
28/09/2018	Ing. José Mejía Viteri	<ul style="list-style-type: none">• Lectura del sensor de proximidad “ultrasónico”.• Movimiento del micro servo para mover la puerta del bus dependiendo de las lecturas del sensor ultrasónico.• Visualización de los datos del GPS en el monitor serial.	80%	
28/09/2018		<ul style="list-style-type: none">• Conexión del sistema Arduino con la Aplicación Móvil.• Envío y recepción de datos de los sensores a la Aplicación Móvil• Envío de datos del GPS a la base de datos.	100%	

1.1. Validación de expertos

1.1.1. Selección de expertos

Para la selección de expertos que validen la estrategia informática para el desarrollo del prototipo de un sistema móvil para el control del parqueadero de la “F.A.F.I”.

Indicadores

A continuación, se detallan los indicadores que han permitido la validación:

- Relevancia, mide la importancia de la propuesta
- Coherencia, mide el nivel de relación entre el objetivo y la propuesta
- Pertinencia, mide si la propuesta es la adecuada
- Impacto, mide los efectos positivos que tendrá la implementación de la propuesta.

1.1.2. Guía de calificaciones

La calificación es la siguiente:

5 puntos excelente

4 puntos muy bien

3 puntos bien

2 puntos regular

1 punto mal

1.1.3. Formulario de Validación

Nombre: Ing. José Mejía Viteri

Indicador	Valoración				
	5	4	3	2	1
Relevancia	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Coherencia	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pertinencia	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Impacto	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>