

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD CON
LOCALIZACIÓN GPS PARA AUTOMÓVILES EN UN ENTORNO CONTROLADO
Y MONITOREADO POR UNA APLICACIÓN DE DISPOSITIVO MÓVIL**

**JORGE YESID RÍOS GÓMEZ
LEONARDO MAURICIO SALAZAR NARVÁEZ**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA – HUILA
2014**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD CON
LOCALIZACIÓN GPS PARA AUTOMÓVILES EN UN ENTORNO CONTROLADO
Y MONITOREADO POR UNA APLICACIÓN DE DISPOSITIVO MÓVIL**

**JORGE YESID RÍOS GÓMEZ
LEONARDO MAURICIO SALAZAR NARVÁEZ**

**Proyecto de grado presentado como requisito para optar
al título de INGENIERO ELECTRÓNICO**

**Asesor
JOHAN JULIAN MOLINA
Docente Programa Ingeniería Electrónica**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA – HUILA
2014**

NOTA DE ACEPTACIÓN

Presidente del Jurado

Jurado

Jurado

Neiva, Diciembre del 2014

DEDICATORIA

Gracias al Señor Jesús por guiarme en el camino de la sabiduría;
A mi madre Mariela, por su amor, apoyo y comprensión;
A mis hermanas Jessica y Hellen, por confiar en mí
A mis familiares y amigos;
quienes hicieron posible la realización de este libro.

JORGE YESID RÍOS GÓMEZ

A mis padres Félix y María Elena, por su constante e incondicional apoyo;
A mis familiares y amigos por levantarme el ánimo para seguir adelante;
Y demás personas que de alguna
u otra forma ayudaron para la culminación de este proyecto;

LEONARDO MAURICIO SALAZAR NARVÁEZ

AGRADECIMIENTOS

Los autores expresan su agradecimiento a:

Nuestra Alma Mater la Universidad Surcolombiana, representada en el personal docente y administrativo, que nos prestaron sus servicios para la realización nuestros estudios de pregrado.

Ing. Julián Molina, Director de trabajo de grado, por brindarnos su apoyo, su guía y además de las herramientas que nos facilitaron la elaboración de este trabajo.

Nuestros profesores Ing. Martín Diomedes Bravo, Ing. Jesús David Quintero; por brindarnos sus conocimientos y su valioso tiempo.

Funcionarios del Tecno parque nodo Neiva, como el Ing. Adrián Chavarro y demás personas que nos facilitaron herramientas y conocimientos para finalizar exitosamente el proyecto.

Nuestros amigos por la confianza depositada en nosotros además de su compañía y colaboración durante el proceso de realización del proyecto.

Todas las personas que directa o indirectamente colaboraron en la realización de este proyecto de grado.

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. OBJETIVOS	18
1.1 OBJETIVO GENERAL	18
1.2 OBJETIVOS ESPECÍFICOS	18
2. MARCO TEÓRICO	19
2.1 DIAGRAMA GENERAL DEL SISTEMA	19
2.1.1 Arduino + Módulo 3G/GPS	19
2.1.2 Satélite GPS	19
2.1.3 Sistema de alarmas GENIUS	20
2.1.4 Dispositivo móvil Android	20
2.1.5 Regulador de voltaje	20
2.1.6 Red GSM	21
2.2 INTRODUCCIÓN A ARDUINO	21
2.2.1 ¿Qué es Arduino?	21
2.2.2 Hardware	21
2.2.3 “Shields” y otras placas de terceros	22
2.2.4 Arduino UNO REV 3	22
2.2.5 Características Arduino UNO	23
2.2.6 Software (Entorno de desarrollo)	26

2.3 MODULO 3G + GPS PARA ARDUINO	28
2.3.1 Características y accesorios	29
2.3.2 Hardware	31
2.3.3 Antenas	32
2.3.4 Usando el módulo 3G con comandos AT	32
2.3.5 Audio	34
2.3.6 Cámara	35
2.3.7 GPS	36
2.3.8 SMS	37
2.3.9 Correo electrónico	37
2.3.10 Almacenamiento interno y externo	37
2.4 ANDROID	38
2.4.1 Arquitectura de Android	39
2.4.2 Las versiones de Android y niveles de API	40
2.4.3 Elección de la plataforma de desarrollo	40
2.4.4 Herramientas para el desarrollo	41
2.4.5 Componentes de una aplicación	41
3. DISEÑO	43
3.1 COMPONENTE DE HARDWARE	43
3.1.1 Diseño circuito de adecuación	43
3.2 COMPONENTE DE SOFTWARE	45
3.2.1 Modelamiento de la aplicación	45

4. LIMITACIONES	51
5. CONCLUSIONES	52
6. RECOMENDACIONES	53
BIBLIOGRAFIA	54
ANEXOS	56

LISTA DE FIGURAS

	Pág.
Figura 1. Diagrama general del sistema	19
Figura 2. Descripción gráfica del regulador de voltaje	21
Figura 3. Placa Arduino UNO	22
Figura 4. Descripción de pines de Arduino UNO	23
Figura 5. Figura IDE Arduino	26
Figura 6. Selección de la placa a usar	27
Figura 7. Monitor serial	27
Figura 8. Módulo 3G/GPS para Arduino + kit de audio y video	29
Figura 9. Diagrama del módulo	31
Figura 10. PCB del módulo	31
Figura 11. Conexiones de antenas del Módulo 3G Arduino	32
Figura 12. Soldadura de la antena al Módulo 3G Arduino	32
Figura 13. Configuración básica sin ATmega	33
Figura 14. Conexión del Arduino con el módulo 3G/GPS	33
Figura 15. Módulo 3G/GPS + kit de audio	35
Figura 16. Diagrama de pines de la cámara	35
Figura 17. Ranura para memoria microSD	38
Figura 18. Arquitectura en capas del sistema Android	39
Figura 19. Simulación del regulador de voltaje L7805CV	43
Figura 20. Diagrama de instalación de alarmas GENIUS	44

Figura 21. Diagrama del circuito de adecuación de alarma GENIUS con Arduino 3G/GPS	45
Figura 22. Comunicación Android-Arduino	46
Figura 23. Comunicación Usuario Sistema	47
Figura 24. Diagrama de clases	48
Figura 25. Diagrama de flujo programación Arduino	49

LISTA DE CUADROS

	Pág.
Cuadro 1. Características Arduino UNO	23
Cuadro 2. Comandos AT básicos	34
Cuadro 3. Especificación de los pines de la cámara	36

LISTA DE ANEXOS

	Pág.
Anexo A. Manual Comandos AT	34
Anexo B. Regulador L7805CV	43
Anexo C. Alarma Genius 2A	44
Anexo D. Manual de usuario	48
Anexo E. Programación en Eclipse	48
Anexo F. Programación Arduino	50

GLOSARIO

API: Interfaz de Programación de Aplicaciones (Application Programming Interface). Conjunto de especificaciones e implementación de comunicación entre componentes software. Trata de un conjunto de funciones o clases y sus métodos cuyo fin es ofrecer acceso a los servicios del sistema.

COMANDOS AT: Instrucciones codificadas que permiten la comunicación entre el humano y un terminal módem o teléfono. A su vez, dispone de varias opciones de configuración del terminal, como son; envío y recepción de mensajes SMS, activación de servicios de ubicación, selección de unidad de almacenamiento, etc.

TELEFONÍA MÓVIL 3G: Tercera generación de transmisión de voz y datos a través de la telefonía móvil. También permite servicios de datos no-voz, entre los cuales están; descarga de programas, intercambio de correos electrónicos y mensajería instantánea (SMS).

TX: Abreviatura que representa la transmisión de datos, que es el proceso de envío y propagación de una señal analógica o digital.

RX: Abreviatura que representa el proceso de recepción de datos de una señal analógica o digital.

RESUMEN

Se diseñó e implementó un sistema de seguridad vehicular en una camioneta Tucson iX35 modelo 2012 de marca Hyundai, que viene con un sistema de alarmas de fabricación GENIUS. Para esto se utilizó un módulo 3G/GPS de cooking hacks, el cual ofrece muchas características y funciones bastante útiles a la hora de proteger, vigilar y localizar la camioneta. El sistema de alarmas fue posible adaptarlo fácilmente con el sistema Arduino UNO y el módulo 3G/GPS. Se desarrolló una interfaz para el aplicativo móvil, transparente para el usuario. De esta manera y a través de botones ubicados en el menú principal de la App, fue posible activar la grabación de video, el cual queda almacenado en la tarjeta micro SD. Además esta placa 3G/GPS cuenta con un dispositivo de localización GPS, SIM 5218, que brindó una opción de rastreo del módulo instalado internamente en la camioneta. Una vez el sistema determina su geolocalización, la placa guarda las coordenadas respectivas y las envía en un mensaje de texto SMS con el fin de alertar al propietario en caso de un intento de robo o algún impacto en el chasis de la camioneta.

Finalmente se obtuvo un sistema de seguridad bastante preciso para la localización GPS junto con una base de datos SQLite, implementada internamente en el dispositivo Android, que registra y accede a cada uno de los eventos sucedidos en el interior de la camioneta, asociados a los videos tomados y las coordenadas recibidas. También este aplicativo móvil permite informar al usuario la hora y fecha de los eventos, además cuenta con mensajes emergentes para informar si los mensajes de texto se enviaron y llegaron exitosamente.

ABSTRACT

It was designed and implemented a car security system on a van Tucson iX35, 2012 model and Hyundai brand which has an alarm system made by GENIUS. Then, It was used a 3G/GPS shield manufacturing Cooking Hacks that offers many features and functions useful in protecting, monitoring and locating it. The alarm system could be easily adapted to the Arduino UNO module and 3G/GPS shield. An interface was developed, transparent to the user. In this way and through buttons on the App's main menu, it was possible to activate video recording, which is stored in the micro SD card. This shield also has a GPS tracking device, SIM 5218, that provides an option of tracking module installed internally in the van. Once the system determines its geolocation, the plate keeps the respective coordinates and sends SMS text messages to alert the owner in the event of an attempted robbery or any impact on the body of the car.

Finally it was obtained an accurate GPS location security system with a SQLite database, implemented internally in the Android device that records and accesses each of the events occurring inside the van, associated with videos taken and the coordinates received. Also this mobile application informs the user the time and date of the event and provides a toasts that reporting if the text messages were sent and successfully received.

INTRODUCCIÓN

Desde el surgimiento de los primeros automóviles, fueron apareciendo también los ladrones de éste tipo de vehículos, por tanto los propietarios buscaron medidas alternativas de seguridad para la protección de sus autos y las personas presentes en estos¹.

El departamento del Huila en el año 2012, ocupó la sexta posición a nivel nacional en delitos relacionados con hurtos de autos y motocicletas, con un incremento del 12% respecto al 2011, los delincuentes se apropiaron de 1062 motocicletas y 56 autos, siendo municipios como Neiva y Pitalito donde presentaron el mayor número de casos². Por ello surge esta idea con el fin de enfrentar esta problemática social desde un enfoque más práctico y novedoso, desarrollando nuevas tecnologías que permitan prevenir y disminuir el robo de autos, reduciendo así costos que se adjuntan a la pérdida del auto y los objetos que se encuentren dentro de este. Incluso de dar el paradero del auto, en caso de que se presentara un secuestro de los ocupantes. Por esto es importante desarrollar un sistema, que pueda ofrecer al usuario un seguimiento del automóvil de manera sencilla y eficiente.

En Suramérica, la empresa Sórensen³, inició sus trabajos desde 1979, desarrollando e instalando sus propios sistemas de seguridad, los cuales fueron más sofisticados a medida que avanzaba la electrónica. Logrando ser una empresa líder reconocida en instalación de sistemas de seguridad en Argentina y en toda América. De allí nació la revolucionaria idea de integrar el celular con un sistema de seguridad de un automóvil. Éste proyecto se denominó COMM-CAR, el cual permite estar al tanto de cualquier anomalía en el vehículo por medio de llamadas o mensajes SMS, además no solo informa del estado del vehículo si no que permite controlarlo remotamente a través de un avanzado menú de opciones que se puede configurar a las necesidades del conductor. COMM-CAR cuenta con funcionalidades como aviso de emergencia por llamada a un número establecido por el usuario, escuchar lo que pasa dentro del auto desde un celular convencional, detener el motor, bloquear puertas y ventanas del vehículo a cualquier distancia disparando el sistema de seguridad y además da una ubicación aproximada en caso de que éste sea robado.

De igual manera, este proyecto hace referencia a un sistema de seguridad para automóviles pero con la diferencia que está basado en las plataformas Android y

¹ <http://ortizayala.wordpress.com/2011/02/26/hello-world/> (último acceso: 16 de Enero de 2014)

² Referenciado por: <http://www.lanacion.com.co/index.php/noticias-regional/huila/item/204438-robo-de-vehiculos-entre-la-inseguridad-y-la-falta-de-conciencia> (último acceso: 26 de Noviembre de 2014)

³ Referenciado por: <http://www.sorensen.com.ar> (último acceso: 26 de Noviembre de 2014)

Arduino. Mediante el entorno de desarrollo Eclipse⁴, se diseña un aplicativo móvil interactivo para celulares con sistema operativo Android, dotado de una interfaz gráfica de fácil acceso y uso, que permite notificar al usuario de algún impacto o intento de robo de su auto a través de un mensaje de texto SMS. La aplicación Android permite realizar la grabación de video por medio de un SMS pre configurado, pulsando un botón, el video tomado se almacena en una tarjeta micro SD. Esto permite tener registros audiovisuales ante una emergencia si el usuario así lo requiera.

A esta aplicación móvil se le diseña una parte del seguimiento de geolocalización en tiempo real, es decir, cuando el usuario necesite la ubicación o paradero de su vehículo, el módulo le envía mensaje al dispositivo móvil con las coordenadas de longitud y latitud, visualizadas mediante la API de Google Maps. El dueño del vehículo puede localizarlo en cualquier momento con la ayuda del GPS incorporado en la tarjeta Arduino leyendo las coordenadas de éste en el dispositivo móvil que muestra la posición de la placa Arduino que viene oculta en el automóvil.

⁴ MURPHY MARK L. Beginning Android 2. Apress. 2010.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Diseñar e implementar un sistema de seguridad con localización GPS para automóviles en un entorno controlado y monitoreado por una aplicación de dispositivo móvil que permita prevenir, proteger y agilizar durante un accidente o robo, la detección y localización de este mediante mensajes de texto enviados al celular pre configurado.

1.2 OBJETIVOS ESPECÍFICOS

- Diseñar una aplicación móvil llamativa, intuitiva, útil y de fácil uso para el control y monitoreo del sistema de seguridad del automóvil desde un móvil con sistema operativo Android.
- Implementar el hardware necesario para el control y adecuación del sistema de seguridad y sus sensores en el automóvil.
- Prevenir, detectar e informar sobre algún robo o accidente de tránsito mediante el sistema GPS integrado con la aplicación diseñada para el automóvil.
- Brindar comunicación bidireccional del sistema Arduino a la aplicación Android, sin que la distancia sea un obstáculo a través de un módulo de conectividad 3G/GPRS.
- Realizar pruebas de campo para verificar el correcto funcionamiento del GPS y del sistema de seguridad.

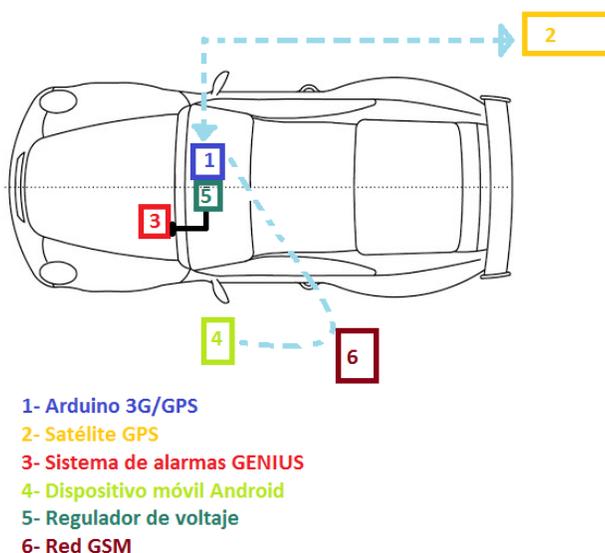
2. MARCO TEÓRICO

2.1 DIAGRAMA GENERAL DEL SISTEMA

El sistema de seguridad en el automóvil permite protegerlo y alertar al propietario cuando algún suceso accione la alarma de la camioneta, además informa sobre la ubicación del vehículo y así mismo graba un corto video para tener un registro audiovisual de lo sucedido. Este sistema está compuesto por diferentes partes esenciales como son; el sistema de alarmas GENIUS de la camioneta, la placa Arduino UNO conectada con el módulo 3G/GPS, el regulador de voltaje, el satélite GPS, el dispositivo móvil Android y la red GSM, que facilita la labor de ubicación.

La Figura 1 presenta el diagrama general del sistema de seguridad en el automóvil y todos sus componentes.

Figura 1. Diagrama general del sistema



2.1.1. Arduino + Módulo 3G/GPS. El Módulo 3G/GPS adaptado a una placa Arduino UNO, la cual es el núcleo de este sistema, es el que sirve de interfaz entre el dispositivo móvil y el sistema de seguridad del vehículo. Viene dotado de antenas 3G y GPS para el envío de datos de manera bidireccional con las estaciones base o los satélites.

2.1.2. Satélite GPS. El Sistema de Posicionamiento Global no es más que una red de satélites que triangulan la posición del usuario por medio de 4 o más satélites, dependiendo de los tiempos de propagación de las señales de satélite.

La precisión de esta geolocalización es en metros e incluso puede llegar a ser de centímetros. En el sistema de seguridad del automóvil, es el módulo GPS, integrado a la placa Arduino, quién se encarga de servir como punto receptor de las señales de satélite⁵.

2.1.3. Sistema de alarmas GENIUS. Los sensores en el auto alertan al sistema principal de alarmas de alguna anomalía dentro o fuera de este. El sensor de impacto detecta cambios de presión en la carrocería y convierte esos cambios en impulsos eléctricos. Cuando exista un impacto o un intruso intente ingresar al vehículo, estos sensores envían al módulo principal GENIUS sus señales eléctricas que se procesan por el microcontrolador y se conectan con el Arduino, cuya función es enviar un SMS para informar al propietario del vehículo, este suceso. El propietario, por medio de una aplicación Android, podrá registrar audiovisualmente y/o rastrear por GPS su vehículo desde el móvil⁶.

2.1.4. Dispositivo móvil Android. Es un teléfono inteligente con sistema operativo Android, de software libre y que permite la conectividad 3G. Son equipos de alta velocidad de procesamiento y almacenamiento para un sin número de aplicaciones. Este celular sirve como interfaz entre el usuario y el sistema de seguridad del automóvil. Específicamente, la aplicación es una interfaz, la cual se diseñó para tener un control remoto distinto al convencional. Este sistema integra el dispositivo móvil con las alarmas del vehículo y además hace posible que el usuario pueda grabar video e incluso saber el paradero de su vehículo.

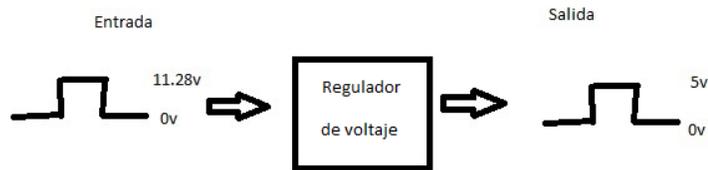
2.1.5. Regulador de voltaje. Es un encapsulado con referencia L7805CV, el cual es un regulador de voltaje positivo, con el objetivo de ajustar los niveles de tensión de la señal de la sirena, de acuerdo con las entradas del Arduino Uno Rev3. Este tiene un voltaje de salida fijado a 5V. Mientras en su entrada, cuando se activa el sistema de alarmas, recibe 12V. A continuación en la Figura 2, se muestra una ilustración del funcionamiento del regulador⁷.

⁵ RIZOS, Chris. University of New South Wales. GPS Satellite Signals. 1999

⁶ Referenciado por: <http://alarmasgenius.com/productos.asp?idcategoria=13&idcategorias=2> (último acceso: 17 de septiembre de 2014)

⁷ Referenciado por: <http://pdf.datasheetcatalog.com/datasheet/stmicroelectronics/2143.pdf> (último acceso: 31 de Octubre de 2014)

Figura 2. Descripción gráfica del regulador de voltaje



2.1.6 Red GSM. La red GSM (Global System for Mobile communications) es una arquitectura basada en conmutación de circuitos, en donde están implicadas las estaciones base, estaciones de control, repetidoras y móviles, en donde existen una comunicación donde el sistema terminal se desplaza y requiere el servicio durante su desplazamiento. Dentro de esta red está incluido el servicio de mensajes cortos denominado SMS (Short Message Service), el cual permite el envío y recepción de mensajes en modo texto de hasta 160 caracteres. Es mediante este servicio de comunicación en el que se basa el sistema de seguridad para comunicar el dispositivo móvil Android con el sistema Arduino 3G/GPS⁸

2.2 INTRODUCCIÓN A ARDUINO

La mayor parte de esta sección es una recopilación de la información de Arduino, describiendo los principales elementos que componen la placa Arduino UNO y el entorno de desarrollo para programarla, es decir la parte hardware y software que actúan sobre Arduino⁹.

2.2.1 ¿Qué es Arduino? Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

2.2.2 Hardware. Al ser Arduino una plataforma de hardware libre tanto su diseño como su distribución puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido alguna licencia. Por eso existen varios tipos de placa oficiales, las creadas por la comunidad Arduino o las no oficiales creadas por terceros pero con características similares.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Se puede usar Arduino para crear objetos interactivos, leyendo

⁸ MOULY M., PAUTET M.-B., The GSM System for mobile Communications, Cell & Sys, France 1992

⁹ <http://arduino.cc/en/Main/ArduinoBoardUno>

datos de una gran variedad de interruptores y sensores a través de sus pines de entrada controlando multitud de tipos de luces, motores y otros actuadores físicos. Los proyectos de Arduino pueden ejecutarse sin necesidad de conectarse a un ordenador o comunicarse con un programa (software) que se ejecute en el ordenador.

2.2.3 "Shields" y otras placas de terceros. Las "Shields" son placas que pueden ser conectadas encima de la placa Arduino y que amplían una nueva función para que sea controlada desde Arduino, para controlar diferentes aparatos, adquirir datos, etc. Las diferentes "shields" siguen la misma filosofía que el conjunto original: son fáciles de montar, y baratas de producir.

Un Shield o escudo es una placa que permite expandir funcionalidades al Arduino, con lo cual se puede conectar motores, o a la red celular, a una red WIFI, a una red Ethernet o tener un MP3 en el Arduino, entre muchos más.

2.2.4 Arduino REV 3. Arduino UNO Rev3 es la última versión disponible de la tarjeta Arduino original. Es una tarjeta electrónica basada en el microcontrolador Atmega328. Dispone de 14 entradas/salidas digitales, 6 de las cuales se pueden emplear como salidas PWM. Dispone también de 6 entradas analógicas, un oscilador de 16 MHz, una conexión USB, un conector de alimentación, un conector ICSP y un pulsador para el reset.

Para empezar a utilizar la placa sólo es necesario conectarla al ordenador a través de un cable USB, o bien alimentarla con un adaptador de corriente AC/DC. También se puede alimentar con una batería. Una de las diferencias más importantes de la tarjeta Arduino UNO respecto a sus predecesoras, es que no utiliza el convertidor USB-serie de la firma FTDI. Por lo contrario, integra un microcontrolador ATmega 8U2 programado como un convertidor o puente de USB a serie.

Figura 3. Placa Arduino UNO



Fuente: <http://arduino.cc/en/Main/ArduinoBoardUno>

2.2.5 Características Arduino UNO. La descripción completa del Arduino UNO se puede encontrar en la página web oficial¹⁰. Lo siguiente es un resumen de las principales características:

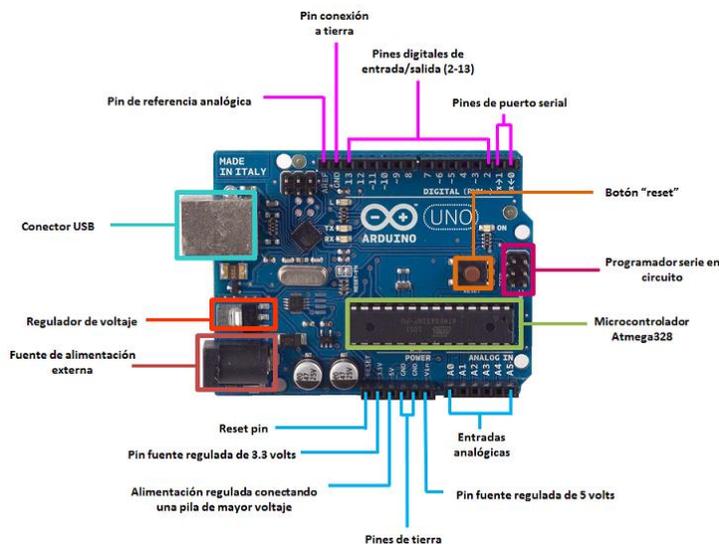
Cuadro 1. Características Arduino UNO

Microcontrolador	Atmega328
Voltaje de funcionamiento	5 V
Voltaje de entrada (recomendada)	7 - 12 V
Voltaje máximo de entrada (límite)	20 V
Pines digitales I/O	14 (de los cuales 6 dan salida PWM)
Pines de entrada analógica	6
Corriente DC por I/O pin	40 mA
Corriente DC para el pin 3.3 V	50 mA
Memoria Flash	32 KB, 0.5 KB son usados por el gestor de arranque
SRAM	2 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Fuente: <http://arduino.cc/en/Main/ArduinoBoardUno>

A continuación se muestra en la figura 4 dónde están ubicados los elementos más importantes que componen la placa Arduino UNO:

Figura 4. Descripción de pines de Arduino UNO



Fuente: <http://arduino.cc/en/Main/ArduinoBoardUno>

Además, algunos de los pines tienen funciones especializadas, entre las cuales tenemos:

¹⁰ <http://arduino.cc> (último acceso: 29 de Febrero de 2014)

Referencia para pines analógicos (AREF): Este pin proporciona un voltaje de referencia para las entradas analógicas.

Pines de tierra (GND): Masa del circuito para pines, es la tensión de referencia de 0 V.

Pines digitales de entrada y salida. En estos pines se conectan las entradas de los sensores. Desde ellos se puede leer la información del sensor o activar la alarma. Hay 14 pines digitales que pueden utilizarse como entrada o salida con las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Operan a 5V. Cada pin proporciona o recibe como máximo 40mA y disponen de una resistencia pull-up (desconectada por defecto) de 20-50 kΩ. Ciertos pines son reservados para determinados usos:

Serie: Pin 0 (RX) y 1 (TX). Usados para recibir (RX) y transmitir (TX) datos a través del puerto serie TTL. Estos pines están conectados a los pines correspondientes del chip de FTDI responsable de la conversión USB-to-TTL.

Interrupciones Externas: Pin 2 y 3. Estos pines se pueden configurar para que interrumpan la ejecución del programa al detectar un flanco o un nivel.

PWM: Pin 3, 5, 6, 9, 10, 11. Proporciona una salida PWM (Modulación por ancho de pulsos) con temporizadores de 8 bits de resolución.

I2C: Pin 4 (SDA) y 5 (SCL). Soporte del protocolo de comunicaciones I2C/TWI. I2C (Inter-Integrated Circuit) o también conocido como TWI (Two wire) o “dos cables” en español, es un protocolo que se utiliza para comunicar circuitos integrados entre sí. Usa dos líneas para transmitir la información: una línea llamada “SDA” para transferir los datos y la otra “SCL” para enviar la señal de reloj¹¹.

SPI: Pines 10-13. Estos pines soportan la librería de comunicación de dispositivos SPI (Serial Peripheral Interface)

LED: Pin 13. Este pin está conectado con un led de la placa. Cuando se le asigna un valor HIGH se enciende, de lo contrario en LOW se apaga.

Conector USB. Existen varios tipos de conectores USB, en concreto esta placa utiliza el tipo B hembra. Con lo cual se necesitará un cable tipo B macho - tipo A macho que deberá conectarse a un conector tipo A hembra (por ejemplo a un ordenador o al cargado de un móvil). La placa se puede alimentar con la tensión del 5V que le proporciona el bus serie USB. Cuando carguemos un programa a la

¹¹ Torrente, O. Curso Práctico de Información. RC Libros. 2013.

placa desde el software de Arduino se inyectará el código del ordenador por este bus.

Botón Reset. Este botón tiene como función reiniciar la ejecución del código en el microcontrolador.

ICSP (In Circuit Serial Programming). Es un conector utilizado en los dispositivos PIC para programarlos sin necesidad de tener que retirar el chip del circuito del que forma parte.

Microcontrolador ATmega328. El microcontrolador es el elemento más importante de la placa. Es donde se instalará y ejecutará el código que se haya diseñado. Ha sido creado por la compañía Atmel, tiene un voltaje operativo de 5V, aunque se recomienda como entrada 7-12V con un límite de 20V. Contiene 14 pines digitales de entrada y salida, 6 pines analógicos que están conectados directamente a los pines de la placa Arduino. Dispone de 32 KB de memoria flash (de los cuales 512 bytes son utilizados por el boot loader). En la memoria flash se instalará el programa a ejecutar. El ATmega328 en las placas Arduino UNO viene precargado con un gestor de arranque (boot loader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. La carga de un nuevo código se realiza del entorno de desarrollo Arduino y la conexión serie/USB. También es posible saltar el gestor de arranque y programar directamente el microcontrolador a través del puerto ICSP (In Circuit Serial Programming). En tal caso, se debe utilizar un programador externo.

Fuente de alimentación externa. La placa puede ser alimentada también mediante corriente continua suministrada por el conector jack de 3.5 mm que podrá recibir entre 7 y 12 V.

Pin de Reset. Es usado para imitar el funcionamiento del botón reset suministrando un valor LOW (0V) para reiniciar el microcontrolador.

Pin de 3.3 V. Desde aquí es posible suministrar 3.3 V a los dispositivos que lo necesiten con una corriente máxima de 50 mA. Es generada gracias al chip FTDI integrado en la placa.

Pin de 5 V. Este pin saca una tensión de 5 V del regulador de la placa. El regulador es necesario puesto que puede ser alimentada con distintos voltajes.

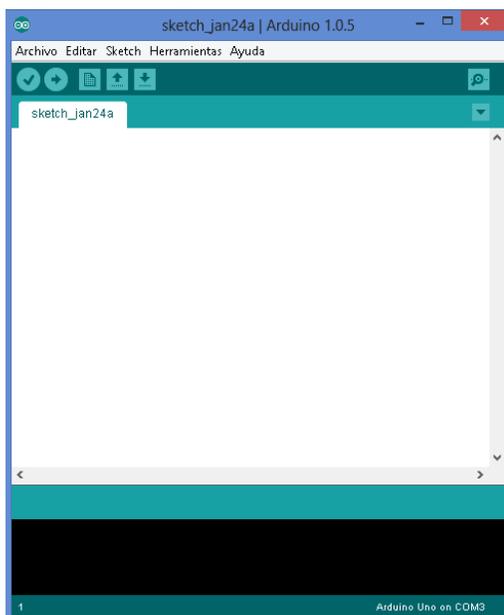
Pin de Vin. Es el voltaje de entrada cuando se usa una fuente de alimentación externa (no tiene en cuenta la conexión USB). Se puede proporcionar voltaje a la placa a través de este pin, o en caso de que se esté utilizando una fuente de alimentación externa tomar el valor que está siendo suministrado.

Pines analógicos. Esta placa contiene 6 pines de entrada analógicos. Los elementos que se conecten aquí suelen tener mayor precisión que los digitales, pero su uso requiere de una lógica levemente mayor. Cada una de estas 6 entradas analógicas proporciona una resolución de 10 bits (1024 valores). Por defecto se mide de tierra a 5V, aunque es posible cambiar la cota superior de este rango usando el pin AREF.

2.2.6 Software (Entorno de desarrollo). La plataforma Arduino tiene un lenguaje propio que está basado en C/C++ y por ello soporta las funciones del estándar C y algunas de C++. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino como Java, Processing, Phyton, Mathematica, Matlab, Perl, Visual Basic, etc¹². Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes anteriormente citados soportan.

El entorno de desarrollo de Arduino es sencillo e intuitivo además puede descargarse gratuitamente desde su página oficial para distintos sistemas operativos. Ha sido implementado con Processing, un lenguaje similar a Java. Está formado por una serie de menús, una barra de herramientas con botones para las funciones comunes, un editor de texto donde se escribe el código, un área de mensajes y una consola de texto. En la Figura 5 se puede apreciar la composición del software de Arduino donde se realiza la programación y se compila.

Figura 5. Figura IDE Arduino



¹² <http://arduino.cc/en/pmwiki.php?n=main/software> (último acceso: 11 de Diciembre de 2014)

Antes que nada es necesario saber que un "sketch", o boceto, es el nombre que usa Arduino para un programa. Es la unidad de código que se sube y se ejecuta en la placa Arduino. El concepto de sketch o boceto es equivalente a proyecto.

Un "sketch" puede contener múltiples archivos (pestañas). Cuando un sketch es compilado, todas las pestañas serán concatenadas juntas para formar el archivo principal del sketch.

Arduino puede utilizar librería y código C/C++. Las pestañas .c o .cpp se compilan por separado y deberían ser incluidas en el sketch usando #include.

Es importante configurar correctamente la placa Arduino que vamos a utilizar, y a través de qué puerto de comunicaciones estará conectada. Para tal efecto, debemos ir al menú "Herramientas" escoger el submenú "Tarjeta" e indicar la placa que usaremos, en nuestro caso "Arduino Uno". En el mismo menú "Herramientas", submenú "Puerto serial", seleccionaremos el puerto correspondiente, en nuestro caso "COM3".

Figura 6. Selección de la placa a usar

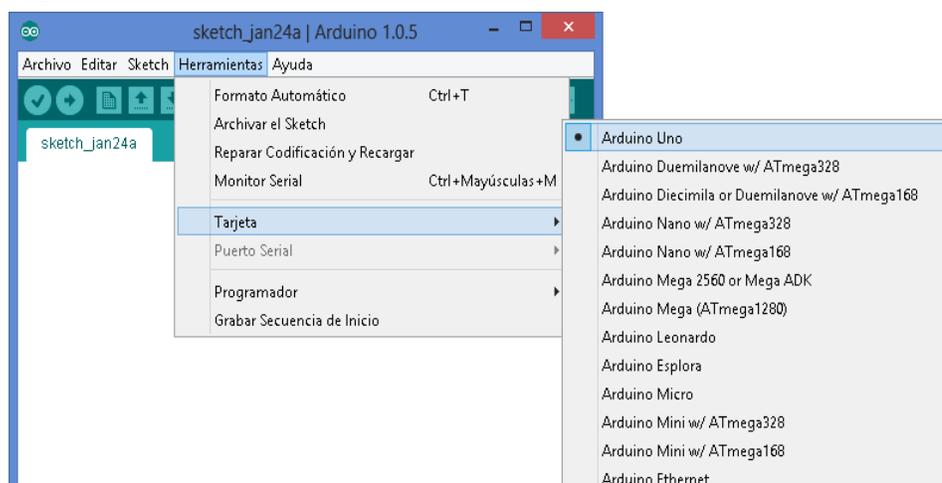
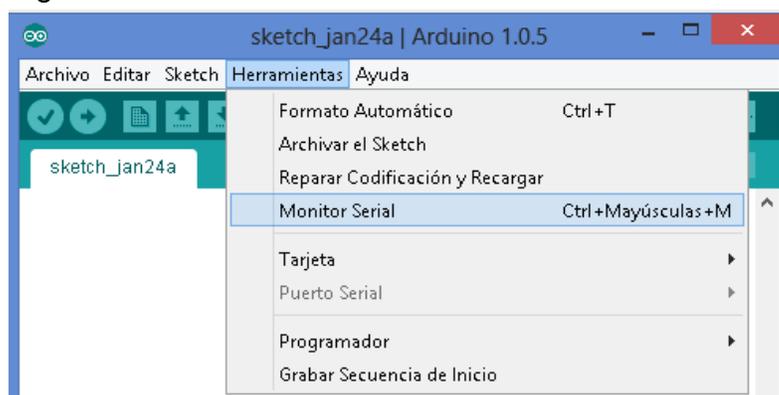


Figura 7. Monitor Serial



La estructura de un programa para Arduino es siempre la misma. Debe existir una función `setup()`, la cual es llamada una vez al inicio de la ejecución, y luego la función `loop()`. Todo el procesamiento se realiza en ésta función.

2.3 MÓDULO 3G + GPS PARA ARDUINO¹³

Este moderno módulo 3G para Arduino permite conectividad a alta velocidad WCDMA y HSPA. La conectividad inalámbrica WCDMA (Wideband Code Division Multiple Access), representa un conjunto de técnicas de acceso múltiple por división de código, el cual permite que todos los usuarios transmitan simultáneamente. Entonces cada usuario se le asigna un código que lo identifica. En esta técnica no existe división en tiempo y en frecuencia, por lo tanto cada uno tiene el mismo ancho de banda. HSPA (High-Speed Packet Access) se refiere a la combinación de tecnologías complementarias a la tercera generación de telefonía móvil (3G)¹⁴. Esta tecnología alcanza velocidades de acceso de 14.4Mbps para descarga y 2Mbps en subida, dependiendo de la congestión de la red.

Este módulo también cuenta con un GPS interno que permite la localización al aire libre, con vista directa al cielo preferiblemente. Combina estándares NMEA y triangulación usando modo A-GPS y S-GPS. El principio de triangulación se basa en la medición de las distancia hacia tres satélites, teniendo en cuenta la velocidad y los tiempos de propagación de la señal. Así se calcula la posición en la tierra con base a las distancias determinadas. Los satélites se usan como puntos de referencia conociendo su posición exacta en cada instante. Esto se conoce como órbita, la cual es muy predecible. Finalmente se corrigen errores debido a la propagación por la atmósfera mediante correcciones matemáticas, modelación o GPS diferenciales¹⁵.

Otro accesorio interesante es que se le puede conectar una cámara de video la cual permite grabación de alta resolución e incluye micrófono, parlante y manos libres. Posee ranura para tarjeta micro SD (32Gb máximo) para almacenar información sea de la red 3G o proveniente del equipo audiovisual. También sirve como reproductor mp3 de los archivos que se guarden en esta memoria.

El moderno hardware interno del módulo permite también enviar y recibir correo electrónico directamente en el Arduino funcionando como cliente SMTP y POP3.

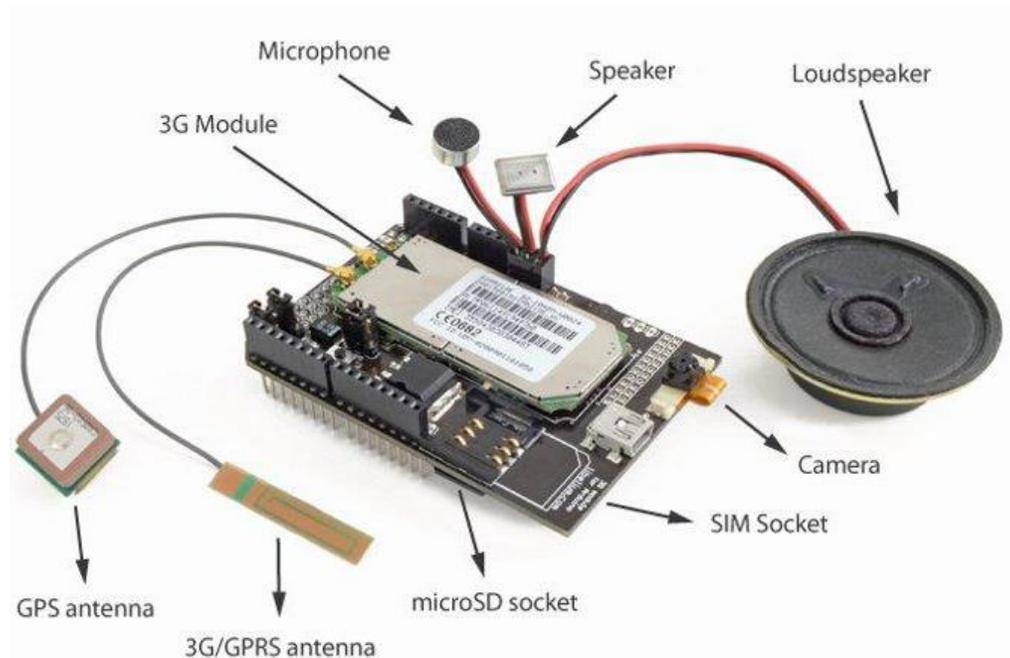
¹³ <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

¹⁴ SAUTER, M. Communications Systems for the Mobile Information Society, Willey. 2006.

¹⁵ A. POZO RUIZ, A RIBEIRO, M.C. GARCÍA-ALEGRE, L. GARCÍA, D. GUINEA, F. SANDOVAL. Sistema de posicionamiento Global (GPS): Descripción, Análisis de errores, aplicaciones y futuro. Instituto de Automática Industrial, Universidad de Málaga

En cuanto al servicio de geolocalización, el GPS interno hace posible esto gracias a sus modo A-GPS y S-GPS. Así entonces, la localización es dada por el GPS a través de especificaciones de la NMEA.

Figura 8. Módulo 3G/GPS para Arduino + Kit de audio y video



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

2.3.1 Características y accesorios. El módulo 3G usa el SIM5218 fabricado por SIMCOM. En cuanto a sus características eléctricas, se encuentra que la fuente de alimentación del SIM5218 es una fuente de voltaje con salida en el rango de 3.4 V a 4.2 V. Se debe procurar no sobrepasar una corriente de 2A. El rango máximo de voltaje es 4.4 V y una corriente de 3A picos. Para las entradas analógicas y digitales, la tensión máxima es de 2.8 V.

En el contexto de radio, el SIM5218 es un motor quad-band GSM/GPRS/EDGE y UMTS que trabaja a frecuencias nominales de GSM, es decir; 850MHz, EGSM 900 MHz, DCS 1800 MHz, PCS 1900 MHz y WCDMA 2100M/1900M/850M. Posee una potencia de salida para UMTS 850/900/1900/2100 de 0.25 W, para GSM 850/900 de 2 W y para DCS 1800/ PCS 1900 de 1 W¹⁶.

Como características extras, posee una interfaz USIM la cual soporta SIM cards de 3 V y 1.8 V, una interfaz USB de alta velocidad cuando se usa como módem,

¹⁶ Manganaro, G. Advances in Analog and RF IC Design for Wireless Communication Systems. Academic Press. 2013

tres canales de audio que incluyen dos entradas para micrófono y tres salidas de audio, una interfaz de cámara que provee un conector FPC y huecos de 2mm. También está abastecido con una interfaz I2C, una entrada ADC de 8 bits con un rango de voltajes de entrada entre los 2.65 V y tierra, una salida de potencia LDO de 2.8 V, una interfaz de 4 bits para tarjeta microSD, tres modos de GPS (stand alone, S-GPS y A-GPS), gestión de archivos, sockets TCP/UDP y POP3/SMTP para correo electrónico.

El módulo Arduino 3G/GPS cuenta con una UART que se puede usar a velocidad máxima, alrededor de 115kbps, así que la comunicación será el mismo ancho de banda como si el Arduino estuviese directamente conectado por USB. Los módulos GPRS disponibles en el mercado permiten velocidades limitadas por la red GPRS, que son alrededor de 20kbps, ahora se puede obtener un rendimiento de descarga seis veces mayor que el anterior. La velocidad de carga se incrementa tres veces, es decir, de 10kbps (GPRS) a 35kbps (3G).

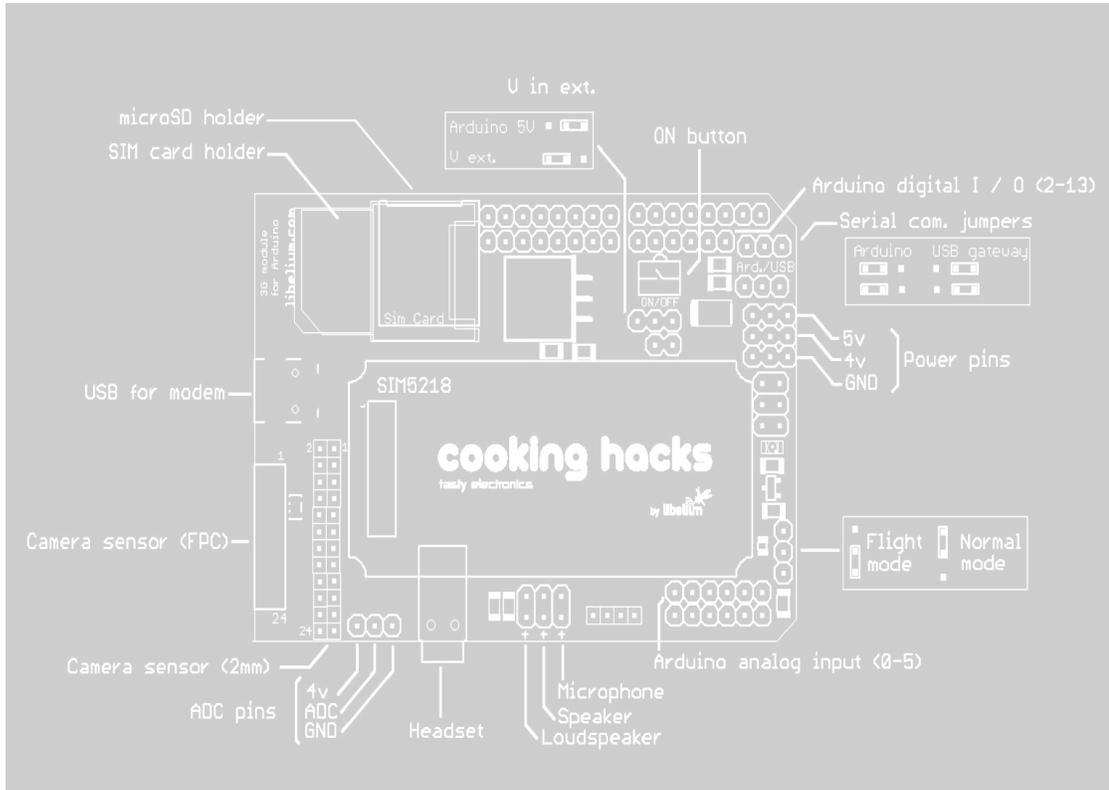
Por otra parte, muchas acciones con las que cuenta el módulo 3G, no están limitadas al microcontrolador de Arduino. Esto significa que se puede controlar todas las acciones de un programa principal que se corre en el Arduino y todos los datos se transmiten desde el módulo 3G a la memoria SD o desde la cámara de video al módulo 3G sin usar la UART y evitando cuellos de botella y asegurando un rendimiento máximo en la velocidad dada por la portadora 3G. Estas velocidades comprenden desde 7.2Mbps en descarga y 5.5Mbps en carga, como se señaló anteriormente.

Lo mejor de esto es que el módulo ofrece teóricamente la posibilidad de realizar video llamadas. Se dice que es teóricamente ya que los fabricantes del módulo (Cooking Hacks), aseguran que pueden existir errores de portadora, según las pruebas realizadas en sus propios celulares. Entonces ellos no pueden asegurar la funcionalidad de video llamada.

Incluso se puede controlar el tipo de servicio de red y la calidad de la señal recibida (RSSI) de las estaciones bases cuando se usan esquemas GSM, GPRS, WCDMA y HSPA dependiendo de la cobertura de la zona.

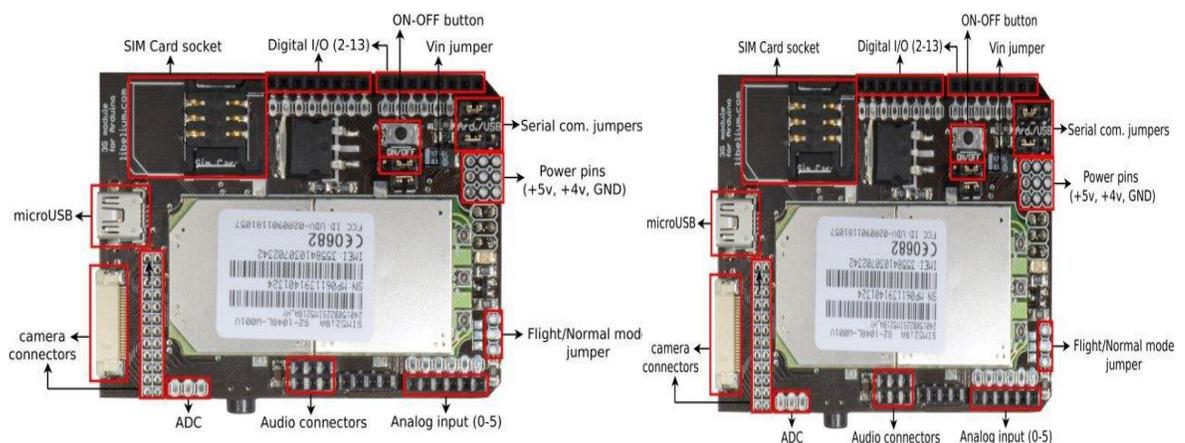
2.3.2 Hardware

Figura 9. Diagrama del módulo



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

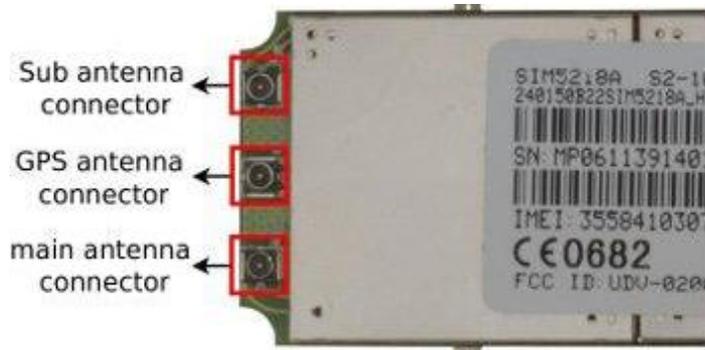
Figura 10. PCB del módulo



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

2.3.3 Antenas. La empresa SIMCOM diseñó la SIM 5218 que tiene tres conectores UFL, dos de ellos para la diversidad de portadores móviles 3G y una para la antena GPS. La impedancia de la interfaz RF es 50 Ω . Es recomendable usar solo el socket principal de la antena 3G, a menos de que se experimenten problemas de cobertura o rendimiento en la red móvil. En este caso se pueden conectar las dos antenas permitiendo diversidad.

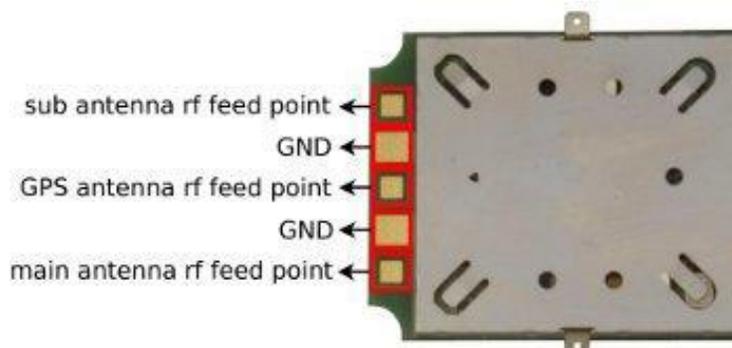
Figura 11. Conexiones de antenas del Módulo 3G Arduino



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

Además, el módulo permite la soldadura de la antena al dispositivo o la fijación de puentes de contacto.

Figura 12. Soldadura de la antena al Módulo 3G Arduino



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

2.3.4 Usando el módulo 3G con comandos AT. Aspectos importantes a considerar; usar letras mayúsculas para los comandos AT, mandar CR (Carriage return) y LF (Line feed) después de un comando AT, ubicar los jumpers de comunicación serial en la derecha, usar fuente de alimentación externa y ubicar los jumpers en la derecha. Si el módulo es alimentado por el Arduino, el jumper de alimentación debe estar en la posición Arduino 5V. Si el módulo es alimentado por la entrada Vin (en el módulo), el jumper debe estar en la posición Vext.

Lo primero que se debe hacer es conectar el módulo al PC directamente (usando Arduino en modo "gateway") y revisando los comandos AT básicos. Los jumpers de comunicación serial deben estar en la posición USB (gateway).

Es necesario que se retire el microcontrolador ATmega de la placa Arduino, tal como se muestra en la siguiente figura sin microcontrolador.

Figura 13. Configuración básica sin ATmega



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

Luego se conecta el módulo al Arduino para trabajar en modo gateway.

Figura 14. Conexión del Arduino con el módulo 3G/GPS



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

Y se conecta la SIM card y el cable USB. Finalmente se conecta el cable USB al PC y se abre el hyperterminal o el monitor serial del Arduino IDE que tenga instalado en el sistema operativo del PC. Se debe tener en cuenta que los comandos AT que se ingresen por el monitor serial del Arduino IDE, se están mandando junto con CR y LF.

Se configura la tasa de baudios a 115200 bps, luego se abre el monitor serial y se presiona el botón "ON" durante dos segundos. Más adelante se escribe en el cuadro de texto AT y se obtendrá como respuesta un OK, lo cual significa que la comunicación con el módulo está funcionando bien. Ahora con el módulo trabajando, se puede revisar algunos comandos AT básicos como.

Cuadro 2. Comando AT básicos

Comando	Respuesta	Descripción
AT	OK	La comunicación con el módulo está funcionando
AT+CPIN="****"	OK	Desbloquea la SIM card con el PIN "****".
AT+COPS?	...	Información del operador

Fuente:<http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

(Ver Anexo A).

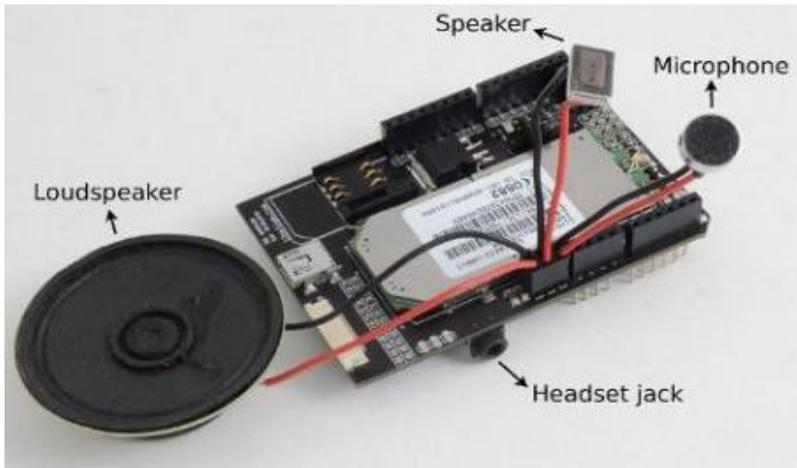
2.3.5 Audio. El módulo SIM5218 dispone de dos entradas analógicas para micrófono y tres salidas analógicas; dos de los parlantes y una para el manos libres. De acuerdo al canal de audio seleccionado (normal, altavoz o manos libres), una entrada y salida específica se activará. Esta selección del canal se realiza gracias al comando AT "AT+CSDVC".

Los canales funcionan de la siguiente manera: El canal normal es mediante el comando (AT+CSDVC=1), en donde se usa el micrófono principal y un altavoz. El canal de auriculares (AT+CSDVC=2) utiliza el conector tipo "Jack". Para el canal manos libres (AT+CSDVC=3), se activa el micrófono principal y el altavoz.

Ambos micrófonos no necesitan alimentación externa ya que la corriente es suministrada directamente por el módulo. Este dispositivo también permite mediante comandos AT la supresión de ruido, el cambio de la ganancia del micrófono y/o cancelación de eco.

Los conectores de audio están situados en la parte inferior de la tarjeta. Los parlantes y el micrófono se conectan a la tarjeta usando los pines de conexión (jumpers).

Figura 15. Módulo 3G/GPS + Kit de audio

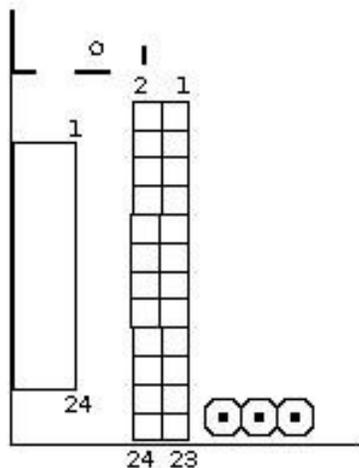


Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

El conector del auricular también está situado en la parte inferior entre la placa Arduino y la tarjeta 3G.

2.3.6 Cámara. El módulo permite conectar una cámara para grabación de video y captura de fotos. La máxima resolución de las imágenes es 640x480 (VGA) y 320x240 (QVGA) con 15 FPS para video. El diagrama de los pines se muestra a continuación.

Figura 16. Diagrama de pines de la cámara



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

Cuadro 3. Especificación de los pines de la cámara

N° Pin	Nombre	N° Pin	Nombre	N° Pin	Nombre
1	NC	9	HSYNC	17	PCLK
2	AGND	10	DVDD	18	DATA6
3	SDA	11	DQVDD	19	DATA2
4	AVDD	12	DATA9	20	DATA5
5	SCL	13	MCLK	21	DATA3
6	RESET	14	DATA8	22	DATA4
7	VSYNC	15	AGND	23	DATA1
8	PWDN	16	DATA7	24	DATA0

Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

2.3.7 GPS. El SIM5218 soporta ambos A-GPS, S-GPS y provee tres modos de operación; modo móvil-asistido (mobile-assited o M-A), modo móvil-basado (mobile-based o M-B) y modo independiente (standalone). A-GPS incluye los modos M-A y M-B.

En el modo M-A, cuando se requiere la posición, la información de la red disponible provee la ubicación del servidor y la asistencia es solicitada por el servidor de localización. Este servidor envía la información de asistencia al teléfono. El GPS obtiene la ubicación por medio de la red de datos disponible. El servidor de localización calcula la posición y da sus resultados de la que se solicitó.

En el modo M-B, los datos de asistencia que provee el servidor de localización no solo se usan cuando se asiste el teléfono midiendo las señales satelitales, también esta información se solicita para calcular la posición del mismo. Por tanto, las medidas del GPS y la red de datos disponible regresan al servidor de localización, el móvil calcula la ubicación en el teléfono y transmite su resultado con la entidad solicitante.

En el modo independiente (standalone), el teléfono demodula los datos directamente de los satélites GPS. Este modo tiene sensibilidad para arranque en frío, y un tiempo más largo de fijación inicial respecto a los modos asistidos. Sin embargo, esto requiere interacción con el servidor y trabaja fuera de cobertura.

La combinación de las mediciones del GPS y la información disponible de red provee; alta sensibilidad de solución que trabaja en todos los ambientes (rural, urbano, en recintos abierto y cerrados) y alta disponibilidad la cual permite el uso de satélites e información de red.

Por lo tanto, mientras que las redes móviles normalmente tienen mal desempeño en zonas rurales y zonas geoméricamente no cubiertas por las células o zonas de recintos cerrados, el GPS del SIM5218 proporciona el tiempo óptimo para fijar la

precisión, la sensibilidad, la disponibilidad y la utilización de la red reducida en ambos ambientes, dependiendo de la condición dada.

En resumen, en el modo standalone, el GPS obtiene la posición dando la latitud, longitud, fecha, tiempo UTC, altitud y velocidad. En los modos S-GPS y A-GPS, el módulo se conecta con el servidor GPS y calcula la posición.

2.3.8 SMS. Este moderno módulo cuenta con un sistema simple para el envío de mensajes de texto conocidos como SMS (short message service) a través de la infraestructura de red GSM, esto facilita la comunicación con otros teléfonos mediante texto, en donde se programa por software el número de destino y otras configuraciones necesarias para este propósito. Por medio de comandos AT se pueden almacenar o leer estos mensajes, es decir, hacer una gestión de los SMS recibidos¹⁷.

2.3.9 Correo electrónico. Para el envío de correo electrónico, es necesario contar con un servidor SMTP y una cuenta de correo. SIMCOM desarrolló este módulo con algunas restricciones para servidores SMTP que usen la encriptación SSL/TTL, es decir, los servidores más populares como Outlook, Gmail y Yahoo, poseen esta seguridad que impide que el módulo se conecte con estos servidores para el envío de e-mails. Es estrictamente necesario contar con un servidor SMTP que no posea seguridad SSL/TTL y así no tener complicaciones con el envío de correos electrónicos.

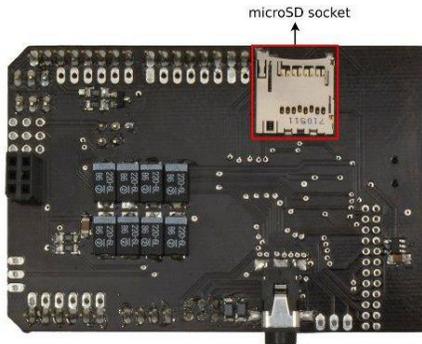
Este módulo también tiene el hardware necesario incorporado para a través del protocolo POP3, se puedan recibir e-mails. Esta funcionalidad al igual que la anterior, requiere un servidor POP3 y una cuenta de correo. Por medio de programación se realiza la asignación de puertos, IPs, y credenciales necesarias para la autenticación con el servidor.

2.3.10. Almacenamiento interno y externo. El sistema de archivos es usado para almacenamiento es jerárquica. El espacio de almacenamiento local es denominado "C:", y la SD es denominada "D:", esto corresponde a los directorios. El módulo crea cuatro direcciones llamadas "Picture", "Audio", "Video" y "VideoCall" automáticamente; La carpeta "Picture" es usada para almacenar las fotos tomadas por la cámara, "Audio" almacena los archivos de audio, "Video" los archivos de video en varios formatos (mp4, avi, etc.), "VideoCall" almacena archivos multimedia los cuales son guardados durante una video llamada. El

¹⁷ http://www.clarin.com/sociedad/primer-SMS-mandan-millones-minuto_0_821917836.html (último acceso: 11 de Diciembre de 2014).

tamaño máximo de la memoria es de 32GB. A continuación se presenta una imagen de la ranura disponible para memorias micro SD.

Figura 17. Ranura para memoria microSD



Fuente: <http://www.cooking-hacks.com/documentation/tutorials/arduino-3g-gprs-gsm-gps>

2.4 ANDROID

Android es un sistema operativo de código abierto basado en Linux y distribuido por Google bajo la licencia Apache. Está diseñado principalmente para dispositivos móviles de pantalla táctil, como smartphones o tablets. Su naturaleza de código abierto y lo permisivo de su licencia hace posible que el software basado en Android sea libremente creado, modificado y distribuido por cualquier desarrollador. Se facilita completo acceso al kit de desarrollo de software o SDK que proporciona todas las herramientas necesarias para implementar, probar y depurar aplicaciones, junto con prestaciones adicionales como librerías API, documentación, emulador, códigos de ejemplo y tutoriales. Además, se permite a los desarrolladores la libre publicación de sus creaciones, facilitando su distribución a través de la plataforma Google Play¹⁸.

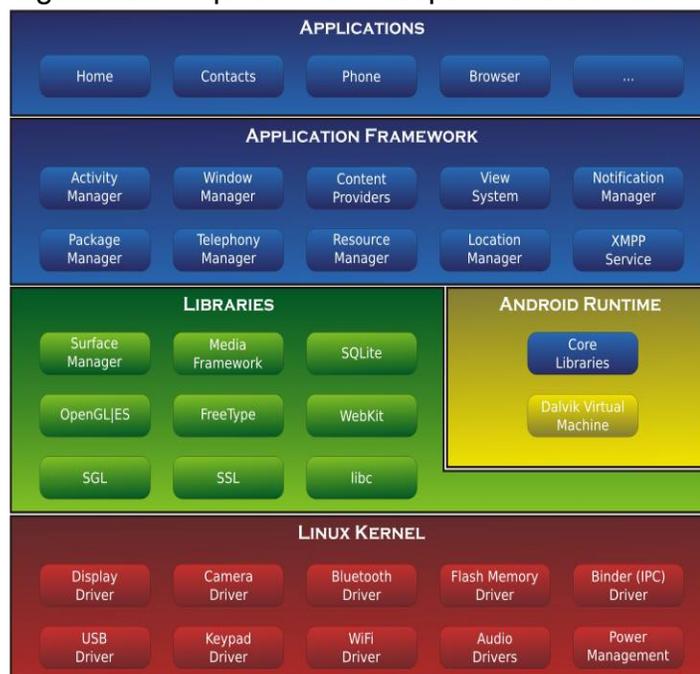
Los programas en entorno Android, denominados comúnmente aplicaciones o apps, se implementan en una versión personalizada del lenguaje de programación Java. Para ello se puede hacer uso de la plataforma libre Eclipse que proporciona entornos de desarrollo integrado tanto para Java como para Android. Además, recientemente ha sido publicado por Google un entorno de desarrollo específico de Android, denominado Android Studio, disponible de forma gratuita desde su web. Debido a que Android Studio se encuentra todavía en fases muy iniciales de su desarrollo, todo el software Android implementado en este proyecto se ha desarrollado mediante la plataforma Eclipse.

¹⁸ GIRONES, Jesús Tomás. El Gran libro de Android

2.4.1 Arquitectura de Android. El sistema operativo Android está conformado por una estructura software dividida en diversos niveles. Cada capa de esta arquitectura proporciona servicios a su capa inmediatamente superior.

- **Núcleo Linux:** Todo el sistema Android se construye sobre un núcleo Linux personalizado por Google. Este nivel se encarga de interactuar con el hardware proporcionando los controladores necesarios. Además actúa como capa de abstracción entre el hardware y el resto de capas software de la arquitectura.
- **Librerías y Android Runtime:** La siguiente capa corresponde a las librerías nativas de Android, en lenguaje C/C++. Al mismo nivel se encuentra la capa de ejecución denominada Android Runtime, que consta de la máquina virtual Dalvik, encargada de ejecutar las aplicaciones, y un conjunto de librerías Java.
- **Marco de Aplicación:** Por encima de los niveles de librerías y ejecución se encuentra la capa marco de aplicación o framework. Esta capa gestiona funciones básicas del dispositivo y proporcionan las herramientas sobre las que se construyen las aplicaciones.
- **Aplicaciones:** Esta es la capa superior de la arquitectura Android. Incluye todas aplicaciones, desde las incorporadas por defecto en Android como las desarrolladas por terceros e instaladas por el usuario.

Figura 18. Arquitectura en capas del sistema Android



Fuente: El gran Libro de Android. Jesús Tomás Girones

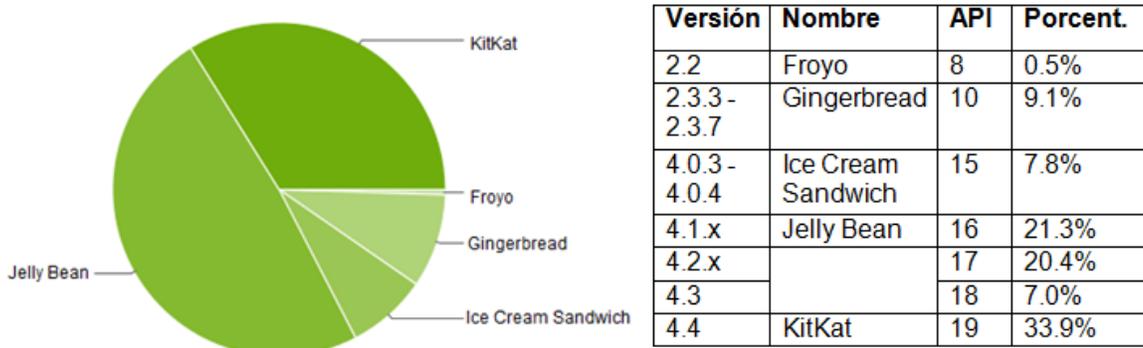
2.4.2 Las versiones de Android y niveles de API. Antes de empezar un proyecto en Android hay que elegir la versión del sistema para la cual se desea realizar la aplicación, es decir, la versión destinada a soportar el aplicativo móvil desarrollado. Es muy importante observar que hay clases y métodos que están disponibles a partir de una nueva versión, si se va a usar es necesario conocer la versión mínima requerida.

Cuando se ha lanzado una nueva plataforma siempre ha sido compatible con las versiones anteriores. Es decir, solo se añaden nuevas funcionalidades y en el caso de modificar alguna funcionalidad no se eliminan sino que se etiquetan como obsoletas, pero se pueden continuar utilizando.

Las plataformas lanzadas hasta la fecha se identifican de tres formas alternativas: versión, nivel de API y nombre comercial. El nivel de API corresponde a números enteros comenzando desde 1. Para los nombres comerciales se han elegido postres en orden alfabético.

2.4.3 Elección de la plataforma de desarrollo. A la hora de seleccionar la plataforma de desarrollo hay que consultar si se necesita alguna característica especial que solo esté disponible a partir de una versión, ya que cada versión de Android tiene una API distinta, en la cual se han desarrollado más librerías, se han actualizado otras o incluso se han abandonado por otras más desarrolladas. Todos los usuarios con versiones inferiores a la seleccionada no podrán instalar la aplicación. Por lo tanto, es recomendable seleccionar la menor versión posible que la aplicación pueda soportar. La creación y evolución de Android ha ocasionado que existan dispositivos con todas las versiones disponibles, para tomar una buena decisión de qué plataforma utilizar es interesante consultarlos porcentajes de utilización de las diferentes versiones de Android:

Gráfica 1. Dispositivos Android según la plataforma instalada



Fuente: <https://developer.android.com/about/dashboards/index.html>

Estos datos son los dispositivos Android según la plataforma instalada que han accedido a Google Play Store durante una semana terminada el 1 de Diciembre de 2014. Tras estudiar la gráfica se destaca el reducido número de usuarios que utilizan la versión 2.2 (0.5 %). Por lo tanto, es buena idea utilizar como versión mínima la 2.3.3 o 3.0 para desarrollar el proyecto, dado a que se daría cobertura al 99.5% de los terminales. Las versiones 4.3 y 4.4 con un 7.0 %, y 33.9 % tienen un porcentaje de usuarios considerable y que viene en aumento.

2.4.4 Herramientas para el desarrollo. Para empezar a programar en Android, lo primero que se hace es preparar el entorno de trabajo. Google ha preparado el paquete de software Android SDK que provee un conjunto de herramientas necesarias que facilitan el desarrollo de aplicaciones en Android. En él se incluye conversor de código, depurador, librerías, emulador, documentación, ejemplos de código, etc. El SDK de Android es compatible con varios sistemas operativos, Windows, Mac y Linux.

La mayoría de desarrolladores prefieren utilizar un IDE, o entorno de desarrollo integrado que integre un editor de texto con todas las herramientas de desarrollo. Para este proyecto se utiliza el Eclipse. El procedimiento para instalar las herramientas necesarias es el siguiente:

2.4.5 Componentes de una aplicación. Los componentes de la aplicación son los bloques esenciales para desarrollar una aplicación Android. Los componentes básicos de una aplicación son: Activities, intents, views, services, content providers y broadcast receivers. Si se dominan bien estos términos se puede saltar directo al código.

- **Vistas (View):** Las vistas son los elementos que componen la interfaz de usuario de una aplicación como, por ejemplo, un botón o una entrada de texto. Todas las vistas van a ser objetos descendientes de la clase View y, por tanto, pueden ser definidos utilizando código Java. Sin embargo, lo habitual va a ser definir las vistas utilizando un fichero XML y dejar que el sistema cree los objetos por a partir de este fichero. Esta forma de trabajar es muy similar a la definición de una página web utilizando código HTML.
- **Layout.** Un Layout es un conjunto de vistas agrupadas de una determinada forma. Hay diferentes tipos de Layouts para organizar las vistas como son de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista. Los Layouts también son objetos descendientes de la clase View. Igual que las vistas los Layouts pueden ser definidos en código, aunque la forma habitual de definirlos es utilizando código XML.

- **Actividad (Activity).** Una aplicación en Android va a estar formada por un conjunto de elementos básicos de visualización, coloquialmente conocidos como pantallas de la aplicación. En Android cada uno de estos elementos, o pantallas, se conoce como actividad. Su función principal es la creación del interfaz de usuario. Una aplicación suele necesitar varias actividades para crear el interfaz de usuario. Las diferentes actividades creadas serán independientes entre sí, aunque todas trabajarán para un objetivo común. Toda actividad ha de pertenecer a una clase descendiente de Activity.
- **Servicio (Service).** Un servicio es un proceso que se ejecuta "detrás", sin necesidad de una interacción con el usuario. Es algo parecido a un demonio en Unix o a un servicio en Windows. En Android se dispone de dos tipos de servicios: servicios locales, que pueden ser utilizados por aplicaciones del mismo terminal y servicios remotos, que pueden ser utilizados desde otros terminales.
- **Intención (Intent).** Una intención representa la voluntad de realizar alguna acción; como realizar una llamada de teléfono, visualizar una página web. Se utiliza cada vez que se quiera lanzar una actividad, un servicio, un anuncio de tipo broadcast, comunicarse con un servicio.

Los componentes lanzados pueden ser internos o externos a la aplicación. También se utilizan las intenciones para el intercambio de información entre estos componentes. En muchas ocasiones una intención no será inicializada por la aplicación, si no por el sistema, por ejemplo, cuando se pide visualizar una página web. En otras ocasiones es necesario que la aplicación inicialice su propia intención. Para ello se crea un objeto de la clase Intent.

- **Receptor de anuncios (Broadcast receiver).** Un receptor de anuncios recibe y reacciona ante anuncios de tipo broadcast. Existen muchos originados por el sistema; como por ejemplo Batería baja, llamada entrante,... Aunque, las aplicaciones también pueden lanzar un anuncio broadcast, no tienen interfaz de usuario, aunque pueden iniciar una actividad para atender a un anuncio.
- **Proveedores de Contenido (Content Provider).** La difusión de información entre teléfonos móviles resulta un tema vital. Android define un mecanismo estándar para que las aplicaciones puedan compartir datos sin necesidad de comprometer la seguridad del sistema de ficheros. Con este mecanismo es posible acceder a datos de otras aplicaciones, como la lista de contactos, o proporcionar datos a otras aplicaciones¹⁹.

¹⁹ DONN FELKER. Android Application Development for Dummies. Wiley 2011.

3. DISEÑO

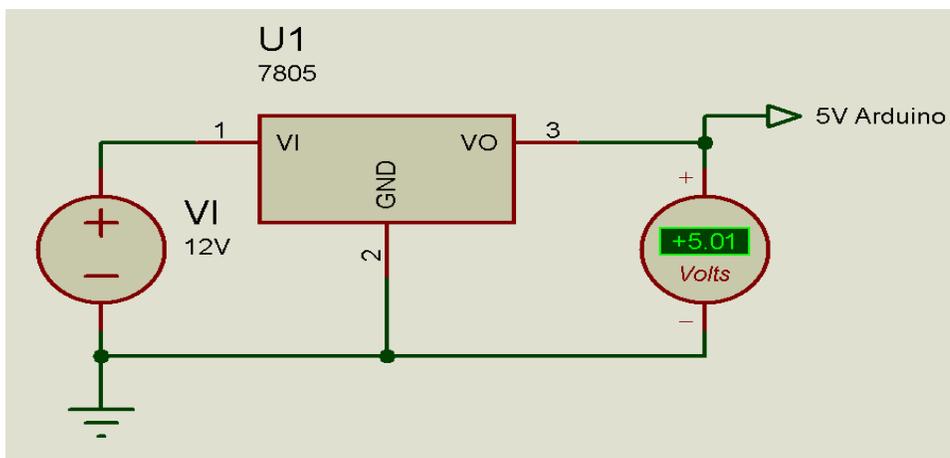
3.1 COMPONENTE DE HARDWARE

El hardware necesario para el desarrollo e implementación del sistema de seguridad vehicular en la camioneta TUCSON iX35 se compone de tres elementos, el primer elemento es el regulador de voltaje L7805CV, luego viene el sistema de alarmas GENIUS que ya está instalado en la camioneta y por último está un chasis, el cual alberga todo el proyecto, es decir, el regulador de voltaje, la placa Arduino UNO junto con el módulo 3G/GPS, esto con el fin de que todo este sistema pase desapercibido por el ocupante o ladrón del vehículo.

3.1.1. Diseño circuito de adecuación. En la Figura 2 presentada al inicio de este documento se puede apreciar qué se requiere regular la tensión de salida de la sirena, por tanto es el regulador de voltaje el indicado para adecuar la señal de salida de la sirena para la entrada del Arduino.

3.1.1.1. Regulador de voltaje. Las entradas analógicas del Arduino permiten voltajes de 0 a 5V para su procesamiento digital, para ello se utilizó un regulador de voltaje con referencia L7805 el cual entrega valores entre estos rangos permitidos. En caso de que no se haya activado la alarma, el voltaje es 0V, cuando se active la alarma, implica un voltaje de entrada en el regulador de aproximadamente 12V y este a su vez entrega un voltaje de 5V a su salida que va conectado al pin digital número 3 del Arduino UNO. A continuación se muestra la simulación del circuito realizado en el software Proteus 8 Professional. (Ver Anexo B).

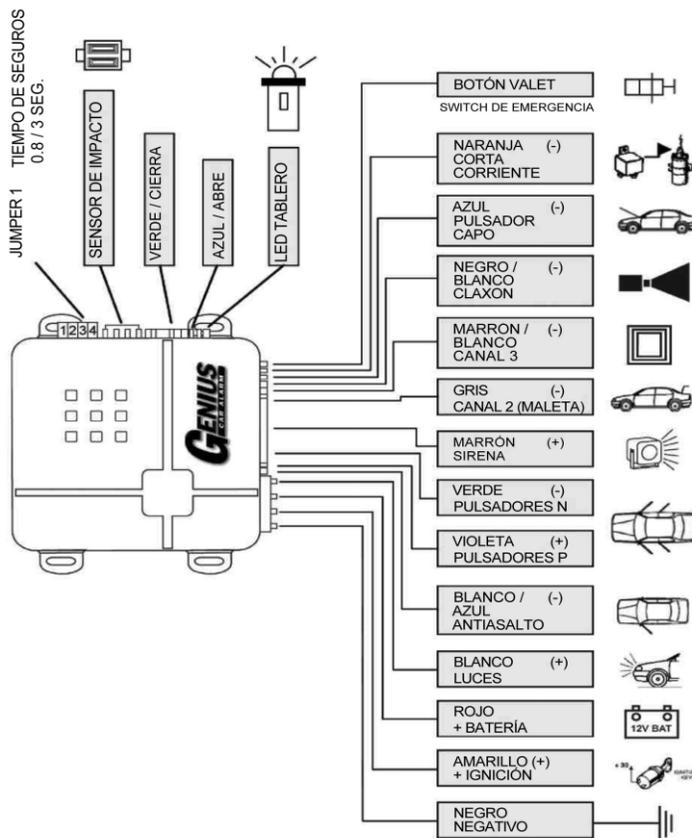
Figura 19. Simulación del regulador de voltaje L7805CV



En donde VI representa el voltaje de la alarma que va a la sirena del sistema de seguridad, en este caso, la señal VI es 12V, por lo tanto el regulador entrega 5.01V a su salida, la cual es interpretada como un 1 lógico o HIGH en el pin digital 3 de la placa Arduino.

3.1.1.2. Sistema de alarma GENIUS. Se adaptó el sistema de alarmas GENIUS que viene instalado en la camioneta Hyundai TUCSON IX35. Está alarma cuenta con un módulo principal donde está el microprocesador, el cual recibe las señales de todos los sensores instalados en el auto, por ejemplo; sensores anti asalto y sensores de impacto. El módulo presenta una serie de salidas, una de ellas es la señal que activa la sirena, representada por un cable de color marrón (café). En la Figura 20 se pueden apreciar los pines de entrada y salida del módulo principal Genius, en donde se especifica el color del cableado y su función. Para más información acerca de datos técnicos del regulador de voltaje, consulte el (Ver Anexo C).

Figura 20. Diagrama de instalación de alarmas GENIUS

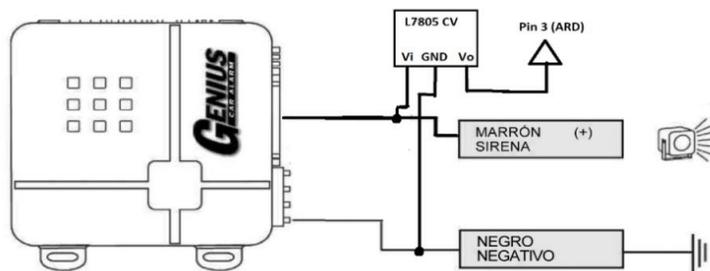


Fuente: Alarmas Genius 2A 4 BOTONES

Entonces cuando el módulo recibe algún pulso, cuya fuente puede ser los sensores de impacto o anti asalto, el micro procesador envía un pulso constante de 12 voltios que va conectado a la sirena. Es aquí el momento cuando la sirena empieza a sonar alertando al propietario del vehículo. Dependiendo de la fuerza del impacto, así mismo es la duración de la alarma, que oscila entre los 30 segundos y hasta 1 minuto. Por otra parte, si se abre una puerta, es el sensor anti asalto el encargado de informar acerca de este suceso. En otras palabras, mantiene información sobre el estado de todas las puertas del vehículo. Luego este pulso de 12V se conecta a la entrada del regulador L7805 y su salida de 5V se conecta a un pin digital del Arduino UNO, exactamente al pin 3.

En conclusión, el sistema Arduino 3G/GPS se conecta a la alarma GENIUS a través del regulador L7805CV, esto se hace conectando el cable marrón (café) de la sirena a la entrada del regulador y el cable negro de la alarma (tierra) se conecta al pin GND del regulador y a su vez, a la tierra de todo el sistema.

Figura 21. Diagrama del circuito de adecuación de alarma GENIUS con Arduino 3G/GPS



3.2 COMPONENTE DE SOFTWARE

Se usaron dos herramientas muy conocidas en el área de la programación de software, Eclipse y Arduino IDE. Son plataformas de desarrollo de software libre y de fácil uso, su descarga es gratuita desde las páginas oficiales de sus creadores.

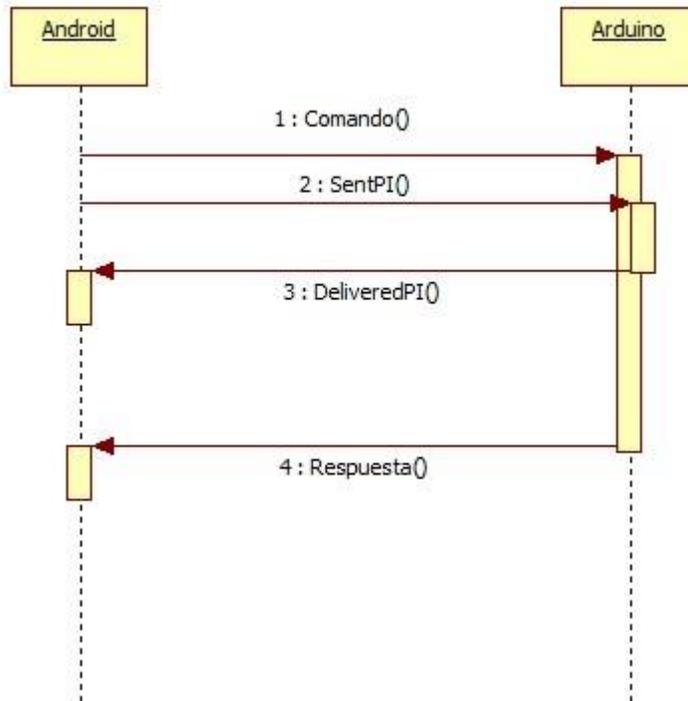
3.2.1 Modelamiento de la aplicación. El modelamiento de la aplicación se elaboró con herramientas UML²⁰.

3.2.1.1. Comunicación Android-Arduino. Para llevar a cabo el proceso de comunicación entre sistemas Arduino y Android se utiliza el servicio de SMS. Es necesario entender primero que comprende Arduino y Android. El sistema Arduino está compuesto por una placa Arduino UNO y un módulo 3G/GPS de cooking

²⁰ <http://users.dcc.uchile.cl/~psalinas/uml/introduccion.html> (último acceso: 12 de Diciembre de 2014)

hacks. Estos dispositivos se transmiten datos a una tasa de baudios de 115200 por serial. Por otra parte el sistema Android está representado en el teléfono inteligente con la aplicación SAFECAR ya instalada. A continuación se muestra en la Figura 22, un diagrama de secuencias de la comunicación entre Arduino y Android de manera general.

Figura 22. Comunicación Android-Arduino

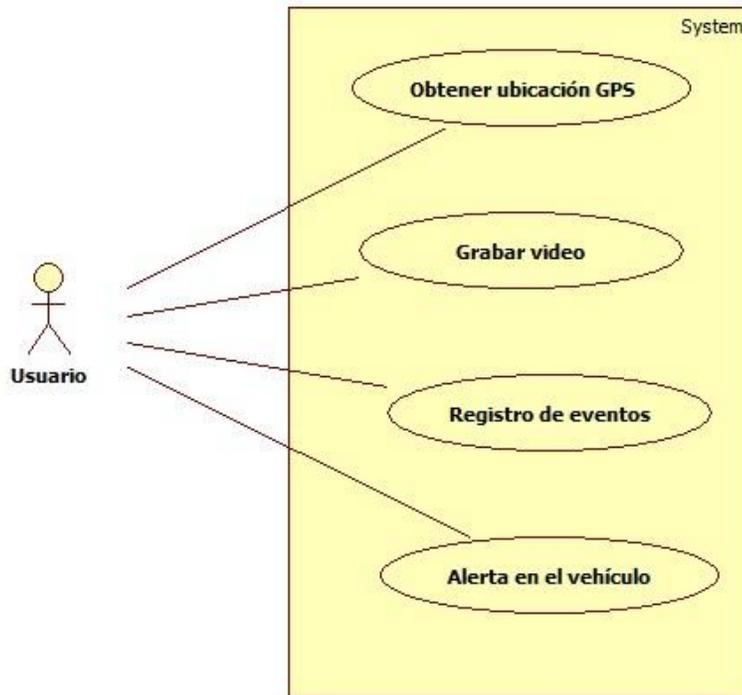


El Comando() es un SMS que indica que función va a realizar el módulo 3G/GPS, si es grabación de video o ubicación GPS. Más tarde, Android envía un mensaje SentPI() el cual confirma que el SMS se transmitió correctamente desde el dispositivo móvil. Luego se genera un DeliveredPI(), que informa si el SMS se recibió con éxito. Finalmente y luego de grabar el video u obtener las coordenadas GPS, Arduino responde al comando inicial de dos maneras: con un SMS que indica si el video ya terminó de grabar o con un SMS con las coordenadas GPS dentro del mensaje²¹.

3.2.1.2. Comunicación Usuario Sistema. En función de representar la comunicación entre el usuario y el sistema de seguridad, se elaboró el siguiente diagrama de casos de uso.

²¹ <http://www.compiletimeerror.com/2013/10/smsmessage-delivery-in-android-with.html#.VIsJwiuUdds>

Figura 23. Comunicación Usuario Sistema



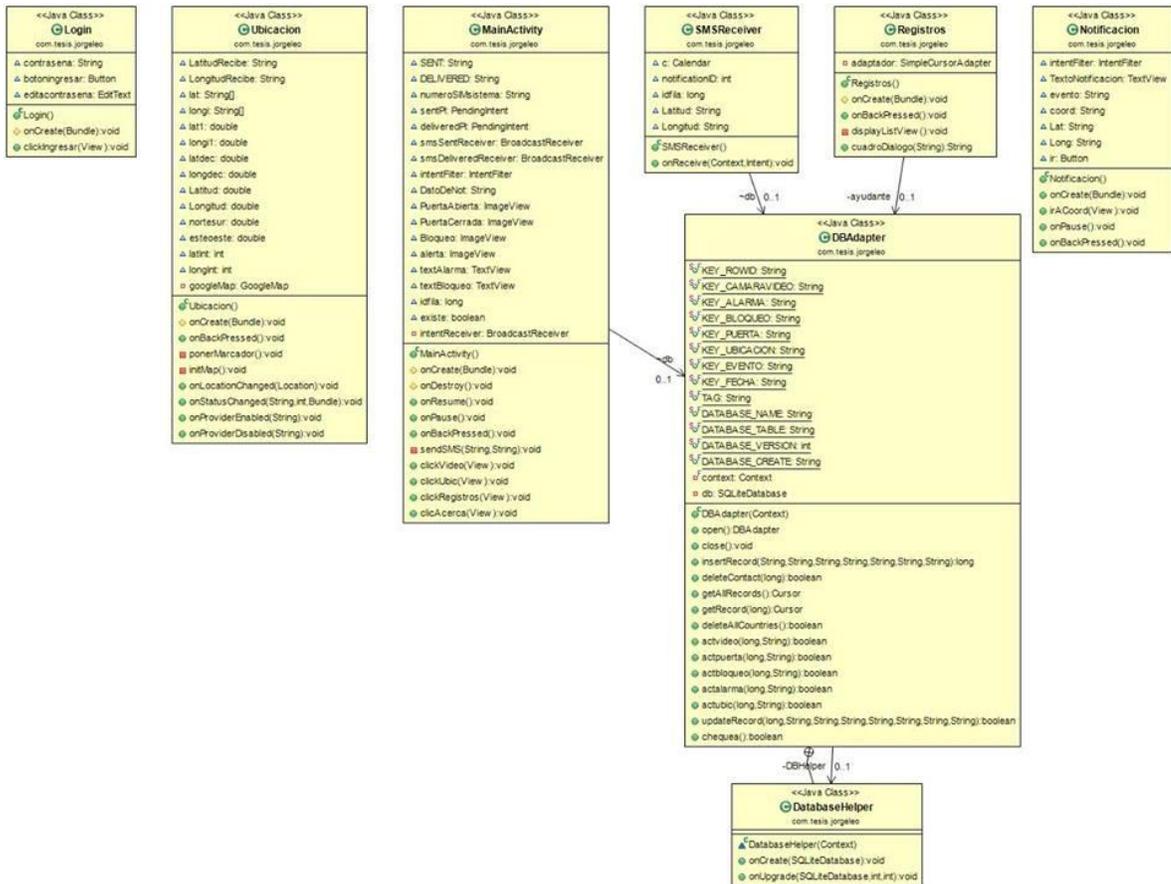
El usuario realiza una serie de acciones, la primera acción basándose en el diagrama de casos de uso es Obtener ubicación GPS, que comprende desde el momento que el usuario presiona el botón “Ubicar vehículo” hasta que finalmente recibe el mensaje SMS con las coordenadas visualizadas en Google Maps. La siguiente acción es Grabar video. En ésta el usuario presiona el botón “Tomar video” y a través de SMS llega el comando al sistema que inicia la captura el video y lo almacena en la tarjeta micro SD, cabe aclarar que el video no se transmite, solo se envía un SMS de confirmación de que el video se grabó correctamente. Registro de eventos es una acción en donde el usuario accede a la base de datos SQLite para poder supervisar los eventos sucedidos desde que se instaló la App en el móvil. Esta base de datos contiene información como fecha, hora y el tipo de evento, además es posible revisar ubicaciones GPS pasadas presionando sobre el evento.

Por último, la acción Alerta en el vehículo es más una manera de informar al usuario que la alarma del carro se ha activado, y así este pueda realizar preventiva, como grabar video, según sea su conveniencia o vaya a ver su carro inmediatamente.

3.2.2.2 Aplicación móvil Android SAFECAR. En función de los requisitos y problema a resolver, se ha definido un modelo de clases, atributos y operaciones.

Mediante el diagrama de clases se representa el esquema de la implementación del sistema de seguridad. En la figura 24 se muestra el diagrama de clases asociado. (Ver Anexo D y E).

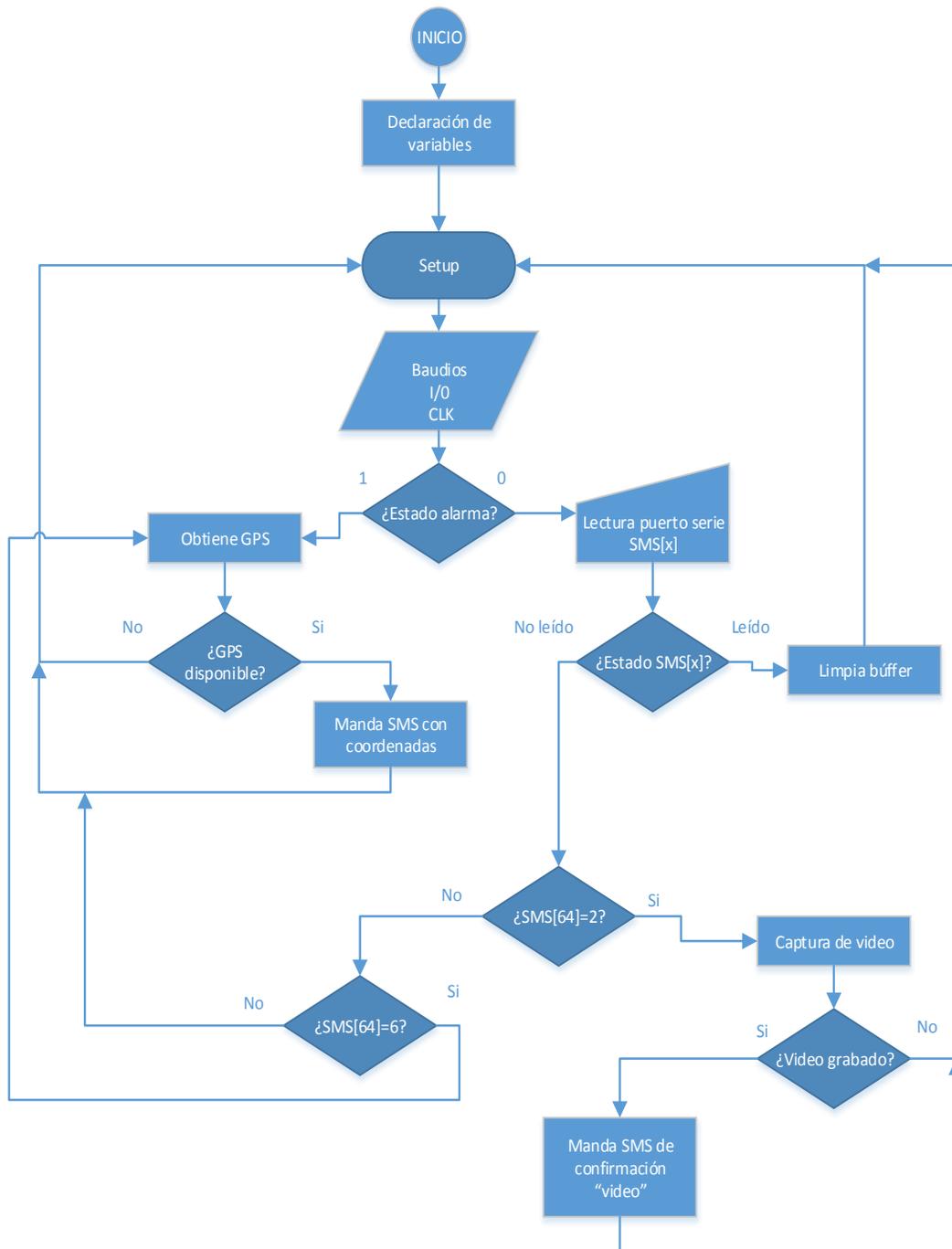
Figura 24. Diagrama de clases



3.2.1.3. Sketch Arduino. Arduino, como se ha podido ver a lo largo de este documento, es un software libre basado en lenguaje C. Esta característica además de su facilidad y versatilidad a lo hora de programar, hace que muchos desarrolladores se inclinen por esta plataforma y así ir reemplazando los micro procesadores convencionales por placas Arduino, las cuales son mucho más potentes, versátiles y útiles para el diseño de nuevos sistemas en el campo de la electrónica. Antes de que se revise los detalles de programación, es necesario analizar el diagrama de flujo del software desarrollado en Arduino IDE para suplir las necesidades y cumplir los objetivos planteados en este proyecto. Para esto se desarrolló una programación rápida y muy confiable para estar constantemente revisando los mensajes de texto que llegan al módulo 3G/GPS. Es claramente apreciable que la estructura de programación del sistema Arduino es similar a

otros sistemas y lenguajes más comunes y populares como C/C++. Declaración de variables, configuraciones y luego el ciclo principal del sketch.

Figura 25. Diagrama de flujo programación Arduino



Se creó un arreglo en el sketch de Arduino para tomar el SMS enviado desde la aplicación móvil y según el botón que se presione, éste ejecuta una acción. Existen dos acciones o eventos; el primer evento es tomar video, en donde el módulo realiza una grabación de video en formato MP4, de resolución 320x240 con 15 segundos de duración.

El segundo evento consiste en mostrar la ubicación GPS, aquí el módulo determina su posición actual y la envía en un mensaje de texto con la siguiente sintaxis: "Ubic:latitudN:longitudW", donde "latitud" y "longitud" vienen siendo los números de la coordenada en formato NMEA. Luego este arreglo SMS[x] es comparado a través de la estructura if y else if, revisando así su contenido, es decir, el cuerpo del mensaje SMS. Antes de esto, se verifica el estado de la alarma (Figura 25); si en la entrada del regulador de voltaje hay un bajo (LOW) se hace la lectura del SMS[x] por el puerto serie, mediante el llamado de la función checkSMS(), que en cuyo caso revisa el búfer de llegada constantemente buscando nuevos SMS entrantes.

En caso de que haya un mensaje marcado como no leído (unread), el sistema revisa este SMS comparando el contenido del mismo con una variable tipo char (Figura 39), Si este mensaje es '2', entonces se realiza la grabación de video mediante la función implementada VIDEO() y esta misma envía un SMS de confirmación de que el video ha sido tomado. Cuando la entrada del regulador de voltaje es un alto (HIGH), se llama a la función get_GPS() y sendSMS() para obtener la ubicación GPS y enviar las coordenadas.

También se llaman estas funciones, si el SMS que llegó es igual a 6 ejecutará las funciones get_GPS() y sendSMS() para precisar la ubicación GPS y mandar las coordenadas a la aplicación Android (Figura 41). En otras palabras, el sistema manda las coordenadas GPS debido a dos estímulos, por solicitud del usuario o por la activación de la alarma en la camioneta.

Finalmente cuando el SMS está marcado como leído (read), se procede a la limpieza del búfer, borrando el SMS que llegó a la memoria de la SIM Card y limpiando a su vez el arreglo SMS[x], esto se realiza debido a que este mensaje es desechable y para no crear cuellos de botella en el búfer. (Ver Anexo F).

4. LIMITACIONES

- El sistema de seguridad no estará disponible si el módulo no cuenta con una cobertura de la red GSM que le permita enviar y recibir SMS sin ningún problema. Por ejemplo en zonas rurales, de difícil acceso y/o geométricamente no cubiertas por las células. Si se cuenta con un acceso lento e inestable a Internet, la capacidad del sistema disminuye debido a que la ubicación GPS a través de Google Maps requiere una velocidad mínima de 128kbps en descarga.
- La persona propietaria del vehículo, necesita tener un saldo disponible tanto en su teléfono móvil como en la SIM card del sistema Arduino 3G/GPS. Además es necesario tener una conexión a Internet en su dispositivo Android, ya sea por WiFi o por plan de datos. Y así disponer de un sistema de seguridad rápido y confiable en su vehículo.

5. CONCLUSIONES

- Se logró diseñar e implementar un sistema de seguridad para un automóvil, el cual consta de una aplicación móvil, amena con el usuario y una placa Arduino + shield 3G/GPS que permite el envío vía SMS al aplicativo móvil de los datos para la geolocalización satelital asistida por red e información sobre la actividad del sistema de alarma.
- La aplicación Android desarrollada en Eclipse permite al usuario acceder y administrar los datos del Arduino+Shield 3G/GPS mediante la implementación de clases y métodos para recepción y extracción de los mensajes de texto entrantes al móvil. La información recibida queda almacenada en una base de datos SQLite, a la cual se ingresa a través del botón "Registro de Eventos" como son la fecha, hora, longitud, latitud.
- El sistema permite la grabación de video en el interior del vehículo, con la limitación de acceder a los videos, ya que se debe extraer la tarjeta micro SD para visualizarlos.
- Es un sistema seguro y adaptable para la implementación en otros vehículos sean particulares o de servicio público para fines tales como; tener un rastreo y control del servicio de taxi, evaluar tiempos de desplazamiento, supervisar el comportamiento de los taxistas y además tener evidencias en caso de un robo, un accidente, secuestro, etc. siendo así un sistema integral debido a que el hardware necesario se consigue a un precio bastante cómodo.

6. RECOMENDACIONES

- Uso de otros protocolos de comunicación (HTTP, FTP, SMTP...) para recibir los registros de video o las coordenadas GPS. Puesto que para el envío de los videos y la ubicación vía email, era necesario implementar un servidor SMTP propio, elevando costos de diseño, y los servidores comerciales como Gmail, Yahoo, Outlook presentan restricciones de tráfico email desde el módulo 3G/GPS a sus servidores. Además la manipulación de los datos mediante mensajes de texto posee menos complejidad y costo de desarrollo.
- Uso de video llamadas para monitoreo del automóvil en tiempo real, ya que los fabricantes de la placa no pueden asegurar la funcionalidad de la video llamada debido a errores de portadora.
- Uso de hardware de menos costo y que tenga más flexibilidad.
- Se recomienda usar un dispositivo móvil con un sistema operativo Android con una versión igual o mayor a 2.3, ya que daría cobertura a más terminales.
- Disponer de un teléfono móvil con plan de datos (WiFi en su defecto) y mensajes de texto.
- Adquirir paquetes de mensajes de texto SMS a bajo costo que ofrecen los operadores de telefonía móvil o realizar recargas a la sim card del módulo según sea la conveniencia del usuario.

BIBLIOGRAFÍA

A. POZO RUIZ, A RIBEIRO, M.C. GARCÍA-ALEGRE, L. GARCÍA, D. GUINEA, F. SANDOVAL. Sistema de posicionamiento Global (GPS): Descripción, Análisis de errores, aplicaciones y futuro. Instituto de Automática Industrial, Universidad de Málaga. (Fecha de consulta, 17 de Marzo de 2014)

ARDUINO TEAM. Arduino Homepage. 2013. Página Web. . DISPONIBLE EN: <http://www.arduino.cc/>. (Fecha de consulta, 29 de Febrero de 2014)

COMM-CAR. DISPONIBLE EN: <http://www.comm-car.com/home.htm>. (Fecha de consulta, 30 de Enero de 2014)

COOKINGHACKS. Documentación y tutoriales Arduino 3G/GPS. DISPONIBLE EN: <http://www.cooking-hacks.com/index.php/documentation/tutorials/arduino-3g-gprs-gsm-gps>. (Fecha de consulta, 15 de Enero de 2014)

DONN FELKER. Android Application Development for Dummies. Wiley 2011. (Fecha de consulta, 10 de Mayo de 2014)

ESPAÑA Fernando, TACÁN Eliana. Tesis; Diseño e implementación de un sistema de control y monitoreo remoto para la iluminación de una vivienda a través de un teléfono móvil con sistema operativo Android y Board Arm Mini 2440. (Fecha de consulta, 19 de Agosto de 2014)

GIRONÉS, Jesús Tomás. El Gran Libro de Android. Marcombo SA. (Fecha de consulta, 28 Junio de 2014)

GALINDEZ Manuel, LEMUS José. Tesis; Diseño e implementación de un sistema móvil sobre la plataforma Android para el acceso fácil y cómodo la sistema de préstamos de la biblioteca de la Universidad Surcolombiana. (Fecha de consulta, 16 de Marzo de 2014)

GONZALEZ, O. Cómo utilizar Arduino con Android. 2010. DISPONIBLE EN: <http://blog.bricogeek.com/noticias/arduino/amarino-como-utilizar-arduino-con-android/#more>. (Fecha de consulta, 14 de Junio de 2014)

GUTIÉRREZ Mauricio, ZAMBRANO, José, GORDILLO Alexander. Tesis; Sistema de caja negra para monitoreo y registro de automotores tipo buseta. (Fecha de consulta, 26 de Febrero de 2014)

MILETTE GREG, Stroud Adam. Professional Android Sensor Programming. Wiley. 2012. (Fecha de consulta, 14 de Abril de 2014)

MURPHY MARK L. Beginning Android 2. Apress. 2010. (Fecha de consulta, 05 de Abril de 2014)

ORTIZ AYALA. Historia de las alarmas de auto. DISPONIBLE EN: <http://ortizayala.wordpress.com/2011/02/26/hello-world/>. (Fecha de consulta, 16 de Enero de 2014)

PARERA, A.M. Sistemas de Seguridad y Confort en Vehículos Automóviles. Boixareu Editores. 2000. (Fecha de consulta, 21 de Febrero de 2014)

ANEXOS

Anexo A.
Manual Comandos AT

Anexo B.
Regulador L7805CV

Anexo C.
Alarma Genius 2A

Anexo D

Manual de Usuario

Anexo E. Programación en Eclipse

Anexo F.

Programación Arduino