

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
FACULTAD DE INGENIERÍA  
INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

DISERTACIÓN PREVIA PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN  
SISTEMAS Y COMPUTACIÓN

SISTEMA VISUALIZADOR DE GEOPOSICIONAMIENTO A TRAVÉS DEL  
TIEMPO

PAUL SEBASTIAN MONTENEGRO ROSERO

DIRECTOR:

Ing. Alfredo Calderón

Quito, noviembre 2020

## CONTENIDO

ÍNDICE DE ILUSTRACIONES.....	84
ÍNDICE DE TABLAS.....	114
RESUMEN.....	115
ABSTRACT.....	126
DEDICATORIA.....	137
AGRADECIMIENTOS.....	148
CAPÍTULO 1. INTRODUCCIÓN.....	159
1.1. JUSTIFICACIÓN.....	159
1.2. PLANTEAMIENTO DEL PROBLEMA.....	169
1.3. ALCANCE.....	1740
1.4. OBJETIVO.....	1740
1.3.1. OBJETIVO GENERAL.....	1740
1.3.2. OBJETIVOS ESPECÍFICOS.....	1740
CAPÍTULO 2. FUNDAMENTOS TEÓRICOS.....	1811
2.1. ESTADO DEL ARTE.....	1811
2.2. HARDWARE LIBRE.....	2114
2.3. BASE DE DATOS.....	2417
2.4. DATA DE GEOPOSICIONAMIENTO.....	2518
2.5. TIEMPO REAL.....	2619
CAPÍTULO 3. HERRAMIENTAS POR USAR Y SU FUNCIONAMIENTO.....	2720
3.1. FUNCIONAMIENTO DE LA RED GPS.....	2720
3.2. ARDUINO Y PROTOTIPOS.....	2821
3.3. MÓDULO SIM808.....	3124
3.4. BASES DE DATOS, CÓMO USAR Y CUÁL USAR.....	3326
3.4.1. MySQL.....	3326
3.4.2. MARIADB.....	3326
3.4.3. CONCLUSIÓN.....	3528
3.5. METODOLOGÍA Y PROCESOS DE DESARROLLO DE SOFTWARE.....	3528
3.5.1. METODOLOGÍA DE CASCADA.....	3528
3.5.2. PROGRAMACIÓN EN TRES CAPAS.....	3629
CAPÍTULO 4. ANÁLISIS Y DISEÑO.....	3831
4.1. ANÁLISIS DEL SISTEMA.....	3932
4.2. DISEÑO DE HARDWARE.....	3932
4.3. DISEÑO DE LA BASE DE DATOS.....	4033

4.4. DISEÑO DE LA APLICACIÓN .....	4134
CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS .....	4134
5.1. DIAGRAMA GENERAL DE CASOS DE USO .....	4235
5.2. SEGUNDO NIVEL DE DIAGRAMA DE CASOS DE USO .....	4335
5.2.1. GESTIÓN DE USUARIOS .....	4336
5.2.2. GESTION DE POSICIONES .....	4336
5.3. CASOS DE USO A DETALLE.....	4437
5.3.1. CREAR USUARIO .....	4437
5.3.2. MODIFICAR USUARIO.....	4638
5.3.3. ELIMINAR USUARIO .....	4740
5.3.4. POSICIÓN ACTUAL .....	4840
5.3.5. HISTORIAL DE POSICIONES.....	4941
5.4. DIAGRAMAS DE SECUENCIA.....	5042
5.4.1. CREAR USUARIO .....	5043
5.4.2. MODIFICAR USUARIO.....	5144
5.4.3. ELIMINAR USUARIO.....	5245
5.4.4. POSICIÓN ACTUAL.....	5346
5.4.5. HISTORIAL DE POSICIONES.....	5447
5.5. DIAGRAMA DE ACTIVIDADES .....	5548
5.5. PRUEBAS.....	5649
CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES.....	5952
6.1. PRECIO ESTIMADO .....	5952
6.2. CONCLUSIONES .....	6053
6.3. RECOMENDACIONES .....	6154
BIBLIOGRAFÍA.....	6558
<a href="#">¡Error! Referencia de hipervínculo no válida.</a> <a href="#">ÍNDICE DE ILUSTRACIONES</a> <a href="#">¡Error! Marcador no definido.</a>	4

## ÍNDICE DE ILUSTRACIONES

<a href="#">Ilustración 1. Foto frontal de la placa Arduino Uno Fuente: Autoría Propia .....</a>	<a href="#">3024</a>
<a href="#">Ilustración 2. Foto lateral de la placa Arduino Uno Fuente: Autoría Propia .....</a>	<a href="#">3124</a>
<a href="#">Ilustración 3. Foto lateral del módulo Sim808 Fuente: Autoría Propia .....</a>	<a href="#">3225</a>
<a href="#">Ilustración 4. Foto frontal del módulo Sim808 Fuente: Autoría Propia .....</a>	<a href="#">3225</a>
<a href="#">Ilustración 5. Diseño en conjunto de Hardware y Software Fuente: Autoría Propia.....</a>	<a href="#">4033</a>
<a href="#">Ilustración 6. Diseño Conceptual de la Base de Datos Fuente: Autoría Propia .....</a>	<a href="#">4033</a>
<a href="#">Ilustración 7. Diagrama General de Casos de Uso Fuente: Autoría Propia .....</a>	<a href="#">4235</a>
<a href="#">Ilustración 8. Diagrama de Casos de Uso en Segundo Nivel de la funcionalidad: Gestión de Usuarios Fuente: Autoría Propia .....</a>	<a href="#">4336</a>
<a href="#">Ilustración 9. Diagrama de Casos de Uso en Segundo Nivel de la Funcionalidad: Gestión de Posiciones Fuente: Autoría Propia .....</a>	<a href="#">4437</a>
<a href="#">Ilustración 10. Relación entre el Administrador y el proceso Crear Usuario Fuente: Autoría Propia .....</a>	<a href="#">4538</a>
<a href="#">Ilustración 11. Relación entre el Administrador y el proceso Modificar Usuario Fuente: Autoría Propia .....</a>	<a href="#">4639</a>
<a href="#">Ilustración 12. Relación entre el Administrador y el proceso Eliminar Usuario Fuente: Autoría Propia .....</a>	<a href="#">4740</a>
<a href="#">Ilustración 13. Relación entre el Administrador y el Cliente con el proceso Posición Actual Fuente: Autoría Propia .....</a>	<a href="#">4841</a>
<a href="#">Ilustración 14. Relación entre el Administrador y el Cliente con el proceso Historial de Posiciones Fuente: Autoría Propia .....</a>	<a href="#">4942</a>
<a href="#">Ilustración 15. Diagrama de Secuencia del proceso Crear Usuario Fuente: Autoría Propia... ..</a>	<a href="#">5144</a>
<a href="#">Ilustración 16. Diagrama de Secuencia del proceso Modificar Usuario Fuente: Autoría Propia.....</a>	<a href="#">5245</a>
<a href="#">Ilustración 17. Diagrama de Secuencia del proceso Eliminar Usuario Fuente: Autoría Propia.....</a>	<a href="#">5346</a>
<a href="#">Ilustración 18. Diagrama de Secuencia del proceso Posición Actual Fuente: Autoría Propia .....</a>	<a href="#">5447</a>
<a href="#">Ilustración 19. Diagrama de Secuencia del proceso Historial de Posiciones Fuente: Autoría Propia .....</a>	<a href="#">5548</a>
<a href="#">Ilustración 20. Diagrama de Actividades Fuente: Autoría Propia .....</a>	<a href="#">5649</a>
<a href="#">¡Error! Referencia de hipervínculo no válida. Ilustración 1. Foto frontal de la placa Arduino Uno</a>	

## ÍNDICE DE TABLAS

<a href="#">Tabla 1. Pruebas de Ejecución .....</a>	<a href="#">5750</a>
<a href="#">Tabla 2. Pruebas de Reacción.....</a>	<a href="#">5750</a>
<a href="#">Tabla 3. Pruebas de Ambiente .....</a>	<a href="#">5851</a>
<a href="#">Tabla 4. Pruebas de Validación .....</a>	<a href="#">5952</a>
<a href="#">Tabla 5. Tabla de Costos.....</a>	<a href="#">6053</a>
<a href="#">¡Error! Referencia de hipervínculo no válida. Tabla 1. Pruebas de Ejecución ¡Error! Referencia de hipervínculo no válida.</a>	<a href="#">51</a>

## RESUMEN

El presente trabajo de disertación busca desarrollar un sistema que permita llevar seguimiento del geoposicionamiento de cualquier objeto al que se lo desee aplicar. Además de esto, el sistema será capaz de llevar un historial, a través del tiempo, de la posición de dicho objeto, lo mismo que será almacenado en una base de datos que se va actualizando constantemente.

Para la implementación del sistema, el uso de los módulos “Arduino” será indispensable ya que serán estos los que permitan tener la información de geoposicionamiento.

El trabajo con base de datos también es uno de los pilares fundamentales del presente trabajo, ya que aquí es donde se guarda toda la información que ~~se~~ reflejase refleja dentro de la aplicación.

Para el sistema virtual es necesario el uso de una metodología ágil para el desarrollo de software. En este caso es la metodología denominada “Cascada”, que nos permite tener una retroalimentación constante y controlar de manera eficiente como se va desarrollando el trabajo.

Para comprobar que el sistema funciona correctamente se han realizado varias pruebas a lo largo de todo el proceso, entre las cuales existe una verificación, que los datos obtenidos sean correctos y que de igual manera se reflejen en pantalla.

Este trabajo permite tener un conocimiento más abundante de los que son bases de datos, geoposicionamiento y hardware libre. Conocimientos que serán de mucha utilidad para el desarrollo de la carrera profesional de un ingeniero en sistemas y computación.

## ABSTRACT

The present dissertation aims to develop a system that will allow the monitoring of the geopositioning of any object to which it is desired to apply it. In addition, the system will be able to keep a history, through time, of the position of such object, which will be stored in a database that is constantly updated.

For the implementation of the system, the use of the "Arduino" modules will be indispensable since they will be the ones that allow having the geopositioning information.

Working with databases is also one of the fundamental pillars of this work, since this is where all the information that will be reflected within the application to be developed will be stored.

For the virtual system it will be necessary to use an agile methodology for software development. In this case it will be the methodology called "Cascade" that allows us to have a constant feedback and to control in an efficient way how the work is being developed.

In order to verify that the system works correctly, several tests have been carried out throughout the process, among which it was verified that the data obtained are correct and that they are also reflected on the screen.

The realization of this work allowed to have a more abundant knowledge of what are databases, geopositioning and free hardware. Knowledge that will be very useful for the development of the professional career of a systems and computer engineer.

## DEDICATORIA

Este trabajo de disertación va dedica a todas aquellas personas que han estado a mi lado durante todos estos años.

A mis padres, quienes han sido el pilar fundamental en mi vida, sin ellos nada de esto hubiera sido posible. Fueron ellos quienes supieron darme la mejor vida posible y de la manera más acertada. Por su gran esfuerzo y por saber motivarme siempre que lo necesitaba. Si no he desistido, es gracias a su gran labor y al constante amor que supieron darme.

A mi hermano mayor por enseñarme gran parte de lo que hoy en día se y ayudarme en mi formación. De la misma forma a mi hermano menor, quien siempre fue la suficiente motivación para ser alguien mejor y así ser un ejemplo a seguir para él.

A mis amigos, tanto a los que están cerca como a los que están lejos. De alguna u otra forma todos ellos han influido para lograr lo que he conseguido.

Para Milena, una mujer muy especial para mí, quien me recordó las cosas por las que debo luchar y los sueños que tengo. Quien me apoyó en uno de los momentos más difíciles de mi vida y me dio la motivación suficiente para poder culminar con este trabajo.

## AGRADECIMIENTOS

A mis padres, ya que gracias a ellos he logrado conseguir todo lo que me he propuesto. De la misma forma, a mis hermanos, su motivación y acompañamiento han sido fundamentales y de vital importancia.

A mis amigos, los cuales han formado gran parte de mi vida y han sabido estar siempre a mi lado.

A mi director: Alfredo Calderón y a mis dos revisores: Damián Nicolalde y Susana Masapanta. Gracias a ellos este trabajo logró llevarse a cabo de la mejor manera.

A mi mejor amigo: Sebastian Erazo. A través de una sana competencia hemos sabido sacar la mejor parte de nosotros. Su ayuda ha sido algo muy importante para poder superarme a mí mismo.

## CAPÍTULO 1. INTRODUCCIÓN

### 1.1. JUSTIFICACIÓN

En la actualidad, el saber dónde se encuentran ubicados ciertos objetos es de mucha utilidad para personas u organizaciones. Pero la necesidad no solo se limita a saber dónde se encuentra, sino también el tiempo de dicha ubicación.

Extraviar o perder bienes es un tema común, que afecta día a día a la población mundial y la falta de un sistema confiable y de costo accesible que ayude a localizarlas es un factor que contribuye a este problema.

El área de aplicación para sistema de geoposicionamiento es extensa, por lo que puede ser usado básicamente por cualquier organización que necesite saber la ubicación de objetos que necesiten localizar y además llevar un registro histórico del tiempo.

### 1.2. PLANTEAMIENTO DEL PROBLEMA

La tecnología ha intentado mejorar día a día para beneficio de las personas. Sin embargo, la propuesta de un visualizador de geoposicionamiento que lleve registro histórico del tiempo es algo que ayudaría en la historia del bien.

Las organizaciones y las personas tienden a extraviar sus bienes, los cuales no pueden ser fácilmente localizados.

Actualmente, muchos de los sistemas de geoposicionamiento trabajan únicamente en base a GPS que se encuentran previamente instalados. Este es el caso de celulares, vehículos, computadoras, etc. Un sistema de este tipo, que pueda ser instalado, es algo necesario.

Una base de datos para guardar registros históricos del geoposicionamiento es algo que en la actualidad ayudaría en lo anteriormente dicho. Múltiples organizaciones necesitan de un sistema que permita llevar control de su logística y de su personal.

Por último, el contar con un sistema de geoposicionamiento que sea confiable, y que tengas las funciones mencionadas anteriormente, viene a ser algo costoso,

de ~~hecho~~hecho, son escasos aquellos sistemas “GPS” que nos brinden resultados precisos, a precios cómodos.

Por lo tanto, se propone realizar las investigaciones necesarias para la elaboración de un sistema de geoposicionamiento que sea confiable y que pueda satisfacer las necesidades planteadas anteriormente, a un costo que sea accesible y que optimice recursos. Es por esto, por lo que resulta relevante visibilizar esta práctica, analizar esta temática y sentar las bases para que abra el camino a otras modalidades y campos de investigación.

### 1.3. ALCANCE

En el presente trabajo investigativo se pretende realizar un análisis de lo que viene a ser “GPS”, Arduino y bases de datos para poder desarrollar un sistema visualizador de geoposicionamiento a través del tiempo. De la misma forma, realizar esquemas y modelos que nos apoyen, y en base a los cuales podamos guiarnos, será algo indispensable. Por lo tanto, se han utilizado estudios descriptivos.

Para el desarrollo de la propuesta, se utiliza un prototipo que junte lo físico con lo virtual, en dónde se vea reflejado aquellas ventajas que presenta el sistema. Dentro del prototipo se aprecia que los componentes usados trabajan en conjunto de una manera eficiente y satisfactoria.

### 1.4. OBJETIVO

#### 1.3.1. OBJETIVO GENERAL

Elaboración de un sistema visualizador de geoposicionamiento en tiempo real, el cual va a llevar un registro histórico de la latitud, longitud, altura y tiempo de cierta ubicación.

#### 1.3.2. OBJETIVOS ESPECÍFICOS

- Analizar el estado del arte de las tecnologías: Hardware Libre, Software Libre, GPS y Bases de Datos, con el fin de trabajar y relacionar las mismas de la manera más adecuada.

- Desarrollar un prototipo funcional que permita gestionar toda la información o data que resulte del geoposicionamiento en tiempo real.
- Generar un estimado de costo del prototipo para demostrar que puede ser accesible por cualquier tipo de organización que necesite de este servicio.

## CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

### 2.1. ESTADO DEL ARTE

Se sabe que la armada estadounidense aplicó esta tecnología de navegación con la ayuda de satélites para proveer a los sistemas de navegación de sus flotas apreciaciones de posiciones actualizadas y precisas. De esta manera surgió el sistema TRANSIT, que quedó en funcionamiento en 1964, y hacia 1967 estuvo disponible, inclusive comercialmente.

En ese tiempo se podía obtener actualizaciones de posición cada 40 minutos y el observador que se utilizaba para lograrlo debía permanecer casi estático para poder obtener información útil.

Posteriormente, en la misma década de los 90's y gracias al desarrollo de los relojes atómicos, se logró diseñar una constelación de satélites. De forma que cada uno de ellos llevaba incorporado uno de estos relojes. Todo se encontraba sincronizado con base en una referencia de tiempo determinado.

En 1973 se fusionaron los programas de la Armada y el de la Fuerza Aérea de los Estados Unidos de América (este último se basaba en una técnica de transmisión codificada que administraba datos precisos usando una señal modulada a través de un código de PRN (Pseudo-Random Noise: ruido pseudoaleatoria), en lo que se dio a conocer como Navigation Technology Program (programa de tecnología de navegación), que posteriormente se renombraría como NAVSTAR GPS.

Entre los años 1978 y 1985 se desarrollaron y lanzaron al espacio once satélites prototipo experimentales NAVSTAR. Tiempo después se lanzarían otras generaciones de satélites los cuales completarían la constelación actual, a la que

se declaró con «capacidad operacional inicial» en diciembre de 1993 y con «capacidad operacional total» en abril de 1995.

En 2009, el gobierno de los Estados Unidos puso a disposición el servicio de determinación de la posición para apoyar las necesidades que tenía la OACI (Organización de Aviación Civil Internacional), y ésta aceptó el ofrecimiento.

Cada satélite GPS emite continuamente un mensaje de navegación a 50 bits por segundo en una frecuencia transportadora de microondas de aproximadamente 1.600 MHz. Las redes Wi-Fi, en comparación funcionan a alrededor de 5000 MHz y 2400 MHz y la radio FM se emite a entre 87,5 y 108,0 MHz. Más concretamente, todos los satélites emiten a 1575,42 MHz (esta es la señal L1) y 1227,6 MHz (la señal L2).

Gracias al reloj atómico que se encuentra en el satélite, la señal GPS proporciona la “hora de la semana” precisa, el número de semana GPS y un informe de estado del satélite. De esta manera puede reducirse si es defectuoso. Cada transmisión lleva 1500 bits de datos codificados y dura 30 segundos. Esta pequeña cantidad de datos se encuentra codificada con una secuencia pseudoaleatoria (PRN) de alta velocidad que viene a ser diferente para cada satélite. Ya que los receptores GPS conocen los códigos PRN de cada satélite pueden decodificar la señal y la pueden distinguir entre diferentes satélites.

El reloj atómico del satélite controla las transmisiones, las cuales son cronometradas para empezar de forma precisa en el minuto y en el medio minuto. La primera parte de la señal GPS indica la relación entre el reloj del satélite y la hora GPS. La siguiente serie de datos proporciona al receptor información de órbita precisa del satélite. De esta manera es como ha progresado la red GPS.

Existen múltiples aplicaciones e ideas para la que se conoce GPS. Entre ellas las más conocidas como: El uso de mapas y localización, rastreo de dispositivos electrónicos, navegación y pilotos automáticos en diferentes tipos de vehículos, etc. Por otra parte, podemos observar que la milicia es quien inicia con los conceptos para esta tecnología a la cual se le ha sacado mucho provecho.

Adicional a esta información se puede agregar que la empresa Apple se encuentra desarrollando una tecnología que funciona a través de GPS, denominada "Air Tags". El concepto de esta tecnología es el siguiente: el Air Tag es un dispositivo tecnológico a manera de un llavero como cualquier otro, el cual podemos ponerlo en cualquier objeto que deseemos ubicar. El Air Tag a través de la red GPS manda su ubicación exacta al usuario a través de una aplicación. De esta forma se puede rastrear el objeto, sin importar su posición.

Como podemos apreciar, la tecnología GPS tiene múltiples aplicaciones y se encuentra en un constante avance tecnológico, al cual se le puede sacar mucho provecho, usándolo de la manera adecuada.

Como ejemplo podemos tomar en cuenta grandes empresas que necesitan intrínsecamente de la tecnología GPS. Tal es el caso de Google, que a través de sus distintos aplicativos nos permite posicionarnos y llevar un registro de nuestros dispositivos. De la misma manera podemos nombrar a la empresa Tesla, la cual está innovando en el campo automotriz con sus vehículos inteligentes que además de ser completamente tecnológicos, brindan la capacidad de un piloto automático, tarea que sin un GPS sería imposible.

Inclusive podemos remontarnos a empresas más pequeñas. Dentro de la ciudad de Tulcán, en la provincia del Carchi, reside una organización dedicada a la seguridad llamada "Centinela del Norte", la misma que se encarga de dar el servicio de salvaguardar bienes; esta empresa necesariamente debe usar el geoposicionamiento para poder dar seguimiento a todo aquellos que necesiten proteger. En la ciudad de Quito una aplicación denominada "EasyTaxi" se encarga de dar el servicio de transporte, tarea que necesita de la tecnología GPS, para poder ubicar a los clientes y llevarlos al lugar que necesiten.

Pero qué es lo que diferencia a nuestro sistema de todo aquello que hemos nombrado anteriormente, cuál es el incremento o plus que se le va a dar. Para empezar, los complejos sistemas que usan las empresas tienden a ser costosos, lo que no facilita que pequeñas y medianas empresas puedan adquirir estos servicios. Nuestro sistema es una alternativa que permite una inversión de bajo costo, tomando en cuenta que cualquier empresa con un departamento de tecnología podría desarrollar sin problema alguno su versión de este prototipo.

Esto gracias a que el proyecto va a ser desarrollado con software y hardware libre, lo cual da camino a que se pueda ajustar a las necesidades de cada cliente, siendo el mismo capaz de modificarlo y agregarle funcionalidades de acuerdo con el fin que vaya a tener.

Por esto, el sistema [estaestá](#) dirigido a cualquier empresa u organización que desee sacarle el máximo provecho a la tecnología GPS para aumentar su competitividad.

## 2.2. HARWDARE LIBRE

Se ha planteado la utilización de hardware libre, debido a la fácil manipulación que tiene el mismo. La documentación del hardware libre está completamente disponible para el público, lo que nos permite conocer todo lo necesario para el desarrollo de cualquier prototipo. Además, las posibilidades que ofrece el hardware libre son mucho más amplias, ya que no existe un limitante para el desarrollo. De esta manera se puede modificar y agregar diferentes características a aquella tecnología que estemos ampliando, dándonos un campo de acción mucho más amplio.

También conocido como hardware de código abierto o electrónica libre, se denomina hardware libre a aquellos dispositivos o aparatos electrónicos cuyas especificaciones y diagramas esquemáticos son de acceso general y público, ya sea de forma gratuita o bajo algún tipo de pago. La misma filosofía que se aplica dentro de lo que es software libre se aplica al hardware libre, en conjunto forman parte de la nombrada cultura libre (corriente de pensamiento que alienta la libertad en la producción y modificación de diferentes tipos de trabajo, promoviendo el contenido libre para distribuir o modificar trabajos y obras creativas; en sí esta cultura se opone a los derechos de autor, ya que se cree que obstaculiza la libre creatividad).

La Declaración de principios del hardware libre (HL) lo define como “aquel hardware cuyo diseño se hace disponible públicamente para que cualquier persona lo pueda estudiar, modificar, distribuir, materializar y vender, tanto el original como otros objetos basados en ese diseño”. Así el hardware libre utiliza elementos y materiales inmediatamente disponibles, procesos estandarizados,

infraestructura abierta, contenido no restringido y herramientas de diseño libres brindando a los/as usuarios/as la posibilidad de controlar su tecnología mientras comparten el conocimiento y alientan el comercio a través del intercambio abierto de diseños. (Lazalde, Torres, & Vila, 2015).

A partir de estos principios, el uso de HL ofrece distintas ventajas a una sociedad y en particular a sus sectores más innovadores. Entre ellas destaca la mejora de la sostenibilidad y soberanía tecnológicas, así como la adopción de las citadas libertades a imagen y semejanza de los principios del software libre. Además, el fortalecimiento de ecosistemas de innovación y producción basados en los principios de HL y análogos favorece el empoderamiento de las comunidades y la utilidad social y apropiabilidad en el uso de las tecnologías, así como su valor educativo al insertarse en procesos de aprendizaje y de trabajo necesariamente colaborativo. Aunque en términos de costes, no exista un abaratamiento como tal en el desarrollo inicial, la democratización del acceso a sus fuentes sí merma la capacidad de concentración del lucro en cuanto a la tecnología. Un ecosistema productivo con estas características puede responder mejor a las necesidades reales de sus comunidades de referencia, dando eficacia a distintos modelos posibles de cogobierno, y acelerar el desarrollo endógeno, reduciendo la dependencia tecnológica de muchos Estados en el capitalismo cognitivo globalizado. (Lazalde, Torres, & Vila, 2015).

González et al. (2003) afirma:

“El software libre ofrece al usuario cuatro libertades: libertad de uso, de estudio y modificación, de distribución, y de redistribución de las mejoras”.

El hardware abierto o libre toma estas mismas ideas para aplicarlas en su campo. Es una propuesta casi tan antigua como la del software libre, sin embargo, su empleo no es tan directo. Compartir diseños hardware es más complicado. No hay una definición exacta (se pueden encontrar referencias a distintos artículos en). Incluso el propio Richard Stallman afirma que las ideas del software libre se pueden aplicar a los ficheros necesarios para su diseño y especificación (esquemas, PCB, etc.), pero no al circuito físico en sí. (González, González, & Gómez, 2003).

El hardware libre presenta una clasificación la cual se ha determinado según la naturaleza del mismo. Dividiéndolo así en dos: Hardware Estático y Hardware Reconfigurable.

**Hardware estático:** Conjunto de elementos materiales de los sistemas electrónicos. Tiene una existencia física (se puede “tocar”). Esta propiedad no la tiene el software. (González, González, & Gómez, 2003).

**Hardware reconfigurable:** Es el que se describe mediante un lenguaje HDL, (Hardware Description Language, lenguaje de descripción hardware) y que permite especificar con todo detalle su estructura y funcionalidad. A partir de este código se generan unos ficheros de configuración (González, González, & Gómez, 2003).

De cierta manera, existen principios para que cierto hardware se considere libre. Estos principios fueron establecidos por la Open Source Hardware Association, los mismos que se resumen de la siguiente manera:

- 1. Documentación:** El hardware debe ser liberado al público con toda su documentación y debe ser posible su modificación.
- 2. Alcance:** En la documentación debe haber constancia clara acerca de ~~cuál~~ cuál parte del diseño se publica bajo licencia.
- 3. Software Necesario:** Si el diseño necesita de una licencia de software, se debe cumplir unos estándares de documentación suficiente y ser publicada bajo el carácter de una licencia de código abierto aprobada por OSI.
- 4. Obras Derivadas:** La licencia debe permitir trabajos derivados y la modificación sin restricción alguna del diseño. Así como el uso, venta, distribución y fabricación de productos creados a partir del diseño original.
- 5. Redistribución libre:** La licencia no debe restringir la venta o distribución libre del proyecto por parte de terceros. De la misma manera no se puede ejercer ningún derecho sobre obras derivadas.
- 6. Atribución:** La licencia debe nombrar avisos de copyright y documentos derivados asociados a los dispositivos. Asimismo, se debe hacer mención del diseñador.

**7. No discriminatoria:** No se debe discriminar a ningún grupo o persona.

**8. No discriminación en función de la finalidad perseguida:** La licencia no debe tener restricción debido al uso que se le va a dar a la misma.

**9. Distribución de la licencia:** No se debe solicitar permisos adicionales para la distribución de la licencia.

**10. La licencia no debe ser específica de un producto:** Esta licencia se hace extensiva de acuerdo con los derechos de productos derivados.

**11. La licencia no debe restringir otro hardware o software:** No se puede poner restricciones a implementaciones de cualquier tipo que se le deseen hacer al diseño.

**12. La licencia debe ser tecnológicamente neutral:** El diseño o la licencia no puede basarse en una tecnología específica, componente o parte, interfaz o material para su uso. (Andrades, 2013).

Las características mencionadas garantizan que nuestro proyecto cumpla con los requerimientos que nos hemos planteado, ya que básicamente se plantea que no puede existir restricciones en el uso del hardware. La idea es que se presente un prototipo funcional, al cual se lo pueda modificar al antojo de quien lo use, sacando el máximo provecho y fomentando la incrementabilidad del mismo.

### 2.3. BASE DE DATOS

Para poder gestionar y guardar toda la información necesitamos de una base de datos confiable y segura, que cumpla con los requisitos de software libre, con el objetivo de que la misma siga con el hilo del proyecto.

Un sistema gestor de bases de datos (SGDB) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos contiene información relevante para una empresa. El objetivo principal de un SGDB es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La

gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Dado que la información es tan importante en la mayoría de las organizaciones, los científicos informáticos han desarrollado un amplio conjunto de conceptos y técnicas para la gestión de los datos. (Silberschatz, Korth, & Sudarshan, 2002).

Las aplicaciones de las bases de datos se encuentran prácticamente en cualquier organismo o entidad que necesite del manejo de grandes cantidades de información y gracias a su gran versatilidad, en muchos gestores de bases de datos se puede guardar información de todo tipo: números, texto, fechas, imágenes, etc.

Vale la pena recalcar que existen bases de datos relacionales (usan tablas para relacionar sus datos) y no relacionales (no hay una relación de los datos en sí). En este caso se ha decidido optar por una base de datos relacional, ya que existen muchas más bases de datos de este tipo que ya han pasado por múltiples versiones que garantizan su funcionamiento. Además, la curva de aprendizaje con una base de datos relacional es menor.

#### 2.4. DATA DE GEOPOSICIONAMIENTO

En base a esta información va a trabajar todo el sistema, por lo tanto es importante conocer un mejor enfoque un punto de vista de la misma. Los inicios de las primeras redes GPS se remontan a años atrás, donde se puede apreciar que los primeros usos que se le dieron fueron básicamente militares. Con el pasar del tiempo se explotó esta tecnología de diferentes maneras, sacándole el provecho hasta para las tareas más cotidianas.

Como se ha comentado anteriormente, GPS significa "Global Positioning System", o lo que es lo mismo "Sistema de Posicionamiento Global". Este sistema, puesto en funcionamiento desde 1973, se desarrolló a partir de los satélites de la constelación NAVSTAR (Navigation Satellite Timing And Ranging), fue desarrollado por el Departamento de Defensa de los Estados

Unidos (DoD), y se lanzó el primer satélite el 22 de febrero de 1978. Este sistema fue desarrollado para mejorar el sistema, de medición distancias DOPPLER, TRANSIT en servicio civil desde 1967. (Peñafiel & Zayas, 2001).

Por razones militares, necesitaban un sistema que tuviese cobertura global, a cualquier hora del día y en cualquier medio funcionase, ya sea, mar, aire o tierra. Se tenía la necesidad, además, que el sistema fuese pasivo, es decir, el usuario no tenía que emitir señal de ningún tipo para no ser delatada su posición. El sistema está pensado para sustituir todos los sistemas de precisión media de navegación civil, Decca, Loran C, Omega, Transit, Tacan, ILS, Radiofaros, etc. Las precisiones esperadas en Navegación se obtienen solamente utilizando un receptor. Si utilizamos dos receptores observando simultáneamente, las precisiones que se pueden alcanzar son ya de 5mm. En este ámbito estaremos hablando de geodesia y de topografía. (Peñafiel & Zayas, 2001).

El sistema se ha declarado oficialmente operativo, por el DoD, en enero de 1994. La constelación proyectada en principio consistente en 8 satélites por cada plano orbital de los tres previstos, fue modificado por motivos presupuestarios, siendo en la actualidad seis órbitas casi circulares con cuatro satélites por cada una. La altitud de los satélites es de unos 20.180 km. cuando están en el Zenit del lugar. Completando cada uno de ellos dos vueltas por cada rotación de 360° a la tierra, es decir el periodo es de 12 horas sidéreas por lo que la configuración de un instante se repite el día anterior debido a la aceleración de las fijas o diferentes entre día sidéreo y el día solar medio. Los seis planos orbitales se suelen definir con las letras A, B, C, D, E, F y dentro de cada órbita cada uno de los satélites con los números 1,2,3,4. Hay otros sistemas de identificarlo, como puede ser por su PRN característico, por número de catálogo de la NASA, fecha de lanzamiento, etc. Además de los satélites lógicamente se ha de disponer de un receptor de la señal enviada por los satélites en tierra, y de algún sistema de control sobre ellos. (Peñafiel & Zayas, 2001).

## 2.5. TIEMPO REAL

Un sistema en tiempo real es aquel sistema informático que interacciona repetidamente con su entorno físico. Además de esto, responde a diferentes

estímulos que percibe dentro de este mismo entorno físico dentro de un plazo de tiempo determinado.

Para que un sistema en tiempo real funcione correctamente, no basta simplemente con que las acciones que realice sean correctas, sino que también deben ejecutarse dentro del intervalo de tiempo especificado. (De La Puente, 2007).

Es importante hablar de un sistema en tiempo real ya que el sistema que se va a presentar va a funcionar de esta manera. Tomando en cuenta que se va a poder verificar la posición de cualquier objeto en el momento que deseemos, la misma posición se va a actualizar a menudo en tiempo real y va a quedar un historial de la determinada posición en la hora especificada.

## CAPÍTULO 3. HERRAMIENTAS POR USAR Y SU FUNCIONAMIENTO

### 3.1. FUNCIONAMIENTO DE LA RED GPS

Se ha hablado históricamente de que es GPS y como ha logrado trascender hasta nuestros tiempos. Pero también es importante saber ~~cómo~~ funciona la red de geoposicionamiento y cómo esta obtiene los datos que van a ser usados por distintos usuarios. Esto nos permitirá tener un mejor enfoque, el cual podremos y debemos aplicar a nuestro prototipo.

El sistema GPS tiene por objetivo calcular la posición de un punto cualquiera en un espacio de coordenadas (x, y, z), partiendo del cálculo de las distancias del punto a un mínimo de tres satélites cuya localización es conocida. La distancia entre el usuario (receptor GPS) y un satélite se mide multiplicando el tiempo de vuelo de la señal emitida desde el satélite por su velocidad de propagación. Para medir el tiempo de vuelo de la señal de radio es necesario que los relojes de los satélites y de los receptores estén sincronizados, pues deben generar simultáneamente el mismo código. Ahora bien, mientras los relojes de los satélites son muy precisos los de los receptores son osciladores de cuarzo de bajo coste y por tanto imprecisos. Las distancias con errores debidos al sincronismo se denominan pseudodistancias. La desviación en los relojes de los

receptores añade una incógnita más que hace necesario un mínimo de cuatro satélites para estimar correctamente las posiciones. (Pozo, y otros, 2000).

En el cálculo de las pseudodistancias hay que tener en cuenta que las señales GPS son muy débiles y se hallan inmersas en el ruido de fondo inherente al planeta en la banda de radio. Este ruido natural está formado por una serie de pulsos aleatorios, lo que motiva la generación de un código pseudo-aleatorio artificial por los receptores GPS como patrón de fluctuaciones. En cada instante un satélite transmite una señal con el mismo patrón que la serie pseudo-aleatoria generada por el receptor. En base a esta sincronización, el receptor calcula la distancia realizando un desplazamiento temporal de su código pseudo-aleatorio hasta lograr la coincidencia con el código recibido; este desplazamiento corresponde al tiempo de vuelo de la señal. Este proceso se realiza de forma automática, continua e instantánea en cada receptor. (Pozo, y otros, 2000).

La utilización de estos códigos pseudo-aleatorios permite el control de acceso al sistema de satélites, de forma que en situaciones conflictivas se podría cambiar el código, obligando a todos los satélites a utilizar una banda de frecuencia única sin interferencias pues cada satélite posee un código GPS propio. Aunque la velocidad de los satélites es elevada (4 Km./s), la posición instantánea de los mismos puede estimarse con un error inferior a varios metros en base a una predicción sobre las posiciones anteriores en un período de 24 a 48 horas. Las estaciones terrestres revisan periódicamente los relojes atómicos de los satélites, dos de cesio y dos de rubidio, enviando las efemérides y las correcciones de los relojes, ya que la precisión de los relojes y la estabilidad de la trayectoria de los satélites son claves en el funcionamiento del sistema GPS. (Pozo, y otros, 2000).

### 3.2. ARDUINO Y PROTOTIPOS

Pasando a la parte de hardware, se ha decidido hacer uso de las placas conocidas como Arduino, en este caso el modelo "Arduino Uno". Vale la pena decir que esta tecnología es de carácter libre y que afortunadamente se ha podido trabajar con la misma en distintas ocasiones previas a este proyecto, por lo que el desenvolverse con la misma no vendría a ser un problema.

Las posibilidades que ofrece Arduino para el desarrollo de cualquier modelo y prototipo son grandes. La versatilidad de esta minicomputadora nos permita dejarnos llevar por la imaginación y realizar lo que se desee. De la misma manera el realizar un sistema GPS es completamente posible con la ayuda de este componente. Por estas razones y las que se exponen a continuación se ha decidido usar Arduino.

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede “sentir” el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el “Arduino Programming Language” y el “Arduino Development Environment”. Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador. Las placas se pueden ensamblar a mano o encargadas preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades. (Rafael, 2009).

- Barato: Las placas Arduino son de un bajo costo. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de 50\$.
- Multiplataforma: El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- Entorno de programación simple y claro: El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.

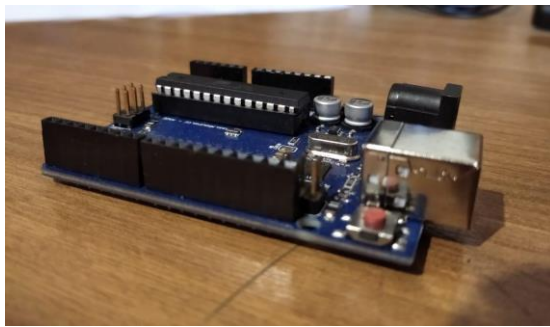
- Código abierto y software extensible: El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados.
- Código abierto y hardware extensible: El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero. (Rafael, 2009).

Arduino se basa en crear diseños, no hablar sobre ellos. Es una búsqueda constante de métodos más rápidos y eficaces de crear mejores prototipos. Se han explorado muchas técnicas de creación de prototipos y desarrollado métodos de pensar con nuestras manos. La ingeniería clásica se basa en un proceso estricto de llegar al punto B desde el punto A; Arduino se deleita en la posibilidad de perderse en el camino y, en su lugar encontrar el punto C. (Banzi & Shiloh, 2016).

La creación de prototipos es la esencia de la filosofía de Arduino: hacemos cosas y creamos objetos que interactúan con otros objetos, persona y redes. El esfuerzo es por encontrar un método más simple y más rápido para crear prototipos, de la forma más barata posible. Son muchos los principiantes que se acercan por primera vez a la electrónica que piensan que tienen que crearlo todo desde cero. Es una pérdida de energía: lo que necesita es poder confirmar rápidamente que algo está funcionando para poder motivarse y seguir adelante o, quizá, incluso motivar a alguien más para que le dé a cambio un montón de dinero para crearlo. Es la razón por la que se ha desarrollado la “creación oportunista de prototipos”: ¿por qué perder tiempo y energía en crear algo desde cero, un proceso que requiere mucho tiempo y conocimientos técnicos, cuando podemos usar dispositivos ya creados y sabotearlos para poder aprovecharnos del duro trabajo realizado por grandes empresas y buenos ingenieros? (Banzi & Shiloh, 2016). [Véase las ilustraciones 1 y 2.](#)



*Ilustración 1. Foto frontal de la placa Arduino Uno  
Fuente: Autoría Propia*



*Ilustración 2. Foto lateral de la placa Arduino Uno  
Fuente: Autoría Propia*

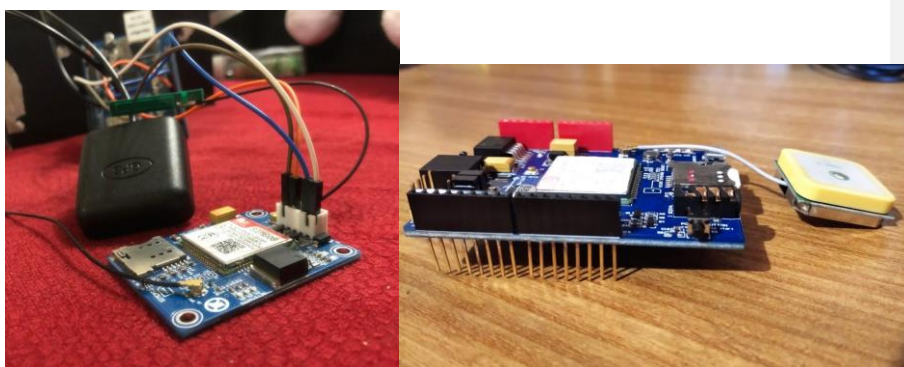
### 3.3. MÓDULOPLACA SIM808

La placa Arduino necesita de complementos para poder trabajar y lograr proyectos innovadores. En este caso el complemento que se va a usar es el módulo~~la placa~~el module sim808, que viene a convertirse en una parte fundamental del proyecto. A través de este módulo~~la placa~~el module se va a obtener toda la data de geoposicionamiento y es la placa Arduino quien la va a gestionar y enviar a la computadora.

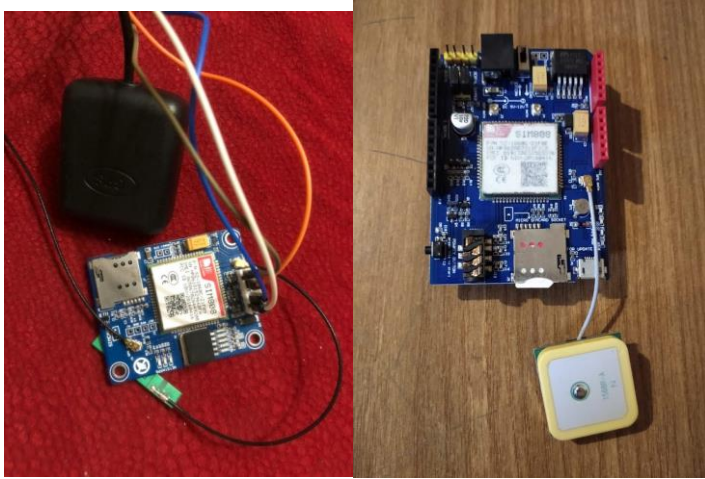
Este módulo~~la placa~~el module se encuentra basado en el chip SIMCOM y nos facilita las funcionalidades de envío y recepción de datos GSM/GPRS. De la misma manera nos ayuda con las funcionalidades de la tecnología GPS de navegación por satélite. Gracias a este módulo~~la placa~~el module y añadiendo una

tarjeta SIM, se tendrá la capacidad de conectarse a la red móvil, enviar y recibir llamadas y/o mensajes, conectarse a Internet y conocer las coordenadas exactas con tiempo UTC (Tiempo Universal Coordinado). (Macho, 2018).

Hablando de manera física, ~~el módulo placa~~ ~~el módulo~~ sim808 viene equipado con dos antenas que deben ser conectadas al mismo. Una nos ayudará a conectarnos a la red celular GSM y la otra nos permitirá conectarnos a la Red GPS. Es aconsejable que las antenas se encuentren en un lugar abierto para que las conexiones sean más confiables. En el caso de la antena GPS esta toma un pequeño tiempo en conectarse a los satélites desde que se la activa a través del IDE de Arduino con los diferentes comandos ATM que deben usarse. [Véase las ilustraciones 3 y 4.](#)



*Ilustración 3. Foto lateral del módulo ~~la Placa~~ Sim808  
Fuente: Autoría Propia*



*Ilustración 4. Foto frontal del módulo ~~la placa~~ Sim808  
Fuente: Autoría Propia*

### 3.4. BASES DE DATOS, CÓMO USAR Y CUÁL USAR

#### 3.4.1. MySQL

Al hablar de bases de datos, existen muchas, las cuales nos presentan diferentes características y posibilidades, entre las opciones se encuentra MySQL la cual es gratuita y libre, por ende, sigue con el hilo de este trabajo investigativo.

MySQL presenta las siguientes características:

- Desarrollado en C/C++.
- Sus ejecutables están disponibles para aproximadamente diecinueve plataformas diferentes.
- La API de la base de datos se encuentra disponible en Python, Perl, Eiffel, C, C++, PHP, Ruby, TCL y Java.
- Los equipos con múltiples procesadores se verán beneficiados ya que MySQL está optimizado para los mismos.
- Tiene destacable velocidad de respuesta.
- Se puede utilizar incrustado en aplicaciones o como cliente-servidor.
- Tiene disponible un amplio conjunto de datos.
- Al soportar múltiples métodos de almacenamiento de las tablas permite prestaciones y rendimiento diferentes para poder optimizar en cada caso.

- Su administración está basada en usuarios y privilegios.
- Han existido casos en los que inclusive se maneja sesenta mil tablas, cincuenta millones de registros y cinco millones de columnas.
- Para su conectividad se abarcan opciones como sockets NT, sockets UNIX y TCP/IP. Además, soporta completamente ODBC.
- Los mensajes de error pueden hacer ordenaciones correctas con palabras con la letra 'ñ' y con palabras acentuadas. Estos mensajes también pueden estar en español.
- Es altamente confiable y estable (Casillas, Gibert, & Pérez, 2014).

#### 3.4.2. MARIADB

MariaDB es una variación, por así decirlo, del sistema de gestión de base de datos MySQL. El propósito de este sistema es otorgar a quien lo use capacidades similares y extendidas en relación con MySQL, con un énfasis especial en mantener el software de forma libre y gratuita a través de la licencia GNU GPL. Tiene aplicaciones muy amplias y puede ser usado por equipos de desarrollo de software, servidores vps o locales y de hosting compartido, sitios web de empresas, bloggers y para testing de aplicaciones. (Salas, 2018).

Inicialmente este gestor de base de datos fue desarrollado por Michael "Monty" Widenius, quien es uno de los fundadores de la compañía MySQL AB. Esta misma compañía fue adquirida por Sun Microsystems a finales de febrero del 2008. Las preocupaciones que Michael Widenius tenía debido a la adquisición de Sun Microsystems por parte de Oracle Corporation lo llevaron al desarrollo de MariaDB, ya que no existía plena certeza de que MySQL seguiría siendo libre. (Salas, 2018)

MariaDB presenta las siguientes extensiones y características nuevas en comparación a MySQL:

- MariaDB controla hasta 32 segmentos clave por clave.
- Se añadió `--abort-sorche-on-error` al cliente mysql.
- Precisión de microsegundos en las listas de procesos.
- Pool de hilos de procesos o ejecución.
- Supresión de tablas.

- [Extensiones de prueba mysqldtest](#)
- [Columnas virtuales](#)
- [El usuario goza de estadísticas extendidas](#)
- [Caché de claves segmentadas](#)
- [Autenticación a través de plugins](#)
- [Especificación de motor de almacenamiento en CREATE TABLE](#)
- [Mejoras a la tabla INFORMATION SCHEMA.PLUGINS](#)
- [Se agregó --rewrite-db como opción en mysqlbinlog al cambiar de base de datos usada](#)
- [Reporte de Procesos para ALTER TABLE y LOAD DATA INFILE \(Salas, 2018\)](#)

#### 3.4.3. CONCLUSIÓN

Al ser los dos sistemas operativos bastante similares (de hecho, porque MariaDB se encuentra basado en lo que es MySQL), se podría escoger cualquiera para la realización del presente trabajo. Las dos bases de datos cumplen con los requerimientos necesarios, que de hecho no son muy grandes, para poder cumplir con las expectativas del sistema.

De la misma manera se puede apreciar que ambas bases de datos son libres y gratuitas, lo que nos permite seguir con el hilo de este trabajo, ya que lo que se busca es que exista la mejor relación coste beneficio para la realización del proyecto.

Se ha optado por usar MySQL para la realización del prototipo, tomando en cuenta que la curva de aprendizaje es menor. Anteriormente se ha tenido el agrado de realizar múltiples trabajos con esta base de datos lo que nos permite tener un mejor dominio de esta, en comparación a MariaDB.

#### 3.5. METODOLOGÍA Y PROCESOS DE DESARROLLO DE SOFTWARE

Para poder desarrollar software se debe tener una metodología a seguir para que el producto final quede como se lo espera, cumpliendo los requerimientos necesarios y funcionando de la manera más óptima posible. De la misma manera hay diferentes tipos y paradigmas de programación, las cuales se ajustan a las

necesidades de los desarrolladores dependiendo del trabajo que tengan previsto. Para este proyecto se ha escogido la metodología de cascada y la programación en tres capas.

### 3.5.1. METODOLOGÍA DE CASCADA

Al hablar del software que se va a desarrollar para controlar este sistema, hablamos de un sistema con un nivel de complejidad no muy alto y que se va a realizar únicamente por una persona.

Por estas razones se ha escogido la metodología de desarrollo de software denominada "Cascada" la cual nos presenta múltiples beneficios y se ajusta a la perfección a las necesidades del proyecto.

El proceso en Cascada consiste en 5 etapas las cuales son las siguientes:

- Definición de Requerimientos.
- Diseño de Sistema y de Software.
- Implementación y Pruebas de Unidades.
- Integración y Pruebas del Sistema.
- Operación y Mantenimiento.

Lo que se busca es lograr responder a las siguientes preguntas a lo largo de todo el proceso. ¿Qué voy a hacer?, ¿Cómo lo voy a hacer?, ¿Cómo se ve completo?, ¿Se hizo bien?.

Este modelo se caracteriza porque es el modelo más simple y conocido. El producto o resultado final solo se ve al final de todo el proceso. Este modelo presenta la facilidad de mejorar el producto final cada que se deba hacerlo o realizar cualquier cambio con facilidad al ser un ciclo que se repite las veces que sea necesario hasta que el resultado final sea el esperado. (Guitierrez, 2011).

Tomando en cuenta los pasos que nos dice el proceso de Cascada, vamos a definir requerimientos funcionales y no funcionales. Seguido a esto se harán diseños en los cuales nos vamos a guiar para desarrollar el software y estructurar el hardware. Diagramas de caso de uso, de secuencia y de actividades serán necesarios para determinar ~~como~~ cómo será la implementación e integración del

software. Por último, se harán pruebas al prototipo finalizado, dando pie a comprobar que el mismo trabaja de acuerdo a los resultados esperados.

Es importante mencionar que vamos a usar la metodología de desarrollo de software para el trabajo en conjunto con el hardware.

### 3.5.2. PROGRAMACIÓN EN TRES CAPAS

Para sistemas que sean confiables, la programación en tres capas es algo que debe ser una de las bases fundamentales para cualquier sistema. La programación en capas consiste en segmentar el código fuente dependiendo de su funcionalidad principal.

Para sacarle el mayor provecho a este tipo de programación se debe seguir una serie de pasos que primero deben ser definidos por el equipo (en este caso una persona) que va a programar el software, luego se deben revisar exhaustivamente estos pasos para así terminar con la implementación. De manera que el producto final sea un sistema seguro y confiable que sea del agrado del cliente. Siguiendo el modelo propuesto se obtiene una herramienta útil, la cual se encuentra dividida en módulos que son fáciles de pulir.

La programación en capas es una técnica propia de la programación por objetos, que hasta el día de hoy ha dado excelentes frutos. Al hablar de las capas, estas se organizan principalmente en tres: La de presentación o frontera, la capa lógica de negocio o control, y la capa de datos. A continuación, vamos a referirnos a ellas. (Vargas & Maltés, 2007).

**Capa de Presentación o Frontera:** Esta viene a ser la presentación del sistema o del programa ante el usuario. Aquí se manejan las interfaces que van a cumplir con el objetivo de facilitar al usuario la interacción con la aplicación. Para lograr esto se utilizan patrones para cada tipo de usuario y para cada tipo de aplicación. La interfaz debe ser fácil de utilizar y amigable, tomando en cuenta que es el usuario final quien va a usar esta parte y de dar la respectiva retroalimentación en caso de que se deba mejorar algo.

La interfaz o interfaces debe ser consciente de los datos que se están manejando, no debe existir más campos de los necesarios, así como la información que se requiera debe ser especificada de manera clara y concisa.

Cada formulario debe contener únicamente lo necesario. Las interfaces deben cumplir con los requerimientos planteados al inicio del proyecto. No se debe excluir información solicitada y no se debe incluir información no solicitada.

Tomando en cuenta la parte técnica, esta capa contiene los objetos que se encargan de comunicar al usuario con el sistema mediante el intercambio de información, capturando y desplegando los datos necesarios para el desempeño de alguna tarea. Aquí los datos se procesan de manera superficial. Esta capa se comunica únicamente con la capa de negocio o control. (Vargas & Maltés, 2007).

**Capa de Lógica de Negocio o Control:** También conocida como la capa de reglas de negocio, ya que en esta se definen todas las reglas a cumplir para que exista una correcta ejecución del programa.

Aquí es donde se encuentra toda la lógica del programa, los objetos encargados para la manipulación de datos existentes, las estructuras de datos y todo el procesamiento de la información ingresada por o solicitada por el usuario en la interfaz.

De cierta manera, esta capa viene a ser el corazón de la aplicación, ya que se comunica con todas las demás capas para poder llevar a cabo todas las tareas. Esta capa recibe los datos que el usuario previamente ingresó al sistema a través de la capa de presentación, luego los procesa y por último crea objetos el caso lo amerite para trabajar con esos datos; esto se conoce como encapsulamiento.

La encapsulación de datos es necesaria para asegurar la consistencia de estos y para obtener información precisa de la base de datos. Solamente se usa la información necesaria, para garantizar que no existan duplicados, tanto en la base como en los reportes que se le entrega al usuario. (Vargas & Maltés, 2007).

**Capa de Datos:** Esta capa es la encargada de realizar las operaciones con la base de datos y con otros sistemas para obtener o ingresar información al sistema.

Como se ha dicho con anterioridad, el manejo de datos debe asegurar la consistencia de estos, de tal manera que los datos que se extraen y que se ingresan deben ser precisos.

Aquí se definen las consultas a realizar en la base de datos, tanto consultas simples como complejas que resultan en reportes más específicos.

Esta capa envía la información directamente a la capa de negocio para que sea ingresada y procesada dentro de objetos según lo amerite. Aquí se puede apreciar nuevamente el encapsulamiento. (Vargas & Maltés, 2007).

## CAPÍTULO 4. ANÁLISIS Y DISEÑO

La primera fase para que cualquier sistema tenga excelentes resultados es un correcto análisis y diseño. Esto nos permitirá enfocarnos en cómo va a funcionar nuestro sistema y bajo qué parámetros. De esta parte depende el éxito de nuestro prototipo ya que de aquí vamos a basarnos para continuar con el resto de los pasos a seguir. Diseñar es proyectarse al resultado final.

### 4.1. ANÁLISIS DEL SISTEMA

Para el análisis se ha optado por delimitar los requerimientos funcionales y no funcionales del prototipo. Los requerimientos funcionales deben especificar cómo va a funcionar nuestro sistema o cómo debe operar el mismo, en sí las funciones que debe cumplir. En cambio, los requerimientos no funcionales establecen bajo qué parámetros va a trabajar el sistema, cuál será la base de datos, cuál será la infraestructura por usar, etc.

#### REQUERIMIENTOS FUNCIONALES

- El sistema debe ser capaz de presentar la posición GPS de cualquier objeto al cual se lo instale.
- Debe funcionar en tiempo real.
- Debe llevar un registro histórico a través del tiempo de las posiciones en las que ha estado dicho objeto.
- La posición se debe presentar a manera de latitud y longitud.
- Debe tener una interfaz de usuario amigable que controle todo el sistema.
- La posición se debe presentar también en un mapa.

#### REQUERIMIENTOS NO FUNCIONALES

- La arquitectura de hardware debe ser con Arduino.
- Para la obtención de la data de geoposicionamiento se usará [el módulo placa-el-módulo-sim808](#).
- La base de datos para el proyecto es MySQL.
- El programa se desarrollará a través de Laravel.
- La metodología de desarrollo de software será la denominada “cascada”.

#### 4.2. DISEÑO DE HARDWARE

Como se dijo anteriormente, los principios de desarrollo que se van a usar, también los vamos a aplicar al hardware del prototipo. Esto quiere decir que también debemos analizar y diseñar ~~como~~ cómo funciona la tecnología en conjunto. Para esta tarea se ha realizado un diagrama en donde se pueden ver como tomarán su lugar las distintas partes que vamos a usar. Véase la figura 1.

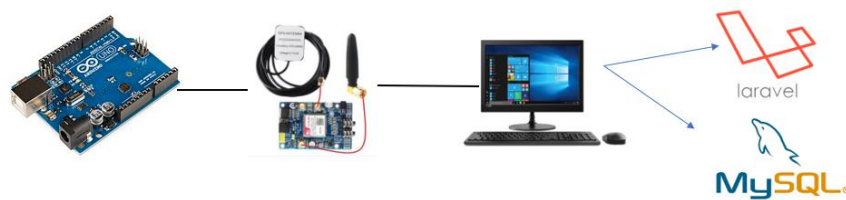


Ilustración 554. Diseño en conjunto de Hardware y Software  
Fuente: Autoría Propia

Dentro de este diseño se debe destacar lo siguiente: Arduino está conectado a un módulo GPS (módulo sim808) el cual será el encargado de darnos la información necesaria. A través de una red inalámbrica se deberá conectar el Arduino con la computadora para enviar los datos. Dentro de la computadora se desarrollará la interfaz de usuario en Laravel, la cual tendrá una conexión directa con la base de datos en MySQL, donde se encontrará toda la data.

#### 4.3. DISEÑO DE LA BASE DE DATOS

Una vez escogida nuestra base de datos, podemos desarrollar el denominado diagrama conceptual, el cual debe mostrar las tablas que va a tener la base,

junto con sus atributos. Atributos que tienen su determinado tipo de dato y longitud de acuerdo a lo requerido para el sistema en sí. De la misma manera la tabla conceptual debe indicar la relación que tienen las tablas que se va a usar. Véase la ilustración 2.

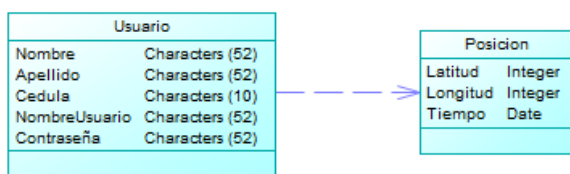


Ilustración 662. Diseño Conceptual de la Base de Datos  
Fuente: Autoría Propia

De hecho, el diseño de la base de datos es bastante simple, pero tiene un par de peculiaridades para el presente trabajo. Para empezar, podemos notar que la posición además de tener las coordenadas básicas de latitud y longitud también va a tener una variable denominada tiempo. Este tiempo va a ser el registro histórico de las posiciones en las que el objeto (al que se ha instalado el sistema) ha estado físicamente.

Además de esto la base de datos debe funcionar con un store procedure que hará que los registros más antiguos de posición se borren constantemente, ya que se va a actualizar cada segundo generando una gran cantidad de datos que hará que el sistema no sea óptimo.

Por último, en la base de datos podemos ver la tabla de usuarios, los cuales van a ser los encargados de usar el sistema. Uno de estos usuarios va a ser el administrador, el cual tiene completo acceso al sistema.

#### 4.4. DISEÑO DE LA APLICACIÓN

Como ya se ha dicho, la presente aplicación se desarrollará en Laravel tratando de realizar una interfaz de usuario lo más amigable posible, para que el usuario final no tenga problemas de usarla. Dentro de la aplicación se debe mostrar la posición actual en tiempo real. Además de esto debe existir un filtro en cual el usuario pueda ingresar una determinada fecha y tiempo, para que de esta manera aparezca en pantalla la posición de los datos ingresados.

## CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS

Para la realización del proyecto se han tomado en cuenta la creación de dos funcionalidades en torno a las cuales va a funcionar el sistema. Al usar la nomenclatura “F”, nos estamos refiriendo a las distintas funcionalidades, las cuales van a estar respectivamente numeradas.

- F.1. Gestión de Usuarios.
- F.2. Gestión de Posiciones.

La parte de Gestión de Usuarios es la que controla todas aquellas personas que van a poder controlar el sistema. Cada uno con sus respectivos diferenciadores (nombre, cédula, etc.). Por otra parte, la Gestión de Posiciones es el sistema en sí, que controla la posición actual del prototipo y mediante la cual se podrá acceder al filtro que nos indica el historial de posiciones.

### 5.1. DIAGRAMA GENERAL DE CASOS DE USO

Con formato: Fuente: 12 pto

Los diagramas de Casos de Uso nos ayudan a visualizar el comportamiento que tiene el sistema, así como la comunicación que existe entre los diferentes actores y funcionalidades que existen. De esta manera se tiene una correcta perspectiva para la implementación.

Tenemos dos actores los cuales van a ser los protagonistas de los casos de uso. El administrador y el cliente. El administrador va a tener completo acceso a las funcionalidades de Gestión de Usuarios y de Gestión de Posiciones. Por su parte el cliente solo tendrá acceso a la funcionalidad de Gestión de Posiciones. Véase la ilustración 3.

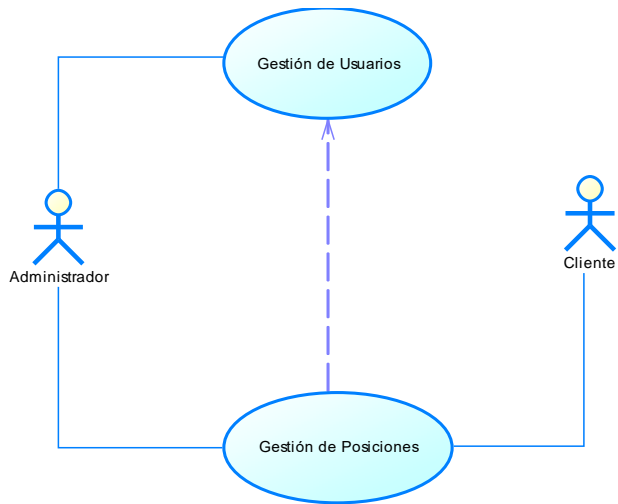


Ilustración 723. Diagrama General de Casos de Uso  
Fuente: Autoría Propia

## 5.2. SEGUNDO NIVEL DE DIAGRAMA DE CASOS DE USO

Un segundo nivel del diagrama de casos de uso nos muestra las sub-funcionalidades que tiene cada una de las funciones, así como su relación con los distintos actores. Véanse las ilustraciones 4 y 5.

### 5.2.1. GESTIÓN DE USUARIOS

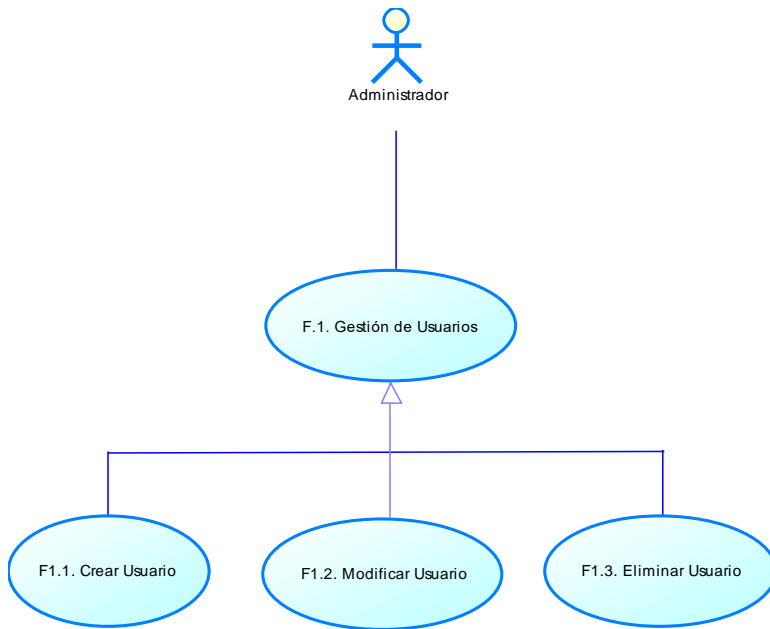


Ilustración 884. Diagrama de Casos de Uso en Segundo Nivel de la funcionalidad: Gestión de Usuarios  
Fuente: Autoría Propia

#### 5.2.2. GESTION DE POSICIONES

La gestión de Posiciones conlleva los procesos: Posición Actual e Historial de Posiciones. Ambas son controladas por el Administrador y por el Cliente. Véase la \_\_\_\_\_ ilustración 5.

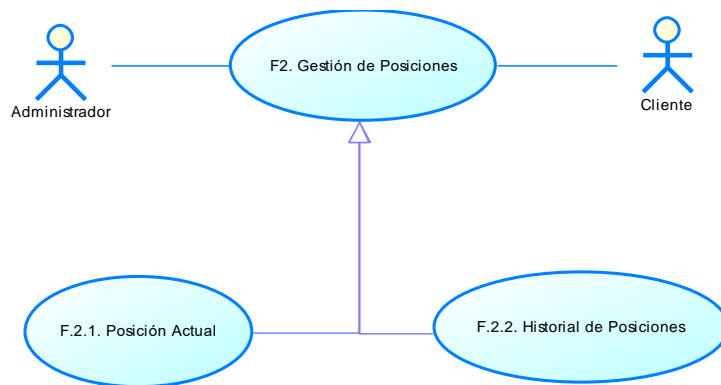


Ilustración 995. Diagrama de Casos de Uso en Segundo Nivel de la Funcionalidad: Gestión de Posiciones  
Fuente: Autoría Propia

### 5.3. CASOS DE USO A DETALLE

Para proceder a explicar todos los procesos, mediante los cuales se encuentra guiada la implementación, usaremos los casos de uso a detalle los cuales nos explican a detalle cómo va a ser todo el flujo de los mismos procesos.

#### 5.3.1. CREAR USUARIO

Este proceso es el necesario para empezar con todo el funcionamiento del sistema, ya que se necesita de un usuario que se vendrá a convertir en administrados y de los usuarios que van a manejar la aplicación. En este proceso se permite crear cada usuario que sea necesario dándole una cédula, nombre, apellido, usuario y contraseña.

**Nombre:** F1.1. Crear Usuario.

**Actores:** Administrador.

**Relación:** Se encuentra directamente relacionado al administrador. Véase la ilustración 6.



Ilustración ~~10.106~~. Relación entre el Administrador y el proceso Crear Usuario  
Fuente: Autoría Propia

### Flujo:

1. El actor selecciona Administrador de Usuarios.
2. El sistema presenta la pantalla de Usuarios.
3. El actor presiona el botón "Crear Usuario".
4. El sistema presenta los campos necesarios para crear un Usuario.
5. El actor ingresa la cédula del Usuario a crear (E1).
6. El actor ingresa el nombre de ingreso del usuario (E2).
7. El actor ingresa la información solicitada.
8. El actor presiona el botón "Crear".
9. El sistema almacena la información.

### Flujo Alternativo:

Casos de Uso 1.2 o 1.3.

### Excepciones:

Causa: E1. Información ya Ingresada.

Mensaje: Cambiar la información introducida.

Causa: E2. Información ya Ingresada.

Mensaje: Cambiar la información introducida.

### 5.3.2. MODIFICAR USUARIO

A través de este proceso el administrador tiene la posibilidad de modificar la información de los usuarios que ya hayan sido previamente creados.

**Nombre:** F1.2. Modificar Usuario.

**Actores:** Administrador.

**Relación:** Se encuentra directamente relacionado al administrador. Véase la ilustración 7.



Ilustración 11447. Relación entre el Administrador y el proceso Modificar Usuario  
Fuente: Autoría Propia

### Flujo:

1. El actor selecciona Administrador de Usuarios.
2. El sistema presenta la pantalla de Usuarios.
3. El actor selecciona Consultar Usuarios.
4. El sistema presenta un filtro de búsqueda.
5. El actor ingresa la información mediante la cual desea buscar.
6. El actor presiona el botón "Buscar".
7. El sistema presenta los Usuarios previamente creados que coincidan con los parámetros ingresados.
8. El actor presiona "Modificar" en el usuario que desea cambiar.
9. El sistema solicita que se introduzca la contraseña del usuario que se desea modificar.
10. El actor ingresa la contraseña (E1).
11. El sistema presenta los campos a modificar.
12. El actor modifica los datos que desea cambiar (E2).
13. El actor presiona el botón "Modificar".
14. El sistema almacena la información.

### Flujo Alternativo:

Casos de Uso 1.1 o 1.3.

### Excepciones:

Causa: E1. Contraseña ingresada es incorrecta.

Mensaje: Contraseña incorrecta.

Causa: E2. Información ya ingresada.

Mensaje: Cambiar la información introducida.

### 5.3.3. ELIMINAR USUARIO

De la misma manera el administrador debe tener la capacidad de los usuarios que ya no sean necesarios dentro del sistema, para eso usamos este proceso.

**Nombre:** F1.3. Eliminar Usuario.

**Actores:** Administrador.

**Relación:** Se encuentra directamente relacionado al administrador. Véase la ilustración 8.

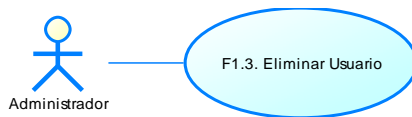


Ilustración 12128. Relación entre el Administrador y el proceso Eliminar Usuario  
Fuente: Autoría Propia

#### Flujo:

1. El actor selecciona Administrador de Usuarios.
2. El sistema presenta la Pantalla de Usuarios.
3. El actor selecciona Consultar Usuarios.
4. El sistema presenta un filtro de búsqueda.
5. El actor ingresa la información mediante la cual desea buscar.
6. El actor presiona el botón "Buscar".
7. El sistema presenta los Usuarios previamente creados que coincidan con los parámetros ingresados.
8. El actor presiona "Eliminar" en el usuario que desea borrar.
9. El sistema pide una confirmación de la acción realizada.
10. El actor confirma la acción de eliminar.
11. El sistema actualiza la información.

#### Flujo Alternativo:

Casos de Uso 1.1 o 1.3.

### 5.3.4. POSICIÓN ACTUAL

Este proceso es por el cual entramos directamente a gestionar la posición actual del GPS.

**Nombre:** F2.1. Posición Actual.

**Actores:** Administrador, Cliente.

**Relación:** Se encuentra directamente relacionado al administrador y el cliente. Véase la ilustración 9.

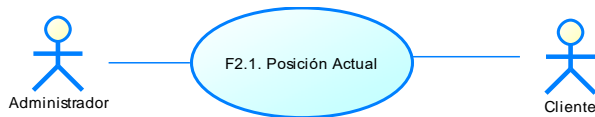


Ilustración 13129. Relación entre el Administrador y el Cliente con el proceso Posición Actual  
Fuente: Autoría Propia

**Flujo:**

1. El sistema solicita un Usuario y Contraseña.
2. El actor escribe un Usuario y Contraseña válidos.
3. El actor presiona el botón “Ingresar” (E1).
4. El sistema presenta la Ventana del GPS.
5. El sistema presenta la posición actual por pantalla.

**Flujo Alternativo:**

Caso de Uso 2.2.

**Excepciones:**

Causa: E1. Usuario ingresado es incorrecto.

Mensaje: Datos mal ingresados.

5.3.5. HISTORIAL DE POSICIONES

A través de este proceso podemos gestionar las posiciones que ha registrado el GPS a través del tiempo. A través de un filtro podemos buscar un registro mediante el parámetro deseado.

**Nombre:** F2.2. Historial de Posiciones.

**Actores:** Administrador, Cliente.

**Relación:** Se encuentra directamente relacionado al administrador y al cliente. Véase la ilustración 10.

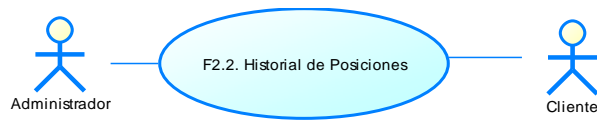


Ilustración ~~144449~~. Relación entre el Administrador y el Cliente con el proceso Historial de Posiciones  
Fuente: Autoría Propia

### Flujo:

1. El sistema solicita un Usuario y Contraseña.
2. El actor escribe un Usuario y Contraseña válidos.
3. El actor presiona el botón "Ingresar" (E1).
4. El sistema presenta la Ventana del GPS.
5. El sistema presenta la posición actual por pantalla.
6. El sistema presenta un buscador para filtrar posiciones.
7. El actor ingresa el parámetro por el cual desea filtrar.
8. El actor presiona el botón "Buscar".
9. El sistema presenta las coincidencias con lo ingresado (E2).

### Flujo Alterno:

Caso de Uso 2.2.

### Excepciones:

Causa: E1. Usuario ingresado es incorrecto.

Mensaje: Datos mal ingresados.

Causa: E2. Información no coincide con la base de datos.

Mensaje: No se encuentran posiciones.

## 5.4. DIAGRAMAS DE SECUENCIA

Para poder explicar a detalle como el actor va a interactuar con las distintas funcionalidades del sistema se usan los diagramas de secuencia. En estos podemos ver cómo paso a paso se desarrollan cada uno de los procesos que hemos tomado en cuenta a través de las interfaces de nuestro sistema.

#### 5.4.1. CREAR USUARIO

Al crear el usuario se deben ingresar los siguientes datos: Cédula, Nombre, Apellido, Nombre de Usuario y Contraseña, en un framework que le permita al cliente realizarlo de manera natural y fluida.

En el caso de la cédula, no pueden existir dos usuarios con la misma cédula. Ni tampoco puede existir dos usuarios con el mismo nombre de usuario, valga la redundancia. Por esto se han realizado controladores que permiten verificar en la base de datos si las cédulas ingresadas o a su vez los nombres de usuario ingresados ya existen.

Además, dentro de este proceso existe una función que hace que un usuario se convierta en administrador, el cual será el único capaz de realizar todas las acciones que un cliente no podría (modificar y eliminar usuarios). Véase la ilustración 11.

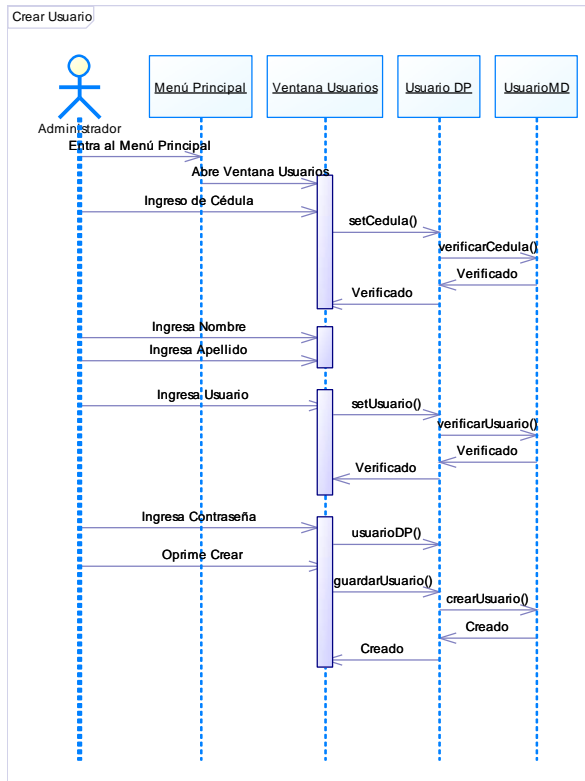


Ilustración 154544. Diagrama de Secuencia del proceso Crear Usuario  
Fuente: Autoría Propia

#### 5.4.2. MODIFICAR USUARIO

Dentro de este proceso se debe nombrar la presencia de un controlador que hace ingresar la contraseña del usuario al cual se lo deseaba modificar. Si la contraseña es correcta, daba acceso a cambiar su información de otra manera no lo permite. Véase la ilustración 12.

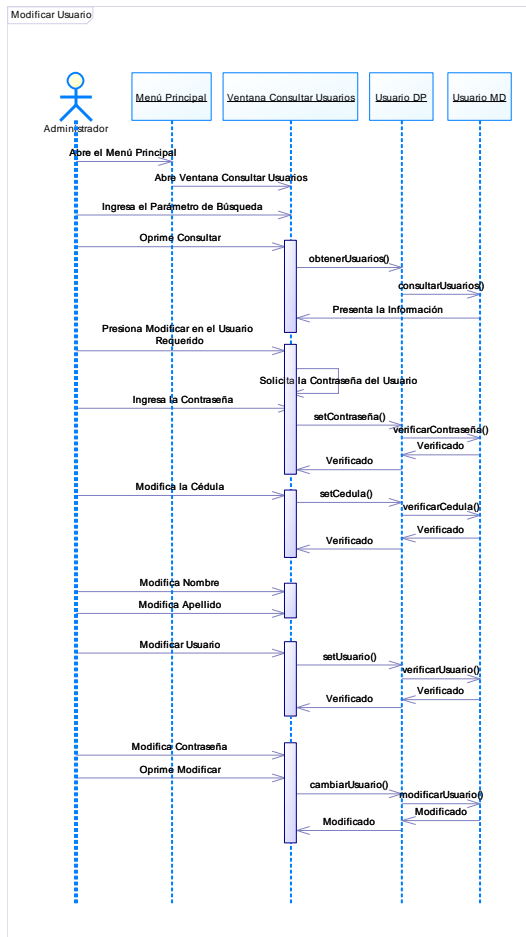


Ilustración 164612. Diagrama de Secuencia del proceso Modificar Usuario  
Fuente: Autoría Propia

### 5.4.3. ELIMINAR USUARIO

En cuanto a lo que se refiere a eliminar usuario, el sistema tiene un borrado lógico que funciona dentro de la base de datos. Al presionar eliminar y confirmar la acción. La base de datos debe cambiar el estado de ese usuario, dejándolo inactivo. Véase la ilustración 13.

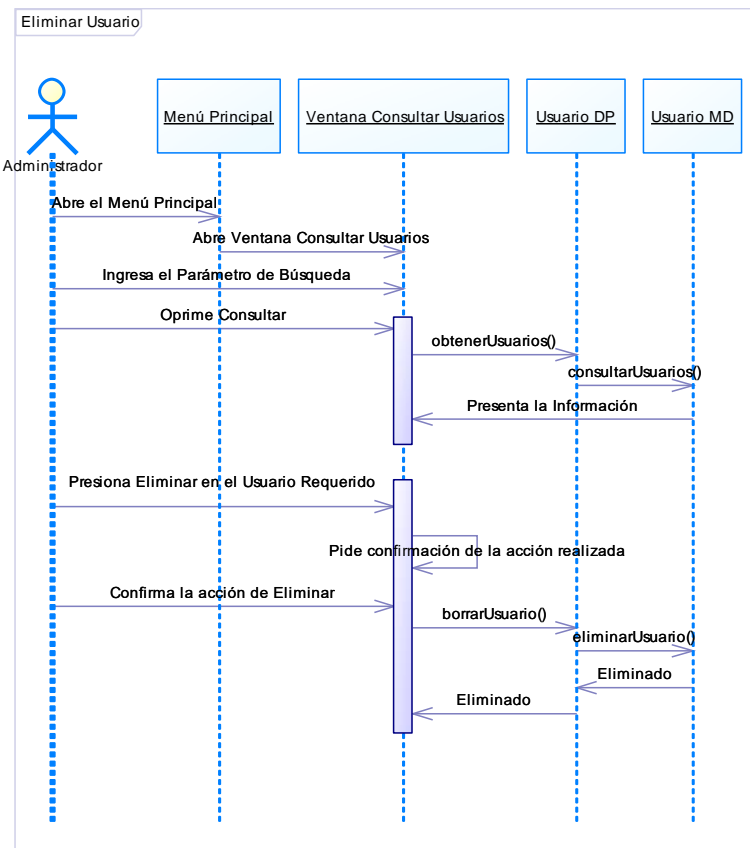


Ilustración 174743. Diagrama de Secuencia del proceso Eliminar Usuario  
Fuente: Autoría Propia

#### 5.4.4. POSICIÓN ACTUAL

Después de haber creado un usuario, el administrador o cliente podrán ingresar directamente a gestionar la posición actual del GPS. El sistema deberá validar la información introducida para saber si el usuario es válido o no y de esta manera se pasará a la pantalla de posiciones. Véase la ilustración 14.

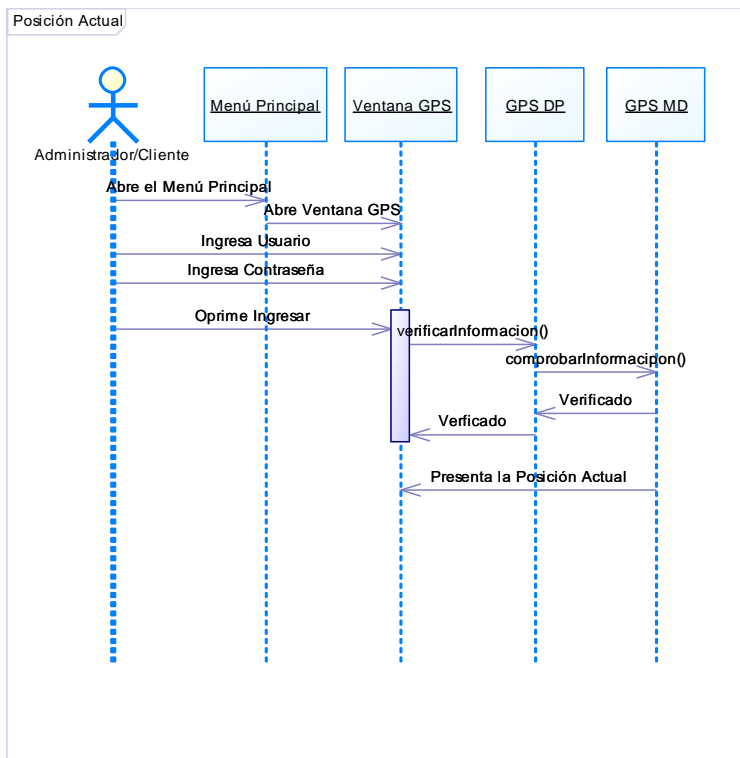


Ilustración 181914. Diagrama de Secuencia del proceso Posición Actual  
Fuente: Autoría Propia

#### 5.4.5. HISTORIAL DE POSICIONES

El sistema automáticamente esta guardando registros durante cada segundo de las distintas posiciones en dónde se ha encontrado el GPS a lo largo del tiempo. Todos estos registros se encuentran en la base de datos y pueden ser filtrados a través de este proceso, el cual permite buscar un registro determinado mediante un parámetro ingresado. Véase la ilustración 15.

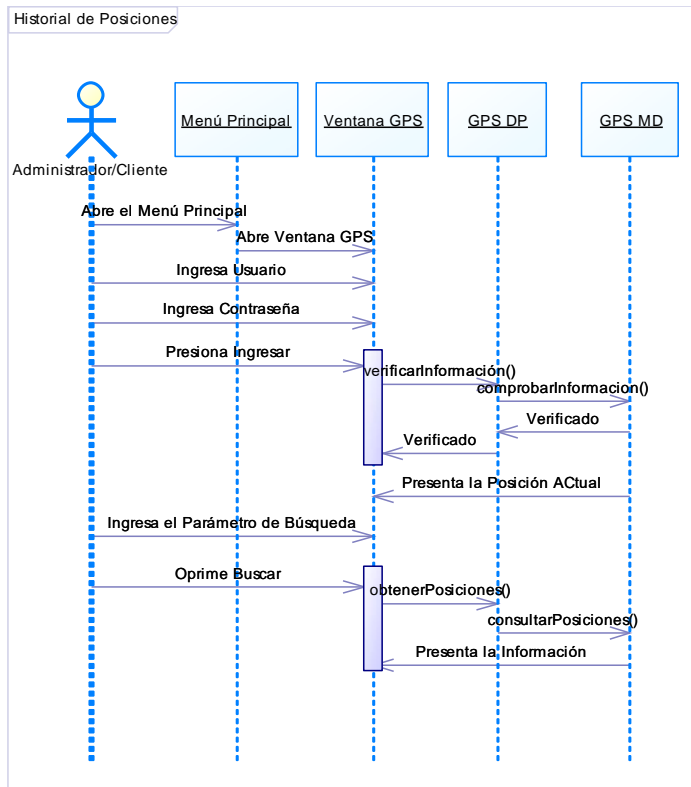


Ilustración 19-15. Diagrama de Secuencia del proceso Historial de Posiciones  
Fuente: Autoría Propia

### 5.5. DIAGRAMA DE ACTIVIDADES

El diagrama de actividades nos va a dar un mejor enfoque de cómo va a funcionar el sistema en conjunto, dividiendo las tareas entre las distintas partes que tiene el mismo: Administrador, Sistema Virtual y el GPS físico. De esta manera vemos como todos los procesos se enlazan desde el comienzo hasta el final. Véase la ilustración 16.

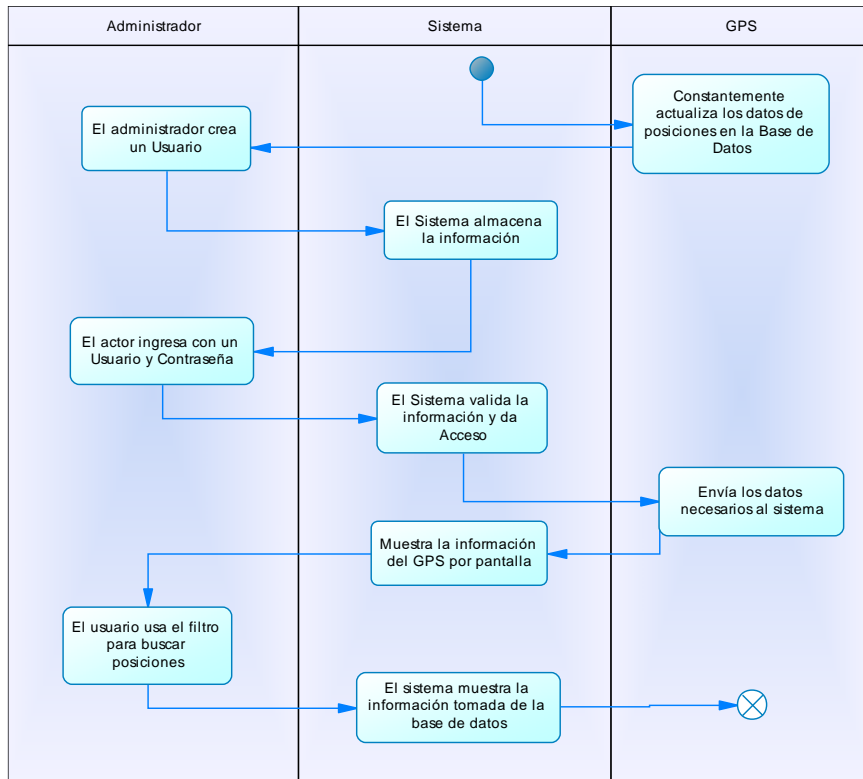


Ilustración 202016. Diagrama de Actividades  
Fuente: Autoría Propia

## 5.5. PRUEBAS

Las pruebas son indispensables dentro de cualquier sistema en desarrollo para verificar que el mismo se encuentra trabajando de la mejor manera. Este sistema no es la excepción, por lo cual se han hecho los debidos análisis que muestran que el mismo no tiene fallas y que gestiona correctamente las funciones que se le han sido asignadas.

Primero se empezó esta etapa con las pruebas de ejecución, que son aquellas que nos muestra como trabaja nuestro hardware. Para esto se puso a trabajar al Arduino varias veces tomando el tiempo de respuesta (tiempo en que se demora en enviar la información al sistema) que el mismo nos da. De esta manera podemos compararlo juntamente con los tiempos de respuesta que tienen otros

modelos de GPS para demostrar que el equipo que estamos usando es óptimo. Véase la tabla 1.

	Tiempo 1	Tiempo 2	Tiempo 3	Registros
Nuestro Prototipo	0.0010	0.0015	0.0018	500
Raspberry	0.0024	0.0085	0.0064	500
Neo 6-M	0.0032	0.0158	0.0142	500
EM-411 GPS	0.0047	0.0142	0.124	500

*Tabla 1. Pruebas de Ejecución  
Fuente: Autoría Propia*

Como se puede apreciar en la tabla, los resultados son los esperados, demostrando así el hardware que estamos usando tiene un aceptable tiempo de respuesta.

Seguido a esto, se prosiguió con las pruebas de reacción. Pruebas en las que se toma en cuenta el tiempo en el que el sistema se demora en precisar donde se encuentra ubicado el GPS. Para esto, tomamos en cuenta tres escenarios diferentes. Mover el GPS a pie, en una bicicleta y en un automóvil. A medida que se avanzaba en distancia, se tomaba el tiempo en el que el prototipo se ubicaba correctamente, dando los siguientes resultados. Véase la tabla 2.

	Tiempo 1	Registros	Tiempo 2	Registros	Tiempo 3	Registros
<b>A pie</b>	0.0000 s	1	0.0007	100	0.0010	150
<b>En bicicleta</b>	0.003 s	10	0.0013	100	0.0019	150
<b>En carro</b>	0.0023 s	20	0.0034	100	0.0048	150

*Tabla 2. Pruebas de Reacción  
Fuente: Autoría Propia*

Como se puede apreciar, los tiempos de reacción en pie son menores a los tiempos en bicicleta. De la misma manera los tiempos en bicicleta vienen a ser

menores que los tiempos en carro. Esto es debido a que se recorre menos distancia en el mismo tiempo, por lo que el GPS tarde menos en ubicarse correctamente. De cualquier manera, los tiempos en los tres casos son aceptables.

Para continuar con esta etapa se realizaron pruebas de ambiente, que nos van a ayudar a verificar si nuestro sistema trabaja adecuadamente en diferentes entornos. Para esto se colocó el GPS en tres diferentes lugares como son: un subsuelo, en un cuarto cerrado y en una azotea. Teniendo los siguientes resultados. [Véase la tabla 3.](#)

Lugar	Respuesta
Subsuelo	Dificultad para funcionar
Cuarto Cerrado	Funciona Correctamente
Azotea	Funciona Correctamente

*Tabla 3. Pruebas de Ambiente  
Fuente: Autoría Propia*

Como se puede apreciar, vemos que, en un subsuelo, el prototipo presenta ciertas dificultades, esto debido a que las conexiones (tanto a Internet como la GPS) no se puede establecer correctamente. Concluyendo que nuestro sistema va a trabajar ininterrumpidamente, siempre en cuando las conexiones sean estables. Vale la pena destacar que el sistema respondió más rápido cuando estuvo en la azotea que cuando estuvo en cuarto cerrado. Esto sucede porque la conexión con las redes es más rápida al estar en un entorno abierto.

Para finalizar, se realizaron pruebas de validación. En las que se debía comprobar que el sistema marque la posición correcta en la que el prototipo se encuentra. De la misma manera, se probó en tres diferentes casos, colocando al prototipo en una casa, en un parque y en un centro comercial, dando los siguientes resultados.

Lugar	Resultado
Casa	Posición Correcta
Parque	Posición Correcta
Centro Comercial	Posición Correcta

*Tabla 4. Pruebas de Validación  
Fuente: Autoría Propia*

El resultado fue el esperado, ya que nuestro GPS marcó siempre la posición correcta, con un margen de error bastante aceptable. En lugares abierto el margen de error se reducía considerablemente, debido a que la señal es mucho mejor y por ende la respuesta es acertada. La mayoría de GPS comerciales muestran siempre un margen de error (esto es apreciable incluso en nuestros celulares), únicamente los profesionales y extensamente costosos son aquellos que nos darán resultados exactos y completamente precisos.

Al haber realizado las pruebas antes mencionadas y haber sometido a nuestro sistema a diferentes ambientes se pudo comprobar la funcionalidad de este garantizando así que los resultados son los esperados.

## CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES

### 6.1. PRECIO ESTIMADO

En este apartado se podrá apreciar los costos de desarrollo ~~de~~ un prototipo funcional del GPS. Para lo mismo se desarrolló una tabla en dónde se puede ver desglosado, los objetos en los que fue necesario invertir. Véase la tabla 5.

OBJETO	COSTO	UNIDADES EMPLEADAS	TOTAL
Arduino Uno	25 dólares	1	25
<del>Módulo</del> <del>Placa</del> Sim808	<del>4050</del> dólares	1	<del>500</del>
Cables	15 centavos	4	0.60
Caja de Madera	3 dólares	1	3
Baterías Recargables más cargador	10 dólares	1	10

Tabla 5. Tabla de Costos  
Fuente: Autoría Propia

Al observar la tabla se puede apreciar que no se emplean demasiados materiales. Esto depende de las especificaciones que cada empresa o persona quiera dar para la fabricación de su respectivo GPS. De la misma manera se puede apreciar que la tecnología empleada no presenta mayor dificultad económica, tomando en cuenta que los materiales empleados son originales y por ende garantizan sus resultados. El precio total del prototipo es de ~~7888~~.60 dólares, precio que puede variar de acuerdo con los materiales empleados. Tomando en cuenta los factores mencionados antes, se puede decir que se tiene un precio accesible por cualquier organización, persona particular o empresa que necesite de un GPS.

## 6.2. CONCLUSIONES

Al realizar una investigación de la tecnología necesaria, se pudo concretar cuales son las herramientas óptimas para el desarrollo del presente trabajo. De esta manera se pudo diseñar el prototipo final, permitiendo que todas las tecnologías en conjunto den como resultado la resolución de las necesidades planteadas.

Al realizar un correcto análisis y tener los diseños necesarios, se pudo desarrollar un prototipo funcional en el cual el hardware y el software trabajan conjuntamente para poder obtener los resultados esperados. Gracias a las pruebas realizadas en el sistema se puede apreciar que la gestión de los datos de geoposicionamiento es acertada.

Vale la pena destacar que las herramientas utilizadas son libres y gratuitas. Esto quiere decir que las empresas o personas que deseen utilizar el prototipo pueden hacerlo sin ninguna complicación. Además de que pueden modificarlo para cumplir los requerimientos que se necesiten.

Tomando en cuenta la inversión necesaria para el desarrollo de este trabajo investigativo se pudo generar un costo estimado del prototipo funcional (valor que es menor a 90 dólares). Concluyendo de esta manera que la herramienta es accesible, ya sea desarrollándola por cuenta propia o por medio de terceros.

### 6.3. RECOMENDACIONES

Se recomienda hacer pruebas con herramientas diferentes a las empleadas, para comprobar que el sistema sea portable. Dicho de otra manera, que funciona en diferentes modelos. Así también se podrá apreciar fallas que puede tener el sistema y corregir las mismas para presentar mejores versiones.

Continuar con las pruebas en el prototipo es algo necesario para comprobar que el mismo funcione de la mejor manera. Solo con pruebas exhaustivas se podrá verificar que no hay errores a corto y largo plazo. Gracias a esto se podrán tomar las acciones pertinentes para corregir los errores, en caso de ser necesario. Las pruebas terminarán en el momento que el prototipo no muestra ningún inconveniente en ningún campo.

De ser posible, se debe promover el uso de las herramientas libres y gratuitas. Estas ofrecen una gran versatilidad a todas aquellas personas que deseen sacar provecho de determinadas tecnologías. Formando un círculo de constate retroalimentación, la tecnología libre nos permite crecer conjuntamente.

Invertir en sistemas que promuevan al desarrollo de la empresa, es algo que se debería hacer sin lugar a duda. La tecnología está tomando gran parte del mundo actual y si una organización quiere mostrarse competitiva ante el mercado, la mejor forma de hacerlo es a través de este medio.

## Anexos

GPS USUARIOS GPS

Listado Gps Nuevo

Latitud	Altitud	Longitud	Fecha	Opciones
11111	1234	123	2020-01-01 00:00:00	Editar
11111	1234	123	2020-01-01 00:00:00	Editar

GPS USUARIOS GPS

Editar Gps: 11111

Latitud  Longitud   
Altitud  Fecha

Guardar Cancelar

GPS USUARIOS GPS

Nuevo Registro Limpiar

Latitud  Longitud   
Altitud  Fecha

Guardar Cancelar

GPS USUARIOS GPS

Nuevo Registro Limpiar

Latitud  Longitud   
Altitud  Fecha

Guardar Cancelar

Listado Usuarios Nuevo

Usuario	Nombre	Identificación	Correo	Condicion	Opciones
sebas6879	sebas	401687934	sebastian6879@hotmail.com	1	Editar
andymera1104	anderson	1720617248	andersonmera6a1@gmail.com	1	Editar
sebas12356	sebastian	1720617245	sebas@gmail.com	1	Editar

Nuevo Usuario Limpiar

Usuario	<input type="text" value="Ingrese Usuario"/>	Password	<input type="text" value="Ingrese password"/>
Nombre	<input type="text" value="Ingrese nombre"/>	Identificacion	<input type="text" value="Ingrese identificacion"/>
Correo	<input type="text" value="Ingrese correo"/>		

Guardar Cancelar

## Editar Usuario: sebas

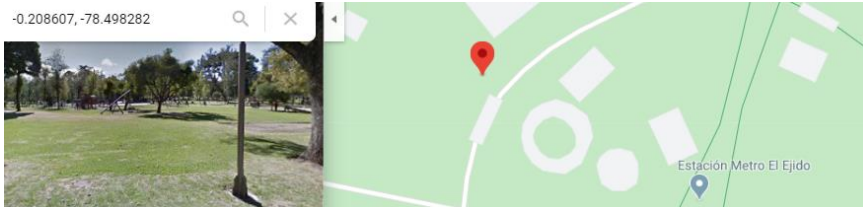
Usuario	<input type="text" value="sebas6879"/>	Password	<input type="text" value="....."/>
Nombre	<input type="text" value="sebas"/>	Identificacion	<input type="text" value="401687934"/>
Correo	<input type="text" value="sebastian6879@hotmail.com"/>		

Guardar Cancelar

## VALIDACIONES DE DATOS

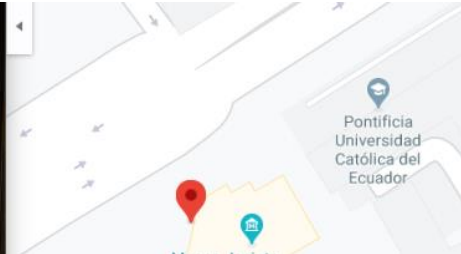
Editar Copiar Borrar 5 -0.208607 -78.498282 1701 2020-02-13 13:00:00 1 NULL NULL

-0.208607, -78.498282



Editar Copiar Borrar 4 -0.209750 -78.492306 1076 2020-02-11 10:00:00 1 NULL NULL

-0.209750 -78.492306



Editar Copiar Borrar 6 -0.209750 -78.492306 1710 2020-03-15 09:00:00 1 NULL NULL

-0.209750 -78.492306



## BIBLIOGRAFÍA

- Andrades, F. (6 de Junio de 2013). *El Diario*. Obtenido de ¿Qué es el Hardware Libre?: [https://www.eldiario.es/turing/Hardware-Libre\\_0\\_139986451.html](https://www.eldiario.es/turing/Hardware-Libre_0_139986451.html)
- Banzi, M., & Shiloh, M. (2016). *Introducción a Arduino*. España: Anaya.
- Blogger*. (12 de Junio de 2015). Obtenido de <http://gpsoncedos.blogspot.com/2015/06/estado-del-arte.html>
- Casillas, L., Gibert, M., & Pérez, Ó. (2014). *Bases De Datos En MySQL*. Cataluña: Universidad Abierta de Cataluña.
- De La Puente, J. (2007). *Introducción A Los Sistemas En Tiempo Real*. Madrid: DIT/UPM.
- González, I., González, J., & Gómez, F. (2003). *Hardware libre: clasificación y desarrollo de hardware*. Madrid, España: Universidad Autónoma de Madrid.
- Guitierrez, D. (2011). *Métodos De Desarrollo De Software*. Mérida: Universidad De Los Andes.
- Lazalde, A., Torres, J., & Vila, D. (2015). *Ecosistemas de innovación y producción basados en hardware libre*. Quito, Ecuador: Flock Society.
- Letelier, P. (2006). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia: Universidad Politécnica de Valencia.
- Macho, J. (25 de Enero de 2018). *Prometec*. Obtenido de <https://www.prometec.net/sim808/>
- Peñafiel, J., & Zayas, J. (2001). *Fundamentos del Sistema GPS y Aplicaciones en la Topografía*. Madrid: Colegio Oficial de Ingenieros Técnicos en Topografía.
- Pozo, A., Ribiero, A., García, M., García, L., Guinea, D., & Sandoval, F. (2000). *Sistema de Posicionamiento Global (GPS): Descripción, Análisis de Errores, Aplicaciones y Futuro*. Madrid: Instituto de Automática Industrial, Consejo Superior de Investigaciones Científicas.
- Rafael, E. (2009). *Guía de Usuario de Arduino*. San Francisco: Creative Commons.
- Salas, M. (15 de Marzo de 2018). *Hosting Latam*. Obtenido de Características de MariaDB, un proyecto derivado de MySQL: <https://www.hostinglatam.cl/caracteristicas-de-mariadb-un-proyecto-derivado-de-mysql/>
- Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Fundamentos de Bases de Datos*. Madrid: McGraw-Hill.