

# **DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

“SISTEMA DE TELEMETRÍA CON CONECTIVIDAD WIFI Y GPRS DE LOS  
DATOS OBTENIDOS POR INTERFAZ OBDII Y UBICACIÓN VEHICULAR  
(GPS)”

# **T E S I S**

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:  
**MAESTRÍA EN INGENIERÍA ELECTRÓNICA**

ING. JUAN ANTONIO BORBOA GRIEGO

DIRECTOR:

M.C. FREDY ALBERTO HERNÁNDEZ AGUIRRE



# Índice

Capítulo 1 .....	6
1.1 Antecedentes.....	6
1.2 Justificación del tema de tesis .....	8
1.3 Planteamiento del problema a solucionar .....	9
1.4 Hipótesis .....	9
1.5 Objetivo general.....	9
1.6 Objetivos específicos .....	10
1.7 Metas y alcances .....	10
1.8 Descripción de la estructura de la tesis.....	10
Capítulo 2 .....	11
2.1 Introducción .....	11
2.2 Estado del arte.....	11
2.2.1 OBDII.....	11
2.2.2. WiFi .....	15
2.2.3. GSM .....	15
2.2.4. GPRS .....	16
2.2.5. GPS.....	16
2.2.6. Servidor de Base de datos.....	16
2.2.7. Arduino.....	17
2.2.8. Alternativas de solución.....	18
2.3. Propuesta de solución. ....	19
Capítulo 3 .....	21
3.1. Introducción .....	21
3.2. Análisis de Módulos.....	21
3.2.1. WiFi .....	21
3.2.2. OBDII.....	24
3.2.3. Bluetooth HC-05 .....	28
3.2.4. GPS NEO 6M.....	30
3.2.5. GPRS .....	31
3.2.6. Servidor de base de datos .....	37
Capítulo 4 .....	40



4.1. Introducción .....	40
4.2. Desarrollo de módulos .....	40
4.2.1. ESP8266.....	40
4.2.2. Inventek .....	42
4.2.3. Bluetooth HC-05 .....	44
4.2.4. OBDII Bluetooth Vgate iCar 2.....	45
4.2.5. OBDII freematics.....	47
4.2.6. GPS NEO 6M.....	48
4.2.5. GPRS .....	49
4.2.6. Servidor de base de datos .....	53
4.3. Selección de módulos.....	55
4.3.1. WiFi .....	56
4.3.2. GPRS.....	56
Capítulo 5 .....	58
5.1. Conclusiones generales. ....	58
5.2 Trabajos Futuros.....	58
Anexos 1. Bandas de frecuencia en México .....	59
Anexos 2. Parámetros del módulo WiFi ESP8266.....	60
Anexos 3. Parámetros del módulo WiFi Inventek .....	61
Anexos 4. Parámetros del módulo SIM800L.....	62
Anexos 5. Parámetros del módulo GSM/GPRS .....	63
Referencias Bibliográficas .....	64



## Índice de figuras.

Figura 2. 1. Terminales del Conector OBDII.....	12
Figura 2. 2. Tipos de “check engine”.....	13
Figura 2. 3. Código de Fallas de sistema OBDII.....	15
Figura 2. 4. Arduino Nano.....	18
Figura 2. 5. Propuesta de solución.....	20
Figura 3. 1. Diagrama a bloques del módulo Esp8266.....	22
Figura 3. 2. Dimensiones Esp8266.....	23
Figura 3. 3. Diagrama a bloques Inventek ISM433362 M3G.....	24
Figura 3. 4. OBDII frematics.....	26
Figura 3. 5. Diagrama interno del módulo HC-05.....	29
Figura 3. 6. Módulo HC-05.....	29
Figura 3. 7. Módulo GPS.....	30
Figura 3. 8. Dimensiones Mini GSM/GPRS Sim800L.....	32
Figura 3. 9. Mini GSM/GPRS Sim800L.....	33
Figura 3. 10. Módulo GPRS A6.....	33
Figura 3. 11. GPRS Shield Seeedstudio.....	35
Figura 3. 12. SIM900 Mini v4.0.....	37
Figura 3. 13. Servidor para la recepción y análisis de datos.....	38
Figura 3. 14. Diagrama Thingspeak.....	38
Figura 4. 1. Diagrama a bloques del funcionamiento de módulo Wi-Fi.....	41
Figura 4. 2. Diagrama a bloques del funcionamiento de módulo Wi-Fi Inventek ISM433362.....	43
Figura 4. 3. Resultados Wifi inventek.....	43
Figura 4. 4. Diagrama a bloques del funcionamiento del módulo Bluetooth.....	44
Figura 4. 5. Resultados Bluetooth HC-05.....	45
Figura 4. 6. Pasos para configurar módulo OBDII Vgate iCar 2.....	45
Figura 4. 7. Resultados OBDII Vgate iCar 2.....	47
Figura 4. 8. Datos recibidos del GPS sin estar conectado al Satélite.....	48
Figura 4. 9. Datos del GPS conectado al satélite.....	48
Figura 4. 10. Datos codificados del GPS.....	49
Figura 4. 11. Módulo GPRS Seeedstudio conectado a la red.....	51
Figura 4. 12. Datos del GPRS y datos de la página de Thingspeak.....	52



Figura 4. 13. Módulo SIM900.....	52
Figura 4. 14. PhpMyAdmin plataforma de base de datos.....	53
Figura 4. 15. Tabla que muestra la plataforma Openshift.....	54
Figura 4. 16. Parámetros del auto.....	55

## Índice de tablas.

Tabla 3. 1. Valores recibidos del módulo GPS.....	30
Tabla 4. 1. Comandos para configurar OBDII.....	46
Tabla 4. 2. Comandos para los datos del OBDII.....	47
Tabla 4. 3. Comparación de módulos WiFi.....	56
Tabla 4. 4. Comparación de módulos GPRS.....	57



# Capítulo 1

## Introducción

---

En este capítulo se presentan los temas preliminares al documento de tesis, como son los antecedentes, la justificación, objetivos y planteamiento del problema, así como una idea general del trabajo de tesis propuesto y el origen del mismo. En esta primera sección se especificará de donde surgió el proyecto y porque es importante realizarse.

### 1.1 Antecedentes

El transporte de carga terrestre constituye una herramienta fundamental e integral en la economía de México ya que contribuye en un 4.9% al Producto Interno Bruto (PIB), y genera 2 millones de empleos directos[1].

En diversas empresas del sector industrial y comercial, la utilización de flotillas es un recurso elemental dentro de las organizaciones, ya que permiten la unión entre proveedores y clientes. Dichas flotillas se definen como un conjunto de vehículos comerciales destinados a un servicio determinado asignados a distintos choferes para distribuir, repartir o proporcionar un servicio[2].

Hoy en día cualquier empresa que quiera sobrevivir y mantenerse en el mundo de los negocios, sean estos de cualquier índole, deberá adoptar y comprometerse con las tendencias que en la actualidad se van marcando[3].

Lo anterior es consecuencia del fenómeno denominado globalización, el cual ha evolucionado y tomado forma con los diferentes hechos históricos acontecidos a nivel mundial, y reconocido con mayor fuerza a partir de la década de los noventa; uniendo a esto los avances de la tecnología en el campo de las comunicaciones, transporte y sobretodo en la informática (con Internet WiFi, *Wireless Fidelity*), los cuales en su conjunto han propiciado que el mundo de hoy sea un lugar más complejo e interrelacionado a



diferencia del pasado, y a su vez han revolucionado la forma de hacer negocios hoy en día[3].

Además, en la actualidad, las facilidades de poder hacer negocios no se limitan a las fronteras de cada país, sino que ahora y desde hace ya un tiempo, las empresas cuentan con un mercado mundial al cual pueden tener acceso, originando con esto la oportunidad de la creación de empresas multinacionales, las cuales han logrado tener una expansión indiscriminada en los últimos años, provocando en cierta medida la monopolización de los mercados, y estableciendo reglas generales de cumplimiento para aquellas empresas que deseen hacer negocios a nivel internacional, tales como son las normas de calidad, entre otras. Es importante que las empresas sean partícipes de la nueva interacción que vive el mundo actual, ya que de lo contrario significaría aislarse, lo cual es sinónimo de desaparecer, por lo que ninguna empresa que desee avanzar o mejorar en el mundo de la distribución de productos o servicios lo podrá hacer realidad[3].

Como se mencionó, para las empresas es de suma importancia la utilización de flotillas para el transporte de su material, productos o servicios, pero para los dueños de las empresas también es importante saber si se está dando un buen uso al presupuesto que se otorga para el transporte. Es por esto que existe la necesidad de poder monitorear lo que está sucediendo con las flotillas. Una de las formas que se ha estado estudiando y mejorando para el monitoreo de flotillas es el uso de el "*On Board Diagnostics*" (Diagnóstico a bordo, OBDII).

En los últimos años los fabricantes de autos han optado por estandarizar los sistemas de diagnóstico a bordo (OBD), recopilándose toda la información necesaria como control de emisiones, codificación de las averías de los componentes del sistema, información del vehículo, entre los más importantes. Por lo tanto, esta información queda accesible a quien lo considere necesario. Pero, ¿Cómo se podría utilizar la información obtenida del OBD-II? Una de las formas que se puede usar es enviando los datos obtenidos por medio de comunicación WiFi. Otra opción posible es por medio de mensajes GSM (*Global System*



*for Mobile communications*) al coordinador de las flotillas que quiera tener una visión clara y en tiempo real sobre lo que sucede con el transporte. Y por último otro medio de transmisión de datos sería por GPRS (*Global Packet Raio Service*) pudiendo utilizar Internet con la ayuda de los llamados “datos” del celular.

Actualmente, en la empresa Integradora Ambiental del Pacífico (INAMPA) existe un prototipo que se encarga de recibir los siguientes datos: posicionamiento global usando GPS (*Global Positioning System*), nivel de gasolina y odómetro por el OBDII. Estos datos son transmitidos de forma inalámbrica e Internet a una red local, con la finalidad de analizarlos y validarlos de acuerdo con las especificaciones o normas de transporte establecidas por la empresa. Para la comunicación de datos se utilizan dos módulos de comunicación inalámbrica con tecnologías WiFi y GSM/GPRS.

## **1.2 Justificación del tema de tesis**

Este proyecto surge por la necesidad de crear un mejor diseño para el prototipo titulado “Control de combustible por interfaz OBDII y posicionamiento global usando redes inalámbricas”, presentado en la convocatoria del programa de estímulos a la investigación, desarrollo tecnológico e innovación 2014; debido a que es indispensable obtener un modelo más compacto que sea adaptable a cualquier tipo de automóvil o transporte de carga. Lo anterior se logrará reduciendo el espacio del circuito impreso, el mismo que es ocupado principalmente por los componentes del módulo WiFi y GSM.

El enfoque de este proyecto es integrar una plataforma en la nube que permita un mejor flujo de información entre el sistema OBDII hacia el usuario, donde se facilite el envío de datos y estos se puedan descargar de manera rápida puesto que el rango de distancia de transferencia de datos de Wifi a la nube supera al obtenido por otro medio.





### **1.3 Planteamiento del problema a solucionar**

El prototipo desarrollado en la empresa INAMPA, para solucionar la problemática que existe del control deficiente de flotillas vehiculares, presenta dos grandes inconvenientes, el primero está relacionado con las dimensiones del prototipo, es demasiado grandes en cuanto a las necesidades de distribución del espacio que debe ocupar en los vehículos, y el segundo inconveniente tiene que ver con la poca adquisición de datos, ya que solamente se pueden obtener 3 parámetros (Odómetro, nivel de gasolina y posicionamiento global).

### **1.4 Hipótesis**

Es posible obtener un prototipo funcional y mejorado con respecto a sus dimensiones y estructura física, así como sustituir el sistema central de procesamiento actual (microcontrolador) por la plataforma Arduino, implicando modificaciones en el código de programación (Firmware) para el módulo GSM/GPRS y WiFi, de tal manera que se pueda realizar la transferencia de información hacia un servidor de base de datos en la nube, logrando con esto desarrollar una propuesta de solución adaptable al sistema de flotillas para su administración y control.

### **1.5 Objetivo general**

Desarrollar un prototipo para la adquisición de datos en flotillas vehiculares a través del puerto OBDII, con ayuda de un módulo de ubicación vehicular GPS para su transmisión por redes inalámbricas WiFi-GSM/GPRS, hacia un servidor de base de datos para la administración de servicios y mantenimiento de las unidades.



## 1.6 Objetivos específicos

1. Analizar y seleccionar los módulos OBD-II, WiFi, GSM/GPRS y GPS.
2. Desarrollar o seleccionar una plataforma de Internet para la transferencia de datos.
3. Desarrollar código para OBD-II, comunicación con WiFi, GSM/GPRS, GPS.
4. Integrar todos los módulos para desarrollar un prototipo.

## 1.7 Metas y alcances

- Integrar una plataforma en Internet para la transferencia de datos lo más cercano a tiempo real vía WiFi.
- Cambiar el uso del módulo GSM para mensajes a GPRS y utilizar datos para Internet.
- Programación C de Arduino para los módulos WiFi, GSM/GPRS, OBDII y GPS.
- Presentación de diseño y propuesta del sistema de control de monitoreo de flotillas.
- Exponer al menos en un congreso o simposio los resultados obtenidos.

## 1.8 Descripción de la estructura de la tesis

El documento de tesis está organizado en 5 capítulos y varios anexos. En el capítulo 2 se presentan los conceptos básicos de los componentes y módulos que se utilizaron durante las pruebas realizadas, así como otras alternativas que se han creado para dar solución a la misma problemática. En el capítulo 3 se describe una introducción en general de cada uno de los módulos y sus características, esto con la finalidad de comparar las diversas opciones y seleccionar la más apropiada de acuerdo con los resultados obtenidos. En el capítulo 4 se describen las pruebas realizadas con los diferentes módulos WiFi, GSM-GPRS y GPS, se exponen los resultados obtenidos y la justificación para cada uno de los módulos seleccionados. Finalmente, en el capítulo 5 se explican las conclusiones y trabajos futuros.



# Capítulo 2

## Propuesta de solución

---

### 2.1 Introducción

Este capítulo abordará los temas enfocados con el estado del arte, iniciando en la sección 2.2.1 con los conceptos preliminares del OBDII, funcionamiento, estructura, códigos de falla, etc. Seguido por los conceptos básicos y definición de Wifi en la sección 2.2.2, además de la transmisión de datos y las velocidades de cada protocolo. En la sección 2.2.3 se exponen conceptos básicos de la tecnología GSM; mientras que, la sección 2.2.4 abordará la tecnología GPRS, continuando en la sección 2.2.5 con GPS y finalizando con los conceptos preliminares, en la sección 2.2.6 se presentan los servidores de base de datos. En la sección 2.2.7 se detallarán algunas alternativas de solución ante la problemática que se presenta sobre la solución a la administración de flotillas. Culminando con el capítulo 2 en la sección 2.3 con la propuesta de solución.

### 2.2 Estado del arte

#### 2.2.1 OBDII

La razón por la cual se desarrolló el estándar OBDII como tal, fue para identificar y diagnosticar las emisiones de los carros que estaban causando bastante daño al medio ambiente en el que vivimos equipándolos con un sofisticado sistema electrónico capaz de detectar dichas emisiones. El problema con la primera versión del primer OBDII fue que, entre más avanzaba el tiempo, más se complicaba todo el sistema del automóvil, provocando así que el primer OBD se volviera obsoleto a causa de las indetectables emisiones que presentaban los autos más nuevos. Pero, así como los automóviles avanzan, también el desarrollo del OBD, creando de esta forma un nuevo y mejorado sistema al que se le llamó OBDII, cercano al año de 1996. Para este año fue obligatorio en



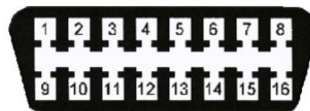
forma legal, que todos los autos contaran con tal sistema. Aunque en Europa, la fecha y los tipos de automóviles en los que se implementó este sistema variaron, al final se requirió que todos los autos contaran con un OBDII[4].

El módulo OBDII opera bajo 3 distintos protocolos:

1. Protocolo ISO (*International Organization for Standardization*) 9141.
2. Protocolo SAE (*Society of Automotive Engineers*) J1850 VPW (*Variable Pulse Width*).
3. Protocolo SAE J1850 PWM (*Pulse-width modulation*).

Los distintos tipos de protocolos del OBD-II se encuentran en determinadas terminales del conector, como se muestra en la Figura 2.1.

#### Terminales del Conector OBDII



1 - Sin uso	9 - Sin uso
2 - J1850 Bus positivo	10 - J1850 Bus negativo
3 - Sin uso	11 - Sin uso
4 - Tierra del Vehículo	12 - Sin uso
5 - Tierra de la Señal	13 - Tierra de la señal
6 - CAN High	14 - CAN Low
7 - ISO 9141-2 - Línea K	15 - ISO 9141-2 - Línea L
8 - Sin uso	16 - Batería - positivo

Figura 2. 1. Terminales del Conector OBDII.

Cuando el OBD-II del vehículo detecta alguna falla para las que fue programado, en el tablero se mostrará una luz de “*Check engine*” o “Checar el motor”, que puede variar dependiendo del año o marca del vehículo, como se puede observar en la Figura 2.2.



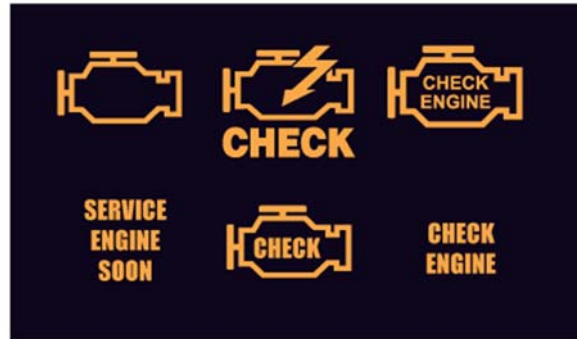


Figura 2. 2. Tipos de “check engine”.

La luz “*Check Engine*” Tiene tres tipos de señales:

- Encendida permanentemente si el problema es más grave.
- Parpadeo constante si el problema es muy grave y puede causar un daño serio si el motor no es apagado de inmediato.
- En todos los casos, se toma una lectura de todos los sensores que es guardada en la computadora central del vehículo[5].

Si la señal de falla es causada por un problema serio, la luz MIL (*Malfunction Indicator Light*) estará encendida hasta que el problema sea resuelto y la luz MIL restablecida (*reset*)[5].

Las fallas intermitentes encienden la luz MIL momentáneamente y se apaga antes de que el problema sea localizado. Sin embargo, si el vehículo completa tres ciclos de manejo sin que vuelva a aparecer el problema, la lectura es borrada [5].

#### 2.2.1.1. Código de Falla (DTC, Diagnostic Trouble Code)

Los códigos de falla están regulados por la norma SAE J1979 y es el estándar que, hoy en día, usan los fabricantes de vehículos. Los códigos de falla constan de 5 caracteres, que son una letra seguida de cuatro números[5].



El primer carácter, que es una letra, indica la función del vehículo de acuerdo a lo siguiente:

- P - Tren motriz o motor y transmisión (*Powertrain*)
- B - Carrocería (*Body*)
- C - Chasis (*Chassis*)
- U - No definido (*Undefined*)

El segundo carácter indica si el código es genérico, definido por SAE o específico, definido por el fabricante del vehículo[5].

- 0 – Genérico para todas las marcas y definido por SAE.
- 1 – Específico definido por el fabricante del vehículo, el código generalmente es diferente para cada fabricante.

Los códigos del 0001 al 0999 son definidos completamente por SAE. Los códigos del 1000 al 1999 son definidos por el fabricante y solo siguen la norma SAE en el formato[5].

El tercer carácter indica el subsistema del vehículo:

- 0 - El sistema electrónico completo.
- 1 y 2 - Control combustión.
- 3 - Sistema de encendido.
- 4 - Control de emisión auxiliar.
- 5 - Control de velocidad y ralentí.
- 6 - ECU (*Engine Control Unit*) y entradas y salidas.
- 7 - Transmisión.

Los caracteres cuarto y quinto indican la falla, la Figura 2.3 describe con un ejemplo del código de falla[5].



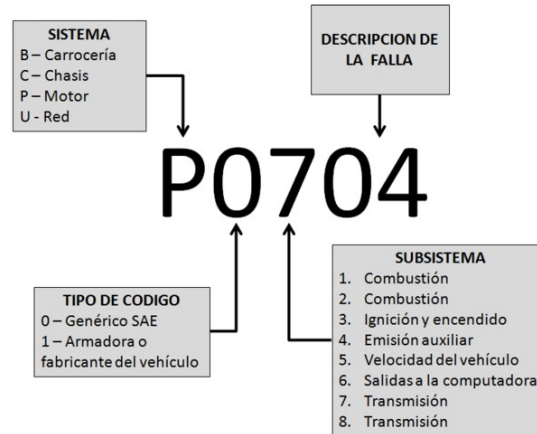


Figura 2. 3. Código de Fallas de sistema OBDII.

## 2.2.2. WiFi

WiFi es una tecnología que permite que una gran variedad de equipos informáticos puedan interconectarse sin necesidad de utilizar cables. Una de las principales ventajas de Wi-Fi es que utiliza el mismo protocolo que Internet (protocolo TCP/IP). Para que un equipo informático pueda comunicarse con el punto de acceso, es necesario un adaptador de red. Esta unidad de radio es la que le ofrece al equipo la posibilidad de comunicarse de forma inalámbrica. Existen otras tecnologías inalámbricas que son más usadas como el GSM.[6].

En la comunicación Wi-Fi existen varias normas de velocidad de comunicación y estas son:

IEEE 802.11b. Norma que permite velocidades de transmisión de hasta 11 Mbps [6].

IEEE 802.11a. Norma que permite velocidades de transmisión de hasta 54 Mbps [6].

IEEE 802.11g. Norma que permite velocidades de transmisión de hasta 54 Mbps [6].

IEEE 802.11a. Norma que permite velocidades de transmisión de hasta 300 Mbps [6].

## 2.2.3. GSM

GSM es un sistema de telefonía móvil digital que se utiliza ampliamente en Europa y otras partes del mundo. GSM utiliza una variación de acceso múltiple por división de tiempo (TDMA, *Time-Division Multiple Access*) y es el más ampliamente utilizado de las tres



tecnologías de telefonía inalámbrica digital (TDMA, GSM y CDMA, *Code Division Multiple Access*). GSM digitaliza y comprime los datos, a continuación, envía a través de un canal con otras dos corrientes de datos de usuario, cada uno en su propia ranura de tiempo. Opera en cualquiera de los 900 MHz o 1800 MHz banda de frecuencia[7].

GSM, junto con otras tecnologías , es parte de la evolución de telecomunicaciones, teléfonos inalámbricos que incluye conmutación de circuitos de alta velocidad de datos (HSCSD), sistema de radio por paquetes (GPRS), Enhanced Data Rates for **GSM** Evolution (EDGE), y el Servicio Universal de Telecomunicaciones Móviles (UMTS) [7].

#### **2.2.4. GPRS**

GSM/GPRS es una tecnología que soporta aplicaciones en red para dispositivos a través del protocolo WAP (*Wireless Application Protocol*), mensajería instantánea, SMS (*Servicio de Mensaje Corto*), MMS (*Servicio de Mensajería Multimedia*) y servicios peer to peer utilizando el protocolo IP (*Internet Protocol*). Para obtener estos servicios es necesario realizar un pago que se basa en megabytes por transferencia [8].

#### **2.2.5. GPS**

El GPS es un sistema de posicionamiento por satélites desarrollado por el Departamento de la Defensa de los E.U., diseñado para apoyar los requerimientos de navegación y posicionamiento precisos con fines militares. En la actualidad es una herramienta importante para aplicaciones de navegación, posicionamientos de puntos en tierra, mar y aire [9].

#### **2.2.6. Servidor de Base de datos**

Es un software que se está ejecutando y puede atender las peticiones de un cliente y devolverle una respuesta si lo requiere. Dicho servidor permite el almacenamiento, modificación y extracción de información, y dependiendo de cómo haya sido su programación se puede modificar parte de la base de datos como borrar, modificar, consultas, analizar datos, etc[10].





### 2.2.7. Arduino

Es una plataforma de código abierto (*open source*) muy fácil de utilizar en un inicio, ya que cuenta con ayuda de bastantes desarrolladores de todo el mundo para seguir mejorando y facilitando la intervención de diferentes usuarios[11]. Es una de las razones por las que se decide utilizar Arduino en el proyecto de tesis ya que existe bastante información para los diferentes módulos que se usarán. Otra de las razones es por la flexibilidad que tiene Arduino para programar distintos módulos con conexión serial.

La plataforma Arduino utilizada es un modelo Nano, que presenta las siguientes características:

- 14 puertos digitales de entrada/salida.
- 6 puertos PWM.
- 8 puertos análogos.
- Utiliza un microcontrolador ATmega328
- Una memoria de 16 KB (*Kilo bytes*), 1 KB de SRAM (*Static Random Access Memory*) y 512 bytes de EPROM (*Erasable Programmable Read-Only Memory*).
- La velocidad de reloj es 16 MHz (*Mega Hertz*).
- Voltaje de entrada en el rango de 7-12 volts.

La plataforma Arduino se utiliza para configurar todos los módulos convirtiéndolo en la CPU (*Central Processing Unit*) del proyecto. El Arduino completo con sus diferentes entradas y salidas y características eléctricas se puede apreciar en la figura 2.4.



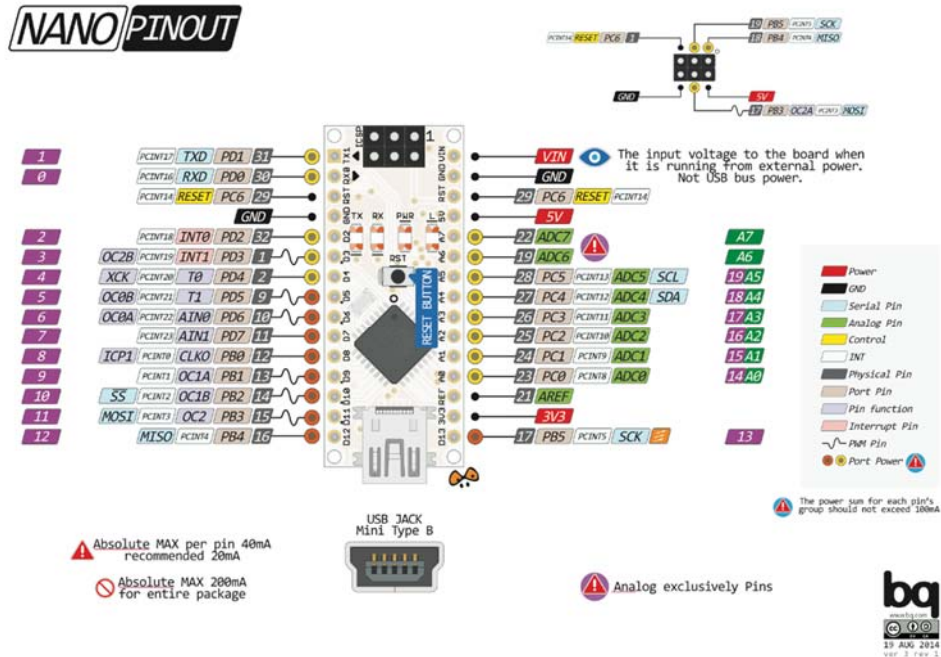


Figura 2. 4. Arduino Nano.

## 2.2.8. Alternativas de solución

### 2.2.8.1. Navman Wireless Director

Navman Wireless Director es una de las soluciones que existen en el mercado acerca de la administración de flotillas. Cuenta con un tablero dinámico en el que se puede monitorear los indicadores de clave de rendimiento o KPI, por sus siglas en inglés, para verificar tendencias de desempeño. Genera información resumida y detallada para identificar y corregir los problemas que se presenten, también cuenta con una aplicación para celular, entre otros [12].



### **2.2.7.2 Origis Administración de flotillas**

Origis ofrece una solución de administración de flotillas, donde se puede programar a las unidades de transporte, distintas alertas para administrar y controlar su velocidad, suavizar el manejo en curvas, localizar en dónde se encuentra georreferenciada cada una de las unidades, optimiza los recursos reduciendo tiempos de espera, horas extra y paradas innecesarias en ruta. De igual forma se puede monitorear en línea el uso del combustible, conociendo el nivel exacto por medio de un sensor, también la capacidad de llenado y así poder determinar el consumo de acuerdo a la distancia recorrida detectando si hay robo o mal uso de este[13].

### **2.2.7.3 Facileasing**

Es una empresa dedicada a la administración de flotillas de autos o camiones. Esta empresa maneja la administración del mantenimiento de los vehículos (con una red de proveedores), manejo administrativo (pago de tenencia, placas, multas, rescate de corralón), planes de pago de gasolina (limitando a los conductores a tomar gasolina si está fuera de su área, horarios o en días no laborables [14].

## **2.3. Propuesta de solución.**

Se propone implementar un circuito, utilizando la plataforma Arduino, con unas dimensiones reducidas a un mínimo de un 40% del prototipo realizado por la empresa INAMPA, además de adquirir más datos del OBD-II del vehículo. Estos datos serán enviados a un servidor de base de datos en la nube utilizando conexión WiFi y GPRS. La propuesta de solución consiste en:

1. Modificar el firmware del microcontrolador para utilizar los módulos Wi-Fi, GSM/GPRS y GPS.
2. Modificar el firmware del microcontrolador de Arduino para obtener una mayor lectura de datos por la interfaz OBDII.
3. Añadir una comunicación con el dispositivo vía puerto serial.



4. Enviar los datos obtenidos del OBDII hacia una plataforma en Internet.

Lo anterior se expone de manera más explícita en la figura 2.5:

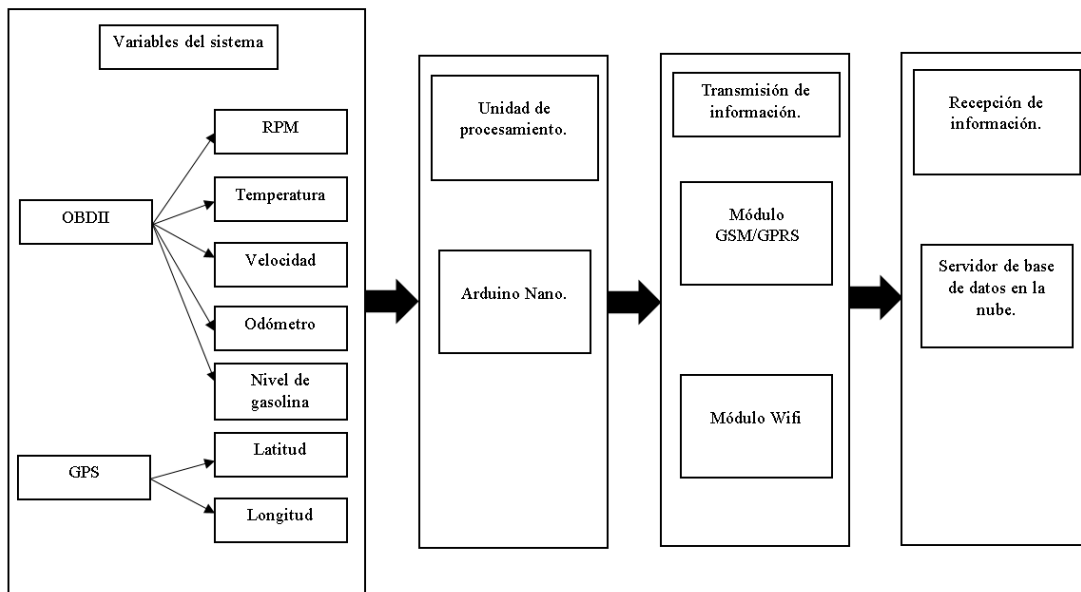


Figura 2. 5. Propuesta de solución.



# Capítulo 3

## Análisis de módulos del prototipo

### 3.1. Introducción

En el siguiente apartado se presentan las características y/o parámetros de los módulos que se seleccionaron para poder realizar el proyecto. De las secciones 3.2.1 a la 3.2.5 se analizan las características, especificaciones y un resumen general de dos módulos WiFi, dos módulos OBDII, un módulo Bluetooth, GPS y GPRS. En la sección 3.2.6 se presentan las características de algunos servidores de base de datos, uno como servidor de área local que se utiliza como base para la nube, y dos servidores en la nube. Un servidor requiere varias modificaciones para su configuración y uso de MySQL entre otros programas, y el otro es una aplicación de servidor para base de datos. La selección de módulos se realizó en base al precio, versatilidad, conocimiento que se tiene del módulo y sus dimensiones.

### 3.2. Análisis de Módulos.

#### 3.2.1. WiFi

##### 3.2.1.1. ESP8266

El primer módulo WiFi que se presenta es el ESP8266 que cuenta con los siguientes parámetros y características presentados por la tabla 1 descrita en Anexos 2 [15]:

- 802.11 b/g/n
- Protocolo TCP/IP integrado.
- PLL, reguladores y administración de energía.
- Diversidad en el soporte de antena.
- WiFi 2.4 GHz, soporte WPA/WPA2
- Soporte a modos de operación STA/AP/STA+AP
- SDIO 2.0, (H) SPI, UART, I2C, I2S, PWM, GPIO



- Protocolo IPv4, TCP/HTTP.
- Alimentación 3.0 ~ 3.6 V.
- Corriente de operación: 170 mA max.
- Alimentación en modo Sleep <10  $\mu$ A, corriente de fuga en este modo < 5  $\mu$ A
- Rango de temperatura en operación -40C ~ 125C

En la Figura 3.1 se muestra en un diagrama a bloques lo que contiene internamente el módulo ESP8266.

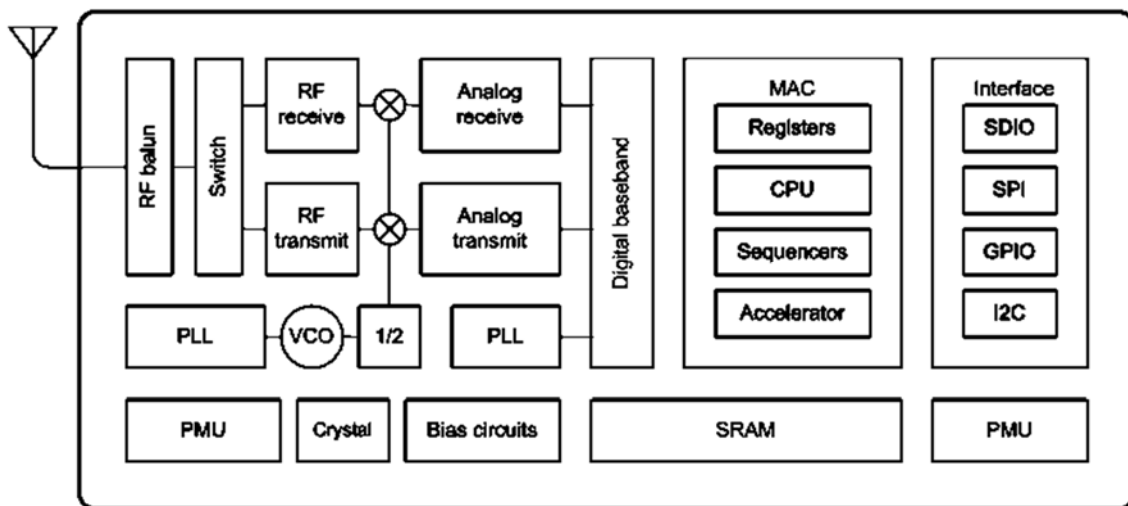


Figura 3. 1. Diagrama a bloques del módulo Esp8266.

Es importante conocer las dimensiones del módulo para poder tener en cuenta el espacio que va a utilizar en el prototipo final, es por esto que en la figura 3.2 se puede observar las dimensiones del módulo ESP8266 a utilizar.



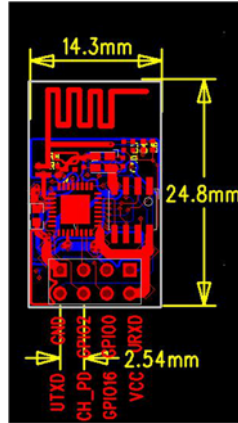


Figura 3. 2. Dimensiones Esp8266.

El módulo Esp8266 es una muy buena opción ya que puede trabajar a altas y bajas temperaturas, es fácil de programar, cuenta con protocolos TCP, las dimensiones son aceptables de acuerdo con las especificaciones del prototipo requerido, además existe bastante información para sus diversas aplicaciones ~~su uso~~ [15]. El módulo es muy usado en aplicaciones de desarrollo para plataformas IoT (*Internet of Things*), es un módulo creado por una empresa China situada en Shanghai, llamada Espressif que se encarga principalmente en el diseño de chips, desarrollo de software/firmware y en marketing, enfocado a proporcionar dispositivos y plataformas de software de IoT en la industria [16].

### 3.2.1.2. Inventek ISM433362 M3G

Este apartado aborda el tema del módulo WiFi Inventek (ISM433362 M3G), sus características y parámetros. En la figura 3.3 podemos observar en gran escala en un diagrama a bloques con lo que cuenta internamente el módulo Inventek ISM433362 M3G.



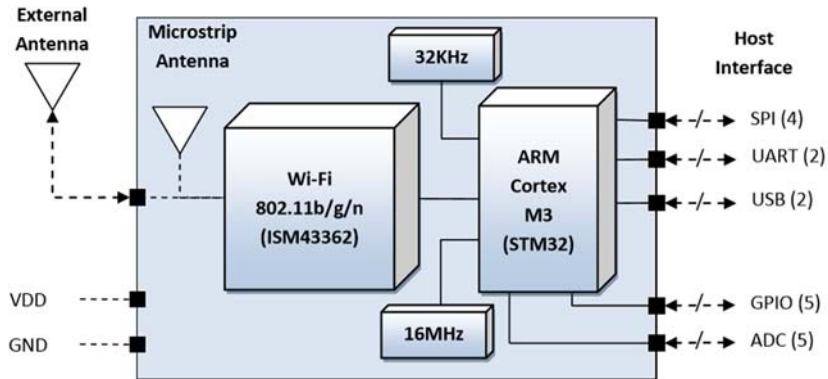


Figura 3. 3. Diagrama a bloques Inventek ISM433362 M3G.

El módulo inventek ISM433362 M3G es una opción también viable para el desarrollo del proyecto ya que cuenta con una aceptable temperatura de operación, ahorro de energía, cuenta con protocolo TCP y además cuenta con dos certificaciones.

## 3.2.2. OBDII

### 3.2.2.1. OBDII Vgate iCar 2

El primer OBDII que se presenta como prueba es uno con conexión Bluetooth para disminuir lo que sería el cableado entre el circuito de control y la conexión del OBDII. El módulo OBDII utilizado es el módulo Vgate icar 2 de la empresa OBD innovations. Este módulo cuenta con las siguientes características:

El Vgate® iCar 2 Bluetooth es una herramienta de diagnóstico y rendimiento del motor para el automóvil. Permite escanear y borrar códigos de motor, ver y guardar datos de rendimiento y motores en tiempo real y mucho más. Cuenta con una función integrada inteligente de reposo automático que lo pone en modo reposo después de 30 minutos, lo que le permite dejarlo enchufado a su vehículo en todo momento.





#### Características generales:

- Función Auto Sleep Mode con Power Switch.
- Utiliza los chips ARM más rápidos y confiables.
- Lee y borra los códigos de diagnóstico de problemas (DTC) y apague la luz del motor de verificación.
- Conexión vía Bluetooth con soporte a Android y iPhone.
- Datos del vehículo en tiempo real, gráficos y parámetros de rendimiento.
- Compatible con todos los vehículos que cumplen con OBDII.
- Conexión inalámbrica a través de Bluetooth® hasta 3 metros.

#### Cobertura de vehículos:

- OBD II (1996+ en EE. UU)
- EOBD (automóviles de gasolina de 2001+ y automóviles diésel de 2003/2004 en Europa)

#### El Vgate iCar 2 soporta todos los protocolos OBDII:

- SAE J1850 PWM (Found on many Ford vehicles)
- SAE J1850 VPW (Found on many GM vehicles)
- ISO 9141-2 (5 baud init, 10.4 kbaud)
- ISO 14230-4 KWP (5 baud init, 10.4 kbaud)
- ISO 14230-4 KWP (fast init, 10.4 kbaud)
- ISO 15765-4 CAN (11 bit ID, 500 kbaud)
- ISO 15765-4 CAN (29 bit ID, 500 kbaud)
- ISO 15765-4 CAN (11 bit ID, 250 kbaud)
- ISO 15765-4 CAN (29 bit ID, 250 kbaud)
- SAE J1939 CAN(29bit ID,250\*Kbaud)



- USER1 CAN(11\*bit ID,125\*Kbaud)
- USER2 CAN(11\*bit ID,50\*kbaud)

### 3.2.2.2. OBDII freematics.

Una de las opciones que se presentan para utilizar el OBDII es un dispositivo que cuenta con cables para la comunicación y alimentación del circuito de control. El cable OBDII fue diseñado por la empresa freematics para desarrollarse con el modulo de Arduino, mega o uno, y que cuenta con una pantalla OLED, aunque se puede comprar los componentes por separado, que muestra los parámetros del carro en tiempo real. Para este proyecto solo se pone a prueba el cable ya que permite una facilidad en lo que es la alimentacion del circuito de control. En la figura 3.4 se muestra el OBDII.



Figura 3. 4. OBDII freematics.

El cable OBDII de la marca freematics cuenta con las siguientes características:

- Acceso a todos los PID estándar OBD-II con el comando extendido ELM327 AT.
- Lectura y borrado de códigos de diagnóstico de problemas del vehículo (solo motor y tren motriz).
- Medición del voltaje de la batería del automóvil.



- Sensor de movimiento MPU-6050 6-DOF incorporado (acelerómetro, giroscopio y temperatura).
- Suministro de energía para el dispositivo host (5 V hasta 2.1 A).
- Interfaz de datos serial UART compatible con microcontroladores de 3.3V y 5V.
- Puerto micro USB para computadora / tableta acceso OBD-II y actualización de firmware.
- Modo de baja potencia a 6mA.

Se admiten vehículos que utilizan los siguientes protocolos de vehículos:

- CAN 500Kbps / 29bit
- CAN 500Kbps / 11bit
- CAN 250 Kbps / 29 bits
- CAN 250 Kbps / 11 bits
- KWP2000 Fast
- KWP2000 5Kbps
- ISO9141-2

### **Conectores:**

El adaptador permanece enchufado en el puerto OBD generalmente ubicado debajo de la columna de dirección o ligeramente a la izquierda del mismo. Un cable sale del adaptador y se divide en un conector de 4 pines, dos conectores de 2 pines, incluidas líneas de alimentación (VCC / GND) y líneas de datos (Rx / Tx).

### **Conector de alimentación** (conector dupont 2.54 de 2 pines)

- Rojo: VCC (se conecta a la VCC de Arduino)
- Negro: GND (conexión a GND de Arduino)

### **Conector de datos UART serie** (conector Dupont 2.54 de 2 pines)

- Blanco: Rx (conectado al Tx serial de Arduino)



- Verde: Tx (conectado a la serie Rx de Arduino)

### **Puerto USB**

- Para la conexión con la computadora o tableta a través de un cable micro USB
- Para salida de potencia (5V / 2A)

El cable UART parece ser una buena opción también ya que aunque ocupa la conexión física para comunicarse con el circuito de control, también cuenta con una alimentación bastante buena para el circuito con lo cual se puede pasar por alto el intento de buscar como alimentar el circuito.

### **3.2.3. Bluetooth HC-05**

En el siguiente apartado se describirán las especificaciones del módulo Bluetooth HC-05, así como información adicional de dicho módulo. La razón por la que se usará este módulo Bluetooth es por que es necesario tener un intermediario entre el OBDII Bluetooth y el Arduino para poder realizar pruebas.

La empresa desarrolladora Guangzhou Información Cheng Tecnología Co., Ltd. se encuentra en distrito de Tianhe, Guangzhou Tianhe Parque de Ciencia y Tecnología, es una empresa dedicada al desarrollo de tecnología basados en software, especializados en productos de comunicación y procesamiento de información [17]. En la figura 3.5 se presenta el diagrama interno del módulo HC-05:



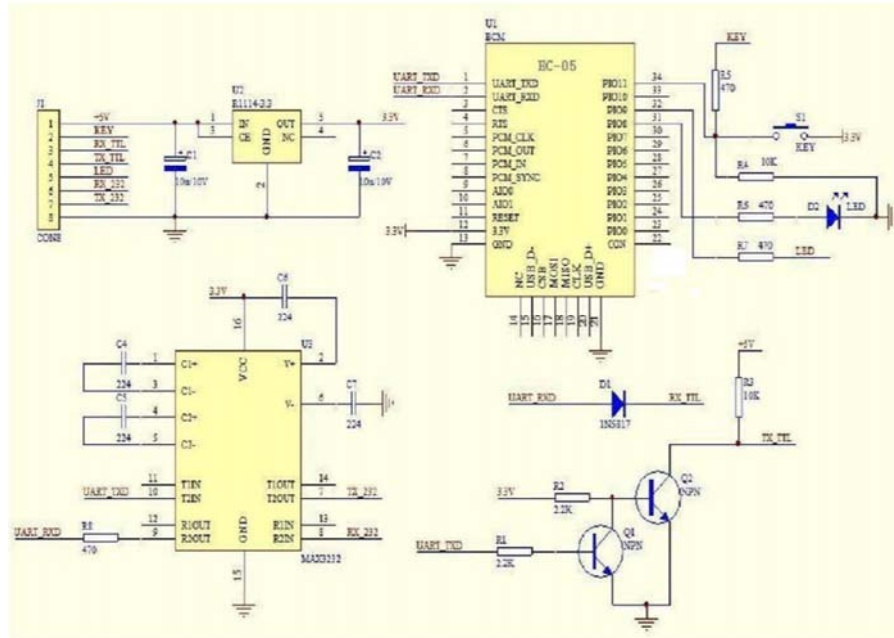


Figura 3. 5. Diagrama interno del módulo HC-05.

El módulo es una buena opción para usarse como intermediario entre el OBDII Bluetooth y el Arduino, ya que cuenta con mucha información sobre su uso y aunque existen otros módulos Bluetooth, este se usará ya que también es fácil de adquirir. El que el módulo Bluetooth sea parte del proyecto al final dependerá del módulo OBDII que se elija. En la figura 3.6 se puede observar el módulo HC-05.



Figura 3. 6. Módulo HC-05.



### 3.2.4. GPS NEO 6M

El módulo NEO 6M en la plataforma Arduino cuenta con una librería llamada TinyGPS que codifica los valores que el GPS siempre está enviando desde que se conecta a su fuente de alimentación y consigue una conexión con los satélites. El módulo GPS es mostrado en la Figura 3.7.



Figura 3. 7. Módulo GPS.

Existen ciertas características de los módulos GPS para la recepción de datos que vienen codificados bajo el protocolo de la NMEA (*National Marine Electronics Association*). En la Tabla 3.1 se puede observar los valores recibidos no codificados del módulo.

Tabla 3. 1. Valores recibidos del módulo GPS.

NMEA Record	Description
GGA	Global positioning system fixed data
GLL	Geographic position – latitude/longitude
GSA	GNSS DOP and active satellites
GSV	GNSS satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed



Características del módulo GPS utilizado:

- Voltaje de alimentación: 3.0 a 5.0 volts.
- Voltaje en pines de comunicación: 3.3 volts.
- Interfaz: UART (*Transmisor-Receptor Asíncrono Universal*), comunicación asíncrona.
- Compatible con la mayoría de librerías para GPS [protocolo NMEA (*National Marine Electronics Association*)].
- Modelo: GY-GPS6MV2

### **3.2.5. GPRS**

En la etapa de envío de datos, además de contar con conectividad Wifi, se tiene tecnología GPRS que se encarga de enviar datos a un servidor en la nube en cualquier momento que el auto se encuentre en movimiento. Para la etapa de GPRS se cuenta con diversos módulos, cuyas características son descritas a continuación.

#### **3.2.5.1. SIM800L**

Mini GSM/GPRS SIM800L, soporta una red cuatribanda, por lo que debería de poderse comunicar con las compañías de teléfono de México. La placa cuenta con un tamaño compacto, como se observa en la Figura 3.8, y bajo consumo de corriente. Se comunica con el microcontrolador a través del puerto UART, apoya al mando, incluyendo 3GPP TS 27.007, 27.005 y SIMCOM mejorado Comandos AT. El módulo permite realizar operaciones por Internet sencillas como HTTP/UDP/FTP. Funciona mediante comandos AT al igual que la mayoría de módulos y shields GSM. Utiliza el mismo chip SIM800L que el módulo FONIA de Adafruit, por lo que se pueden utilizar las mismas librerías. [18]



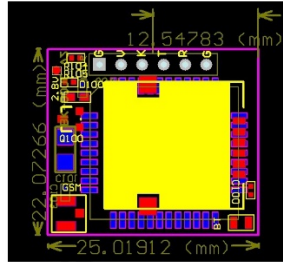


Figura 3. 8. Dimensiones Mini GSM/GPRS Sim800L

Algunas de las características, que vienen descritas en la sección de anexos 4 son las siguientes:

- Quad-banda de 850/900/1800 / 1900MHz.
- Enviar y recibir datos GPRS (TCP / IP, HTTP, etc.).
- Enviar y recibir mensajes SMS.
- Escanear y recibir emisiones de radio FM.
- Se conecta a cualquier red GSM global con cualquier tarjeta SIM 2G.
- Tension de trabajo 3.5~4.2V.
- Dimensiones: 25x23x1 mm.
- Peso: 4.7 g.
- Marca: OEM.

El módulo cuenta con varios pines que sirven para la comunicación hacia otro dispositivo por medio de UART y también cuenta con otros pines que funcionan para utilizar el módulo como un teléfono celular, por ejemplo, micrófono y bocinas, todos los pines se muestran la figura 3.9.





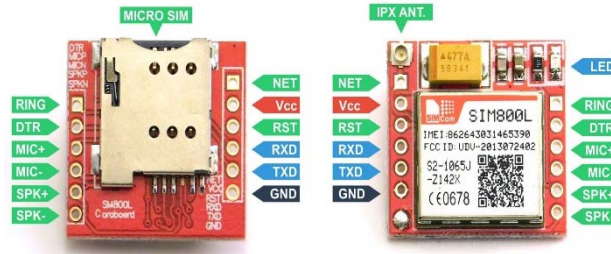


Figura 3. 9. Mini GSM/GPRS Sim800L.

### 3.2.5.2. GPRS A6

El mini GSM/GPRS, como se observa en la figura 3.10, es una tarjeta de desarrollo en base al módulo GPRS A6. Soporta cuatribanda GSM/GPRS de transmisión remota de red, cuenta con bajo consumo de energía de courant que quiere decir que puede llegar hasta los 3 mA en modo de espera. Puede comunicarse por medio tecnología UART.[19]

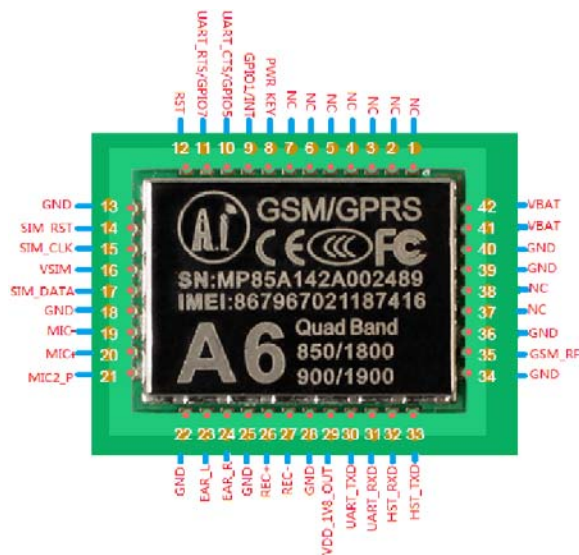


Figura 3. 10. Módulo GPRS A6.



### Parámetros generales:

- Dimensiones: 22.8 \* 16.8 \* 2.5 mm.
- Rango de temperatura: -30°C ~ 80°C.
- Voltaje de operación: 3.3 ~4.2 V.
- Corriente en standby: 3 ma.
- Soporte a 4 bandas GSM/GPRS: 850, 900, 1800, 1900 MHz.
- GPRS clase: 10.
- Transferencia de datos de bajada: 85.6 kbps.
- Transferencia de datos de subida: 42.8 kbps.
- Soporte a comandos AT: GSM07.07, 07.05.
- Soporte a comandos AT: Standard AT y TCP/IP.
- Peso: 3 g.

### 3.2.5.3. Shield GPRS Seeedstudio

El módulo shield GSM/GPRS, que se aprecia en la Figura 3.11, creado por la empresa Seeedstudio llegando a desarrollar hasta 3 versiones de la misma placa, mejorando cada vez el consumo o transmisión y recepción del shield. La versión a utilizar es la 2, la cual cuenta con un módulo SIM900 de bajo consumo, de cuádruple banda y una antena PCB compacta. El módulo fue creado para adaptarse y poder ser usado en las placas Arduino UNO y Arduino Mega, aunque puede programarse utilizando cualquier otro Arduino, la versión 2 se creó con la finalidad de ser shield de Arduino.





Figura 3. 11. GPRS Shield Seedstudio.

Algunas de las características y parámetros del Shield, que vienen descritas en la tabla 4 en la sección de anexos, se pueden observar a continuación:

- Comunicación UART o SoftwareSerial.
- Quad band: 850/900/1800/1900MHz
- Soporta comunicación: Standard - GSM 07.07 & 07.05 and Enhanced - SIMCOM AT Commands.
- Temperatura de operación: -40°C to +85 °C.
- Soporta protocolos: 0710 MUX protocol, embedded TCP/UDP protocol, FTP/HTTP, FOTA, MMS, embedded AT.
- Alimentación: 5v via 5V pin, 6.5~12v via Vin pin
- Dimensiones: 68.58 \* 53.34mm.

#### 3.2.5.4. SIM900 Mini v4.0

##### Descripción

Módulo SIM900 Mini v4.0 que puede soportar red europea (850MHZ, 900MHZ, 1800MHZ, 1900MHZ). Sus principales características son:

1. Seis elementos para garantizar el rendimiento y la estabilidad:



Núcleo: nueva versión mejorada original SIM900A (no el tipo de lastre y reformado).

Condensador de filtro de fuente de alimentación: condensadores de tántalo de chip de calidad 1000uf.

Circuito anti-interferencia: TVS tubo de aumento, antiestático.

Conector: cabezal IPMA cabeza MMA y aguja de cobre estañado.

Proceso de producción: tecnología de chip industrial (alta temperatura para garantizar que el producto no se dañe).

Embalaje: bolsa antiestática y bolsa de burbuja a prueba de golpes.

2. Completamente funcional: teléfono, SMS, comunicación de red GPRS, mensajería multimedia (MMS), ubicación de la estación base (LBS), multifrecuencia de doble tono (DTMF)

GPRS: Protocolo TCP / IP incorporado, IP pública directa y fija de las comunicaciones de la computadora, amplia supervisión de datos.

LBS: ubicación de la estación base; latitud y longitud a través del módulo de lectura en serie, y avise a la policía que usted es un teléfono celular que es el mismo. A través de esto, podemos saber dónde están sus aplicaciones de equipos.

DTMF: Multifrecuencia de tono dual, es decir, durante una llamada a través del puerto en serie que se puede leer con otra tecla, y presionamos 10086, voz automatizada para que seleccionemos presionar 1, 2, 3 de la misma manera. Podemos controlar el dispositivo a través de esta acción, el control remoto solo.

Las funciones anteriores se basan en el puerto serie del nivel TTL, no es necesario simular con el serial serial IO. La depuración es mejor utilizar las palabras del TTL 232 o 3232 para probar el mercado, muchos de PL2303 buen uso.





Figura 3. 12. SIM900 Mini v4.0.

### 3.2.6. Servidor de base de datos

Finalmente, la etapa de análisis de los datos, consiste en la recepción de datos a un servidor, esta es una plataforma que ayuda a visualizar de una forma ordenada y con un formato agradable a la vista del usuario los datos que se reciban de los módulos WiFi y GPRS.

#### 3.2.6.1. Openshift

Red Hat® OpenShift es una plataforma de aplicaciones en contenedores, una imagen que representa openshift esa mostrada en la figura 3.13, que proporciona a la empresa docker y kubernetes. Independientemente de la arquitectura de sus aplicaciones, OpenShift le permite diseñar, desarrollar e implementar de forma fácil y rápida en casi cualquier infraestructura, pública o privada. Ya sea de forma local, alojada o en una nube pública, usted tiene una plataforma galardonada para obtener su próxima gran idea para comercializar antes de que su competencia lo haga [20].





Figura 3. 13. Servidor para la recepción y análisis de datos.

### 3.2.6.2. Thingspeak

ThingSpeak™ es un servicio de plataforma de análisis IoT que le permite agregar, visualizar y analizar flujos de datos en vivo en la nube. ThingSpeak, proporciona visualizaciones instantáneas de los datos publicados por sus dispositivos a ThingSpeak. Con la capacidad de ejecutar el código MATLAB® en ThingSpeak, puede realizar análisis y procesamiento en línea de los datos a medida que ingresa. ThingSpeak se usa a menudo para prototipos y sistemas de IoT de prueba de concepto que requieren análisis[21]. En figura 3.14 se muestra un diagrama a bloques de cómo funciona thingspeak.

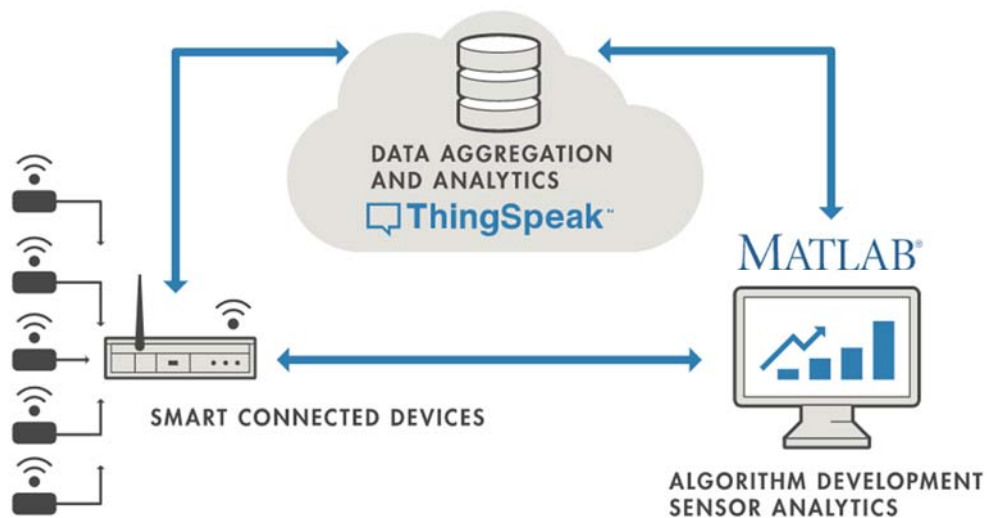


Figura 3. 14. Diagrama Thingspeak.



### **Funciones principales de Thingspeak**

ThingSpeak permite agregar, visualizar y analizar flujos de datos en vivo en la nube. Algunas de las capacidades clave de ThingSpeak incluyen la capacidad de:

- Configurar fácilmente los dispositivos para enviar datos a ThingSpeak utilizando los populares protocolos de IoT.
- Visualizar los datos de su sensor en tiempo real.
- Datos agregados a solicitud de fuentes de terceros.
- Usar el poder de MATLAB para dar sentido a sus datos de IoT.
- Ejecutar análisis de IoT de forma automática en función de los horarios o eventos.
- Prototipo y construcción de sistemas IoT sin configurar servidores o desarrollar software web[21].

En general, ambos softwares en la nube parecen buenas opciones para trabajar, openshift permite enviar documentos y sistemas para configurar el espacio en la nube y hacer la página web de una forma que le parezca mejor al usuario. Thingspeak por otra parte es una plataforma en Internet que te permite enviar datos que ya se encuentran estructurados y acomodados dentro de las configuraciones que pide la página, pero también hace que esto se vuelva aún más sencillo de enviar los datos.



# Capítulo 4

## Desarrollo y selección de módulos

### 4.1. Introducción

El capítulo 4 trata los temas del desarrollo de cada uno de los módulos propuestos en el capítulo 3. Iniciando con los puntos 4.2.1 y 4.2.2, el funcionamiento de los módulos WiFi. En el punto 4.2.3 se desarrolla el método para crear la comunicación entre el módulo OBDII Bluetooth y el Arduino. Siguiendo con los puntos 4.2.4 y 4.2.5 que muestran el desarrollo de cómo establecer una comunicación entre el OBDII del auto y la computadora usando dos alternativas, un OBDII Bluetooth y otra OBDII con cable.

### 4.2. Desarrollo de módulos

#### 4.2.1. ESP8266

En esta sección se desarrollan los pasos para conectar el módulo ESP8266 a Internet y enviar datos. El módulo WiFi necesita ciertos pasos para poderse conectar a una red y después tener acceso a Internet. Lo primero que se realiza es la configuración del módulo como cliente, después se establece el nombre de la red WiFi y su contraseña. Una vez conectado a la red se realiza la solicitud del módulo para tener acceso al servidor en la nube. En el paso final, el módulo WiFi trata de enviar una trama, cuya longitud contiene los datos requeridos para que el servidor pueda reconocer los parámetros del OBDII y GPS, para lo cual utiliza etiquetas y sus respectivos valores.





A continuación se detallan los pasos para la configuración del módulo WiFi:

1. Establecer que el puerto trabaja a 115200 bauds.
2. Resetear el módulo.
3. Configurar el módulo a 9600 bauds.
4. Configurar el módulo como cliente.
5. Configurar Internet usuario y contraseña para después conectarse.
6. Obtener dirección IP del módulo.
7. Configurar el módulo para múltiples conexiones.
8. Conectarse a la dirección TCP del servidor.
9. Enviar la longitud de la trama.
10. Enviar el dato.
11. Repetir pasos desde el 5.

En la figura 4.1 se muestra un diagrama a bloques que presenta los pasos a seguir para el envío datos.

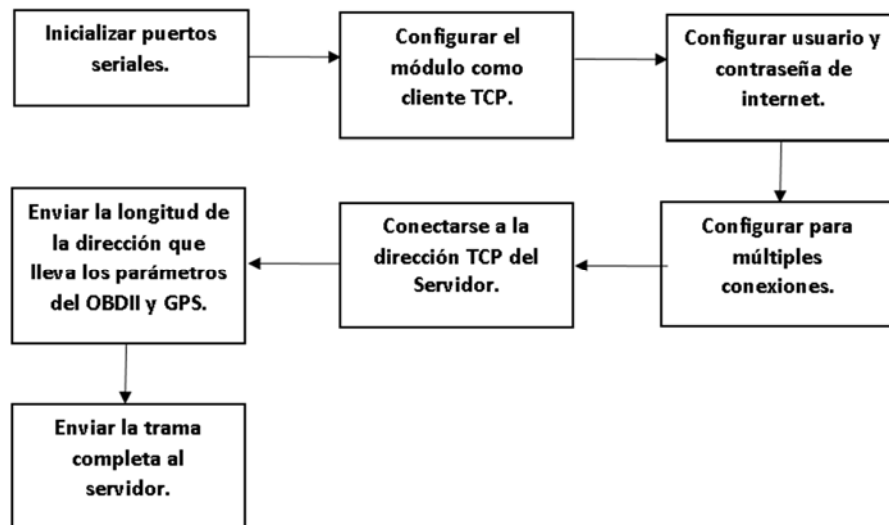


Figura 4. 1. Diagrama a bloques del funcionamiento de módulo Wi-Fi.



Después de enviar el dato correctamente se deberán repetir los tres últimos pasos para que el dato sea enviado cada cierto tiempo.

#### **4.2.2. Inventek**

En este apartado se hablará sobre las características físicas del módulo Wifi Inventek (ISM433362 M3G) para poder enviar datos hacia un cliente configurando el modulo como servidor TCP.

Primeramente se configura el módulo conectándolo a una red WiFi, y después se realizan los siguientes pasos:

- 1.- Elegir el paquete que se va a usar.
- 2.- Configurar el módulo WiFi como TCP server.
- 3.- Configurar el número de puerto que se utilizará.
- 4.- Se elimina cualquier cliente que se pudo haber conectado antes para no tener errores.
- 5.- Iniciar el server.
- 6.- Definir el tiempo máximo de espera para la escritura.
- 7.- Definir la longitud del dato a enviar.
- 8.- Enviar el dato.

Después de enviar el dato correctamente se deberán repetir los dos últimos pasos cada cierto tiempo.

La figura 4.2 muestra un diagrama a bloques donde se presentan los pasos a seguir para el envío de datos hacia un cliente.



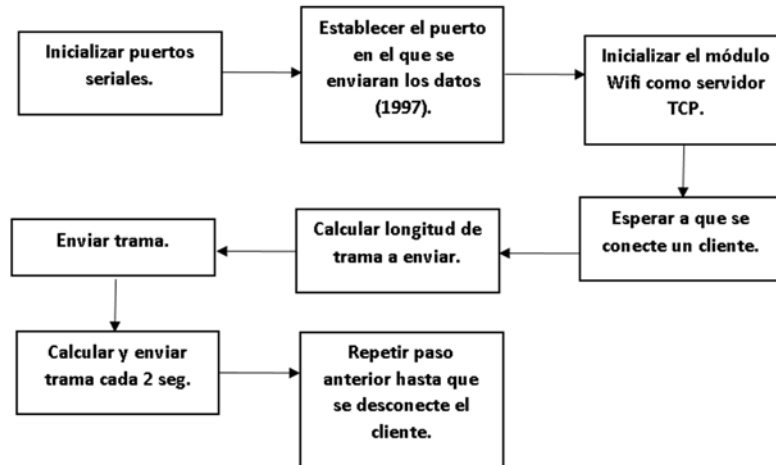


Figura 4. 2. Diagrama a bloques del funcionamiento de módulo Wi-Fi Inventek ISM433362.

Para realizar pruebas con el módulo WiFi se utilizó un potenciómetro para enviar valores variables de voltaje y se mostraban en un celular que contaba con la ayuda de una aplicación TCP *client*. El resultado de la variación de voltaje se muestra en una pantalla en forma de fila y en la parte superior de la aplicación se puede observar la dirección IP que se usó y también el puerto de comunicación, como se muestra en la figura 4.3.

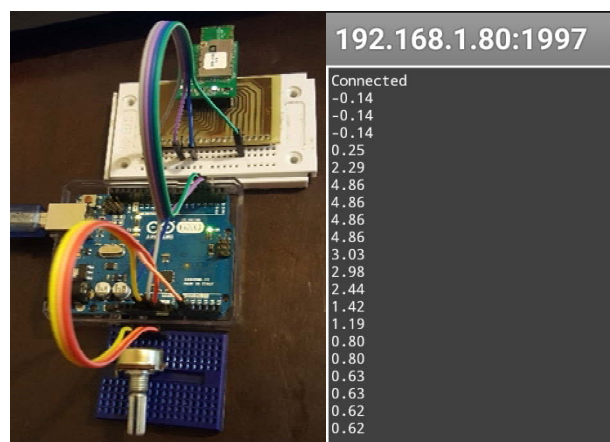


Figura 4. 3. Resultados Wifi inventek.



### 4.2.3. Bluetooth HC-05

En este apartado se explica sobre la configuración del módulo Bluetooth del Arduino para poder conectarse al módulo Bluetooth del OBDII.

En la figura 4.4 muestra un diagrama a bloques que presenta los pasos a seguir para la conexión con el módulo OBDII.

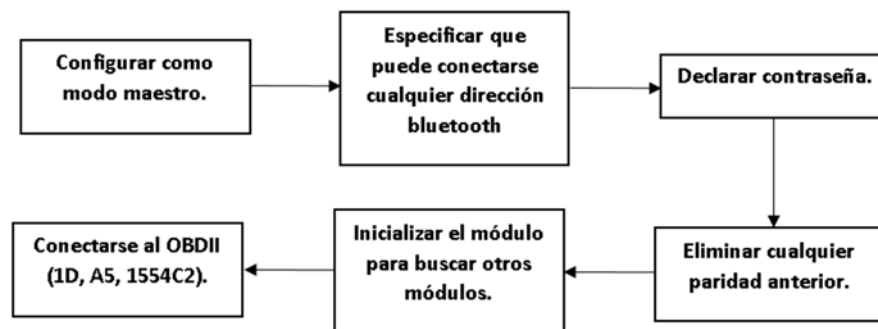


Figura 4. 4. Diagrama a bloques del funcionamiento del módulo Bluetooth.

Los pasos para habilitar la opción del módulo Bluetooth del Arduino para poder conectarse con el Bluetooth del OBDII serían los siguientes:

- 1.- Elegir el rol del módulo (Rol = 0, Modo esclavo; Rol = 1, Modo maestro).
- 2.- Elegir el modo de apareamiento (CMODE = 0, conectarse con la dirección especificada; CMODE = 1, conectarse con cualquier dirección).
- 3.- Declarar la contraseña.
- 4.- Eliminar cualquier paridad anterior.
- 5.- Inicializar el módulo para que habilite la opción de buscar otros módulos Bluetooth.
- 6.- Establecer la dirección del módulo Bluetooth del OBDII (1D, A5, 1554C2).

Una vez lograda la comunicación con el módulo OBDII, se procede salir de los comandos AT para el Bluetooth del Arduino y trabajar completamente en recibir los datos enviados del Bluetooth del OBDII.



En la figura 4.5 se muestra un ejemplo de cómo se obtienen los datos en esta sección de configurar el módulo Bluetooth.

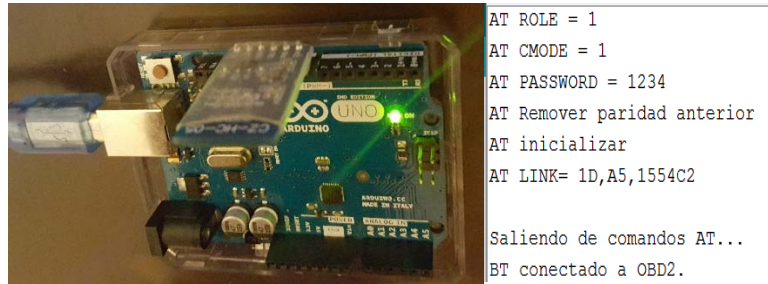


Figura 4. 5. Resultados Bluetooth HC-05.

#### 4.2.4. OBDII Bluetooth Vgate iCar 2

En este apartado se habla sobre la configuración del módulo Bluetooth del OBDII de algún automóvil (en este caso un Jetta, Volkswagen, 2012).

En la figura 4.6 muestra un diagrama a bloques que presenta los pasos a seguir para el envío de datos.



Figura 4. 6. Pasos para configurar módulo OBDII Vgate iCar 2.

En el caso del módulo OBDII, se necesita un medio para enviar datos, es por esto que se decidió utilizar un OBDII con Bluetooth integrado. Entonces, se utiliza la plataforma Arduino para codificar los datos que se envían del Bluetooth del OBDII (Modo esclavo) hacia el Bluetooth del Arduino (HC-05, Modo maestro).



Después de que se establece la comunicación correctamente entre el módulo OBDII del automóvil y el Arduino, se realizarán los siguientes pasos para configurar el OBDII utilizando los comandos que se muestran en la tabla 4.1:

- 1.- Reiniciar el módulo.
- 2.- Establecer las respuestas completas del OBDII.
- 3.- Agregar espacios entre las respuestas.
- 4.- Permite datos de más de 7 bytes.
- 5.- Establecer como modo automático la selección de protocolo.
- 6.- Apagar la repetición de comandos.

Tabla 4. 1. Comandos para configurar OBDII.

Comandos para configurar OBDII	Datos
ATZ	Reiniciar el módulo.
ATH1	Visualizar encabezados.
ATS1	Habilitar espacios en respuestas.
ATAL	Permitir datos de más de 7 bytes.
ATSP0	Modo automático la selección de protocolo.
ATE0	Deshabilitar la repetición de comandos.

Una vez inicializado el módulo estará preparado para obtener datos del automóvil usando los comandos correspondientes, dependiendo del parámetro que se quiere obtener.

En la tabla 4.2 se muestran los comandos enviados al OBDII y los datos de entrega correspondientes.



Tabla 4. 2. Comandos para los datos del OBDII.

Comandos OBDII	Datos
0105	Temperatura.
010D	Velocidad del vehículo.
010C	Revoluciones por minuto.

En la Figura 4.6 se muestran las respuestas de la configuración del OBDII y los datos que se obtienen del mismo.

```

Initializando.....
OBD2 inicializado.....
SEARCHING...
Temperatura: 160 C
Real : 41 00 BE 3E B8 11 >41 05 6A >
Velocidad: 10 km/h
Real : 41 0D 0A >
RPM: 1278 rpm
Real : 41 0C 13 F8 >

Temperatura: 77 C
Real : 41 05 6B >
Velocidad: 18 km/h
Real : 41 0D 12 >
RPM: 1752 rpm
Real : 41 0C 1B 63 >

Temperatura: 79 C
Real : 41 05 6D >
Velocidad: 21 km/h
Real : 41 0D 15 >
RPM: 964 rpm
Real : 41 0C 0F 11 >

Temperatura: 79 C
Real : 41 05 6D >
Velocidad: 10 km/h
Real : 41 0D 0A >
RPM: 1215 rpm
Real : 41 0C 12 FE >

```

Figura 4. 7. Resultados OBDII Vgate iCar 2.

#### 4.2.5. OBDII freematics.

Este tipo de OBDII que utiliza cableado para conectarse al automóvil utiliza los mismos comandos e instrucciones que aparecen en la sección de OBDII Bluetooth (tabla 4.1). La diferencia entre uno y otro es simplemente que este OBDII solo ocupa la configuración inicial para después poder enviar los comandos para recibir los datos, que serían los que se encuentran en la tabla 4.2



#### 4.2.6. GPS NEO 6M

Una vez alimentado el GPS y estando conectado al Arduino, el módulo no espera y comienza a enviar datos pero estos datos tienen 2 detalles, el primero es que el GPS no se encuentra al principio conectado a los satélites y lo que envía son datos que no tienen relevancia y prácticamente se reciben nada más las etiquetas de lo que serán los datos importantes del GPS, como se observa en la figura 4.8.

```
$GPGLL,,,,,142501.00,V,N*49
$GPRMC,142502.00,V,,,,,,280417,,,N*75
$GPVTG,,,,,,N*30
$GPGGA,142502.00,,,,,0,03,4.77,,,,,*51
$GPGSA,A,1,14,26,31,,,,,,4.88,4.77,1.00*02
$GPGSV,3,1,12,01,,,27,03,,,34,04,45,089,33,11,,,20*4E
$GPGSV,3,2,12,13,,,23,14,25,073,26,22,,,29,23,,,33*40
$GPGSV,3,3,12,26,49,103,33,31,33,039,33,46,52,155,,48,48,220,40*79
```

Figura 4. 8. Datos recibidos del GPS sin estar conectado al Satélite.

Después de unos momentos y cuando el módulo se encuentre en un espacio no tan alejado del exterior, se realiza la conexión a los satélites logrando de esta manera recibir los datos necesarios para conocer la posición del GPS, como se puede observar en la figura 4.9.

```
$GPGLL,2907.25992,N,11059.94306,W,143426.00,A,D*75
$GPRMC,143427.00,A,2907.26005,N,11059.94279,W,0.292,,280417,,,D*61
$GPVTG,,T,,M,0.292,N,0.542,K,D*2C
$GPGGA,143427.00,2907.26005,N,11059.94279,W,2,05,2.09,217.2,M,-31.6,M,,0000*6B
$GPGSA,A,3,26,14,31,01,22,,,,,,2.91,2.09,2.02*01
$GPGSV,4,1,15,01,27,230,20,02,,,18,03,,,19,04,45,083,25*76
$GPGSV,4,2,15,11,08,216,,14,23,077,31,16,43,142,32,22,82,214,33*7B
$GPGSV,4,3,15,23,,,31,26,50,096,28,31,29,040,36,32,06,092,16*4E
$GPGSV,4,4,15,46,52,155,,48,48,220,41,51,56,172,*4F
```

Figura 4. 9. Datos del GPS conectado al satélite.

Obtenidos los datos completos del GPS y tomando como referencia la tabla 3.1 se pueden obtener los parámetros para conocer la posición, en este caso la latitud y la longitud.





Utilizando la librería TinyGPS de Arduino correspondiente al módulo de GPS, solo es necesario conocer los comandos correctos para obtener dichos parámetros:

- Comando para obtener la latitud:

```
flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6
```

- Comando para obtener la longitud:

```
flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6
```

Con la ayuda de estos dos comandos se pueden obtener los datos necesarios para conocer la posición del módulo, en la figura 4.10 se muestra el formato de la información obtenida del GPS de una manera más sencilla.

```
GPS=LAT=29.120996, LON=-110.999053  
  
GPS=LAT=29.120994, LON=-110.999053  
  
GPS=LAT=29.120992, LON=-110.999053  
  
GPS=LAT=29.120992, LON=-110.999053  
  
GPS=LAT=29.120988, LON=-110.999053
```

Figura 4. 10. Datos codificados del GPS.

Existen otros comandos, por ejemplo como reconocer la conexión del GPS con los satélites, número de datos enviados, etc., pero para esta aplicación es suficiente con la latitud y longitud.

## 4.2.5. GPRS

### 4.2.5.1. SIM800L

El módulo SIM800L presenta problemas al intentar conectarse a Internet de alguna red GSM, aunque para las llamadas y mensajes si se puede conectar, en la cuestión de Internet



este módulo no soporta la transmisión de datos ya que es un módulo de datos 2G y desde 2016 se eliminó la red 2G en algunas compañías y en otras en el año 2017.

#### **4.2.5.2. GPRS A7**

El módulo GPRS A7 es otro módulo que no soporta las frecuencias de comunicación para la transmisión de datos de las redes de telefonía celular de México. El módulo se conecta a llamadas y mensajes pero para la cuestión de datos no se recibe una respuesta positiva. Se presenta el mismo problema del caso anterior con el módulo SIM800L, que nada más soporta red 2G.

#### **4.2.5.3. Shield GPRS Seeedstudio**

El módulo GPRS es un shield diseñado para Arduino UNO y mega, creado por Seeedstudio, el módulo permite hacer llamadas, mensajes y hacer uso del GPRS del celular.

Para poder utilizar los datos de Internet del celular, primero se tiene que hacer una comunicación entre el módulo GPRS y el Arduino que es lograda con la ayuda de la librería "GPRS\_Shield\_Arduino.h". Una vez establecida la comunicación y se configura el módulo el siguiente paso es conectarse a la red del celular usando:

```
gprs.join(F("internet.movistar.mx"))
```

Esta instrucción aparece en la librería para solicitar el APN de la compañía celular que se está utilizando y que puede variar. Después de conectarse ~~estar conectado~~ a la red, en el módulo GPRS se activa un led rojo, indicando que la conexión se ha realizado, como se muestra en la Figura 4.11.





Figura 4. 11. Módulo GPRS Seeedstudio conectado a la red.

Lo siguiente es conectarse a la dirección IP de la página de base de datos que se desea utilizar, en este caso thingspeak, utilizando la siguiente instrucción:

```
gprs.connect(TCP,"184.106.153.149", 80)
```

Se decide primero si es TCP o UDP, que en este caso thingspeak es TCP, después se coloca la dirección IP de la página entre comillas, y por último en que puerto de comunicación se encuentra la página. Cuando el módulo GPRS se conecta a la página de Internet lo siguiente sería enviar una trama que contenga la solicitud necesaria para que thingspeak la pueda identificar y actualizar los datos. Utilizando la siguiente solicitud:

```
GET /update?api_key=HC9KPPSOFCIYRIWU&field2=58\r\n\r\n.
```

Una vez que se hayan realizado todos los pasos, el módulo GPRS habrá enviado los datos correspondientes a las gráficas que se están solicitando, como se muestra en la Figura 4.12.



COM5 (Arduino/Genuino Uno)

```
Inicializacion correcta.  
IP Address is  
100.117.32.153  
Conectado a Thingspeak  
Esperando para extraer...  
fetch over...
```

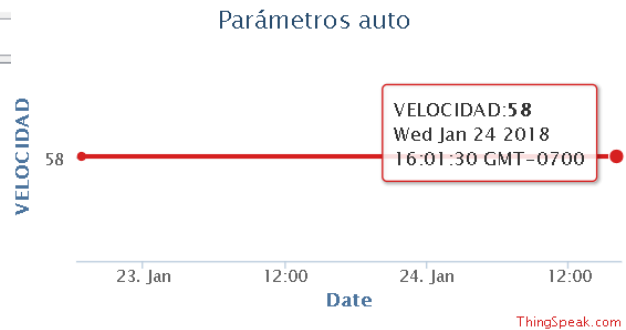


Figura 4. 12. Datos del GPRS y datos de la página de Thingspeak.

#### 4.2.5.4. SIM900

El módulo SIM900 es un GPRS con el que se logró establecer comunicación con la red celular. El módulo utilizado se muestra en la figura 4.13 que cuenta con una antena para recibir y enviar la información. Los pasos a seguir para establecer comunicación y poder enviar datos son los siguientes:

1. Inicializar el módulo a 2400.
2. Realizar la conexión con la red celular.
3. Conectarse con la dirección IP de la página en Internet.
4. Enviar trama con la solicitud que contenga los datos.



Figura 4. 13. Módulo SIM900.



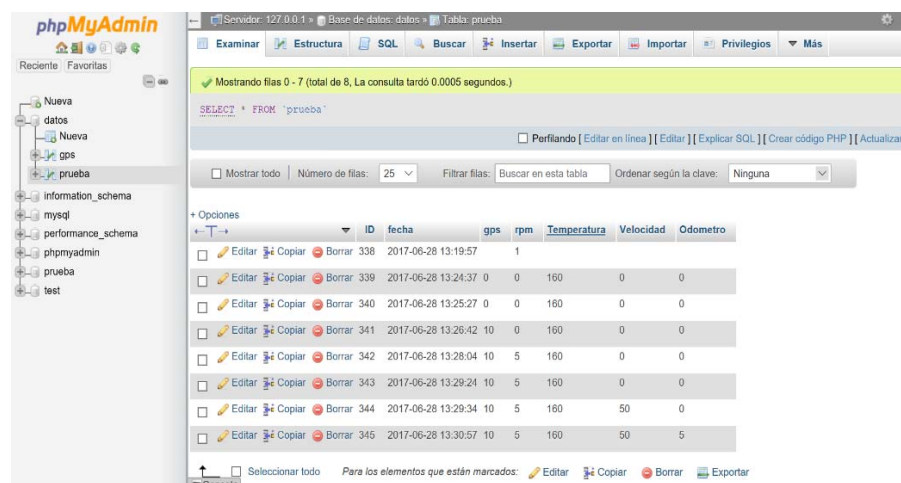
## 4.2.6. Servidor de base de datos

### 4.2.6.1. Openshift

Openshift es una plataforma de Internet en la nube que permite hacer uso de varios programas de computadora como lo son php, ruby, MySQL, etc. Que sirven para programar en el área de la nube, crear tu propio espacio en Internet y enviar datos. En el proyecto se utilizó openshift como primera opción para realizar la recepción de datos que se envían desde el OBDII.

Primeramente se necesitan enviar librerías e información de las distintas formas de trabajar, como lo son: php, ruby, MySQL, etc. Con ayuda del sistema de gestión de bases de datos MySQL se pudo realizar una tabla con los siguientes datos: GPS, RPM, Temperatura, Velocidad y Odómetro. A partir de que se reciben los datos se van organizando en su respectiva etiqueta de la tabla.

La plataforma phpMyAdmin proporcionada por MySQL, como se muestra en la figura 4.14, para base de datos en donde primero se envían los datos para después ser organizados de la forma que más le parezca al interventor.



ID	fecha	gps	rpm	Temperatura	Velocidad	Odometro
338	2017-06-28 13:19:57		1			
339	2017-06-28 13:24:37	0	0	160	0	0
340	2017-06-28 13:25:27	0	0	180	0	0
341	2017-06-28 13:26:42	10	0	160	0	0
342	2017-06-28 13:28:04	10	5	160	0	0
343	2017-06-28 13:29:24	10	5	160	0	0
344	2017-06-28 13:29:34	10	5	160	50	0
345	2017-06-28 13:30:57	10	5	160	50	5

Figura 4. 14. PhpMyAdmin plataforma de base de datos.



Por último, en la figura 4.15 se puede observar una tabla donde se organizan los datos enviados a phpMyAdmin por medio de programación en php y subido a la nube:



ID	Fecha	Gps	RPM	Temperatura	Velocidad	Odómetro
338	2017-06-28 13:19:57		1			
339	2017-06-28 13:24:37	0	0	160	0	0
340	2017-06-28 13:25:27	0	0	160	0	0
341	2017-06-28 13:26:42	10	0	160	0	0
342	2017-06-28 13:28:04	10	5	160	0	0
343	2017-06-28 13:29:24	10	5	160	0	0
344	2017-06-28 13:29:34	10	5	160	50	0
345	2017-06-28 13:30:57	10	5	160	50	5

Figura 4. 15. Tabla que muestra la plataforma Openshift.

#### 4.2.6.2. Thingspeak

Thingspeak es una plataforma en Internet de base de datos que ya cuenta con una estructura y un modelo de comunicación entre el usuario que necesite enviar sus datos para almacenarlos y después analizarlos, y la nube. Thingspeak utiliza los datos recibidos mostrándolos en gráficas para que sea más digerible para el usuario como se puede observar en la figura 4.16.





Figura 4. 16. Parámetros del auto.

Para poder enviar datos a la plataforma se necesita hacer una solicitud a la página utilizando el siguiente formato:

GET [https://api.thingspeak.com/update?api\\_key=HC9KPPSOFClYRIWU&field1=0](https://api.thingspeak.com/update?api_key=HC9KPPSOFClYRIWU&field1=0)

En la parte de “api\_key” se utiliza una contraseña que es brindada por thingspeak para poder entrar a ese espacio en la nube.

### 4.3. Selección de módulos.

El único módulo que por defecto se eligió sin hacer una comparación fue el GPS NEO 6M, ya que cuenta con un reducido tamaño y cumple con las características necesarias para el proyecto y no existieron fallas al momento de usarlo. Además de también optar por la plataforma Thingspeak sobre la de Openshift, ya que la plataforma de Openshift se actualizó de manera inesperada y cambió totalmente la interfaz y la forma de trabajar, provocando que al momento de intentar subir toda la información nuevamente a la nube no se pudo ya que requería ciertos pasos nuevos para poder realizar la comunicación.

En el caso de los módulos OBDII se decidió utilizar el que contaba con cable ya que este módulo aportaba para realizar la alimentación del circuito, el OBDII puede suministrar hasta 2.1 ampere y el circuito utiliza en total 1.8 Ampere.



### 4.3.1. WiFi

El módulo WiFi Esp8266 es la mejor opción para este proyecto porque según sus características que se comparan en la tabla 4.3 se puede observar que en cuestión tamaño es más reducido, así como también la cuestión de temperatura y corriente máxima es mayor y menos respectivamente que el del módulo WiFi Inventek ISM que parecer la mejor opción por cuestión de tamaño, 433362 M3G.

Tabla 4. 3. Comparación de módulos WiFi.

	Módulos WiFi	
	Esp8266	Inventek ISM433362 M3G
Protocolo	IPv4, TCP/HTTP	TCP, UDP, IPv4, ARP, ICMP, DHCP client
Modos de operación	STA/AP/STA+AP	STA/AP/STA+AP
Comunicación	SDIO 2.0, (H) SPI, UART, I2C, I2S, PWM, GPIO	SPI, UART, USB (HID y VCP)
Voltaje/ Corriente max.	3.6 V / 170 mA	3.3 V $\pm$ 5% / 300 mA
Modo Sleep	<10 $\mu$ A	No aplica
Rango de temperatura en operación	-40C ~ 125C	-35°C a +80°C
Dimensiones	14.3mm x 24.8mm	14.5mm x 30mm
Funciona	Si	Si

### 4.3.2. GPRS.

El módulo GPRS SIM900 es la mejor opción en base a la tabla 4.4 en ella se muestra que existen dos módulos GPRS que no funcionan por el tipo de frecuencia a la que trabajan y que para este proyecto es vital. Los otros dos que si funcionan y envían datos a Internet, el módulo Shield GPRS Seeedstudio es demasiado grande y ocupa mucho espacio, en cambio el módulo SIM900 tiene un tamaño, un consumo y temperaturas aceptables y es





por esta razón que se opta por dicho módulo. En la tabla 4.2 se puede observar una comparación entre los módulos GPRS.

Tabla 4. 4. Comparación de módulos GPRS.

	Módulos GPRS			
	SIM800L	GPRS A6	Shield GPRS Seedstudio	SIM900
Frecuencias	850/900/1800 / 1900MHz	850/900/1800 / 1900MHz	850/900/1800 / 1900MHz	850/900/1800 / 1900MHz
Protocolos	TCP / IP, HTTP	TCP / IP, HTTP	TCP / IP, HTTP	TCP / IP, HTTP
Tarjeta SIM para conexión GPRS	2G	2G	3G	3G
Dimensiones	25x23x1 mm.	22.8 x 16.8 x 2.5 mm.	68.58 x 53.34mm.	49 x 35 mm
Voltaje / corriente max.	4.2V / 500 mA	4.2 V / 800 mA	12v / 1 A	5v / 1A
Temperatura	No	-30°C ~ 80°C.	-40°C to +85 °C.	-40° ~ 85°
Comunicación	UART	UART	UART	UART
Funciona	No	No	Si	Si



# Capítulo 5

## Conclusiones

### 5.1. Conclusiones generales.

Se logró obtener un prototipo capaz de realizar la lectura de datos del automóvil y por medio de redes inalámbricas enviarlos hacia una plataforma en la nube. El prototipo realizado al momento cuenta con muy buenas mejoras al anterior. Se logró obtener una comunicación entre el dispositivo y la nube que permite guardar los datos con conexión a Internet, pudiendo de esta manera realizar observaciones y analizar la información desde cualquier dispositivo que cuente con conexión a la red de Internet. Con los dispositivos que se cuentan de momento se puede reducir las dimensiones del prototipo final y llegar a menos del 50% del tamaño anterior.

### 5.2 Trabajos Futuros.

El trabajar con una plataforma de Arduino puede ayudar bastante en el avance que se le puede realizar a un proyecto, es por esto que uno de los cambios que se pueden realizar es mejorar el punto débil de utilizar Arduino y cambiar el Arduino convencional que ya viene soldado en la placa y trabajar nada más con el microcontrolador Atmega328.

Mejorar la plataforma en la nube que pueda reconocer cuando haya un problema inesperado y poder realizar alguna acción con respecto a lo que se presente que puede variar desde tener alarmas o cambiar los estados de cómo se presenten los datos.

Verificar si es realmente necesario utilizar el módulo Wi-Fi y quedarse con el módulo GPRS ya que dicho módulo puede realizar el mismo trabajo y puede ser que este de más el uso de WiFi porque los datos enviados al servidor no representar mucho gasto en el consumo de datos de Internet.



## Anexos 1. Bandas de frecuencia en México

	Banda / Frecuencia 3G	Banda / Frecuencia 4G
<b>Telcel</b>	<b>B2/1900MHz</b> <b>B5/850MHz</b>	<b>B4/ 1700/2100MHz</b>
<b>AT&amp;T</b>	<b>B2/1900 MHz</b> <b>B5/850 MHz</b> <b>B4/1700/2100MHz</b>	<b>B4/1700/2100MHz</b>
<b>Movistar</b>	<b>B2/1900MHz</b> <b>B5/850MHz</b>	<b>B2/1900MHz</b>



## Anexos 2. Parámetros del módulo WiFi ESP8266

Categorías	Objetos	Valores
<b>Parámetros WIFI</b>	Certificados	FCC/CE/TELEC/SRRC
	Protocolos Wifi	802.11 b/g/n
	Rango de frecuencia	2.4-2.5 (2400M-248305M)
	Tx	802.11 b: +20dBm
		802.11 g: +17dBm
		802.11 n: +14dBm
	Rx sensibilidad	802.11 b: -91dBm (11Mbps)
		802.11 g: -75dBm (54Mbps)
802.11 n: -72 dBm (MCS7)		
Tipos de antena	PCB rastreo, conector IPEX, chip cerámico	
<b>Parámetros Hardware</b>	Bus periférico	Uart/SDIO/SPI/I2C/I2S/IR control remoto GPIO/PWM
	Voltaje de operación	3.0~3.6V
	Corriente de operación	Valor promedio: 80mA
	Rango de temperatura	-40°~125°
	Rango de temperatura ambiente	Temperatura normal
	Tamaño del paquete	5x5 mm
	Interface externa	N/A
<b>Parámetros de software</b>	Modo Wifi	Station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encriptación	WEP/TKIP/AES
	Actualización de firmware	UART descarga/OTA (vía Internet)
	Desarrollo de software	Desarrollo de servidores en la nube/SDK para desarrollo de firmware
	Protocolos network	IPv4, TCP/UDP/HTTP/FTP
Configuración del usuario	Comandos AT, servidor en nube, Android/iOS App	



## Anexos 3. Parámetros del módulo WiFi Inventek

<b>Estándar de la red</b>	IEEE 802.11b/g/n
<b>Frecuencia RF</b>	2.4 GHz ~ 2.497 GHz
<b>Tasa de transferencia de datos</b>	802.11n: 6.5, 13, 19.5, 26, 39, 52, 58.5, 65 Mbps
	802.11g: 6, 9, 12, 18, 24, 36, 48, 54 Mbps
	802.11b: 1, 2, 5.5, 11 Mbps
<b>Técnicas de modulación</b>	802.11 g/n: 64-QAM, 16-QAM, QPSK, BPSK
	802.11 b: CCK, DQPSK, DBPSK
<b>Interfaces de host</b>	SPI, UART, USB (HID y VCP)
<b>Protocolos de red</b>	TCP, UDP, IPv4, ARP, ICMP, DHCP client
<b>Funciones WLAN</b>	Modo ahorro de energía, roaming automatico, auto-ratio
<b>Configuración</b>	Comandos AT, marco SPI
<b>Voltaje de alimentación</b>	3.3 ± 5%
<b>Paquete</b>	44-pin LGA, 14.5mm x 30mm
<b>Temperatura de operación</b>	-35°C a +80°C
<b>Certificación</b>	FCC y CE



## Anexos 4. Parámetros del módulo SIM800L

Características	Implementación
Alimentación	3.4 ~ 4.3 V
Ahorro de energía	En modo sleep 0.7 ma
Bandas de frecuencia	Quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900.
Poder de transmisión	Clase 4 (2W) GSM 850 y EGSM 900
	Clase 1 (1W) a DCS 1800 y PCS 1900
Conectividad GPRS	GPRS multi-slot clase 12 (default)
	GPRS multi-slot clase 1-12 (opción)
Rango de temperatura	-40°C ~ 85°C
GPRS Datos	Transferencia de datos de bajada GPRS: 85.6 kbps
	Transferencia de datos de subida GPRS: 85.6 kbps
	Protocolo PAP para conexión PPP
	Protocolo TCP/IP integrado
	Rangos de transmisión CSD: 2.4, 4.8, 9.6, 14.4 kbps
SMS	MT, MO, CB, Text y modo PDU
	Almacenamiento SMS: Sim card
Interfaz SIM	Soporta SIM card: 1.8 V, 3 V
Antena externa	Antena de almohadilla
Puerto serie	Interfaz de módem con líneas de estado y control, no balanceado, asíncrono.
	1200 bps a 115200 bps
	Puede ser usado para comandos AT o datos stream
	Soporte RTS/CTS
	Multiplexado acorde a GSM 07.10 protocolo multiplexor
	Actualización de firmware
Funciones	Comandos AT
Características físicas	Tamaño: 15.8 * 17.8 * 2.4 mm
	Peso: 1.35 g



## Anexos 5. Parámetros del módulo GSM/GPRS

<b>Compatible</b>	Arduino UNO / MEGA Seeduino directamente; Otra placa principal a través de puentes
<b>Interfaz seleccionable</b>	UART, serie de software
<b>Soporta banda cuádruple</b>	850/900/1800 / 1900MHz
<b>Soporte de comunicación</b>	Estándar - GSM 07.07 y 07.05 y mejorado - Comandos SIMCOM AT
<b>Temperatura de operación</b>	-40 ° C a +85 ° C
<b>Soporte de protocolo</b>	0710 protocolo MUX, protocolo TCP / UDP integrado, FTP / HTTP, FOTA, MMS, incrustado AT
<b>Certificación de SIM900</b>	CE, IC, FCC, ROHS, PTCRB, GCF, ICASA, REACH, AT & T
<b>Dimensiones</b>	68.58 * 53.34 mm
<b>Fuente de alimentación</b>	5v a través de 5V pin, 6.5 ~ 12v a través de Vin pin



## Referencias Bibliográficas

- [1] T. Flotillas, “Flotillas.” [Online]. Available: <http://www.topflotillas.com/info-general/>. [Accessed: 21-Dec-2015].
- [2] A. Flotillera, “Definicion de la flotilla de distribucion.” [Online]. Available: <http://www.alianzaflotillera.com/noticias/definicion-de-la-flotilla-de-distribucion/>. [Accessed: 21-Dec-2015].
- [3] M. D. López Prado, “Administración.” Tesis, 2006. [Online]. Available: <http://www.uteq.edu.mx/tesis/administracion/0100000060.pdf>. [Accessed: 22-Dec-2015].
- [4] Asif Faiz, C. S. Weaver, and M. P. Walsh, *Air pollution from motor vehicles: standards and technologies for controlling emissions*. World Bank Publications, 2009.
- [5] e-auto, “Manuales, ¿Qué es OBDII?” [Online]. Available: [http://e-auto.com.mx/manual\\_detalle.php?manual\\_id=119](http://e-auto.com.mx/manual_detalle.php?manual_id=119). [Accessed: 10-Dec-2015].
- [6] J. A. C. Falcón, *Wi-Fi lo que se necesita conocer*. Alfaomega, 2016.
- [7] T. Target, “GSM (Global System for Mobile communication).” [Online]. Available: <http://searchmobilecomputing.techtarget.com/definition/GSM>. [Accessed: 06-Jul-2016].
- [8] X. He, J. Montillet, R. Fernandes, M. Bos, and K. Yu, “Review of current GPS methodologies for producing accurate time series and their error sources,” *J. Geodyn.*, vol. 106, pp. 12–29, 2017.
- [9] INEGI, “Sistema de Posicionamiento Global (GPS).” [Online]. Available: <http://www.inegi.org.mx/geo/contenidos/geodesia/gps.aspx?dv=c1>. [Accessed: 01-Jun-2016].
- [10] M. U. Zhou, Y. Tang, Z. Tian, and X. Geng, “Semi-Supervised Learning for Indoor Hybrid Fingerprint Database Calibration With Low Effort,” vol. 5, 2017.
- [11] L. T. Ojeda, “¿Qué es arduino?,” *Articulo*, 2016. [Online]. Available: <http://arduino.cl/que-es-arduino/>.
- [12] N. Wireless, “Soluciones.” [Online]. Available: <https://www.navmanwireless.com.mx/soluciones-administracion-flotillas-gps/navman-wireless-director/funciones/>. [Accessed: 29-Dec-2015].
- [13] Origis, “Administración de Flotillas.” [Online]. Available: [http://www.origis.com/index.php?option=com\\_k2&view=itemlist&layout=category&task=category&id=6&Itemid=2041](http://www.origis.com/index.php?option=com_k2&view=itemlist&layout=category&task=category&id=6&Itemid=2041). [Accessed: 29-Dec-2015].
- [14] Cnnexpansion, “CÓMO AHORRAR EN EL MANEJO DE FLOTILLAS.” [Online]. Available: <http://www.cnnexpansion.com/negocios/2009/05/29/ahorra-gastos-en-el-manejo-de-flotillas>. [Accessed: 29-Dec-2015].
- [15] E. Systems and I. O. T. Team, “ESP8266EX Datasheet,” vol. Version 4., pp. 8–14, 2015.
- [16] E. Team, “Espressif,” 2017. [Online]. Available:





- <http://espressif.com/en/company/about-us/who-we-are>. [Accessed: 10-Jun-2017].
- [17] L. Guangzhou Información Cheng Tecnología Co., “Wavesen.” [Online]. Available: <http://www.wavesen.com/about.asp>. [Accessed: 12-Sep-2017].
- [18] SIMCom, “*SIM800L Hardware Design*”, *SIM800I Datasheet*. .
- [19] Electrodragon, “GSM GPRS A6 Module.” [Online]. Available: [http://www.electrodragon.com/w/GSM\\_GPRS\\_A6\\_Module](http://www.electrodragon.com/w/GSM_GPRS_A6_Module).
- [20] R. Hat, “Openshift.” [Online]. Available: <https://www.openshift.com/>.
- [21] Ruby, “Thingspeak,” 2010. [Online]. Available: [https://thingspeak.com/pages/learn\\_more](https://thingspeak.com/pages/learn_more). [Accessed: 07-Jul-2017].

