# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS

# PROYECTO DE GRADO

"SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS"

**CASO:** GOBIERNO AUTÓNOMO MUNICIPAL DE VIACHA

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INGENIERÍA DE SISTEMAS

**MENCIÓN: GESTION Y PRODUCCIÓN**


**POSTULANTE :** ISRAEL GUILLERMO QUISPE MAMANI

**TUTOR METODOLÓGICO:** M.SC. ING. ENRIQUE FLORES BALTAZAR

**TUTOR ESPECIALISTA:** LIC. FREDDY SALGUEIRO TRUJILLO

**TUTOR REVISOR:** ING. ROSA VERASTEGUI ONTIVEROS


**EL ALTO – BOLIVIA**

**2020**

**DEDICATORIA**

Primeramente el presente trabajo la dedico a Dios por siempre estar a mi lado y darme la fortaleza y voluntad para poder culminar y cumplir una de mis metas.

A mis padres Vicente y Cristina, por el infinito amor demostrado, el sacrificio realizado y la muestra de ejemplo para luchar en la vida sin importar las circunstancias.

A mi hermana Estefani, mi hermano Ever y mi abuelita quienes siempre están para darme su mano y apoyo.

A mi novia Luz Yorgelis quien de la mejor forma posible ha ido apoyándome impulsándome para que esta meta lo cumpla, gracias por el inmenso amor infinito e incondicional.

## AGRADECIEMIENTOS

**A Dios Todopoderoso**

Quien me dio las fuerzas, sabiduría, coraje y sobre todo el entusiasmo para llegar hasta donde estoy y siempre me guio y protegió durante esta parte de mi vida. Te agradezco Señor Jesús y te pido que me sigas dando ese entusiasmo de aprender y seguir en todas las facetas de mi vida.

**A mis Padres Vicente y Cristina**

Ya que sin su apoyo incondicional, no hubiese sido posible cumplir con este objetivo, pues con mucho esfuerzo, sacrificio y amor, me dieron siempre el respaldo y confianza necesaria durante toda la carrera. Este éxito se los dedico a ellos especialmente.

**A mi Abuela, Hermana y Hermano**

Quienes siempre han estado apoyándome de la mejor forma posible, acompañándome, ayudándome, colaborándome con mucho cariño incondicionalmente.

**A mis Tutores**

De la misma manera agradezco al Ing. Enrique Flores Baltazar, Lic. Freddy Salgueiro Trujillo e Ing. Rosa Verastegui Ontiveros, quienes fueron mis guías durante el desarrollo de mi proyecto de grado, gracias por la paciencia, la orientación y el tiempo dedicado.

**A mi novia Luz Yorgelis**

Quien con mucho amor, lealtad y respeto, me ha impulsado a seguir adelante, ofreciéndome su apoyo incondicional dentro de la realización del presente trabajo.

## RESUMEN

El presente documento es una guía de todo el proceso realizado para la implementación de un Sistema basado en internet de las cosas aplicado a la geo localización de vehículos recolectores de basura, la cual coadyuvará en la toma de decisiones y el mejoramiento del servicio de recojo de basura en el Municipio de Viacha.

En la primera parte denominada marco preliminar, se realiza una análisis de la problemática a abordar junto a una revisión de los antecedentes de la institución, y a partir de ello, se define los objetivos para dar solución a la problemática, así también a partir de lo información obtenida se describe la justificación ya sea social, técnica y económica. Definimos las herramientas que se utilizará con conceptos muy claros y precisos, y por ultimo definimos los límites y alcances del presente proyecto.

En la segunda parte la cual es el Marco teórico, definimos toda a parte teórica sobre la cual se basa el presente proyecto, definiendo conceptos, herramientas, metodologías, etc.

En la etapa de análisis y desarrollo aplicamos toda esa base teórica definida, en el apartado anterior, así también hacemos uso de las métricas de calidad y la aplicación de políticas de seguridad al prototipo terminado.

Finalmente llegamos a las conclusiones y recomendaciones la cual se establecen como en la parte final del presente, en conclusiones hacemos una revisión de todos los objetivos cumplidos satisfactoriamente, por otro lado las recomendaciones describen los objetivos a futuro del presente proyecto.

**Palabras clave: UWE, geo localización, arduino, broker, gps.**

## ABSTRACT

This document is a guide to the entire process carried out for the implementation of a system based on the Internet of Things applied to the geo-location of garbage collection vehicles, which will help in decision-making and the improvement of the collection service. garbage in the Municipality of Viacha.

In the first part called the preliminary framework, an analysis of the problem to be addressed is carried out together with a review of the institution's background, and from there, the objectives are defined to solve the problem, as well as from what Information obtained describes the justification, whether social, technical or economic. We define the tools to be used with very clear and precise concepts, and finally we define the limits and scope of this project.

In the second part, which is the theoretical framework, we define the entire theoretical part on which this project is based, defining concepts, tools, methodologies, etc.

In the analysis and development stage we apply all that theoretical base defined in the previous section, so we also make use of quality metrics and the application of security policies to the finished prototype.

Finally we come to the conclusions and recommendations which are established as in the final part of this, in conclusions we review all the objectives satisfactorily fulfilled, on the other hand the recommendations describe the future objectives of this project.


**Keywords: UWE, geolocation, arduino, broker, gps.**

# INDICE DE CONTENIDO

## INDICE DE FIGURAS

# INDICE DE TABLA

# CAPITULO I

# MARCO PRELIMINAR

## 1.1. INTRODUCCIÓN

En la actualidad se hace importante el intercambio de información entre objetos cotidianos con internet para el monitoreo de las mismas, esta idea ha llevado a la creación de un concepto muy interesante, Internet of Things (Internet de las cosas). La misma hace referencia a poder conectar ciertos tipos de objetos que usamos en nuestra vida cotidiana al internet para el tratado de los datos.

La idea de intercambiar este tipo de información es para mejorar la usabilidad de tales objetos cotidianos y tener datos que puedan darnos mayor certeza de lo que las personas necesitan. Para esta idea, por ejemplo, se pretende usar protocolos de comunicación ligeras lo más sencillos posibles sin dejar de lado la seguridad. Para esto se hace uso del protocolo MQTT (Message Queue Telemetry Transport) la cual trabaja muy bien en dispositivos con recursos y conectividad limitados y para este caso en particular en el dispositivo de hardware abierto Arduino Mega junto al módulo de conectividad GSM/GPRS SIM 900

El presente proyecto pretende hacer uso de estas tecnologías aplicado a la geo localización en tiempo real de los vehículos encargados en la recolección de basura teniendo conocido características principales de la misma. Así también pretende monitorear las rutas por cumplir, los tiempos de llegada y de esta manera coadyuvar en la toma de decisiones para el mejoramiento del servicio.

Para el cumplimiento de este proyecto se dividirá en dos partes, la parte del software y la parte del hardware. En cuanto al hardware se pretende hacer uso del Arduino versión Uno, la cual es una plataforma abierta para el desarrollo de prototipos de hardware; Agregado a la misma haciendo uso de la placa SIM900 GSM/GPRS para la comunicación y el dispositivo GPS NEO-6M para la geo localización. Para la parte del software tenemos las tecnologías como NestJs para el Backend, que se encarga de la interacción con la base de datos en PostgreSQL, Angular para consumo de los datos generados por el Backend.

Todas estas tecnologías trabajaran sobre la tecnología NodeJs junto a Socket.io para los datos en tiempo real.

## 1.2. ANTECEDENTES

### 1.2.1. Antecedentes de la institución

La localización geográfica del Municipio de Viacha está en el departamento de La Paz, Provincia Ingavi primera sección, a continuación describimos algunos datos relevantes del lugar de estudio en la siguiente tabla.

**Tabla Nº 1.1:** Descripción de ubicación geográfica

| Departamento | Provincia | Municipio | Coordenadas Geográficas | Altura msnm |
|---|---|---|---|---|
| **La Paz** | Ingavi | Viacha | 16º39'12'' Latitud Sur<br>68º18'05'' Longitud Oeste | 3560 m.s.n.m[1] |

**Fuente:** (Delfín, 2012)

El Gobierno Autónomo Municipal de Viacha, municipio milenario e histórico; promueve el desarrollo humano integral, productivo industrial – agropecuario, competitivo, respetuoso de la madre tierra y el medio ambiente, construyendo un eco ciudad planificada; un municipio transparente, participativo, eficaz, eficiente, pujante y promotor del desarrollo Departamental y Nacional, en base a los siguientes pilares:

Viacha, promoviendo el Desarrollo Humano Integral,

Viacha, Productiva – Competitiva,

Viacha, Eco Ciudad Industrial,

Viacha, en armonía con la Madre Tierra,

Viacha, Unida por el desarrollo Municipal, Departamental y Nacional

La gestión de residuos sólidos, está a cargo del Gobierno Municipal de Viacha, operativamente depende de la Dirección de Medio Ambiente, responsable de supervisar y velar por el cumplimiento en la prestación de los servicios y paralelamente del control de actividades inherentes a la temática ambiental.

[1] m.s.n.m.: metros sobre el nivel del mar.

### 1.2.2. Antecedentes de proyectos

Haciendo referencia a trabajos realizados a nivel internacional, nacional y local se puede citar los siguientes:

**Antecedentes de proyectos nacionales**

- En el proyecto denominado “Sistema domótico de seguridad perimetral basado en Arduino” de (Choque, 2016), se diseña el prototipo de hardware basado en tecnologías libres haciendo uso de la plataforma de desarrollo de hardware Arduino versión Uno, junto a un módulo de comunicación GSM (SIM900) la cual se encarga de enviar alertas al usuario. Este sistema hace uso de otros módulos complementarios como ser sensores y relés para el control de seguridad.
- Por otro lado en el proyecto denominado “Internet de las Cosas, control y seguimiento de un automóvil”, se observa que se implementa un sistema de seguimiento para los automóviles a través de una aplicación Android. La misma hace uso de la metodología Mobil-D orientada al desarrollo de aplicaciones móviles. En cuanto al hardware hace uso de la placa electrónica de desarrollo Arduino junto al módulo GSM (SIM900) la cual es esencial para la comunicación. (Quispe, 2016)
- (Limachi, 2016) nos muestra la implementación de un sistema de administración remota para el monitoreo de sensores en base a radiofrecuencia aplicando Arduino, para tal efecto, en la parte principal uso un modelo en v para el desarrollo del software, es necesario aclarar, el lenguaje de programación utilizado es Java. Para la comunicación de radiofrecuencia utiliza el modulo RF basado en el chip Nordic Nrf24L01.
- En el proyecto de (Apaza, 2016) se realiza la construcción de un dispositivo constituido por elementos electrónicos (Hardware) y el desarrollo de un aplicación (Software), una herramienta capaz de brindar información sobre la composición de los estratos del subsuelo en base a la resistividad eléctrica. Para el desarrollo de la aplicación hace uso de la

metodología Mobile-D, haciendo uso en la parte electrónica del dispositivo Arduino como plataforma de desarrollo, el proyecto esta titulado como: “Prospección geo eléctrica de agua subterránea mediante dispositivos móviles y Arduino”.

**Antecedentes de proyectos internacionales**

- En el proyecto de (Biundo, 2008) denominado “Diseño de un Sistema de Seguridad Y Monitoreo de Vehículos”, se realiza la implementación de un sistema que nos ofrece la posibilidad de adaptar un sistema de monitoreo y seguridad a los automóviles para que se pueda realizar el control de la ubicación geográfica y la velocidad de vehículos con el uso de un modem y un sistema de localización que dotara de la información necesaria en el sistema, el diseño del hardware se la realiza en base a diagramas de bloques de cada etapa del proyecto diseño y simulación con el software PROTEUS y para el software de control se la implementa en base a diagramas de flujo y diagramas de estado para estudiar el comportamiento del sistema.
- (Astudillo & Delgado, 2012) Se realiza la implementación de un “Sistema de Localización Monitoreo y Control Vehicular Basado en los Protocolos Gps/Gsm/Gprs”, la cual consiste en el usos de los protocolos de comunicación GPS y GSM/GPRS en conjunto con los servidores de aplicaciones web y sockets las cuales en conjunción valga la redundancia tienen el objetivo de proveer la funcionalidad de optimizar rutas para el eficiente uso de la gasolina, controlar la velocidad de los vehículos y así también ofrece la posibilidad de obtener reportes detallados del monitorio de las mismas. Para la implementación de la base de datos que contendrá toda la información se hizo el uso de MYSQL como lenguaje de programación se hace el uso de PHP junto a otras varias tecnologías como los son HTML5, JAVASCRIPT Y AJAX.

- (De León, 2013) Con el título “Control de Navegación con Arduino de un Modelo de Barco”, donde se realizar el uso de la plataforma de hardware libre ARDUINO junto a su entorno de desarrollo integrado para la programación del Atmega 2560 el cual es el cerebro de todo, en cuanto al hardware esta plataforma estará acompañado de sensor y actuadores que le darán un grado de autonomía a una maqueta de barco.

## 1.3. PLANTEAMIENTO DEL PROBLEMA

La dirección de medio ambiente del Gobierno Municipal de Viacha realiza el monitoreo del cumplimiento de rutas de los carros basureros en coordinación con los encargados de conducir los carros basureros apelando al trabajo consciente de los mismos.

La población no cuenta con un medio que le permita verificar la ubicación y el tiempo de llegada del carros basureros para de esta manera preparar los residuos sólidos y entregar a tiempo las mismas, el perjuicio hacia el municipio es: al no depositar los residuos en los carros basureros los pobladores proceden a depositarlos, valga la redundancia, en lugares improvisados no autorizados causando males al medio ambiente.

### 1.3.1. Problema principal

Actualmente el servicio de recojo de basura es parte esencial, necesaria e importante en el Municipio de Viacha, pero la misma cuenta con falencias y la principal radica en que no existe un buen monitoreo de los carros basureros encargados de ofrecer el servicio de recojo, lo que genera ineficiencia en el manejo de la basura y daños a nuestro medio ambiente.

### 1.3.2. Problemas Secundarios

- Ausencia de un monitoreo en tiempo real de los vehículos recolectores de basura.
- Falta de información, a la población en general, de las rutas establecidas para los vehículos recolectores de basura.

- Ausencia de reportes de rutas y puntos de acopio de basura autorizada para la información de la población en general.
- Dificultad en la toma de decisiones para el mejoramiento del servicio de recolección de basura.

### 1.3.3. Formulación del problema

¿De qué manera un SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS coadyuvar en el mejoramiento del servicio de recojo de basura en el municipio de Viacha?

## 1.4. OBJETIVOS

### 1.4.1. Objetivo general

Implementar un sistema de geo localización para el monitoreo de los vehículos encargados del recojo de basura aplicando en el concepto de internet de las cosas, haciendo uso de tecnologías libres que coadyuve al mejoramiento del servicio.

### 1.4.2. Objetivos Específicos

- Diagnosticar el funcionamiento actual del servicio de recojo de basura.
- Realizar el análisis y el diseño del sistema en base a los requerimientos de la institución.
- Analizar los costos de acuerdo a la implementación del presente proyecto.
- Implementar el diseño del prototipo para la geo localización correspondiente de los vehículos encargados del recojo de basura.
- Aplicar los conceptos de internet de las cosas para la conectividad del prototipo de geo localización.
- Generar reportes de rutas realizadas por los vehículos encargados del recojo de basura.

## 1.5. JUSTIFICACIÓN

### 1.5.1. Técnica

El presente proyecto pretende proveer una herramienta monitoreo implementado con tecnologías libres acorde a las necesidades de la Dirección de Medio Ambiente del Gobierno Autónomo Municipal de Viacha y así también a la necesidad de la población, la cual permita mejorar el recojo de basura y así también a la toma de decisiones. Esta contribuirá con la información necesaria acerca del recojo de basura así como las rutas que realizara los vehículos a partir de un hardware de localización para mayor eficiencia en el recojo de basura.

La institución cuenta con el equipo necesario y la predisposición para la implementación del proyecto de esta manera justificándose técnicamente.

### 1.5.2. Social

El presente proyecto ofrecerá una plataforma bajo la cual toda persona podrá acceder para verificar que carros basureros están más cerca y poder realizar el recojo correspondiente y puntual de la basura ya que no existirá ningún tipo de búsqueda a los carros basureros, el sistema como tal ofrecerá la posibilidad de localizarlos ofreciendo mayor eficiencia en el manejo de la basura de esta manera beneficiándose el Municipio de Viacha, por lo tanto se justifica porque se constituirá en un aporte de gran ayuda para el mejoramiento del servicio de recojo de basura.

### 1.5.3. Económica

En cuanto al hardware necesario, ya que la plataforma es libre, el costo no será muy elevado por ser una tecnología bastante extendida, el software que se pretende implementar mediante herramientas basadas en software libres para aligerar mucho más los costos de desarrollo del presente proyecto.

Así también el sistema a implementar pretende coadyuvar en la toma de decisiones pudiendo analizar las rutas para hacer uso de la menor cantidad de recurso por parte del municipio.

## 1.6. METODOLOGÍA

### 1.6.1. Método de Recolección de datos

Para la recolección de datos, se la realizará a través de una observación del cumplimiento del servicio, así también de cuestionarios a los encargados de supervisar y administrar el servicio, y por ultimo una entrevista con el encargado de la Unidad de Medio Ambiente para tener conocimiento de todo el proceso realizado del servicio de recojo de basura en el Municipio de Viacha.

### 1.6.2. Método de Ingeniería

En el desarrollo de aplicaciones Web es muy importante un debido análisis y diseño, en los cuales el modelamiento y planificación empleados de una manera adecuada van a permitir obtener un producto de calidad que satisfaga todos los requerimientos del cliente.

La metodología por la cual se opta se describe como UWE la cual define modelos representados gráficamente por diagramas en UML,

**UML-Based Web**

Se trata de un método que hace uso de técnicas procedentes de la orientación a objetos para especificar aplicaciones hipermedia. UWE plantea un enfoque iterativo y progresivo cuyas actividades fundamentales son el análisis de requisitos y el diseño conceptual, de la navegación y de la presentación.

La Ingeniería Web basada en UML (UML-Based Web Enginering, UWE), es un proceso de desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web, las cuales permiten una adecuada planificación del proyecto.

Las aplicaciones Web tienen características especiales como los requerimientos del cliente y el entorno en el que operan entre otros aspectos, para lo cual UWE ha definido varias vistas especiales como modelos de navegación y presentación, modelos que representan de una manera gráfica la funcionalidad, usabilidad y representación de la aplicación Web.

Para esto será utilizar todas las fases de UWE que son:

**Análisis de requisitos:** El objetivo es encontrar los requisitos funcionales de la aplicación web para representarlos como casos de uso. Esto da lugar a los diagramas de casos de uso.

**Modelo conceptual:** Se construye el modelo conceptual del dominio de la aplicación considerando los requisitos reflejados en los casos de uso. El resultado es el diagrama de clases de dominio.

**Modelo navegacional:** Se obtienen el modelo de espacio de navegación y el de estructura de navegación, que muestra como navegar a través del espacio de navegación.

**Modelo de presentación:** Se obtienen una serie de vistas de interfaz de usuario representadas mediante diagramas de interacción UML.

**Modelo de procesos:** El paquete de proceso proporciona un modelo de elementos para integrar procesos de negocio en un modelo de uso de web de UWE.

### 1.6.3. Métricas de Calidad

Se evaluará la calidad de software usando la norma ISO 9126. La norma ISO 9126 es un estándar internacional para la evaluación de la calidad del Software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad de software. La calidad de software se refiere a: “Los factores de un producto de software que contribuyen a la satisfacción completa y total de las necesidades de un usuario u organización”. (Académicos Intercontinentales, 2019)

Las métricas de calidad proporcionan una indicación de cómo se ajusta al software, a los requerimientos implícitos y explícitos del cliente.

La normativa define seis características de la aplicación, las cuales son:

- ✓ **Funcionalidad:** Adecuación, exactitud, interoperabilidad y seguridad.
- ✓ **Confiabilidad:** Madurez, tolerancia a fallas y recuperabilidad.
- ✓ **Usabilidad:** Entendibilidad, capacidad de aprendizaje y operabilidad.
- ✓ **Eficiencia:** Comportamiento en tiempos y comportamiento de recursos.
- ✓ **Mantenibilidad:** Analizabilidad, modificabilidad, estabilidad y capacidad de prueba.
- ✓ **Portabilidad:** adaptabilidad, instabilidad y reemplazabilidad.

### 1.6.4. Costos

Una parte muy importante en la elaboración de software y hardware es la estimación de costos, hacemos referencia a la estimación de costos, a evaluar los esfuerzos ya sean económicos de tiempo y de personal que serán necesario en función al tamaño del proyecto. Para la misma existen diferentes metodologías, pero la principal y la más extendida es COCOMO (Modelo Constructivo de Costos). A continuación veremos las características principales.

#### 1.6.4.1. COCOMO (Modelo Constructivo de Costos)

El Modelo Constructivo de Costes COCOMO, es utilizado en proyectos de software para estimar los costes del mismo en función de tres submodelos: básico, intermedio y detallado (Escuela de Organizacion Industrial, 2012).

**Figura Nº 1.1:** Modelo Constructivo de Costos COCOMO


**Fuente:** (Escuela de Organizacion Industrial, 2012)

## 1.7. HERRAMIENTAS

### 1.7.1. Software

#### 1.7.1.1. Angular

Angular es un framework de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitarnos el desarrollo de aplicaciones web SPA y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y óptima.
Otro propósito que tiene Angular es la separación completa entre el front-end y el back-end en una aplicación web. (Robles, 2019) .

#### 1.7.1.2. NodeJS

Node.js es un entorno JavaScript de lado de servidor que utiliza un modelo asíncrono y dirigido por eventos. Es una Máquina Virtual tremendamente rápida y de gran calidad escrita por gente como Lars Bak, uno de los mejores ingenieros del mundo especializados en VMs(Virtual Machines). No olvidemos que V8 es actualizado constantemente y es uno de los intérpretes más rápidos que puedan existir en la actualidad para cualquier lenguaje dinámico. Además las capacidades de Node.js para I/O (Entrada/Salida) son realmente ligeras y potentes, dando al desarrollador la posibilidad de utilizar a tope la I/O del sistema (Dev Code, 2019).

#### 1.7.1.3. Sequelize

Cuando hacemos algún desarrollo del lado del backend una de las tareas más comunes que podemos realizar es manipular bases de datos(Insertar, buscar, actualizar, borrar), para esto generalmente se escribe directamente la consulta SQL en el lenguaje de programación y asi conseguir los datos, un ORM (Object-Relational mapping) nos permite convertir tablas de una base de datos en entidades en un lenguaje de programación orientado a objetos, lo cual agiliza bastante el acceso a estos datos (EDteam, 2019).

#### 1.7.1.4. Gestor de Base de Datos PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional de objetos (ORDBMS), desarrollado en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley. Es compatible con un gran del estándar SQL y ofrece muchas características modernas:

- ✓ Consultas complejas
- ✓ Llaves extrajeras
- ✓ Desencadena
- ✓ Vistas actualizables
- ✓ Integridad transaccional
- ✓ Control de ocurrencia multiversion

Además, el usuario puede ampliar PostgreSQL de muchas maneras, por ejemplo, agregando nuevos

- ✓ Tipos de datos
- ✓ Funciones
- ✓ Operadores
- ✓ Funciones agregadas
- ✓ Método de índice
- ✓ Lenguaje de procedimiento

Y debido a la licencia liberal, PostgreSQL puede ser utilizado, modificado y distribuido por cualquier persona de forma gratuita para cualquier propósito, ya sea privado, comercial o académico (The PostgreSQL Global Development Group, 2020).

#### 1.7.1.5. Leaflet

Leaflet es la biblioteca de JavaScript de código abierto líder para mapas interactivos aptos para dispositivos móviles, tiene todas las características de mapeo que la mayoría de los desarrolladores necesitan. Leaflet está diseñado teniendo en cuenta la simplicidad, el rendimiento y

la usabilidad. Funciona de manera eficiente en todas las principales plataformas de escritorio y móviles, se puede ampliar con muchos complementos , tiene una API hermosa, fácil de usar y bien documentada y un código fuente simple y legible al que es un placer contribuir (Agafonkin, 2020).

#### 1.7.1.6. NestJS

Nest(o NestJS) es un framework para crear aplicaciones de Backend sobre NodeJS de manera eficiente y escalable. Utiliza JavaScript progresivo, está basado y completamente construido en TypeScript. (Mysliwiec & Staron, 2019).

#### 1.7.1.7. JWT (Json Web Token)

JSON Web Token (JWT) es un estándar abierto ( RFC 7519 ) que define una forma compacta y autónoma para transmitir de forma segura información entre las partes como un objeto JSON. Esta información puede ser verificada y confiable porque está firmada digitalmente. Los JWT se pueden firmar usando un secreto (con el algoritmo HMAC) o un par de claves pública / privada usando RSA o ECDSA (Auth0, 2020).

Aunque los JWT pueden cifrarse para proporcionar también el secreto entre las partes, nos centraremos en los tokens firmados. Los tokens firmados pueden verificar la integridad de las reclamaciones que contiene, mientras que los tokens encriptados ocultan esas reclamaciones de otras partes. Cuando los tokens se firman utilizando pares de clave pública / privada, la firma también certifica que solo la parte que posee la clave privada es la que la firmó

**Estructura de un JWT**

La estructura básica de los Tokens web está compuesta por 3 partes esenciales separados por punto:

- Encabezamiento
- Carga útil
- Firma

Por lo tanto, un JWT normalmente se parece a lo siguiente: xxxxx.yyyyy.zzzzz

**Cabecera:** La primera parte es la cabecera del token, que a su vez tiene otras dos partes, el tipo, en este caso un JWT y la codificación utilizada.
Comúnmente es el algoritmo HMAC SHA256

**Carga útil:** El Payload está compuesto por los llamados *JWT Claims* donde irán colocados los atributos que definen nuestro token. Existen varios que puedes consultar aquí, los más comunes a utilizar son:

- **sub:** Identifica el sujeto del token, por ejemplo un identificador de usuario.
- **iat:** Identifica la fecha de creación del token, válido para si queremos ponerle una fecha de caducidad. En formato de tiempo UNIX
- **exp:** Identifica a la fecha de expiración del token. Podemos calcularla a partir del iat. También en formato de tiempo UNIX.

**Firma:** La firma es la tercera y última parte del JSON Web Token. Está formada por los anteriores componentes (Cabecera y Carga útil) cifrados en *Base64* con una clave secreta (almacenada en nuestro backend). Así sirve de *Hash* para comprobar que todo está bien.

#### 1.7.1.8. QGis

QGIS es un Sistema de Información Geográfica (SIG) de Código Abierto licenciado bajo GNU - General Public License . QGIS es un proyecto oficial de Open Source Geospatial Foundation (OSGeo). Corre sobre Linux, Unix, Mac OSX, Windows y Android y soporta numerosos formatos y funcionalidades de datos vector, datos ráster y bases de datos. (QGIS, 2020).

### 1.7.2. Hardware

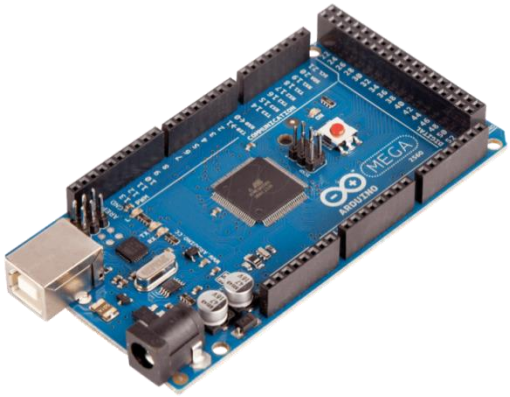
#### 1.7.2.1. Arduino Mega 2560

Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un

sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida: activar un motor, encender un LED, publicar algo en línea. Puede decirle a su placa qué hacer enviando un conjunto de instrucciones al microcontrolador de la placa. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en el cableado ) y el software Arduino (IDE) , basado en el procesamiento. (Arduino, 2020)

El Arduino Mega 2560 es una placa de microcontrolador basada en el ATmega2560 . Tiene 54 pines de entrada / salida digital (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP, y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a una computadora con un cable USB o enciéndalo con un adaptador de CA a CC o una batería para comenzar. La placa Mega 2560 es compatible con la mayoría de los escudos diseñados para la Uno y las antiguas placas Duemilanove o Diecimila. (Arduino, 2020)

**Figura Nº 1.2:** Arduino Mega 2560




**Fuente:** (www032)

### 1.7.2.2. Módulo GSM/GPRS SIM 900

Módulo de sistema mínimo SIM900 con todos los componentes básicos para hacer funcionar el módulo GSM/GPRS SIM900 de la marca SIMCOM. Es ideal para evaluar el funcionamiento de dicho modulo e incluirlo en prototipos o

sistemas finalizados con el cableado adecuado. Se comunica con el microcontrolador a través de una interfaz serial y comandos AT. El módulo soporta reset y encendido mediante hardware. Los módulos Simcom también se utilizan en shields para arduino, por lo que podemos utilizar este módulo también con arduino o cualquier microcontrolador o computadora como el Raspberry Pi o PC de escritorio. (Geek Factory, 2020).

#### 1.7.2.3. Modulo GPS NEO-6M

Módulo GPS para Arduino y microcontroladores, basado en el receptor de la marca Ublox modelo NEO 6M, el módulo incluye su antena cerámica para colocarse directamente sobre el PCB, por lo que ya viene listo para operar sin requerir más accesorios.

El GPS para Arduino puede funcionar con un voltaje de alimentación en el rango de 3.0 a 5.0 volts, mientras que las señales que entran y salen son de 3.3 volts, por lo que se requiere un convertidor de niveles si un arduino o microcontrolador de 5 volts va a comunicarse (transmitir) hacia el módulo GPS. Si solamente se desean recibir los datos NMEA basta con conectar el pin TX con el RX de arduino y recibir los datos que envía el módulo, en este caso, no hace falta conversión de niveles por que el arduino reconoce los 3.3 volts como nivel alto. (Geek Factory, 2020).

## 1.8. LÍMITES Y ALCANCES

### 1.8.1. Limites

El proyecto en su aplicación se enmarca estrictamente en el Municipio de Viacha, en un trabajo conjunto del Gobierno Autónomo de Viacha y la Dirección de Medio Ambiente. Por lo tanto la funcionalidad del proyecto se delimita geográfica al Municipio de Viacha como tal.

Así también el presente proyecto se realizará en función a los requerimientos de la institución, existirá un administrador quien tendrá acceso al sistema para su administración correspondiente y si fuese necesario se realizara las

modificaciones necesarias, siempre respetando los requerimientos primordiales de la institución, estará orientado al manejo de las ubicaciones de los vehículos encargados del recojo de basura, monitoreo del cumplimiento de las rutas, la posibilidad de notificar la ubicación a los usuarios y pudiéndose registrar datos característicos de la basura, la cual coadyuve en la toma de decisiones.

Es necesario aclarar que si bien se desarrollará un prototipo de hardware esta nos ofrecerá las funciones necesarias y básicas, dando como prioridad a la plataforma que procesará los datos por ende las características propias del prototipo como la alimentación y otras más profundas no son parte del objetivo de desarrollo del presente proyecto.

### 1.8.2. Alcances

En la presente se describirá la implementación de un sistema de monitoreo la cual pretende, como parte esencial, el monitoreo de los vehículos encargados del recojo de basura la cual coadyuve en la toma de decisiones y el mejoramiento del servicio. Por otro lado se pretende implementar una interfaz amigable para todos los usuarios, así también para los administradores del sistemas, de tal manera, que puedan agregar y registrar datos importantes y puedan generar informes necesarios a la hora de ofrecer el servicio.

Es necesario tomar en cuenta que este sistema será factible en entornos donde exista muy buena señal, ya que en conjunto trabajarán Arduino Mega, el módulo GSM/GPRS SIM 900 y el modulo GPS NEO-6M y las mismas harán uso de comunicación móvil a través de una SIMCARD que provee el módulo GSM/GPRS SIM 900, esto de acuerdo a una empresa de telefonía móvil en particular. Se hace bastante énfasis en el desarrollo del software que es parte muy importante la cual consta de los siguientes módulos: Modulo de Mapa de Geo localización, Modulo de puntos de Acopio, Modulo de Reportes y Modulo de control de usuarios, y no tanto así al hardware y más aún en temas como la alimentación propia del dispositivo.

## 1.9. APORTES

Se podrá monitorear rutas de los carros basureros en base a su ubicación para así de esta forma poder optimizar rutas para el recojo de basura.

Nos proporcionara una plataforma para la notificación de la ubicación de los carros basureros a través de una aplicación web para de esta manera realizar un recojo más efectivo de la basura.

Se podrán recolectar datos de la generación de basura en el municipio, almacenándolos en una base de datos la cual coadyuvará en la toma de decisiones para mejorar el servicio.

# CAPITULO II

# MARCO TEÓRICO

Este apartado tiene por objetivo realizar una descripción de los conceptos y definiciones sobre la cual se desarrollará el proyecto, tomándolo como base teórica. En esta sección se desarrollarán conceptos como Sistemas de control, Sistemas de monitoreo, Plataformas de hardware para el desarrollo de proyectos electrónicos, y así también sistemas de comunicación como son el protocolo de comunicación MQTT y el protocolo de comunicación GSM/GPRS. Por otro lado, también se hace referencia al uso de dispositivos receptores GPS.

## 2.1. SISTEMA

En muchos documentos académicos, según diferentes autores ya sea en el ámbito de la Ingeniería de Sistemas o particularmente en la teoría general de sistemas se hace referencia a este término obteniendo muchas definiciones y conceptualizaciones. Johansen (1993) define sistema como: “Un conjunto de partes coordinadas y en interacción para alcanzar un conjunto de objetivos” (p.54).

Un sistema es un conjunto de "elementos" relacionados entre sí, de forma tal que un cambio en un elemento afecta al conjunto de todos ellos. Los elementos relacionados directa o indirectamente con el problema (Garcia, 2019). Sin embargo, al definir que se trata de un conjunto de elementos es necesario aclarar que un elemento puede, también, referirse a un subsistema que a su vez agrupa otro conjunto de subelementos que trabajan en pro del subsistema general, entonces sistema también se puede entender como un conjunto de subsistemas las cuales se interrelacionan entre ellas para un fin en común.

## 2.2. ANALISIS DE SISTEMAS

Es un conjunto o disposición de procedimientos o programas relacionados de manera que juntos forman una sola unidad. Un conjunto de hechos, principios y reglas clasificadas y dispuestas de manera ordenada mostrando un plan lógico en la unión de las partes. Un método, plan o procedimiento de clasificación para

hacer algo. También es un conjunto o arreglo de elementos para realizar un objetivo predefinido en el procesamiento de la Información. Esto se lleva a cabo teniendo en cuenta ciertos principios:

- Debe presentarse y entenderse el dominio de la información de un problema.
- Defina las funciones que debe realizar el Software.
- Represente el comportamiento del software a consecuencias de acontecimientos externos.
- Divida en forma jerárquica los modelos que representan la información, funciones y comportamiento.

El proceso debe partir desde la información esencial hasta el detalle de la Implementación.

La función del Análisis puede ser dar soporte a las actividades de un negocio, o desarrollar un producto que pueda venderse para generar beneficios. Para conseguir este objetivo, un Sistema basado en computadoras hace uso de seis (6) elementos fundamentales:

- Software, que son Programas de computadora, con estructuras de datos y su documentación que hacen efectiva la logística metodología o controles de requerimientos del Programa.
- Hardware, dispositivos electrónicos y electromecánicos, que proporcionan capacidad de cálculos y funciones rápidas, exactas y efectivas (Computadoras, Censores, maquinarias, bombas, lectores, etc.), que proporcionan una función externa dentro de los Sistemas.
- Personal, son los operadores o usuarios directos de las herramientas del Sistema.
- Base de Datos, una gran colección de informaciones organizadas y enlazadas al Sistema a las que se accede por medio del Software.
- Documentación, Manuales, formularios, y otra información descriptiva que detalla o da instrucciones sobre el empleo y operación del Programa.

- Procedimientos, o pasos que definen el uso específico de cada uno de los elementos o componentes del Sistema y las reglas de su manejo y mantenimiento.

Un Análisis de Sistema se lleva a cabo teniendo en cuenta los siguientes objetivos en mente:

- Identifique las necesidades del Cliente.
- Evalúe que conceptos tiene el cliente del sistema para establecer su viabilidad.
- Realice un Análisis Técnico y económico.
- Asigne funciones al Hardware, Software, personal, base de datos, y otros elementos del Sistema.
- Establezca las restricciones de presupuestos y planificación temporal.
- Cree una definición del sistema que forme el fundamento de todo el trabajo de Ingeniería.

Para lograr estos objetivos se requiere tener un gran conocimiento y dominio del Hardware y el Software, así como de la Ingeniería humana (Manejo y Administración de personal), y administración de base de datos (Monografias.com, 2020).

## 2.3. DISEÑO DE SISTEMAS

El diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad. Los principios de diseño establecen una filosofía general que guía el trabajo de diseño que debe ejecutarse. Deben entenderse los conceptos de diseño antes de aplicar la mecánica de éste, y la práctica del diseño en sí lleva a la creación de distintas representaciones del software que sirve como guía para la actividad de construcción que siga. El objetivo del diseño es producir un modelo o representación que tenga resistencia, funcionalidad y belleza (Pressman, Ingeniería de Software - Un enfoque práctico, 2003, pág. 183).

## 2.4. SIG (SISTEMA DE INFORMACION GEOGRAFICA)

De acuerdo a (Olaya, 2014) define como: “Sistema de información diseñado para trabajar con datos referenciados mediante coordenadas espaciales o geográficas. En otras palabras, un SIG es tanto un sistema de base de datos con capacidades específicas para datos georreferenciados, como un conjunto de operaciones para trabajar con esos datos. En cierto modo, un SIG es un mapa de orden superior” (p.7).

Así también el autor (Geoinnova Formación SIG y Medio Ambiente, 2020) define a un SIG como: “Es un conjunto de componentes específicos que permiten a los usuarios finales crear consultas, integrar, analizar y representar de una forma eficiente cualquier tipo de información geográfica referenciada asociada a un territorio. La información geográfica va a ser aquella información que tiene algún componente espacial, es decir, una ubicación, y además, una información atributiva que nos detalle más sobre ese elemento en cuestión. Esa ubicación se podrá definir con un nombre de una calle, por ejemplo, o con coordenadas espaciales”.

### 2.4.1. Componentes de un SIG

A continuación, veremos, cada uno de los componentes que conforman un Sistema de Información Geográfica (SIG).

1) **Datos.**

   Los datos son la materia prima para trabajar con los Sistemas de Información Geográfica (SIG). Sin ellos, no podremos construir productos de información o mapas que nos ayuden a hacer nuestros análisis y tomar las decisiones en nuestra organización. Esos datos podrán venir de diferentes fuentes: sensores remotos, GPS, fotografías aéreas, archivos formatos shapefile, archivos CAD, archivos Excel, etc.

   Esta información geográfica será el inicio de partida para empezar a trabajar con los SIG, los cuales nos permitirán analizarla y extraer toda la

información posible para plasmarla en un mapa que nos ayude a la interpretación de esa información.

2) **Software.**

Para el correcto análisis e interpretación de la información geográfica es necesaria la participación de un software SIG que tenga la potencia y funcionalidad de trabajar con información de este tipo.

Hoy en día existe bastante software SIG en el mercado que nos ponen a disposición herramientas SIG para el tratamiento de la información geográfica. A continuación, nombraremos los más comunes y/o utilizados.

ArcGIS es actualmente la tecnología de referencia en los Sistemas de Información Geográfica (SIG). Esta tecnología ha sido desarrollada y mejorada año tras año por la compañía propietaria ESRI (Enviromental Systems Research Institute) desde hace más de 30 años. Actualmente ofrece una plataforma a nivel escritorio, servidor, online y aplicaciones que permite una interoperabilidad completa a la hora de trabajar con los Sistemas de Información Geográfica (SIG). En el próximo artículo podrás ver mejor qué es ArcGIS.

3) **Hardware.**

Como es lógico, para poder utilizar algún software mencionado anteriormente es necesario un ordenador o hardware. Dependiendo de las características de esta máquina, obtendremos un mayor o menor rendimiento a la hora de realizar nuestros análisis. Dentro de las características del hardware a tener en cuenta para análisis de información geográfica con software SIG deberíamos incluir las siguientes:

- Sistema operativo: Windows, Mac, Linux.
- RAM
- Disco duro
- CPU: 64 o 32 bits.

- Tarjeta gráfica (para visualizaciones 3D)

4) **Personas.**

Una vez tenemos los datos y con qué analizarlos, necesitamos saber cómo. Aquí es donde entramos en juego los **profesionales SIG**. Y es que el profesional SIG es un perfil muy cuestionado (y demandado) en los últimos años, ya que existen muchas tareas dentro de un análisis SIG, las cuales necesitan de uno o varios profesionales, incluso profesionales temáticos. Dentro de los **perfiles SIG** podemos encontrar dos perfiles fundamentales:

- Técnico/Analista SIG. Profesional que se encarga de realizar análisis geográficos y obtener resultados acorde con la investigación o proyecto que se esté llevando a cabo.
- Programador SIG. Desarrollador de partes funcionales de un SIG de escritorio (o de servidor) y /o de aplicativos web para la visualización de mapas. Además de eso, y dado que los SIG están creciendo tantísimo hoy en día, se pueden encontrar perfiles como Administrador SIG, Gerentes de cuenta SIG, o directores SIG. Todo dependerá de las necesidades de los proyectos.

5) **Procesos.**

Un SIG exitoso opera de acuerdo a un buen diseño de reglas de implementación y de negocios, que son los modelos y prácticas de operación únicas para cada organización.

Al igual que en todas las organizaciones relacionadas con la tecnología sofisticada, las nuevas herramientas sólo se pueden utilizar con eficacia si se integran adecuadamente en toda la estrategia empresarial de la organización Para hacer esto correctamente, se requiere no sólo de las inversiones necesarias en hardware y software, sino también en el reciclaje y / o contratación de personal para utilizar la nueva tecnología en el contexto de la organización adecuada. La aplicación de su SIG sin

tener en cuenta el compromiso organizacional adecuado resultará en un sistema sin éxito.

Simplemente no es suficiente para una organización la compra de un ordenador con algún tipo de software SIG, contratar a algunos individuos entusiastas y esperar un éxito inmediato (Geoinnova Formación SIG y Medio Ambiente, 2020).

### 2.4.2. Aplicaciones

Las aplicaciones más comunes de los SIG en la administración incluyen la gestión urbanística, la distribución de los servicios sanitarios, el planeamiento de los servicios de transporte urbano y gestión de datos catastrales entre otros.

**Administración y gestión**

En Administración y gestión el uso de los sistemas de información geográfica garantiza la integración del componente espacial a la información gestionada y facilita la toma de decisiones políticas.

Las aplicaciones más comunes de los SIG en la administración incluyen la gestión urbanística, la distribución de los servicios sanitarios, el planeamiento de los servicios de transporte urbano y gestión de datos catastrales entre otros.

**Actividades físicas y deportivas**

La creación de rutas de senderismo, circuitos de ciclismo o maratón y la optimización de tramos a partir de mapas de carreteras o sendas son algunas de las aplicaciones en las actividades físico deportivas recreativas. Para programar este tipo de actividades el uso de los sistemas de información geográfica se hace indispensable, resultando tedioso e ineficaz trabajar sin ellos.

**Agraria**

El sector agrario es uno de los sectores más dependientes del uso de sistemas de información geográfica. En el área forestal resulta indispensable tener un conocimiento amplio de estas herramientas para el reconocimiento, gestión y

evolución del entorno. El trabajo sobre grandes áreas forestales sin el uso de software GIS es hoy en día difícil, tedioso y costoso. Trabajando con estos sistemas podremos hacer un reconocimiento de una amplia zona sin la necesidad de visitarla y en un corto periodo de tiempo, además, el conocimiento de los SIG permite al usuario trabajar con LIDAR o Teledetección, dos herramientas que cobran valor año a año en la gestión forestal resolviendo situaciones de trabajo que previamente suponían plazos y costes muy elevados con sencillez y rapidez; cálculo de biomasa, estudio de incendios. El área agrícola también tiene una gran dependencia sobre estas tecnologías. La agricultura de precisión pasa por manejar sistemas de información geográfica; gestión de información proveniente de drones o teledetección, control y reconocimiento del cultivo en grandes áreas reduciendo los costes y el plazo empleado. En el área de la jardinería también se mejoran los procesos con el uso de SIG, cuando esta se realiza en grandes áreas o en muchas pequeñas áreas, por ejemplo, en la gestión de la jardinería de una pequeña ciudad por parte de una empresa, con el uso de sistemas de información geográfica, se haría un estudio de las rutas más cortas para la poda o corte del césped y se haría un control espacial del regadío para garantizar la optimización del trabajo.

**Comercio y Marketing**

Los avances tecnológicos y el hecho de que buena parte de las actividades humanas tengan un componente locacional, provocan que en casi cualquier ámbito laboral se puedan mejorar los procesos desarrollados añadiendo Sistemas de Información Geográfica al trabajo diario. Uno de los ejemplos es el Comercio y Marketing, las grandes empresas y cada vez más las pequeñas empresas añaden el uso de SIG para el desarrollo de su actividad, contando con la información de la ubicación del cliente que ya compró un producto se puede estudiar bajo un software SIG el objetivo centralizado de márketing y la optimización de rutas de reparto.

**Edificación y obra civil**

El proceso de planificación del territorio y la ciudad, supone manejar grandes volúmenes de información gráfica y descriptiva. Los Sistemas de Información Geográfica son la tecnología que nos permite manejar dicha información y su análisis. Las aplicaciones de los Sistemas de información Geográfica en edificación y obra civil son la planificación y diseño de obras civiles, gestión de redes de servicios públicos, planes de protección ambiental, ordenamiento territorial y urbano y análisis de riesgos.

**Electricidad y electrónica**

En el área de electricidad y electrónica el uso de SIG es preciso para gestionar todos los datos tomados en campo y diseñar las redes eléctricas o cualquier red de telecomunicaciones. Actualmente gestionar toda la información disponible y diseñar estas redes con cualquier otra herramienta es un proceso muy tedioso y costoso, las interacciones entre las distintas redes, la optimización de la red y la gestión de toda la información disponible en cada una de las redes hacen que el SIG sea indispensable para el desarrollo y gestión de las redes. Además, la supervisión de las redes, quita de nidos, reparación de torres, etc. se optimiza por completo mediante el uso de un software SIG. Las empresas de electricidad y electrónica que diseñan y gestionan redes mejoran de forma radical los plazos y costes mediante el uso de estas herramientas, el conocimiento de estas por parte del trabajador hace que tenga un valor añadido.

**Energía y agua**

El uso de los SIG ha sido ampliamente difundido para la gestión del agua urbana. Un paso de desarrollo en este campo ha sido utilizar información SIG no solo para mapear y realizar consultas, sino para analizar tendencias y tomar decisiones mediante las aplicaciones que brindan los análisis espaciales. Actualmente, la gestión de las redes de abastecimiento y saneamiento en una ciudad es un proceso que no se comprende sin sistemas de información

geográfica, tratar de diseñar y gestionar una red compleja de estas características sin esta herramienta significa ampliar plazos y presupuesto. Además, los sistemas de información geográfica son una herramienta muy útil para la gestión de cuencas hidrográficas; tratamiento y estudio de datos, análisis de crecidas, tiempos de concentración y diagnosis de pendientes.

**Hostelería y turismo**

En el área de turismo, la importancia de los SIG radica en el manejo que puede darse a la información geográfica, haciendo posible la representación del territorio. En esta área los SIG han permitido por ejemplo la elaboración de bases de datos turísticas, mapas, videos, aplicaciones y páginas web en donde se muestran los atractivos, algunas veces en tiempo real, así como sus coordenadas exactas. Esto se ha realizado con el fin de fortalecer y/o potencializar el turismo dentro de un territorio específico.
La generación de rutas o senderos, la administración y gestión de zonas turísticas y la creación de planes de visita para el turista son tres usos potenciales de SIG en el área turística.

**Industrias alimentarias**

Los Sistemas de Información Geográfica son ampliamente empleados en los procesos de toma de decisiones y en el control de diversos recursos. En la industria alimentaria se utilizan tanto en la fase de transporte como en la fase inicial de gestión de ubicaciones de almacenes, sedes y proveedores conforme a criterios de distancias, socioeconómicos y telecomunicaciones.

**Seguridad y medio ambiente**

Los Sistemas de Información Geográfica son la herramienta clave para la gestión de cualquier recurso ya sea natural o antrópico. Permiten desarrollar cartografía básica con la que gestionar los recursos para, posteriormente, analizarla, representarla y plantear estrategias de gestión de manera coherente, optimizada y viable. Generación de cartografía básica, modelos predictivos de

distribución de especies, gestión de biodiversidad, evaluaciones de impacto ambiental, gestión hidrológica, incendios y gestión forestal, gestión de espacios naturales y gestión de plagas y especies invasoras. Además, mediante la generación bases de datos sólidas se realizan análisis de los cambios en el territorio y el paisaje. Por ello ambos métodos de trabajo se convierten para cualquier profesional vinculado con el medio ambiente en las herramientas necesarias y obligatorias para la toma de decisiones en la ordenación del territorio y el paisaje.

**Transporte y mantenimiento de vehículos**

Los SIG, gracias a su capacidad de gestión espacial, son sistemas indicados y muy aptos para prestar su ayuda a los sistemas de transporte ya que permiten controlar parámetros en diferentes variables; Mantenimiento y Conservación de Infraestructuras, Tráfico, Gestión, Impactos nuevas infraestructuras, Sistemas de Navegación para vehículos. La gestión del transporte público es uno de los usos potenciales de los Sistemas de Información geográfica, permitiendo la gestión de las distintas rutas, optimización de distancias y tiempos de ruta, definición de paradas conforme a criterios de distancia, densidad de población y accesibilidad y análisis del tráfico. Además, en el transporte de mercancías también tiene un papel importante el uso de esta herramienta, optimizando tiempos, distancias, organizando el recorrido y ordenando los destinos, de esta forma, al implementar los sistemas de información geográfica en una empresa de transporte (formación, el software es libre), empezará a revertir económicamente al estar todas las rutas optimizadas.

### 2.4.3. Modelo de Datos

Los datos son, como ya sabemos, una parte imprescindible del SIG, ya que sin ellos las aplicaciones SIG y los restantes elementos que se encuentran en torno a estas no tienen utilidad alguna. Necesitamos conocer el área geográfica que estudiamos en un SIG (es decir, tener datos sobre ella), para así poder proceder a dicho estudio.

No obstante, convertir esa área geográfica y la información acerca de ella en un dato susceptible de ser incorporado a un SIG no resulta una tarea sencilla.

Desde los orígenes de los SIG, una de las preocupaciones principales ha sido la de representar de la mejor manera posible toda la información que podemos extraer de una zona geográfica dada, de tal modo que pueda almacenarse y analizarse en el entorno de un SIG. Este proceso de representación, que ya desde el inicio planteaba problemas a los creadores de los primeros SIG, ha sido el responsable en gran medida de la arquitectura y forma de los SIG actuales, y a él se debe en buena parte el desarrollo que han experimentado tanto los SIG en sí como las disciplinas afines.

Describir los enfoques teóricos existentes para convertir la realidad relativa a una variable dada en una capa que la contenga de la forma más precisa posible y pueda ser empleada en un SIG es el objeto de este capítulo. Este proceso implica la construcción de un modelo (el dato geográfico), que representa la realidad y puede servir para conocer esta en profundidad a través de análisis que no se llevan a cabo sobre dicha realidad, sino sobre el modelo en sí.

El problema principal reside en el hecho de que el detalle real que encontramos en la naturaleza es prácticamente infinito, mientras que la representación y almacenamiento de esa realidad es finito. Se hace necesario extraer una serie de elementos y valores característicos, los cuales en última instancia se recogerán como valores interpretarse como el anteriormente citado modelo. El camino que lleva desde la realidad hasta ese conjunto de meros valores numéricos pasa por tres niveles:

Establecimiento de un modelo geográfico. Es decir, un modelo conceptual de la realidad geográfica y su comportamiento.

Establecimiento de un modelo de representación. Es decir, una forma de recoger el anterior modelo conceptual y sus características propias, reduciéndolo a una serie finita de elementos.

Establecimiento de un modelo de almacenamiento. Es decir, un esquema de cómo almacenar los distintos elementos del modelo de representación.

El modelo geográfico es un ente puramente conceptual (de alto nivel), mientras que el de almacenamiento es más un concepto técnico inherente a la naturaleza informática del SIG (de bajo nivel) (Olaya, 2014, p. 79-80).

### 2.4.4. Monitoreo

El término monitoreo podría definirse como la acción y efecto de monitorear. Pero otra posible acepción se utilizaría para describir a un proceso mediante el cual se reúne, observa, estudia y emplea información para luego poder realizar un seguimiento de un programa o hecho particular. Esta palabra monitoreo no se encuentra en el diccionario de la real academia y viene de la voz «monitor» que recolecta imágenes y vídeos directamente desde las filmadora o cámaras las cuales permite la correcta visualización de una serie de hechos por medio de una pantalla; es decir que el monitor ayuda y permite a inspeccionar, controlar y registrar una circunstancia o situación; y allí es donde nace monitoreo para poder realizar el hecho, o que usualmente se dirige a los procesos en lo que refiere como, cuando y donde dan lugar las actividades, quien las realiza y cuantos individuos o entidades podría beneficiar. Y el verbo de éste es «monitorear» que es la acción como tal de supervisar y controlar a través de un monitor (ConceptoDefinicion, 2020).

#### 2.4.4.1. GPS (Sistema de posicionamiento global)

El sistema de posicionamiento global tiene por objetivo proveernos la ubicación geográfica y la hora de un receptor en particular de manera gratuita y en cualquier parte del mundo siempre y cuando se cuente con un receptor apropiado. Fue creado por el Departamento de Defensa de Estados Unidos a finales de la década de 1960, fue bastante extendida en el uso de la navegación marítima y aeronáutica.

El funcionamiento consiste básicamente en un sistema de satélites orbitando a la tierra a una altura de 22 Km., las cuales están ubicadas de manera estratégica alrededor del globo terráqueo, los dispositivos receptores capturan los datos enviados de al menos 4 satélites a partir de las cuales se realizar el cálculo del receptor hacia los satélites, habiendo realizado dicho cálculo se realizar un proceso denominado trilateración obteniendo como resultado la latitud, longitud, altura y la hora local.

Hoy en día los receptores GPS están bastante extendidos en su uso, está al alcance de todo el público en general por el bajo costo y las altas prestaciones, es bastante usada en la navegación en automóviles, aviones, etc., en las áreas de geología, topografía, redes móviles, etc.

#### 2.4.4.2. Dispositivos receptores GPS

Como se había mencionado el sistema de posicionamiento global consiste en un conjunto de satélites los cuales enviar datos que permiten calcular la ubicación en cualquier parte de globo terráqueo, estos datos son recibidos por receptores especializados los cuales a partir de los datos realizan el cálculo correspondiente obteniendo como resultado la latitud, longitud y altura dentro de una área geográfica. Existe gran variedad de dispositivos receptores desde los más especializados hasta los más genéricos

Los teléfonos inteligentes tienen integrados receptores que nos proporcionan la ubicación del dispositivo. Por otro lado al ser este un proyecto especializado se hará uso de un módulo específico la cual será acoplada a la plataforma Arduino para el diseño del dispositivo de geo localización correspondiente.

En el siguiente punto se realizará una descripción de las principales características del módulo a ser utilizado.

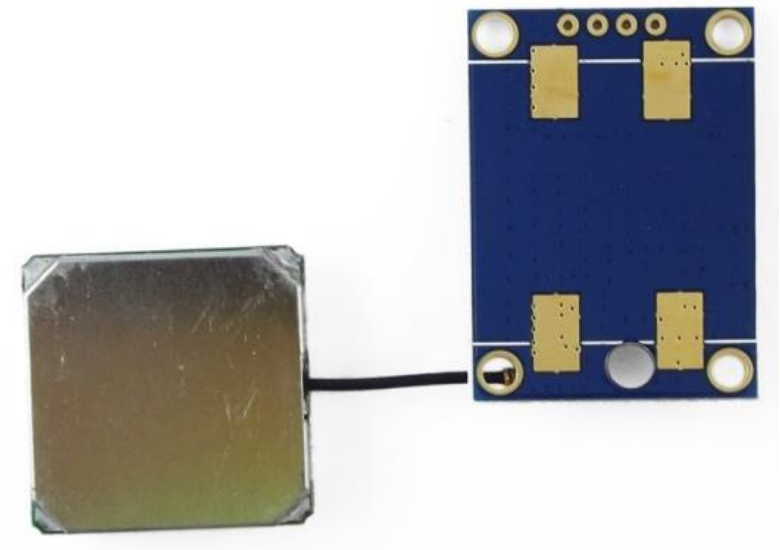
##### 2.4.4.2.1. Módulo GPS NEO-6M

Este dispositivo se caracteriza por el fácil acoplamiento a diferentes plataformas de desarrollo de proyectos electrónicos, ya sea Arduino, Raspeberry Pi u otros.

En las Figura 2.1 y Figura 2.2 se puede observar las características físicas de la misma a través de una toma frontal y otra trasera del dispositivo respectivamente.

**Figura Nº 2.1:** Módulo GPS NEO-6M, Vista Frontal


**Figura Nº 2.2:** Módulo GPS NEO-6M, Vista Trasera



**Fuente:** (Naylamp Mechatronics, 2020)

Algunas de las características más importantes de este módulo son descritos a continuación.

**Tabla Nº 2.1:** Características técnicas GPS NEO-6M

| CARACTERISTICAS | DEFINICIÓN |
|---|---|
| Voltaje de alimentación | 3-5 VDC |
| Interface | Serial UART 5V |
| Antena | Cerámica |
| EEPROM | Para guardar datos de configuración cuando el módulo se desenergice |
| Batería | Si (Respaldo) |
| Frecuencia de refresco | 5Hz |
| Soporte | SBAS(WAAS, EGNOS, MSAS, GAGAN) |
| Indicador | Led |
| Tamaño de antena | 25mm x 25mm |
| Tamaño de agujeros de montaje | 3mm |
| Baudrate | 9600 bps |

**Fuente:** (Naylamp Mechatronics, 2020)

### 2.4.5. Control

Fundamentalmente cuando hablamos de control se entiende como la capacidad de recolectar y analizar información con la cual a través de alguna herramienta se pretende encaminar uno o varios procesos hacia su objetivo.

Como se había mencionado en el anterior punto, monitoreo es una herramienta para el control ya que al realizar monitoreo se obtiene información trascendente de lo que se está haciendo, esto nos ayudará a la toma de decisiones con la cual podremos realizar algún tipo de cambio sobre el proceso para de esta manera corregir ya sea una falla particularmente o una característica no planificada.

Complementando la definición se tiene que: “El control es un proceso por medio del cual se modifica algún aspecto de un sistema para que se alcance el

desempeño deseado en el mismo. La finalidad del proceso de control es hacer que el sistema se encamine completamente hacia sus objetivos. El control no es un fin en sí mismo, es un medio para alcanzar el fin, o sea mejorar la operación del sistema" (Web profit, 2020).

## 2.5. CIRCUITO ELECTRONICO

Son placas compuestas por materiales semiconductores, materiales activos y pasivos, cuyo funcionamiento depende del flujo de electrones para la generación, transmisión, recepción, almacenamiento de información, entre otros. Esta información puede consistir en voz o música como en un receptor de radio, en una imagen en una pantalla de televisión, o en números u otros datos en un ordenador o computadora (EcuRed, 2020).

## 2.6. INTERNET DE LAS COSAS (IOT)

En la actualidad la tecnología ha ido evolucionando de gran manera, adentrándose e inmiscuyéndose más en el diario vivir de las personas. Así también es notorio el uso extendido de dispositivos inteligentes a partir del bajo costo que implica, ofreciéndonos cada vez mayor capacidad de conectividad.

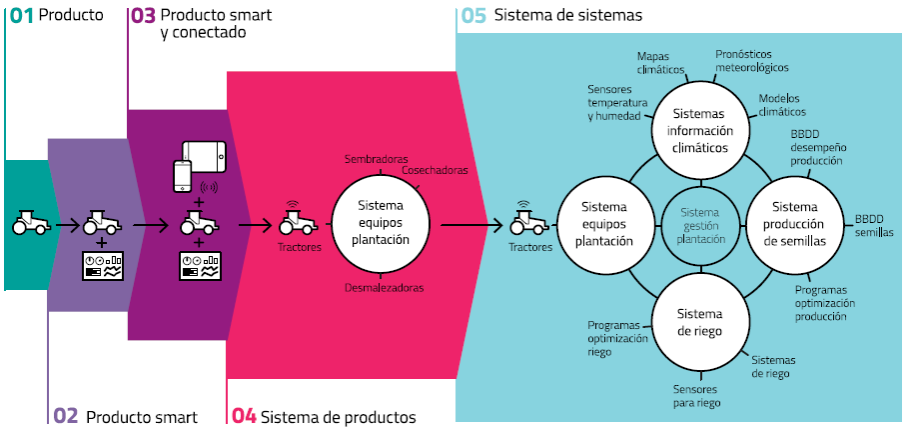
Si bien el concepto del Internet de las cosas es relativamente nueva, existe amplia documentación que describen las características principales de la misma, según la: Corporación de Fomento de la Producción (Corfo, 2018) "El Internet de las Cosas se refiere a la interconexión digital de objetos cotidianos con Internet" (p.23). Sin embargo, el internet de las cosas no solamente implica la interconexión con internet sino la interconexión entre dispositivos ya sea a través de conexión wifi, bluethoot, etc., no necesariamente implica tener acceso a internet.

El internet de las cosas tiene como premisa fundamental la interconexión entre dispositivos cotidianos con dispositivos inteligentes también la interconexión con internet, un dispositivo cotidiano puede enviar datos a un servidor, la cual llega

a ser monitoreado a partir de esta poder realizar algún tipo de acción todo esto en tiempo real.

En la Figura Nº 2.3 se puede observar la implicación que tiene adentrarse en el concepto del internet de las cosas, ya que hoy en día ya no solo implica construir un prototipo o producto funcional sino este debe permitir interconectarse con internet para poder enviar y recibir datos para el control y monitoreo correspondiente.

**Figura Nº 2.3:** Producto cotidiano conectado




**Fuente:** (Corfo, 2018).

## 2.7. ARDUINO

Consiste en una plataforma electrónica de hardware abierto y libre, la cual está basada en el micro controlador Atmega, está orientada al desarrollo de prototipos de proyectos electrónicos ya que gracias a su fácil programación, bajo costo y simplicidad se ha ido extendiendo ampliamente su uso. La misma consiste en un conjunto de puertos de entradas analógicas y digitales, junto a salidas digitales a través de la cual se puede ir controlando todo tipo de dispositivos.

Otra de las características que nos ofrece es el uso de un completo entorno de trabajo la cual hace más flexible el uso y control de la plataforma. Esta

desarrollado bajo la idea de que pueda ser usado por aficionados en electrónica hasta expertos en robótica y control.

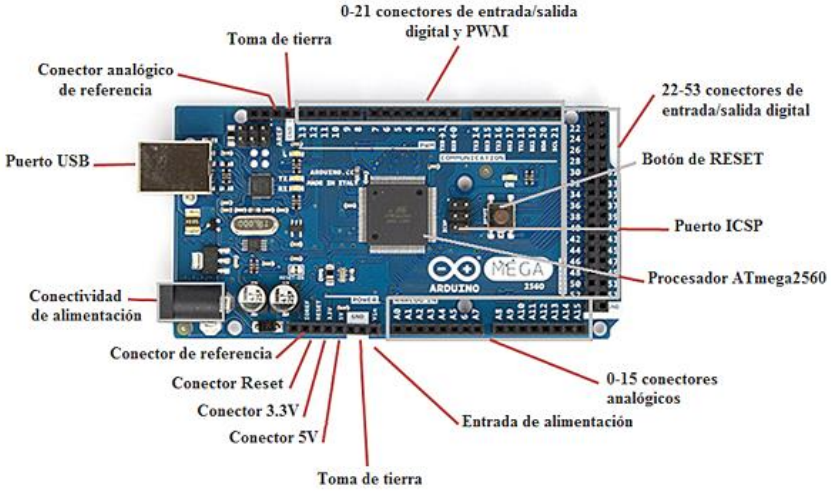
Una de las características principales, valga la redundancia, es su fácil programación, utiliza el IDE Arduino, que la provee la misma fabricante y está basado en un lenguaje programación Arduino muy parecido al lenguaje C. El funcionamiento de la misma no exige que tenga que estar conectado a un ordenador ya que esta plataforma tiene la capacidad de almacenar las rutinas de procesos en memoria.

En el tiempo se han ido lanzando diferentes versiones de la plataforma Arduino proporcionando variedad en función a las características y prestaciones requeridas. En el particular no será necesario describir las demás ya que la presente está basada en el uso del Arduino Mega 2560.

### 2.7.1. Arduino Mega 2560

Si bien hay diferentes versiones de la plataforma Arduino en este caso se hará referencia a la versión denominada Arduino Mega 2560. Esta versión es la más completa. En comparación a otras versiones el costo es bastante bajo y las prestaciones son suficientes para el desarrollo del presente proyecto. A continuación, se ilustra la siguiente imagen con la descripción física:

**Figura Nº 2.4:** Arduino Mega 2560




**Fuente:** (Gallardo, 2010)

#### 2.7.1.1. Características

Dentro de las principales características técnicas tenemos las siguientes:

**Tabla Nº 2.2:** Características principales Arduino Mega 2560

| CARACTERISTICA | DESCRIPCION |
|---|---|
| Microcontrolador | ATmega2560 |
| Voltaje Operativa | 5V |
| Voltaje de Entrada (Recomendado) | 7-12V |
| Pines de Entradas/Salidas Digital | 54(15 son salidas PWM) |
| Pines de Entrada Análogas | 16 |
| Memoria Flash | 256 Kb (8 kb bootloader) |
| SRMA | 8 Kb |
| EEPROM | 4 Kb |
| Velocidad de Reloj | 16 Mhz |

**Fuente:** (Artero, 2013)

**Alimentación**

El voltaje de funcionamiento de la plataforma incluyendo el micro controlador y demás componentes es de 5V. La alimentación puede ser de 2 maneras ya sea a través de una fuente externa o a través de un cable USB conectado a la computadora, el diseño es lo suficientemente inteligente para elegir la alimentación que esté disponible. Por otro lado, la alimentación soportada en teoría es de 5 a 12V, pero por temas de estabilidad y seguridad se recomienda mantenerlo en 5V.

**Entradas y Salidas digitales**

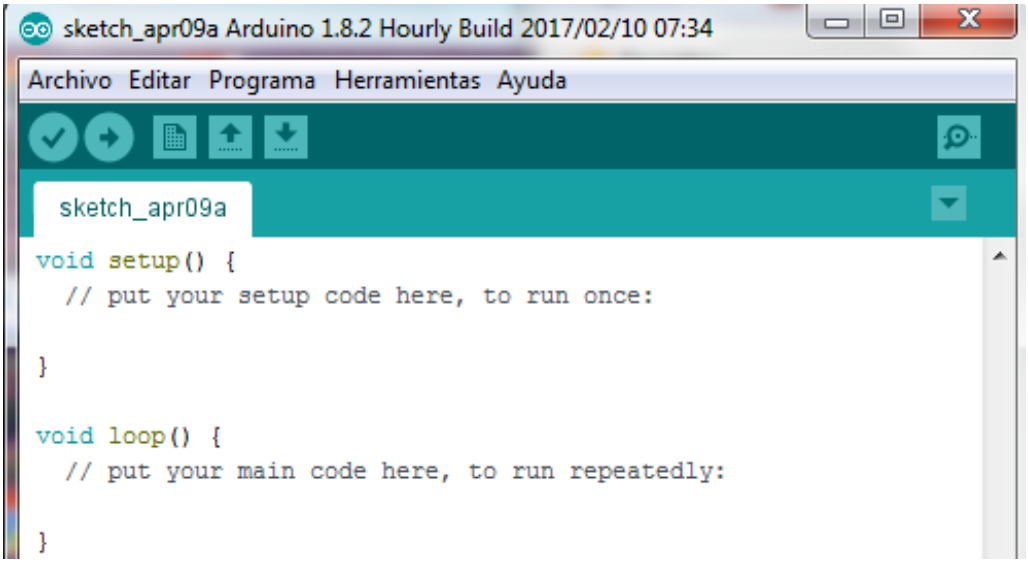
La placa Arduino dispone de 54 pines hembra, ya sea de entrada o salida, según lo requerido, ya que es posible configurarlos de las dos formas.

Todos estos pines-hembra digitales funcionan a 5 V, pueden proveer o recibir un máximo de 40 mA y disponen de una resistencia “pull-up” interna de entre 20 KΩ y 50 KΩ que inicialmente está desconectada (salvo que nosotros indiquemos lo contrario mediante programación software). (Torrente, 2013, p.90)

### 2.7.2. Entorno de desarrollo Arduino

La comunidad encargada de dar soporte a la plataforma de desarrollo de hardware abierto Arduino, además de la documentación, nos proporciona un IDE (Entorno de Desarrollo Integrado) completo de desarrollo, en la cual los programadores puedan desarrollar programas para el control del mismo. Estos programas se denominan “sketch”. El IDE, ya mencionado, nos ayuda a escribir y editar los programas comprobando que no haya errores para luego realizar la compilación necesaria y cargarlo al micro controlador de la placa Arduino.

**Figura Nº 2.5:** Entorno de Desarrollo Integrado - Arduino




**Fuente:** (Artero, 2013)

### 2.7.3. Estructura básica de un programa

Un programa diseñado para ejecutarse sobre un Arduino (un “sketch”) siempre se compone de tres secciones:

- **La sección de declaración de variables globales:** ubicada directamente al principio del sketch.
- **La sección llamada “void setup()”:** delimitada por llaves de apertura y cierre.
- **La sección llamda “void loop()”:** delimitada por llaves de apertura y cierre.

La primera sección del skecth (que no tiene ningún tipo de símbolo delimitador de inicio o de final) está reservada para escribir, tal como su nombre indica, las diferentes declaraciones de variables que necesitemos.

En el interior de las otras dos secciones (es decir, dentro de sus llaves) deberemos escribir las instrucciones que deseemos ejecutar en nuestra placa, teniendo en cuenta lo siguiente:

- **Las instrucciones escritas dentro de la sección “void setup()”** se ejecutan una única vez, en el momento de encender (o resetear) la placa Arduino.
- **Las instrucciones escritas dentro de la sección “void loop()”** se ejecutan justo después de las de la sección “void setup()” infinitas veces hasta que la placa se apague (o se resetee). Es decir, el contenido de “void loop()” se ejecuta desde la 1ra instrucción hasta la última, para seguidamente volver a ejecutarse desde la 1ra instrucción hasta la última, y así una y otra vez.

#### 2.7.3.1. Tipos de datos

A continuación describimos los tipos de datos aceptados y manejados en Arduino con especificación de las características de las mismas, sin dejar de lado la capacidad de cada una de los tipos de datos.

**Tabla Nº 2.3:** Tipos de datos aceptados por la placa Arduino

| Tipo de Dato | Característica |
|---|---|
| **boolean** | Las variables de este tipo solo pueden tener dos valores: cierto o falso. |
| **char** | El valor que puede tener una variable de este tipo es siempre un solo carácter (una letra, un dígito, un signo de puntuación, etc) |
| **byte** | El valor que pude tener una variable de este tipo es siempre un numero entero entre 0 y 255 |
| **int** | El valor que puede tener una variable de este tipo es un número entero entre -32768($-2^{31}$) y 32767($2^{15}$-1). |
| **word** | Las variables de tipo “word” en la placa Arduino Due ocupan 4 bytes para almacenar su valor. |
| **short** | El valor que pude tener una variable de este tipo para todos los modelos es un número entero entre -32768($-2^{15}$) y 32767($2^{15}$-1). |
| **long** | El valor que puede tener una variable de este tipo para todos los modelos de placa es un número entero entre -2.147.483.648 y 2.147.483.647 gracias a que utilizan 4bytes de memoria para almacenar. |
| **unsigned long** | El valor que puede tener una varibale de este tipo paratodos los modelos de placa es un número entero entre 0 y 4.294.967.295 ($2^{32}$-1). |
| **float** | El valor que puede tener una variable de este tipo es un número decimal. Los valores “float” posibles pueden ir desde el número -3,4028235x$10^{38}$ hasta el número 3,4028235x$10^{38}$ |
| **double** | Es un sinónimo exactamente equivalente del tipo “float” |
| **array** | Es una colección de variables (arreglo) de un tipo concreto. |

**Fuente:** (Artero, 2013)

### 2.7.4. Conectividad

Se denomina conectividad a la capacidad de establecer una conexión: una comunicación, un vínculo. El concepto suele aludir a la disponibilidad que tiene de un dispositivo para ser conectado a otro o a una red.

Es importante distinguir entre este concepto y el de conexión: mientras que la conectividad se mantiene igual a lo largo de la vida de un dispositivo hasta que se actualizan o mejoran sus partes, las conexiones comienzan y terminan, y

dentro de un mismo contexto (haciendo uso de una misma red de equipos con conectividades invariables) pueden tener características diferentes cada vez.

Por ejemplo, con un mismo teléfono móvil, cuya conectividad es siempre la misma, establecemos muchas conexiones a lo largo de la semana y los resultados suelen ser muy variables, ya sea por la presencia de humedad en el aire, por las tormentas o la fuerza del viento.

La idea de conectividad inalámbrica alude a las comunicaciones que se establecen sin cables. De este modo, el emisor y el receptor no se vinculan a través de un medio físico, sino que apelan a ondas que viajan por el espacio. Un teléfono celular (móvil), por citar un caso, puede conectarse con otro mediante la tecnología Bluetooth. Así se puede enviar información de un equipo a otro sin emplear cables (Definicion.DE, 2020).

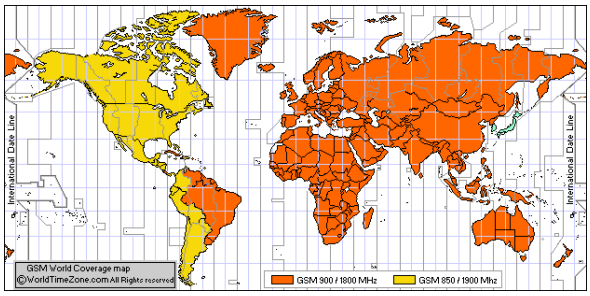
#### 2.7.4.1. Gsm/Gprs

La red GSM (Global System for Mobile Communicatioins) es el estándar más usado para la comunicación de teléfonos móviles o portátiles. Se denomina estándar "de segunda generación" (2G) porque, a diferencia de la primera generación de teléfonos portátiles, las comunicaciones se producen de un modo completamente digital. Posteriormente fueron introducidos respectivamente los estándares UMTS (Universal Mobile Telecomunications Systems) y LTE (Long Term Evolution), con capacidades adicionales de conexión a internet y mayor velocidad de transferencia de datos (Punto Flotante, 2020).

Básicamente existen 4 bandas, que son estándares a nivel mundial: 850, 900, 1800 y 1900 Mhz. En la mayoría de los países de Europa, Asia, Australia, Medio Oriente y Africa, se emplean las bandas de 900-1800 Mhz. En los Estados Unidos, Canadá, México y la mayor parte de Centro y Sudamérica se usan las bandas de 850-1900 Mhz. En México se emplea la banda de 1900 Mhz.

Los módems GSM cuatri-banda permiten operar en cualquiera de las 4 bandas mencionadas. Mediante los comandos AT, es posible configurar el módem GSM cuatri-banda para operar en la banda deseada. En la Figura 2.11 de abajo se muestra la asignación de pares de frecuencias GSM a nivel global.

**Figura Nº 2.6:** Mapa de líneas GSM




**Fuente:** (WorldTimeZone, 2020)

### 2.7.4.2. Módulo GSM/GPRS SIM900

El módulo GSM/GPRS es un dispositivo de comunicación, a través de una tarjeta SIM de forma que podamos comunicarnos con él como si se tratase de un teléfono móvil. Este dispositivo, de fácil acoplamiento a la tarjeta Arduino, no ofrece la posibilidad de enviar y recibir llamadas y SMS, como también, la más interesante, de conectarnos a internet (Prometec, 2020).

**Figura Nº 2.7:** Módulo GSM/GPRS SIM900


**Fuente:** (Prometec, 2020)

#### 2.7.4.2.1. Características

El módulo SIM900 es un shield ultra compacto y confiable, basado en el chip SIM900, compatible con Arduino Mega. Puede trabajar en frecuencias gsm/gprs

de 850/900/1800/1900 MHz. Tiene un bajo consumo y un diseño muy pequeño con grandes posibilidades. Se controla y configura mediante protocolo UART, usando comandos AT. Algunas de las características más significativas son las siguientes:

- ✓ Totalmente compatible con Arduino
- ✓ Conexión con el puerto serial
- ✓ Quad-Band 850/900/1800/1900 Mhz
- ✓ GPRS multi-slot 10/8 GPRS mobile station clase B
- ✓ Compatible GSM fase 2/2+Clase 4(2W (AT) 850/900 MHz)
- ✓ Clase 1 (1 W(AT) 1800/1900 MHz) TCP/UP embebido
- ✓ Soporta RTC
- ✓ Consumo de 1.5 mA(susp)
- ✓ Frecuencia 850/900/850/900

Por otro lado se describirá algunas especificaciones técnicas del módulo:

- ✓ Voltaje de alimentación: 9V – 20V
- ✓ Comunicación UART
- ✓ Bandas de frecuencia: 850/900/1800/1900 MHz
- ✓ Multi-GPRS Slot: clase 10/8
- ✓ Estación móvil: GPRS Clase B
- ✓ Fase: GSM 2/2+
- ✓ Pila embebida TCP/UDP: Carga de datos a un servidor
- ✓ Puerto Serie: Libre selección
- ✓ Porta batería para batería: CR1220
- ✓ Altavoz y tomas de auriculares: 2 conectores
- ✓ Temperatura de operación: -40ºC a +80ºC

#### 2.7.4.2.2. Comandos AT

Los tradicionales módems siempre han traido un conjunto de comandos para poder comunicarnos con ellos y configurarlos a través del puerto serie de

ordenador al cual están conectados. A estos comandos se les llama AT (de attention). Los móviles actuales suelen tener la capacidad de comportarse como módems de la red de telefonía móvil, por lo que también aceptan estos comandos, a parte de otros comandos específicos para telefonía, denominados comandos AT+**.** Ayudándonos de la conexión Bluetooth del móvil, podemos usar estos comandos para interactuar con el mismo desde nuestro ordenador. (Sancho, 2020)

De acuerdo a lo definido los comandos at nos permitirán comunicarnos con módems de comunicación, en el particular caso se hará uso del módulo SIM900 para lo cual, a continuación, detallaremos los comandos a usar.

**Tabla Nº 2.4:** Descripción de los comandos AT para el módulo SIM900

| COMANDO | DESCRIPCIÓN |
|---|---|
| **AT** | Comprueba estado de módulo |
| **AT+CPIN=”XXXX”** | Introducir el PIN de la SIM. XXXX como ejemplo. |
| **AT+CREG?** | Comprueba la conexión a la red. |
| **ATDXXXXXXX;** | Realia un llamada XXXXXXX por el numero a llamar |
| **ATA** | Descuelga una llamada |
| **ATH** | Finaliza llamada |
| **AT+CMGF=1** | Configura el modo texto para enviar o recibir mensajes. Devuelve “>” como inductor. |
| **AT+CMGF=”XXXXXXXXX”** | Nro al que vamos a enviar el mensaje |
| **AT+CLIP=1** | Activamos la identificación de llamada |
| **AT+CNMI=2,2,0,0,0** | Configuramos el módulo para que muestre los SMS por el puerto serie. |
| **AT+CGATT=1** | Conectamos a la red GPRS |
| **AT+CSTT=”APN”,”usuario”,”contraseña”** | Definimos APN, usuario y contraseña |
| **AT+CIICR** | Activamos el perfil de datos inalámbricos |
| **AT+CIFSR** | Obtenemos nuestra IP |
| **AT+CIPSTART=”TCP”,”direccionIP”,”puerto”** | Indicamos el tipo de conexión, dirección IP y puerto al que realizamos la conexión. |
| **AT+CIPSEND** | Preparamos el envío de datos. Devuelve “>” como inductor. |
| **AT+CIPCLOSE** | Cerramos la conexión |
| **AT+CIPSHUT** | Cierra el contexto PDP del GPRS |

**Fuente:** (SIMCom, 2015)

#### 2.7.4.3. Mqtt (Message Queue Telemetry Transport)

Este es un protocolo de comunicación, la cual se aplicará en el presente proyecto denotando características muy importantes e interesantes. “El protocolo MQTT es un protocolo de red ligero que se utiliza para publicar y suscribir Mensajería, optimizada para dispositivos remotos inalámbricos que pueden tener limitaciones ancho de banda y conectividad intermitente” (Lampkin, 2014, p.303).

Algunas de las características definidas en el protocolo MQTT son las siguientes:

- Niveles y flujos de calidad de servicio (QoS)
- Determinación de Qos
- Efecto QoS en el rendimiento
- ID de cliente
- Suscriptores duraderos y no duraderos
- Persistencia MQTT
- Encabezado MQTT
- Mantener vivo
- Reintentar la entrega de mensajes
- Bandera retenida en mensajes
- Pila TCP /IP

##### 2.7.4.3.1. Eclipse Mosquito

Eclipse Mosquitto es un intermediario de mensajes de código abierto (con licencia EPL / EDL) que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT. Mosquitto es liviano y es adecuado para su uso en todos los dispositivos, desde computadoras de una sola placa de baja potencia hasta servidores completos (Eclipse Foundation, 2020).

El protocolo MQTT proporciona un método ligero para llevar a cabo la mensajería utilizando un modelo de publicación / suscripción. Esto lo hace

adecuado para la mensajería de Internet de las cosas, como con sensores de baja potencia o dispositivos móviles como teléfonos, computadoras integradas o microcontroladores.

## 2.8. INGENIERIA DE SOFTWARE

(Pressman, Ingeniería de Software - Un enfoque práctico, 2003) Afirma “Ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales” (p.30).

Así también de acuerdo al Diccionario de la Real Academia Española (DRAE) define el término Ingeniería como: “conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía”.

En este contexto, la ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable hasta el desarrollo, operación, y mantenimiento del software.

### 2.8.1. Objetivos de la ingeniería de software

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software, para:

- ✓ Mejorar la calidad de los productos de software
- ✓ Aumenta la productividad y trabajo de los ingenieros del software
- ✓ Facilitar el control del proceso de desarrollo del software
- ✓ Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- ✓ Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

### 2.8.2. Metodologías del desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software.

Es como un libro de recetas de cocina, en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además que personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

#### 2.8.2.1. Metodología UWE

El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; y herramientas de soporte para la generación semi-automática de Aplicaciones Web. (Blanco, 2020)

De acuerdo a lo definido anteriormente la metodología UWE nos proporciona una extensión del uso de los modelos UML aplicado al análisis y diseño de sistemas orientados a la web, nos proporciona una guía de la manera de encarar este tipo de proyectos.

**Modelos de UWE**

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro de marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación.

Los principales artefactos que produce el método de diseño de UWE son los siguientes:

- ✓ Un Modelo de Requerimientos que captura los requerimientos del sistema, en este caso es aplicable el modelo de casos de uso.
- ✓ Un Modelo conceptual para el contenido (modelo de contenido).
- ✓ Un Modelo de Proceso.
- ✓ Un Modelo de Navegación que comprende la estructura de la navegación.
- ✓ Un Modelo de Presentación que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo de flujo de la presentación, modelo de interface abstracta de usuario, y modelo de ciclo de vida del objeto).

#### 2.8.2.1.1. Modelo de Requerimientos

El primer paso para el desarrollo de un sistema web que se especificará con UWE es realizar la identificación de los requerimientos y plasmarlos en un modelo de requerimientos.

Los requerimientos pueden ser documentados en diferentes niveles de detalle para este caso, UWE propone dos niveles de granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son modeladas con casos de uso UML. Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se delimiten las responsabilidades y acciones de los actores involucrados.

Los casos de uso fueron propuestos por el Proceso de Desarrollo de Software Unificado (RUP) para capturar los requerimientos del sistema.

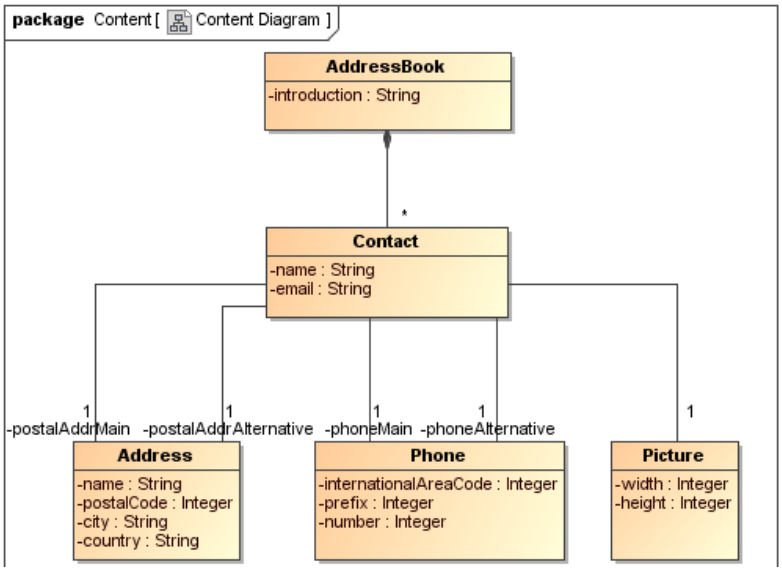
**Figura Nº 2.8:** Modelo de Caso de Uso UWE


**Fuente:** (Schmuller, 2010)

### 2.8.2.1.2. Modelo de contenido

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio Web. Su objetivo es construir un modelo de contenido del dominio de la aplicación considerando los requisitos reflejados en los casos de uso.

**Figura Nº 2.9:** Modelo de contenido UWE



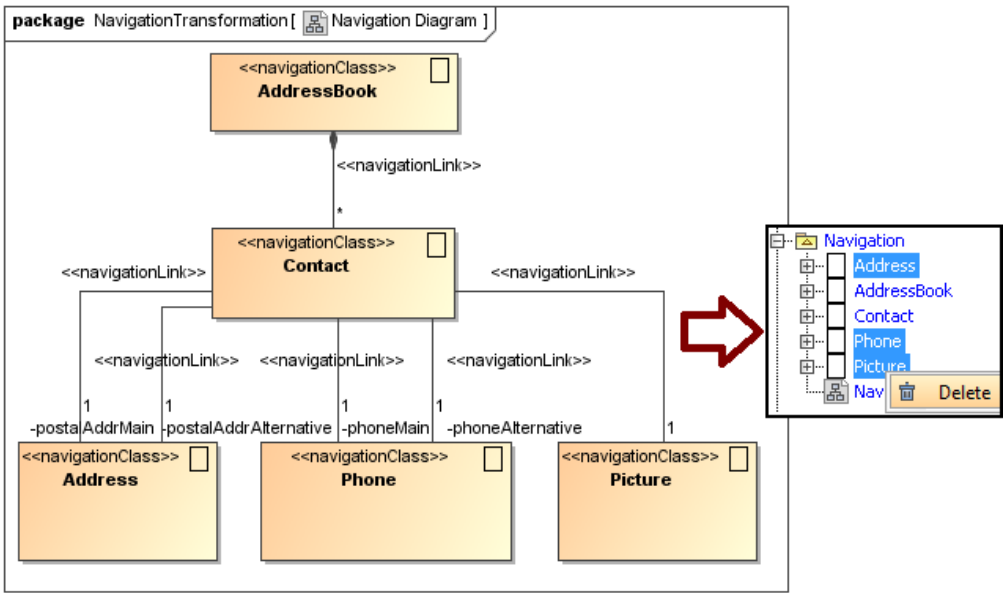

**Fuente:** (Web Engineering Group, 2020)

### 2.8.2.1.3. Modelo de navegación

En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos (nodes) y enlaces (links).

Nodos son unidades de navegación y están conectados por medio de enlaces. Las mismas pueden ser presentadas en diferentes páginas o en una misma.

**Figura Nº 2.10:** Modelo de navegación UWE

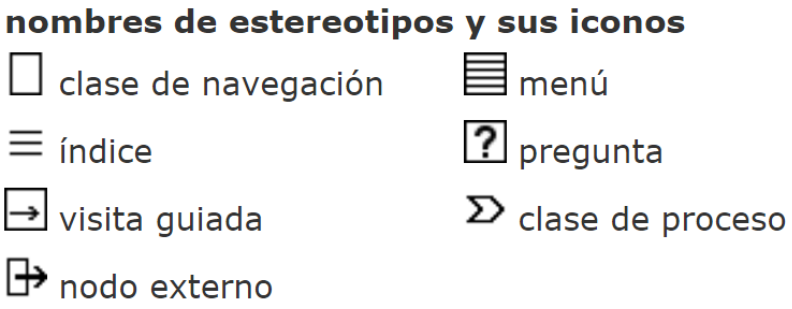



**Fuente:** (Web Engineering Group, 2020)

En este modelo se especifica la relación interna del sitio web, es decir cómo se relaciona cada página web con las demás, con lo cual, en definitiva es como se navega por el sitio web.

Para ello, se utilizan unidades de navegación llamadas nodos conectados por enlaces de navegación.

**Figura Nº 2.11:** Nombres de estereotipos y sus iconos



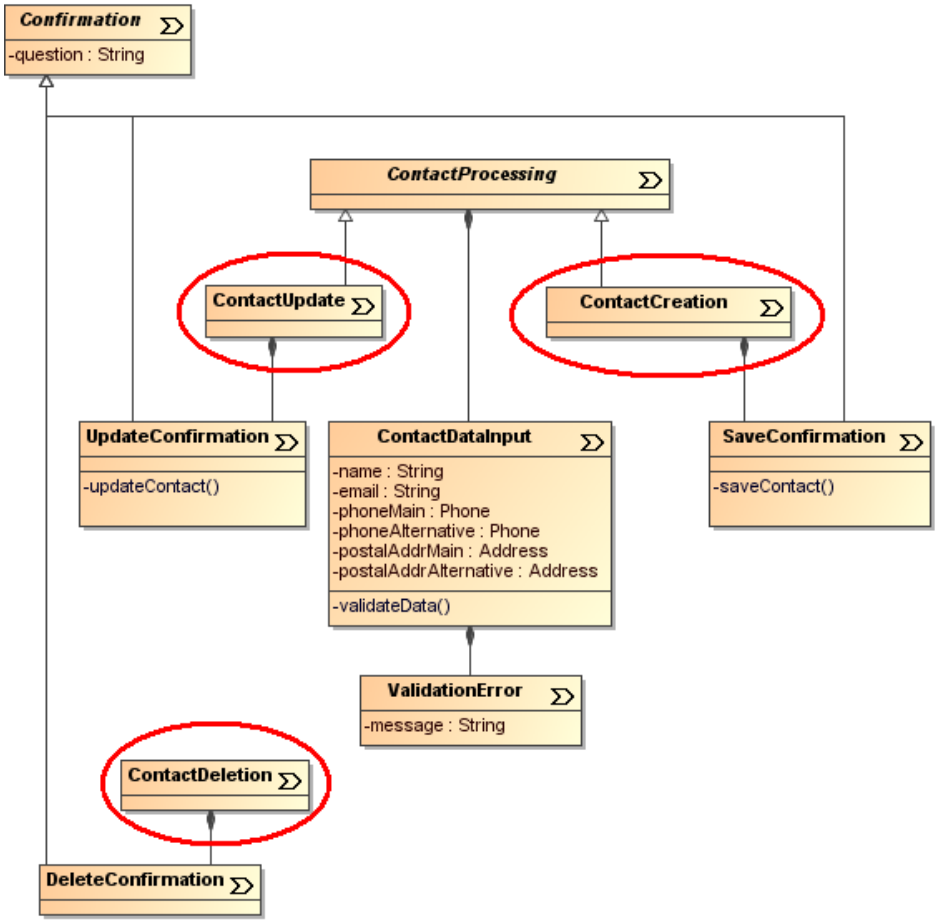

**Fuente:** (Web Engineering Group, 2020)

#### 2.8.2.1.4. Modelo de proceso

En este modelo se definen las acciones que realizan las clases del proceso (operacionales) específicas en el modelo de navegación, como se explicó anteriormente, el modelo del proceso se divide en dos partes:

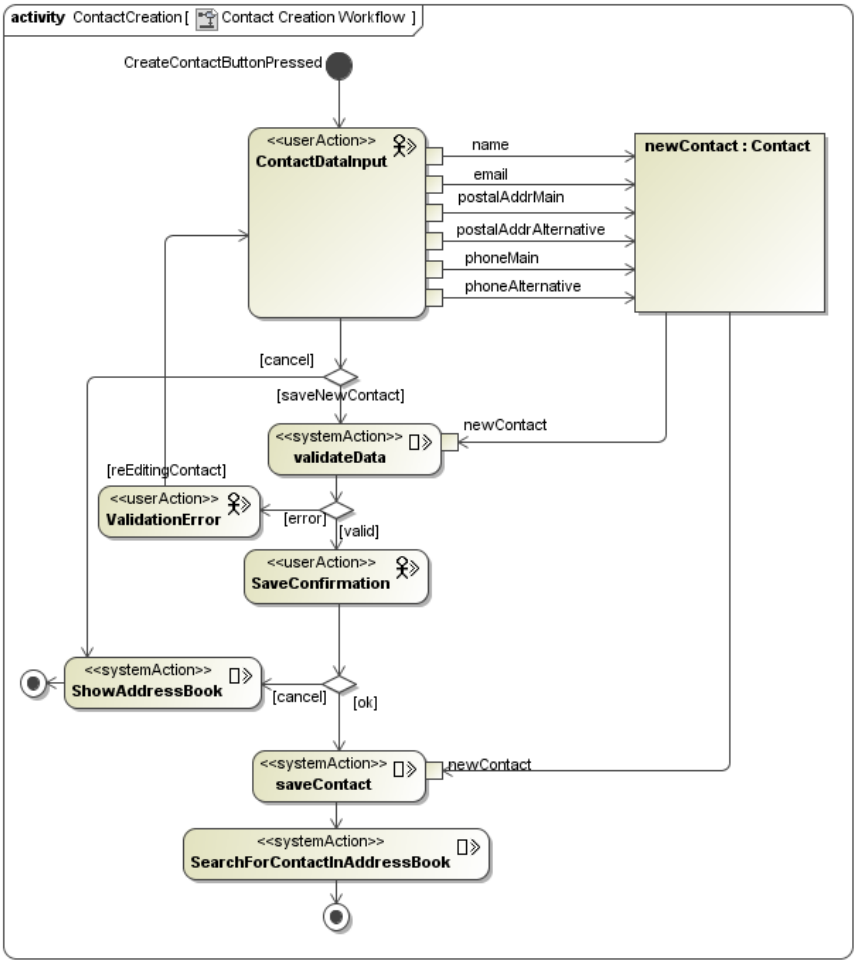
- ✓ Modelo de estructura del proceso: en el cual se incluyen las relaciones entre las clases de proceso, se crea un diagrama de clases donde cada una se presenta con un estereotipo de clases de proceso.
- ✓ Modelo de flujo del proceso: la conducta de un proceso es representado mediante un diagrama de actividades UML, describiendo el flujo de una clase del proceso, lo que sucede cuando un usuario navega hacia una clase de proceso.

**Figura Nº 2.12:** Modelo de estructura del Proceso




**Fuente:** (Web Engineering Group, 2020)

**Figura Nº 2.13:** Modelo de estructura del Proceso



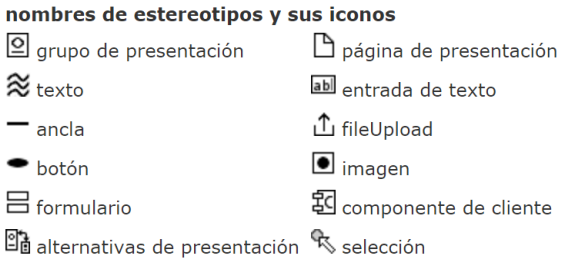

**Fuente:** (Web Engineering Group, 2020)

### 2.8.2.1.5. Modelo de presentación

El modelo de Navegación no indica cuáles son las clases de navegación y de proceso que pertenece a una página web, para esto se describe la utilización del diagrama de Presentación.

En este modelo presentan las clases de navegación de procesos que pertenecen a cada página web. Las notaciones de UWE son más claras y están mejor documentadas. UWE tiene como ventaja adicional que es un profile de UML, es decir, un modelo navegacional en un diagrama de clases UML con estereotipos.

**Figura Nº 2.14:** Nombre de estereotipos y sus iconos




**Fuente:** (Web Engineering Group, 2020)

### 2.8.3. Software en tiempo real

#### 2.8.3.1. Websockets

“La API y el protocolo de WebSocket se definen en RFC 6455. WebSocket le ofrece un canal de comunicaciones bidireccional, dúplex completo que opera a través de HTTP a través de un solo socket” (Lombardi, 2015, p.1).

Anteriormente se utilizaban peticiones http continuas en intervalos de tiempos, de esta manera, obteniendo datos actualizados en tiempo real aparentemente. Los websockets nos ofrecen la posibilidad de una comunicación bi-direccional sin tener que estar haciendo peticiones continuas, así también nos ofrece seguridad en la capa de transporte de datos del tipo SSL (Lombardi, 2015).

#### 2.8.3.2. RESTful Web Services

Buscando una definición sencilla, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad pero también mucha complejidad. A veces es preferible una solución más sencilla de manipulación de datos como REST ( BBVA Open4U, 2020).

Algunas de las características propias son:

- Esta asociados a información

- Permiten listar, crear, leer, actualizar y borrar información
- Para las operaciones anteriores necesitan una URL y un método HTTP para accederlas
- Usualmente regresan la información en formato JSON.
- Retornan códigos de respuesta HTML, por ejemplo 200, 201, 404, etc

### 2.8.4. Modelo Vista Controlador (MVC)

En líneas generales, MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

#### 2.8.4.1. Modelo

Es la capa donde se trabaja con los datos, por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes selects, updates, inserts, etc.

#### 2.8.4.2. Vista

Las vistas, como su nombre nos hace entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación, por ejemplo, en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

#### 2.8.4.3. Controlador

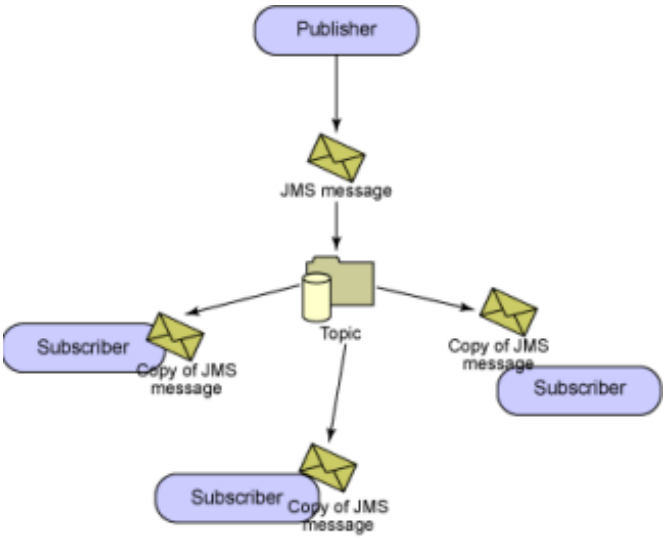
Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

En realidad es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo (DesarrolloWeb, 2020).

### 2.8.5. Patrón Subscribe/Publish

El sistema publish/subscribe es un paradigma de mensajes asíncronos donde los que envían (Publisher) mensajes no están programados para enviar sus mensajes a receptores específicos (Suscriptores), sino que se envían a algún tipo de servidor. Los mensajes publicados se caracterizan por clases, sin tener constancia de los suscriptores que pueda haber. Los suscriptores expresan interés en una o más clases, y solo reciben mensajes de ese mismo interés, sin tener constancia de qué publicadores hay. Esta relación independiente entre publicadores y subscriptores puede permitir una mayor escalabilidad (Danielle, 2020).

**Figura Nº 2.15:** Esquema básico del patrón Publicador/Suscriptor




**Fuente:** (Danielle, 2020)

## 2.9. CALIDAD DE SOFTWARE

Es muy importante definir el concepto de calidad y la misma se podría definir como: La totalidad de características de un producto, proceso o servicio que cuenta con la habilidad de satisfacer necesidades explícitas o implícitas (Académicos Intercontinentales, 2019).

La calidad de software se refiere a: “Los factores de un producto de software que contribuyen a la satisfacción completa y total de las necesidades de un usuario u organización” (Académicos Intercontinentales, 2019).

Así también Pressman (2010) afirma que la calidad de software se define como: “El proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan” (p. 340).

Por lo tanto podemos definir que la calidad de software cumple con los requisitos de acuerdo al modelo de calidad aplicado de esta manera satisfaciendo las necesidades del usuario o cliente.

### 2.9.1. Métricas de calidad de software ISO-9126

El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave de software de cómputo. Este sistema identifica seis atributos clave de la calidad:

a) **Funcionalidad:** Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
b) **Confiabilidad:** Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.
c) **Usabilidad:** Grado en el que el software es fácil de usar, según lo indican los siguientes subatributos: entendible, aprendible y operable.

d) **Eficiencia:** Grado en el que el software emplea óptimamente los recursos del sistema, según lo indican los subatributos siguientes: comportamiento del tiempo y de los recursos.
e) **Facilidad de recibir mantenimiento:** Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.
f) **Portabilidad:** Facilidad con la que el software puede llevarse de un ambiente a otro según la indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.

Igual que otros factores de la calidad de software estudiados en las subsecciones anteriores, los factores ISO 9126 no necesariamente conducen a una medición directa. Sin embargo, proporcionan una base útil para hacer mediciones indirectas y una lista de comprobación excelente para evaluar la calidad del sistema. (Pressman, 2010)

#### 2.9.1.1. Funcionabilidad

La métrica orientada a la función, utiliza una medida de la funcionalidad entregada por la aplicación como un valor de normalización. Ya que la funcionalidad no se puede medir directamente es necesario derivar mediante otras medidas directas como el punto función.

**Punto Función**

La métrica de punto función (PF), se puede usar como medio para predecir el tamaño de un sistema que se va a obtener de un modelo análisis. Los puntos de función se obtienen utilizando una relación empírica basada en medidas cuantitativas del dominio de información de software y valorización subjetiva de la complejidad del software.

Para poder determinar la funcionalidad del sistema se debe determinar cinco características del dominio de información.

- **Número de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada, estas aplicaciones pueden ser: insertar, actualizar, borrar datos del sistema.
- **Número de salidas de usuario.** Se cuenta cada salida que proporciona al usuario de información orientada a la aplicación. En este contexto la salida se refiere a informes, datos de pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.
- **Número de peticiones de usuario.** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata de forma de salida interactiva. Se cuenta cada petición por separado.
- **Número de archivos.** Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).
- **Número de interfaces externas.** Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

**Tabla Nº 2.5:** Cálculo de punto función no ajustado

| Parámetros de Medición | Cuentas | Factor de Ponderación | | | Total |
|---|---|---|---|---|---|
| | | Simple | Medio | Complejo | |
| **Nº De entradas de Usuario** | | 3 | 4 | 6 | |
| **Nº de Salidas de Usuario** | | 4 | 5 | 7 | |
| **Nº de Peticiones de Usuario** | | 3 | 4 | 6 | |
| **Nº de Archivos de Operación** | | 7 | 10 | 15 | |
| **Nº de Interfaces Externos** | | 5 | 7 | 10 | |

**Fuente:** (Pressman, 2010)

El punto función se puede calcular mediante la siguiente ecuación:

$$PF = Cuenta\ Total * (X + Y * \sum F_i)$$

Dónde:

**Cuenta Total:** es la suma de todas las entradas obtenidas en Nº de Entradas, Nº de Salidas, Nº de Peticiones, Nº de Archivos y Nº de Interfaces Externas.

**X:** nivel de confiabilidad del sistema es de (0.65).

**Y:** nivel de error igual a (0.01).

**$F_i$:** (i=1 a 14): Son los valores de ajuste de complejidad según las respuestas a las preguntas destacadas en la siguiente tabla.

**Tabla Nº 2.6:** Tabla para factores de complejidad

| Nº | Factores de complejidad | Valor |
|---|---|---|
| **1** | ¿Requiere el sistema copias de seguridad y de recuperación fiables? | |
| **2** | ¿Se requiere comunicación de datos? | |
| **3** | ¿Existen funciones de procesamiento distribuido? | |
| **4** | ¿Es crítico el rendimiento? | |
| **5** | ¿Se ejecuta el sistema en un entorno operativo, existente y fuertemente utilizado? | |
| **6** | ¿Requiere el sistema entrada de datos interactiva? | |
| **7** | Facilidad Operativa | |
| **8** | ¿Se actualizan los archivos maestros de forma interactiva? | |
| **9** | ¿Son complejos las entradas, las salidas, los archivos o las peticiones? | |
| **10** | ¿Es complejo el procesamiento interno? | |
| **11** | ¿Se ha diseñado el código para ser reutilizable? | |
| **12** | Facilidad de instalación | |
| **13** | ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones? | |
| **14** | Facilidad de cambio | |
| | **Total** | |

**Fuente:** (Pressman, 2010)

Cada una de las preguntas anteriores es respondida usando una escala con rangos de 0 (no importante o aplicable) hasta cinco (abosultamente esencial).

Los valores constantes de la ecuación y los factores de peso que se aplican a las cuentas de los dominios de inforamción se determinan empiricamente.

El indicador de funcionalidad se lo encuentra calculando la siguiente relación.

$$\%\boldsymbol{Funcionalidad} = \$\boldsymbol{PF} = \frac{\boldsymbol{PF_{Calculada}}}{\boldsymbol{PF_{maxima}}} * \mathbf{100}\%$$

### 2.9.1.2. Confiabilidad

La confiabilidad es la cantidad de tiempo que el software está disponible para su uso, es decir, la cantidad de tiempo que el sistema está en funcionamiento dentro de la institución y esté libre de fallas; se puede considerar que mientras mayor es el número de fallas, menor confiabilidad, pero a menor número de fallas mayor será la confiabilidad.

Por lo tanto, para poder medir la confiabilidad del sistema, se toma la siguiente fórmula que calcula la confiabilidad del sistema.

$$\boldsymbol{F(t) = f * e^{\left(-\frac{\lambda}{10} * t\right)}}$$

Dónde:

**f:** es la funcionalidad del sistema ya calculada.

**-λ:** es la probabilidad de error que puede tener el sistema.

**t:** tiempo que dura una gestión en el sistema y la probabilidad de que el sistema esté libre de fallos es: P(T>=t) = 1-F(t).

#### 2.9.1.3. Usabilidad

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que debería invertir el usuario para utilizar el sistema.

- **Comprensibilidad.** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- **Facilidad de Aprendizaje.** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
- **Operabilidad.** Agrupa los conceptos que evalúan la operación y el control del sistema.

Para realizar el cálculo de usabilidad del sistema aplicamos la siguiente ecuación:

$$\boldsymbol{FU = \frac{\left(\frac{\sum x_i}{n}\right) * 100}{N}}$$

Dónde:

**N:** número de la población

**n:** número de la muestra

Por lo tanto, el esfuerzo necesario para aprender a operar el sistema, preparando los datos de entrada e interpretando los datos de salida (resultados) de un programa, será dado por la usabilidad.

#### 2.9.1.4. Factibilidad de mantenimiento

Es la facilidad con que una modificación puede ser realizada en el sistema. Las modificaciones puedes incluir correcciones, mejoras o adaptar si su entorno cambia, o mejorar si el cliente desea un cambio de requisitos. Para este fin Pressman nos sugiere el Índice de Madurez del Software (IMS) para determinar

la estabilidad de un producto software. Dicha IMS es calculada por la siguiente ecuación:

$$IMS = \frac{[MT - (F_a + F_c + F_d)]}{M_T}$$

Dónde:

**$M_T$:** número de módulos de la versión actual.

**$F_c$:** número de módulos en la versión actual que se han cambiado.

**$F_a$:** número de módulos en la versión actual que se han añadido.

**$F_d$:** Número de módulos de la versión anterior que se han borrado en la versión actual.

#### 2.9.1.5. Potabilidad

La portabilidad se define como la característica que posee un software para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad es menor la dependencia del software con respecto a la plataforma.

### 2.9.2. Método de Estimación de Costos (COCOMO)

La elaboración de software conlleva varios procesos que son muy importantes para obtener un buen producto, y en este caso nos referimos particularmente al proceso de estimación de costos del software. “Una de las tareas de mayor importancia en la administración de proyectos de software es la estimación de costos. Si bien es una de las primeras actividades, inmediatamente posterior al establecimiento de los requerimientos, se ejecutan regularmente a medida que el proyecto progresa con el fin de ajustar la precisión en la estimación” (Gómez, Migani, & Otazú, 2008, p.4).

La estimación de costos de software tiene dos usos en la administración de proyectos:

- Durante la etapa de planeamiento: Permitir decidir cuantas personas son necesarias para llevar a cabo el proyecto y establecer el cronograma adecuado.
- Para controlar el progreso del proyecto: Es esencial evaluar si el proyecto está evolucionando de acuerdo al cronograma y tomar las acciones correctivas si fueran necesario.

COCOMO hace referencia a un modelo de estimación de costos propuesto por Barry Bohem. La misma permite realizar estimaciones en función del tamaño de software, y de un conjunto de factores de costo y de escala.

#### 2.9.2.1. Modos de Desarrollo

De acuerdo a Gómez, Mingani, & Otazú (2008) Todo proyecto de desarrollo corresponde a un de los siguientes modos tomando en cuenta la variable KSLOC que es una unidad de medida, donde 1 KSLOC equivale e mil líneas de código fuente.

a) **Modo Orgánico:** En esta clasificación se encuentran proyectos desarrollados en un ambiente familiar y estable. El producto a elaborar ese relativamente pequeño y requiere pocas innovaciones tecnológicas en lo que refiere a algoritmos, estructuras de datos e integración de hardware, no sobrepasa los 50 KSLOC.
b) **Modo Semiacoplado:** Es un modelo para productos de software de tamaño y complejidad media, las mismas tienen un tamaño que llega a 300 KSLOC.
c) **Modo Empotrado:** En esta clasificación están incluidas proyectos de gran envergadura que operan en un ambiente completo con altas restricciones de hardware, software y procedimientos operacionales, tales como los sistemas aéreos.

**Tabla Nº 2.7:** Valores y respectivas formular por modos de desarrollo

| Modo de Desarrollo | A | B | C | D | Esfuerzo | Cronograma |
|---|---|---|---|---|---|---|
| **Orgánico** | 3.20 | 1.05 | 2.50 | 0.38 | $PM = 3.20 \times (KSLOC)^{1.05}$ | $TDEV = 2.5 \times (PM)^{0.38}$ |
| **Semiacoplado** | 3.00 | 1.12 | 2.50 | 0.35 | $PM = 3.00 \times (KSLOC)^{1.12}$ | $TDEV = 2.5 \times (PM)^{0.35}$ |
| **Empotrado** | 2.80 | 1.20 | 2.50 | 0.32 | $PM = 2.80 \times (KSLOC)^{1.20}$ | $TDEV = 2.5 \times (PM)^{0.32}$ |

**Fuente:** (Gómez, Migani, & Otazú, 2008)

### 2.9.2.2. Modelos de Desarrollo

#### 2.9.2.2.1. Modelo Básico

Esta estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usado dos variables predictivas y denominadas factores de costo: el tamaño del software y el modo de desarrollo. Las ecuaciones básicas son:

**Esfuerzo:**

$$\boldsymbol{PM = A \times (KSLOC)^{B}}$$

Dónde:

$\boldsymbol{PM}$: es el esfuerzo estimado. Representa los mese-persona necesarios para ejecutar el proyecto.

$\boldsymbol{KSLOC}$: es el tamaño del software a desarrollar en miles de líneas de código.

$\boldsymbol{A\ y\ B}$ : son coeficientes que varían según el Modo de desarrollo (Orgánico, Semiacoplado, Empotrado).

**Cronograma:**

$$\boldsymbol{TDEV = C \times (PM)^{D}}$$

Dónde:

$\boldsymbol{TDEV}$: representa los meses de trabajo que se necesitan para ejecutar el proyecto.

$\boldsymbol{C\ y\ D}$: son coeficientes que varían según el Modo de Desarrollo (Orgánico, Semiacoplado, Empotrado).

#### 2.9.2.2.2. Modelo Intermedio

Comparando con el modelo anterior éste provee un nivel de detalle y precisión superior, por lo cual es más apropiado para la estimación de costos en etapas de mayor especificación. Incorpora un conjunto de quince variables de predicción que toman en cuenta las variaciones de costos no consideradas el modelo básico.

Los factores seleccionados se agrupan en cuatro categorías:

a) Atributos del producto de software
   - RELY: Confiabilidad Requerida
   - DATA: Tamaño de la Base de Datos
   - CPLX: Complejidad del producto

b) Atributos del hardware
   - TIME: Restricción del tiempo de ejecución
   - STOR: Restricción del almacenamiento Principal
   - VIRT: Volatilidad de la Máquina Virtual
   - TURN: Tiempo de Respuesta de la computadora expresado en horas

c) Atributos del personal involucrado en el proyecto
   - ACAP: Capacidad del Analista
   - AEXP: Experiencia en Aplicaciones Similares
   - PCAP: Capacidad del Programador
   - VEXP: Experiencia en la máquina virtual
   - LEXP: Experiencia en el Lenguaje de Programación

d) Atributos propios del proyecto
   - MODP: Prácticas Modernas de Programación
   - TOOL: Uso de Herramientas de software
   - SCED: Cronograma de Desarrollo Requerido

**Tabla Nº 2.8:** Valores de los atributos de Coste

| ATRIBUTOS | VALOR | | | | | |
|---|---|---|---|---|---|---|
| | MUY BAJO | BAJO | NOMINAL | ALTO | MUY ALTO | EXTRA ALTO |
| ATRIBUTOS DE SOFTWARE | | | | | | |
| **Fiabilidad** | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 | |
| **Tamaño de Base de Datos** | | 0.94 | 1.00 | 1.08 | 1.16 | |
| **Complejidad** | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| ATRIBUTOS DE HARDWARE | | | | | | |
| **Restricciones de tiempo de ejecución** | | | 1.00 | 1.11 | 1.30 | 1.66 |
| **Restricciones de memoria virtual** | | | 1.00 | 1.06 | 1.21 | 1.56 |
| **Volatilidad de la máquina virtual** | | 0.87 | 1.00 | 1.15 | 1.30 | |
| **Tiempo de respuesta** | | 0.87 | 1.00 | 1.07 | 1.15 | |
| ATRIBUTOS DE PERSONAL | | | | | | |
| **Capacidad de análisis** | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 | |
| **Experiencia en la aplicación** | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 | |
| **Calidad de los programadores** | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 | |
| **Experiencia en la máquina virtual** | 1.21 | 1.10 | 1.00 | 0.90 | | |
| **Experiencia en el lenguaje** | 1.14 | 1.07 | 1.00 | 0.95 | | |
| ATRIBUTOS DE PROYECTO | | | | | | |
| **Técnicas actualizadas de programación** | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 | |
| **Utilización de herramientas de software** | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 | |
| **Restricciones de tiempo de desarrollo** | 1.22 | 1.08 | 1.00 | 1.04 | 1.10 | |

**Fuente:** (Gómez, Migani, & Otazú, 2008)

El proceso de estimación del esfuerzo puede sintetizarse en los siguientes pasos:

- Se calcula el esfuerzo nominal $PM_{Nominal}$, al igual que en el modelo Básico, donde los únicos factores de costo son el tamaño y el modo de desarrollo.
- Se determina el Factor de Ajuste del Esfuerzo (FAE) según la fórmula:

$$\boldsymbol{EAF} = \prod_{\boldsymbol{i}=1}^{\boldsymbol{15}} \boldsymbol{EM_I}$$

Dónde:

- Cada $EM$, llamado factor multiplicador de esfuerzo, es el valor que corresponde a cada atributo de acuerdo al grado de influencia (Muy Bajo, Bajo, Nominal, Alto, Muy Alto, Extra Alto) en el esfuerzo del desarrollo del software como se puede observar en la Tabla Nº 2.6.
- Finalmente, se ajusta el esfuerzo nominal aplicando el FAE.

$$\boldsymbol{PM = A \times EAF \times (KSLOC)^B}$$

La Tabla Nº 2.7 muestra la variación de la ecuación de estimación de esfuerzo y cronograma según los tres modos de desarrollo.

**Tabla Nº 2.9:** Ecuación de estimación de esfuerzo y cronograma

| MODO DE DESARROLLO | ESFUERZO NOMINAL | ESFUERZO AJUSTADO | CRONOGRAMA |
|---|---|---|---|
| **Orgánico** | $PM_{nominal} = 3.2 \times (KSLOC)^{1.05}$ | $PM = 3.2 \times EAF \times (KSLOC)^{1.05}$ | $TDEV = 2.5 \times (PM)^{0.38}$ |
| **Semiacoplado** | $PM_{nominal} = 3.0 \times (KSLOC)^{1.12}$ | $PM = 3.0 \times EAF \times (KSLOC)^{1.12}$ | $TDEV = 2.5 \times (PM)^{0.35}$ |
| **Empotrado** | $PM_{nominal} = 2.8 \times (KSLOC)^{1.20}$ | $PM = 2.8 \times EAF \times (KSLOC)^{1.20}$ | $TDEV = 2.5 \times (PM)^{0.32}$ |

**Fuente:** (Gómez, Migani, & Otazú, 2008)

### 2.9.2.2.3. Modelo Detallado

El Modelo Detallado provee los medios para generar estimaciones con mayor grado de precisión y detalle.

Las ecuaciones fundamentales de este modelo con similares a las del modelo intermedio, la única diferencia reside en el cálculo del Factor de Ajuste de Esfuerzo $(EAF)$. El procedimiento incluye el cálculo de un Factor de Ajuste del Esfuerzo al nivel de módulo $(EAF_M)$ y otro al nivel de subsistema $(EAF_S)$.

## 2.10. HERRAMIENTAS

### 2.10.1.Angular

Angular es un framework Javascript potente, muy adecuado para el desarrollo de aplicaciones frontend modernas, de complejidad media o elevada. El tipo de aplicación Javascript que se desarrolla con Angular es del estilo SPA (Single Page Application) o también las denominadas PWA (Progressive Web App).

El framework Angular ofrece una base para el desarrollo de aplicaciones robustas, escalables y optimizadas, que promueve además las mejores prácticas y un estilo de codificación homogéneo y de gran modularidad.

Aunque ofrece principalmente una base para el desarrollo de la parte frontal, la programación Javascript del lado del cliente, también aborda técnicas de desarrollo de la parte del backend, para la implementación del Server Side Rendering. A esta parte se le llama Angular Universal.

El desarrollo en Angular se hace por medio de TypeScript (aunque también se podría desarrollar con Javascript, todas las guías y recomendaciones se basan en usar TypeScript), un superset del lenguaje Javascript que ofrece muchas herramientas adicionales al lenguaje, como el tipado estático o los decoradores.

**Desarrollo de aplicaciones SPA**

Angular es un framework especializado en la creación de aplicaciones SPA, Single Page Application. Este tipo de aplicaciones se muestra en una sola página y sus pantallas son simplemente vistas que se van intercambiando. con ello se obtiene una mejor experiencia de usuario, más parecida a la de aplicaciones de escritorio.

Para conseguirlo angular ofrece un sistema de routing entre sus herramientas. Es altamente configurable y muy potente, qué se puede configurar fácilmente con una estrategia de Lazy loading. Así se obtienen aplicaciones muy optimizadas, que cargan muy rápido, tanto en el primer acceso como cuando se acceden a otras pantallas, o páginas, de la aplicación. (CriarWeb S.L., 2020)

**Figura Nº 2.16:** Logo Angular

**Fuente:** (CriarWeb S.L., 2020)

### 2.10.2. Nodejs

Node.js es un entorno de tiempo de ejecución multiplataforma de código abierto para desarrollar aplicaciones de red y del lado del servidor. Las aplicaciones Node.js están escritas en JavaScript y pueden ejecutarse dentro del tiempo de ejecución de Node.js en OS X, Microsoft Windows y Linux.

Node.js también proporciona una rica biblioteca de varios módulos JavaScript que simplifica el desarrollo de aplicaciones web usando Node.js en gran medida.

**Características**

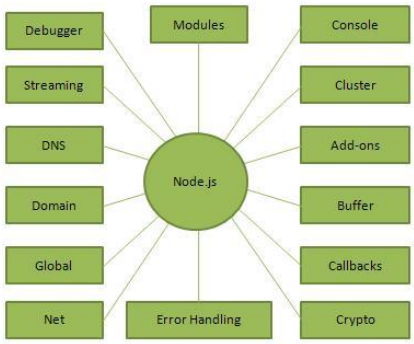
Las siguientes son algunas de las características importantes que hacen de Node.js la primera opción de arquitectos de software.

- **Asíncrono y controlado por eventos**: todas las API de la biblioteca Node.js son asíncronas, es decir, sin bloqueo. Básicamente, significa que un servidor basado en Node.js nunca espera a que una API devuelva datos. El servidor pasa a la siguiente API después de llamarlo y un mecanismo de notificación de Events of Node.js ayuda al servidor a obtener una respuesta de la llamada API anterior.
- **Muy rápido**: al estar construido en el motor JavaScript V8 de Google Chrome, la biblioteca Node.js es muy rápida en la ejecución de código.
- **Rosca única pero altamente escalable**: Node.js utiliza un modelo de una sola rosca con bucle de eventos. El mecanismo de eventos ayuda al servidor a responder sin bloqueos y hace que el servidor sea altamente

escalable en comparación con los servidores tradicionales que crean hilos limitados para manejar las solicitudes. Node.js usa un solo programa de subprocesos y el mismo programa puede proporcionar servicio a un número mucho mayor de solicitudes que los servidores tradicionales como el Servidor Apache HTTP.

- **Sin** almacenamiento en búfer: las aplicaciones de Node.js nunca almacenan en búfer ningún dato. Estas aplicaciones simplemente generan los datos en fragmentos.
- **Licencia**: Node.js se publica bajo la licencia MIT. (Tutorials Point, 2020)

**Figura Nº 2.17:** Partes de Nodejs




**Fuente:** (Tutorials Point, 2020)

### 2.10.3. Sequelize

Sequelize es un ORM basado en la promesa para Node.js y io.js. Es compatible con los dialectos PostgreSQL , MySQL , MariaDB , SQLite y MSSQL y cuenta con soporte de transacciones sólidas, relaciones, replicación de lectura y más. (RIP Tutorial, 2020).

### 2.10.4. Postgresql

PostgreSQL es un poderoso sistema de base de datos relacional de objetos de código abierto que usa y extiende el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a

1986 como parte del proyecto postgres en la Universidad de California en Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central.

PostgreSQL se ha ganado una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto de características robustas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para brindar soluciones innovadoras y de alto rendimiento. PostgreSQL se ejecuta en todos los principales sistemas operativos , cumple con ACID desde 2001 y tiene complementos potentes como el extensor de base de datos geoespaciales PostGIS . No sorprende que PostgreSQL se haya convertido en la base de datos relacional de código abierto elegida por muchas personas y organizaciones.

**Características de Postgresql**

PostgreSQL viene con muchas características destinadas a ayudar a los desarrolladores a crear aplicaciones, a los administradores a proteger la integridad de los datos y a crear entornos tolerantes a fallas, y a ayudarlo a administrar sus datos sin importar cuán grande o pequeño sea el conjunto de datos. Además de ser gratuito y de código abierto , PostgreSQL es altamente extensible. Por ejemplo, puede definir sus propios tipos de datos, crear funciones personalizadas, ¡incluso escribir código desde diferentes lenguajes de programación sin recompilar su base de datos!

PostgreSQL intenta cumplir con el estándar SQL donde dicha conformidad no contradice las características tradicionales o podría conducir a malas decisiones arquitectónicas. Se admiten muchas de las características requeridas por el estándar SQL, aunque a veces con una sintaxis o función ligeramente diferente. Se pueden esperar más avances hacia la conformidad con el tiempo. A partir del lanzamiento de la versión 12 en octubre de 2019, PostgreSQL se ajusta a al menos 160 de las 179 características obligatorias para SQL: conformidad con 2016 Core. Al momento de escribir este artículo,

ninguna base de datos relacional cumple con la plena conformidad con este estándar.

A continuación se muestra una lista inagotable de varias características que se encuentran en PostgreSQL, y se agregan más en cada versión principal:

### Tipos de datos

- Primitivas: Entero, Numérico, Cadena, Booleano
- Estructurado: fecha / hora, matriz, rango, UUID
- Documento: JSON / JSONB, XML, clave-valor (Hstore)
- Geometría: Punto, Línea, Círculo, Polígono
- Personalizaciones: compuesto, tipos personalizados

### Integridad de los datos

- ÚNICO, NO NULO
- Claves primarias
- Llaves extranjeras
- Restricciones de exclusión
- Cerraduras explícitas, cerraduras de aviso

### Concurrencia, rendimiento

- Indexación: B-tree, Multicolumn, Expresiones, Parcial
- Indexación avanzada: GiST, SP-Gist, KNN Gist, GIN, BRIN, índices de cobertura, filtros Bloom
- Sofisticado planificador / optimizador de consultas, escaneos de solo índice, estadísticas de varias columnas
- Transacciones, transacciones anidadas (a través de puntos de guardado)
- Control de concurrencia de versiones múltiples (MVCC)
- Paralelización de consultas de lectura y creación de índices de árbol B
- Particionamiento de tabla

- Todos los niveles de aislamiento de transacciones definidos en el estándar SQL, incluido Serializable
- Justo a tiempo (JIT) compilación de expresiones

### Fiabilidad, recuperación ante desastres

- Registro de escritura anticipada (WAL)
- Replicación: asíncrona, síncrona, lógica
- Recuperación de punto en el tiempo (PITR), en espera activa
- Espacios de mesa

### Seguridad

- Autenticación: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificado y más
- Robusto sistema de control de acceso
- Seguridad de nivel de columna y fila
- Autenticación multifactor con certificados y un método adicional.

### Extensibilidad

- Funciones y procedimientos almacenados
- Lenguajes de procedimiento: PL / PGSQL, Perl, Python (y muchos más)
- Expresiones de ruta SQL / JSON
- Contenedores de datos externos: conéctese a otras bases de datos o secuencias con una interfaz SQL estándar
- Interfaz de almacenamiento personalizable para mesas
- Muchas extensiones que proporcionan funcionalidad adicional, incluido PostGIS

### Internacionalización, Búsqueda de texto

- Soporte para juegos de caracteres internacionales, por ejemplo, a través de intercalaciones de UCI
- Cotejos insensibles a mayúsculas y minúsculas

- Búsqueda de texto completo. (The PostgreSQL Global Development Group, 2020)

### 2.10.5. Leaflet

Es una moderna biblioteca JavaScript de código abierto para mapas interactivos adaptable para móviles, una popular opción para crear tu propio mapa deslizante.

Está desarrollado por Vladimir Agafonkin, previamente a través de CloudMade pero desarrollado ahora mediante Mapbox, con un equipo de contribuidores dedicados. Es bastante menos complejo que OpenLayers y a diferencia de aquél se centra en el rendimiento, la facilidad de uso, una API simple, pequeño tamaño y soporte móvil.

Se usa en el mapa del sitio web principal de OSM, así como en Flickr, aplicaciones móviles de Wikipedia, foursquare, craigslist, IGN, Washington Post, The Wall Steet Journal, Geocaching.com, City-Data.com, StreetEasy, Nestoria y Skobbler, entre otros. (OpenStreetMap Wiki Contributos, 2020)

### 2.10.6. Nestjs

Está resultando bastante común ver cómo en eventos puramente de Angular, el famoso framework para el desarrollo front, se están colando charlas sobre NestJS. Un framework para el desarrollo en el lado del servidor. Esto es debido a que se inspira en los principios básicos de Angular.

Esta herramienta está creciendo en popularidad a un ritmo vertiginoso entre los desarrolladores de NodeJS, además sus creadores están cuidando mucho la documentación que están elaborando.

**Aportes**

Nest aporta un gran punto de partida para el desarrollo con NodeJS. En nuestra opinión, el desarrollo con Node puede resultar un poco caótico y si no se tiene

una buena metodología y estructura puede que acabemos creando un proyecto difícil de mantener y evolucionar.

NestJS soluciona por completo este problema. Además nos permite integrarlo con todo el ecosistema de librerías que rodea este famoso lenguaje de alta concurrencia y todo esto usando Typescript de fondo. (DIGITAL55, 2020).

# CAPITULO III

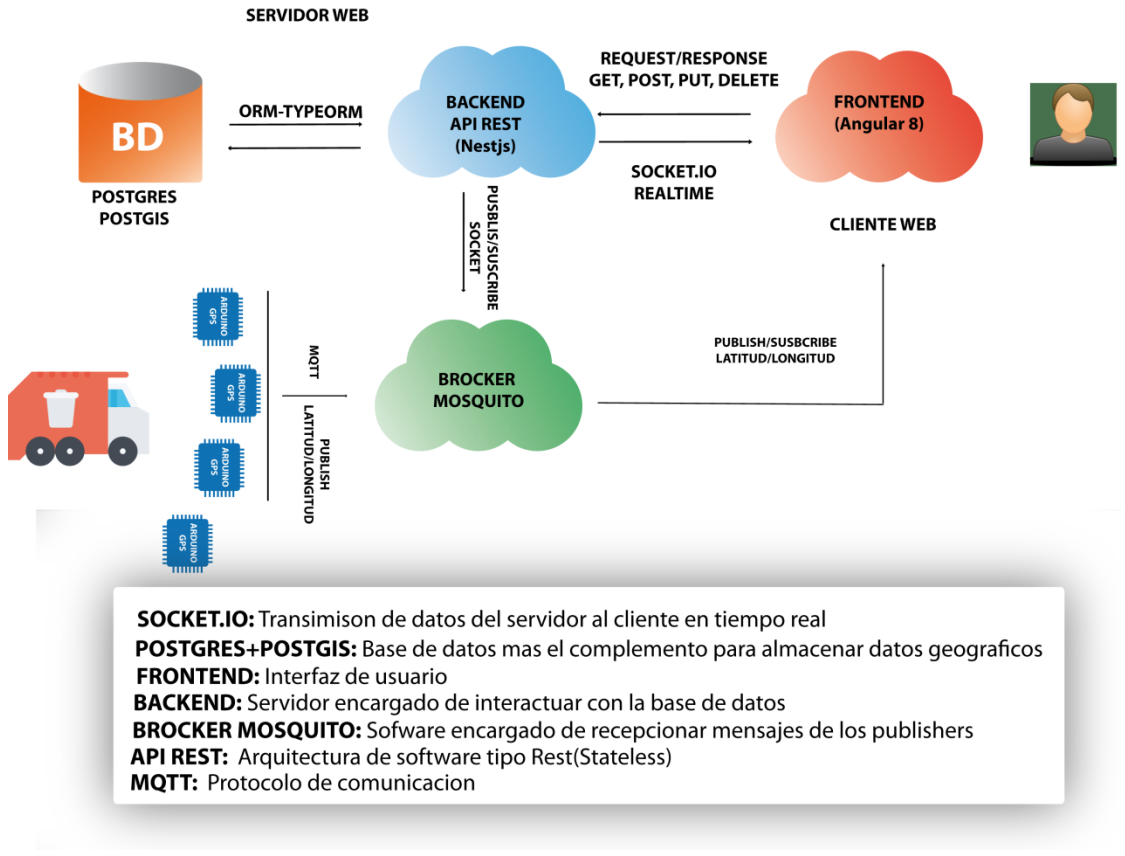
# MARCO APLICATIVO

A continuación, abordaremos el proceso de análisis, diseño, desarrollo e implementación del software aplicando la Metodología UWE. Por otro lado describirá el desarrollo del hardware levantando los servicios correspondientes en función a la base teórica ya definida.

## 3.1. ESQUEMA DEL PROYECTO

En este apartado se muestra una representación gráfica básica del funcionamiento del sistema y la interactuación de la misma.

**Figura Nº 3.1:** Esquema de proyecto Sistema de Geo localización




**Fuente:** (Elaboración propia)

## 3.2. METODOLOGIA UWE

### 3.2.1. Fase de Inicio

Dentro de esta fase, se aborda la estructura organizacional y funcional de la Unidad de Medio Ambiente perteneciente al Gobierno Autónomo Municipal de Viacha, describiendo los procesos que se realizan identificando los actores, para de esta manera, comenzar el desarrollo del sistema propuesto.

#### 3.2.1.1. Análisis de la situación actual

##### 3.2.1.1.1. Investigación preliminar

En este proceso del presente trabajo se presentan tareas adicionales que nos permita tener una visión global y exhaustiva de la institución y todo lo que conlleva los procesos que se cumplen, valga la redundancia, dentro de la institución.

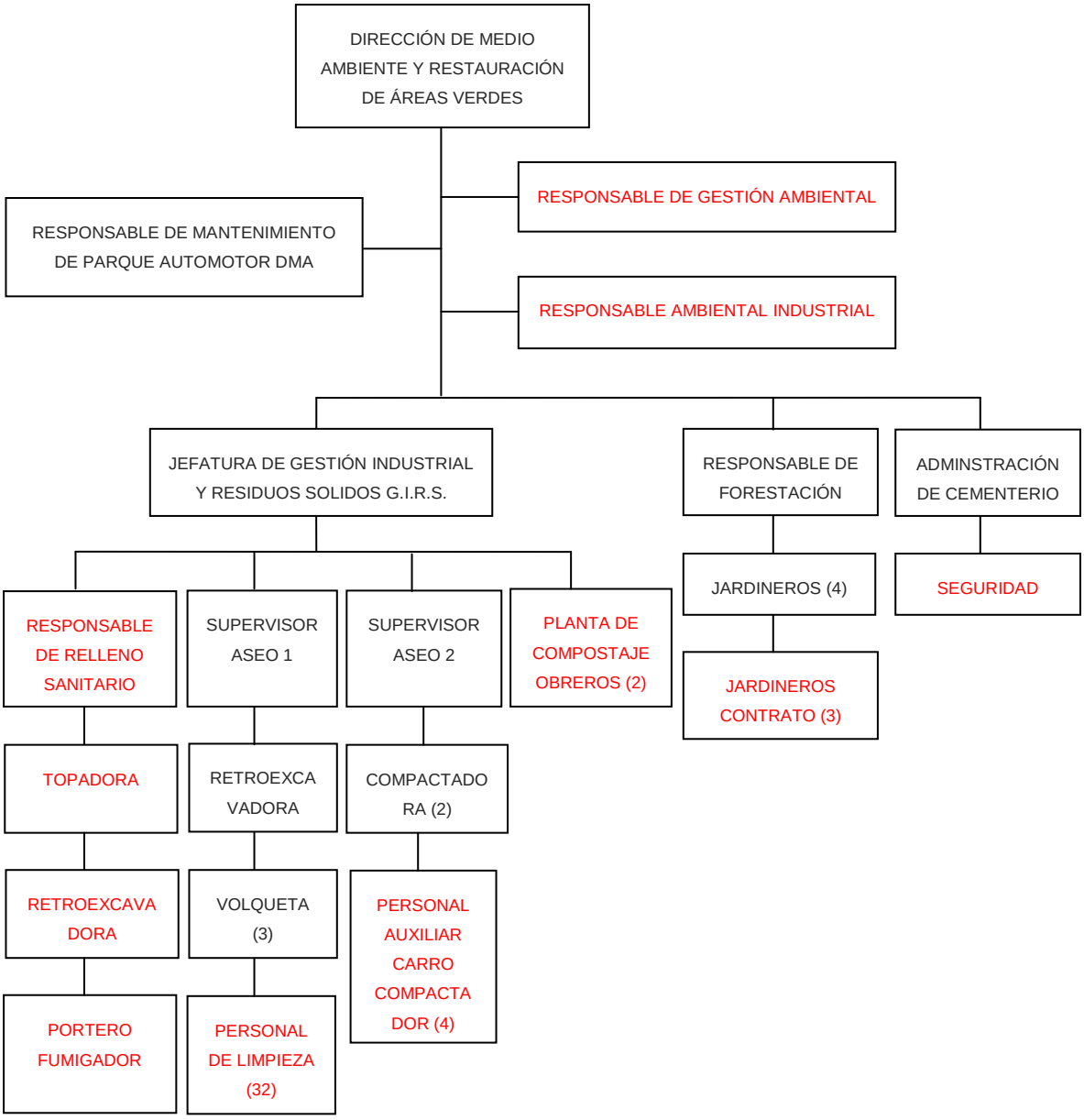
Por lo tanto, en esta etapa se hizo la revisión correspondiente de documentos proveídos por la institución, así también, se realizó las entrevistas necesarias tomando apuntes de puntos importantes acerca del proceso que conlleva la recolección de basura mediante los vehículos encargados de las mismas.

##### 3.2.1.1.2. Estructura organizacional

Si bien en el presente trabajo se habla del Gobierno Autónomo Municipal de Viacha, como sabremos, la estructura organizativa es bastante grande y compleja, es por eso que básicamente se hará énfasis en la unidad correspondiente a la gobernación de Viacha encargado de ejecutar y controlar el recojo de basura a través de los vehículos motorizados.

Por lo tanto, a continuación se realiza una descripción de la estructura organizativa de la Unidad de Medio Ambiente del Gobierno Autónomo Municipal de Viacha.

**Figura Nº 3.2:** Estructura Organizacional – Dirección de Medio Ambiente




**Fuente:** (Delfín, 2012)

### 3.2.1.1.3. Descripción de funciones

Habiendo entendido la estructura organizacional del lugar o caso de estudio se precedió a la identificación de cargos y su respectiva función quienes están involucrados en el sistema.

**Tabla Nº 3.1:** Descripción de funciones de cargos identificados

| CARGO | FUNCION |
|---|---|
| **RESPONSABLE DE GESTIÓN AMBIENTAL** | Encargado de ejecutar y controlar los proyectos de protección del medio ambiente, a través de procesos de recolección de basura con diferentes estrategias u otros tipos de proyectos que tengan que ver con el medio ambiente. |
| **RESPONSABLE AMBIENTAL INDUSTRIAL** | Encargado de ejecutar y controlar proyectos y/o estrategias para la protección del medio ambiente en cuanto a la incidencia que tienen las industrias de producción en todo de tipo de sectores, ya que el impacto causado es bastante alto y complejo. |
| **SUPERVISOR DE ASEO** | Encargado de controlar y dar solución a los procesos de recolección de basura entendiendo las tareas inherentes como limpieza y separación de basura. |
| **VOLQUETA (CONDUCTOR)** | Hace referencia al personal que conduce los vehículos encargados de transportar la basura, primero recolectando las mismas en rutas pre-establecidas y luego llevándolos a al relleno sanitario correspondiente de la Gobernación. |
| **PERSONAL AUXILIAR CARRO COMPACTADOR** | Personal que coadyuva en la recolección y compactación de la basura recibida para su posterior tratamiento y/o almacenado en el relleno sanitario. |

**Fuente:** (Delfín, 2012)

### 3.2.1.2. Identificación de actores

Para la presente se darán a conocer los usuarios identificados en el sistema, en este caso, aquellos que estarán involucrados, tomando en cuenta el análisis realizado de los cargos existentes en la estructura orgánica de la Unidad de Medio Ambiente, es en base a ello que se identificarán los actores principales y más.

**Tabla Nº 3.2:** Identificación de actores

| | DESCRIPCIÓN |
|---|---|
| **ADMINISTRADOR** | Persona encargada de:<br>➢ Habilitar usuarios de tipo: Operador, Supervisor, conductor con los roles y permisos correspondientes. |
| **SUPERVISOR** | Persona encargada de:<br>➢ Controlar las rutas realizadas por los vehículos.<br>➢ Solucionar problemas sobre el servicio.<br>➢ Realizar el manejo de rutas ya sea de adición, modificación o eliminación.<br>➢ Verificar estadísticas, informes.<br>➢ Realizar el manejo de la información de los conductores, vehículos.<br>➢ Registrar información necesaria concerniente a la basura. |
| **USUARIO** | Usuario normal de la población quien se encarga de realizar el monitoreo de los vehículos obteniendo:<br>➢ La ubicación<br>➢ Los conductores y ayudantes encargados.<br>➢ Datos del vehículo.<br>➢ Muestra de las rutas a cumplir<br>➢ Tiempos de llegada y distancias. |
| **DISPOSITIVO** | Hacemos referencia al dispositivo de desarrollo de hardware, la cual realizara el envio correspondiente de datos al servidor como ser: ubicación en latitud y longitud, hora, cada cierto tiempo con las señalizaciones correspondientes. |

**Fuente:** Elaboración propia en base a (Delfín, 2012).

### 3.2.1.3. Modelo de negocio

En este punto se abordó las técnicas necesarias para la identificación de los procesos del servicio, desde el punto de vista del usuario final. Se entiende como una manera narrativa a través de la cual se podrá entender todos los procesos y sus relaciones.

El modelo de negocio explica gráficamente, a través, de diagramas, la interacción de los actores identificados y los procesos o tareas que las relacionan.

**Figura Nº 3.3:** Modelo de caso de uso del negocio


**Fuente: (**Elaboración propia)

Dentro de la institución de estudio, se identificó los casos de uso que describen los procesos de administración, estos procesos se describen a continuación:

- **Administración de roles y permisos:** Este proceso representa el manejo que se debe realizar para el acceso al sistema por parte de los diferentes usuarios.
- **Registro y configuración de dispositivo:** Como se puede observar el encargado de este proceso o caso de uso es el administrador, ya que

este proceso es muy importante y delicado para el funcionamiento del hardware de geo localización.

- **Registro de Vehículos:** Este proceso describe los datos a registrar, característicos del vehículo encargado de recolectar la basura, así también los encargados de conducir la misma y los ayudantes relacionados. Este proceso lo puede realizar los roles administrador y supervisor.
- **Registro de rutas y puntos de acopio:** En este proceso se realizan las operaciones básicas de las rutas y puntos de acopio, se puede entender por ejemplo como el registro de rutas en un mapa la cual se almacena en una base de datos para su posterior uso.
- **Registro de datos de basura:** Este proceso hace referencia al registro de la información de la basura como fechas, cantidades, viajes o cualquier evento que tenga que ver con estos datos.
- **Geo localización de vehículo:** Este proceso lo realiza el acto dispositivo la cual está conectado al servidor para enviarle datos de ubicación del vehículo al cual está instalado como ser latitud longitud distancia y hora.

#### 3.2.1.4. Requerimientos funcionales y no funcionales

La ingeniería de requerimientos es una parte muy importante del desarrollo de software, ya que la misma nos permite identificar las funcionalidades que debería cumplir el sistema para colmar las expectativas de los usuarios que participan en la misma. Así también se definen las restricciones sobre el sistema.

La descripción clara de los requerimientos a cumplir en el sistema nos evita de ambigüedades, una mejor construcción del software consistente y compacto.

Las funciones que debe realizar se clasifican en tres categorías como se detallan en la siguiente tabla

**Tabla Nº 3.3:** Categoría de las funciones

| CATEGORIA DE LA FUNCIÓN | SIGNIFICADO |
|---|---|
| **Evidente** | Esta categoría hace referencia a lo que debe realizarse, y el usuario debería saber que se realizó, a través de diferentes tipos de avisos, notificaciones, salidas, informes, etc. |
| **Oculto** | Debe realizarse aunque no necesariamente el usuario debe verificar la misma. Esto puede hacer referencia a procesos como por ejemplo el registro de información, almacenamiento o algún tipo de persistencia de datos, servicios necesarios para la comunicación hardware software, etc. |
| **Superflua** | Su inclusión no repercute significativamente en el costo ni en otras funciones. |

**Fuente:** (Elaboración propia)

### 3.2.1.4.1. Requerimientos funcionales

Los requerimientos funcionales nos permiten identificar las funciones primarias propias del sistema las cuales cumplirán los requerimientos de acuerdo al usuario. Una función se describe como un conjunto de entradas, procesos o comportamientos y salidas. Los requerimientos funcionales, pueden ser: cálculos, detalles técnicos, manipulación de datos como la interactuación con la base de datos o archivos.

Estos requerimientos funcionales se deben cumplir en función al modelo de negocio descrito anteriormente, las mismas están complementadas con los requerimientos no funcionales.

Los requerimientos funcionales se describen en la siguiente tabla:

**Tabla Nº 3.4:** Requerimientos funcionales

| REF. | FUNCIÓN | CATEGORIA |
|---|---|---|
| **R1** | Acceso al sistema por roles de usuario (Administrador, Supervisor, Usuario) | Evidente |
| **R2** | Registrar, modificar, o eliminar usuarios del sistema. | Evidente |
| **R3** | Registrar, modificar, o eliminar puntos de acopio para recolección de basura. | Evidente |
| **R4** | Registrar, modificar, o eliminar datos de vehículos encargados de la recolección de basura. | Evidente |
| **R5** | Desplegar información de ubicación de vehículo a través de la latitud y longitud del dispositivo de geo localización en tiempo real. | Evidente |
| **R6** | Registrar, modificar o eliminar, datos de dispositivo de hardware de geo localización relacionado a los vehículos. | Evidente |
| **R7** | Notificación de vehículos cerca para la recolección de basura. | Evidente |
| **R8** | Reporte de listado de rutas generales de recolección de basura. | Evidente |
| **R9** | Visualización de puntos de acopio para el recojo de basura. | |

**Fuente:** (Elaboración propia)

### 3.2.1.4.2. Requerimientos no funcionales

**Tabla Nº 3.5:** Requerimientos no funcionales

| REF. | FUNCIÓN | CATEGORÍA |
|---|---|---|
| **R1** | La información en tiempo real de la geo localización otorgada al usuario sea correcta. | Evidente |
| **R2** | El sistema debe tener seguridad en el acceso a la información. | Evidente |
| **R3** | El sistema funcionará correctamente en navegadores que soporten HTML5. | Evidente |
| **R4** | El sistema debe tener una interfaz amigable y funcional para el usuario normal, y el usuario de tipo administrador. | Evidente |

**Fuente:** (Elaboración propia)

### 3.2.2. Fase de elaboración

#### 3.2.2.1. Análisis y diseño del sistema web

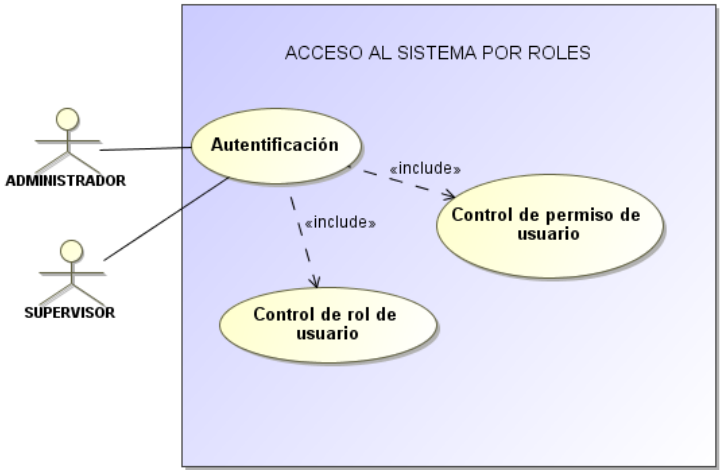
En este punto se realizó el análisis de los casos de uso intervinientes en el proceso para la geo localización de vehículos encargados del recojo de basura.

##### 3.2.2.1.1. Análisis del sistema

**a) Diagramas de casos de uso**

**DIAGRAMA DE CASO DE USO:** ACCESO AL SISTEMA POR ROLES

**Figura Nº 3.4:** Acceso al sistema por roles




**Fuente:** (Elaboración propia)

**Tabla Nº 3.6:** Caso de uso: acceso al sistema por roles

| **Caso de Uso: Acceso al sistema por roles** | |
|---|---|
| **Actores:** | Administrador, Supervisor |
| **Tipo:** | Primario Esencial |
| **Descripción:** | Este caso de uso describe el acceso al sistema, tomando en cuenta el rol del usuario y los permisos que han sido designados para restringir las operaciones que puede realizar. El caso de usuario normal no necesitar acceso con credenciales al sistema ya que se restringirá a solo visualizar datos. |

**Fuente:** (Elaboración propia)

**DIAGRAMA DE CASO DE USO:** ADMINISTRACION DEL SISTEMA

**Figura Nº 3.5:** Administración del sistema


**Fuente:** (Elaboración propia)

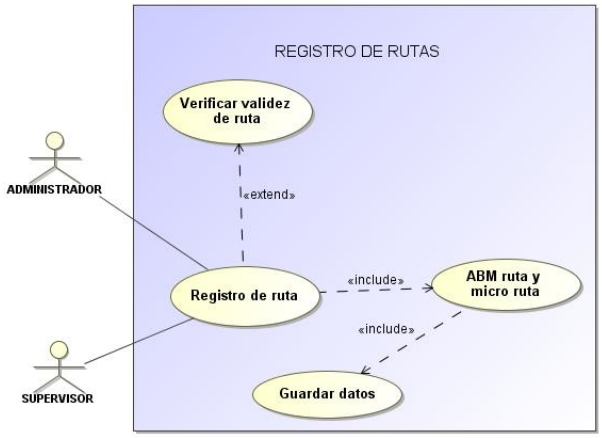
**Tabla Nº 3.7:** Caso de uso: Administración del sistema

| **Caso de uso: Administración del sistema** | |
|---|---|
| **Actores:** | Administrador |
| **Tipo:** | Primordial esencial |
| **Descripción:** | El usuario de tipo administrador tiene los niveles correspondientes para agregar otros usuarios, asignar tareas y roles, así también, habilita y deshabilita funciones en función a lo necesario. |

**Fuente:** (Elaboración propia)

**DIAGRAMA DE CASO DE USO:** REGISTRO DE RUTAS

**Figura Nº 3.6:** Registro de rutas




**Fuente:** (Elaboración propia)

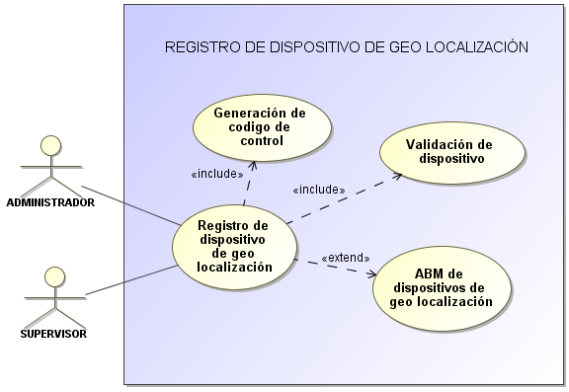
**Tabla Nº 3.8:** Caso de Uso: Registro de rutas

| **Caso de Uso: Registro de rutas** | |
|---|---|
| **Actores:** | Administrador, Supervisor |
| **Tipo:** | Primario Esencial |
| **Descripción:** | El personal responsable del registro de rutas, tiene la función de tomar los datos correspondientes y así también la de trazar las rutas para su posterior cumplimiento de la misma por parte de los vehículos encargados del recojo de basura. La misma será verificada y está compuesta por micro rutas, pudiendo realizar operaciones como de agregación, eliminación, modificación, y listado. |

**Fuente:** (Elaboración propia)

**DIAGRAMA DE CASO DE USO**: REGISTRO DE DISPOSITIVO

**Figura Nº 3.7:** Registro de dispositivo de geo localización




**Fuente:** (Elaboración propia)

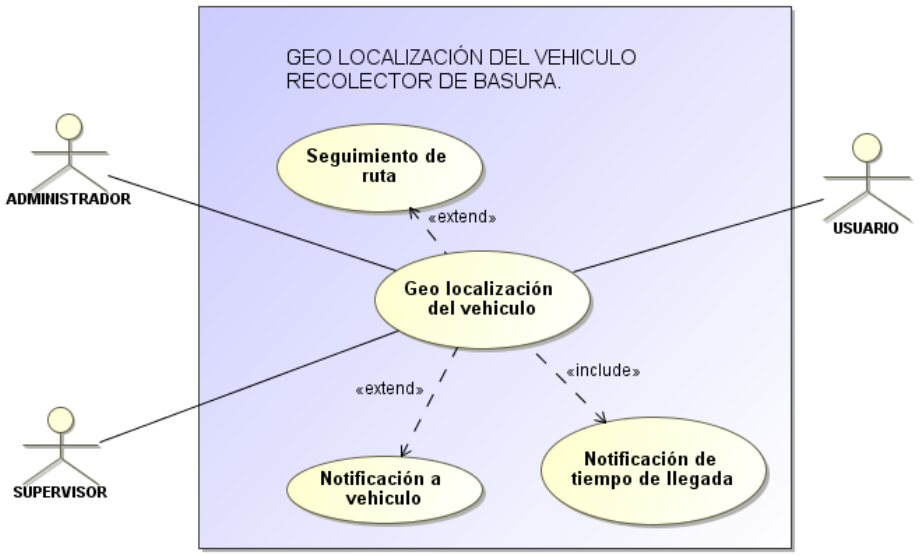
**Tabla Nº 3.9:** Caso de uso: Registro de dispositivo de geo localización

| **Caso de uso: Registro de dispositivo de geo localización** | |
|---|---|
| **Actores:** | Administrador, Supervisor |
| **Tipo:** | Primario esencial |
| **Descripción:** | Los usuarios, en este caso, de tipo Administrador y Supervisor, tendrán la función de registrar dispositivos de geo localización que estarán instaladas en los vehículos que harán la recolección de basura. Se generará un código para poder validad el dispositivo dentro de todo el sistema, y así también se podrá realizar las operaciones básicas de: Agregar, Modificar, Eliminar, Listar todos los dispositivos. |

**Fuente:** (Elaboración propia)

**DIAGRAMA DE CASO DE USO:** GEO LOCALIZACIÓN DEL VEHICULO

**Figura Nº 3.8:** Geo localización del vehículo recolector de basura




**Fuente:** (Elaboración propia)

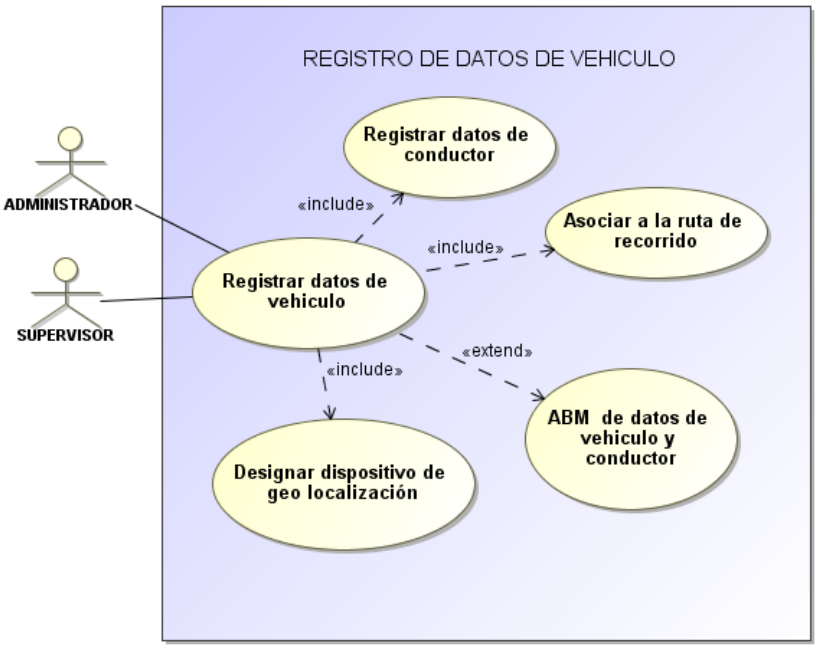
**Tabla Nº 3.10:** Caso de uso: Geo localización del vehículo

| **Caso de uso: Geo localización del vehículo** | |
|---|---|
| **Actores:** | Administrador, Supervisor, Usuario |
| **Tipo:** | Primario Esencial |
| **Descripción:** | La funcionalidad comprende en la visualización de la geo localización de los vehículos en un respectivo mapa, notificando ubicación, tiempo de llegada, y aquellos más cercanos. Por otro lado la misma estará correlacionada con las rutas previamente definidas por lo tanto tendrá que ir cumpliendo la misma. Así también los administradores, operadores o supervisores podrán enviar notificaciones al dispositivo. |

**Fuente:** (Elaboración propia)

**DIAGRAMA DE CASO DE USO:** REGISTRO DE DATOS DE VEHICULO.

**Figura Nº 3.9:** Registro de datos de vehículo




**Fuente:** (Elaboración propia)

**Tabla Nº 3.11:** Caso de uso: Registro de datos de vehículo

| **Caso de uso: Registro de datos de vehículo** | |
|---|---|
| **Actores:** | Administrador, Supervisor. |
| **Tipo:** | Primario Esencial |
| **Descripción:** | El personal designado como: Administrador, Supervisor y Operador, proceden a tomar los datos característicos del vehículo, así también los datos del conductor designado. Por otro lado este registro asociará a un dispositivo de geo localización y una ruta a cumplir para de esta manera ofrecer la información necesaria en el sistema. |

**Fuente:** (Elaboración propia)

**DIAGRAMA DE CASO DE USO:** REGISTRO DE PUNTOS DE ACOPIO.

**Figura Nº 3.10:** Registro de puntos de acopio


**Fuente:** (Elaboración propia)

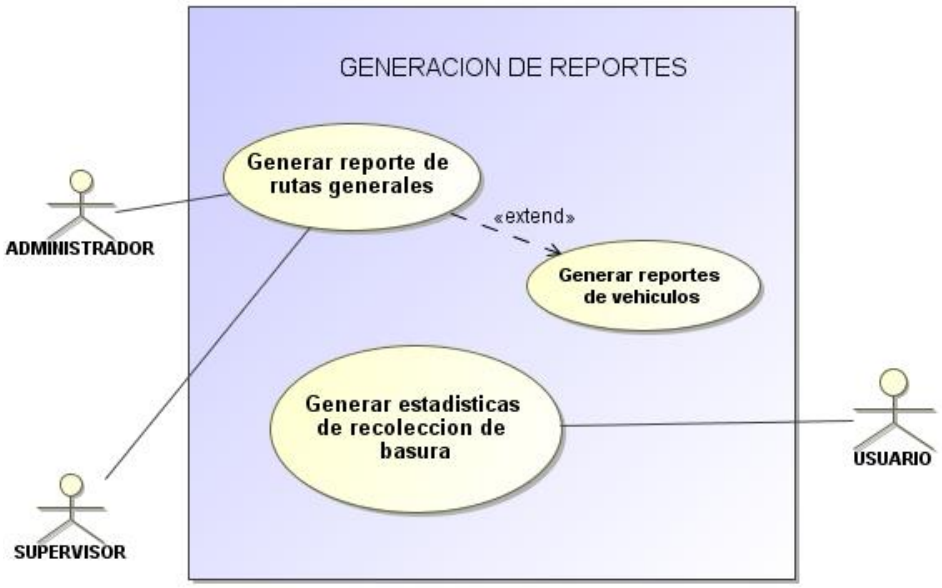
**Tabla Nº 3.12:** Caso de uso: Registro de puntos de acopio

| **Caso de uso: Registro de punto de acopio** | |
|---|---|
| **Actores:** | Administrador, Supervisor, Usuario |
| **Tipo:** | Primario Esencial |
| **Descripción:** | Los usuarios en este caso: Administrador, Supervisor y Operador, realizaran el registro de puntos de acopio dentro de un mapa mostrando la ubicación del Municipio de Viacha. El Usuario normal podrá visualizar este mapa para ver los puntos autorizados. Por otro lado se tendrá las opciones de Adicionar, Modificar, Eliminar y Listar los respectivos datos. |

**Fuente:** (Elaboración propia)

**DIAGRAMA DE CASO DE USO:** GENERACION DE REPORTES.

**Figura Nº 3.11:** Generación de reportes




**Fuente:** (Elaboración propia)

**Tabla Nº 3.13:** Caso de uso: Generación de reportes

| **Caso de uso: Generación de reportes** | |
|---|---|
| **Actores:** | Administrador, Supervisor, Usuario |
| **Tipo:** | Primario Esencial |
| **Descripción:** | Los usuarios que administran el sistema generan informes de diferentes tipos relacionados con: dispositivos de geo localización, vehículos y conductores, rutas y micro rutas, así también el usuario visitante podrá generar estadísticas acerca de la basura generada. |

**Fuente:** (Elaboración propia)

### b) Diagramas de secuencia

En este apartado se implementa los diagramas secuenciales posteriores a haber realizado la ingeniería de requerimientos y los casos de uso general y especifico habiendo entendido de manera estática la funcionalidad que tendrá el

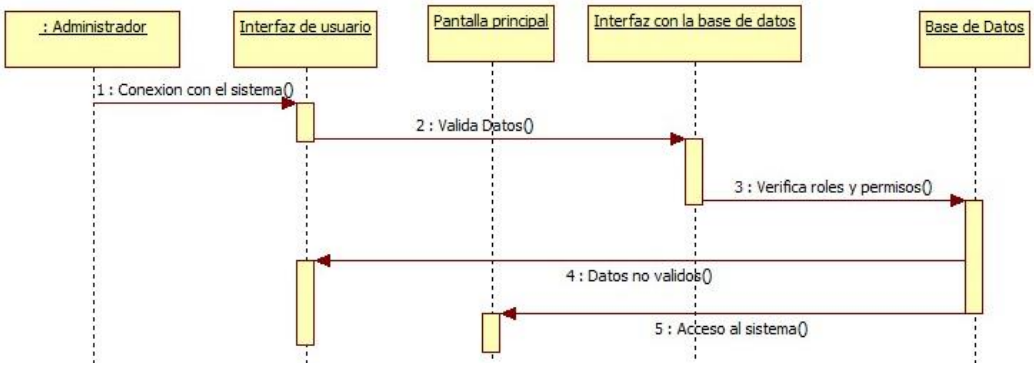
sistema. Los diagramas de secuencias nos muestran gráficamente los eventos y operaciones que se realizan e influyen los actores en el sistema.

**DIAGRAMA DE SECUENCIAS:** ACCESO AL SISTEMA POR ROLES

El primer diagrama de secuencia hace referencia a el acceso al sistema este solo afecta a los usuarios: administrador, supervisor y operador, ya que ellos realizaran las operaciones sobre los datos del sistema, por otro lado el usuario normal simplemente podrá monitorear directamente los vehículos y puntos de acopio para el recojo de basura pudiendo ver también estadísticas acerca de la basura.

Aquellos usuarios como: administrador, Supervisor y Operador realizaran el ingreso de nombre de usuario y contraseña en un formulario, se verificara la existencia en la base de datos y si es así podrá acceder a las funcionalidades del sistema y caso contrario su acceso será rechazado.

**Figura Nº 3.12:** Diagrama de secuencias: Acceso a al sistema por roles



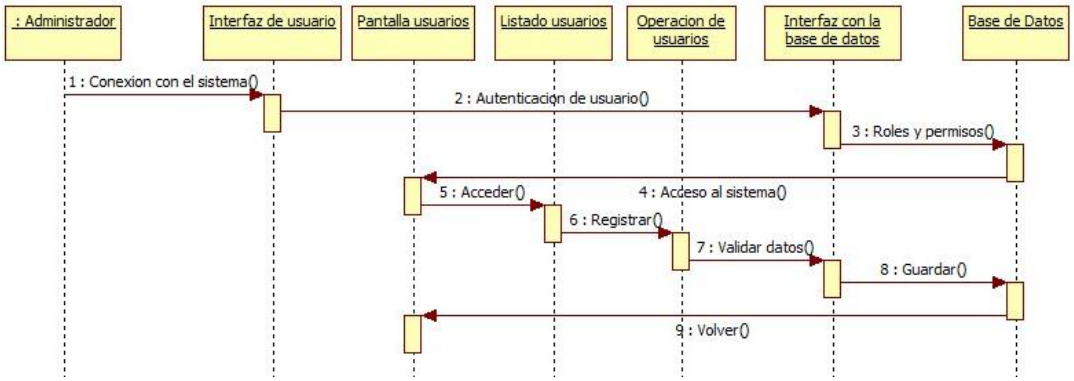

**Fuente:** (Elaboración propia)

**DIAGRAMA DE SECUENCIAS:** ADMINISTRACIÓN DEL SISTEMA

Este diagrama describe la administración del sistema en cuanto al usuario permisos y roles. El usuario de tipo Administrador ya registrado será quien dará de alta a nuevos usuarios ya sea de tipo Supervisores y Operadores, otorgándole los permisos correspondientes.

Describe el proceso de listado de los usuarios, con las operaciones de registro, modificación eliminación, y el ABM correspondiente para la asignación de permisos. Para agregar un nuevo usuario se deberá ingresar datos básicos de la persona en un formulario y el sistema lo almacenará en la base de datos.

**Figura Nº 3.13:** Diagrama de secuencias: Administración del sistema



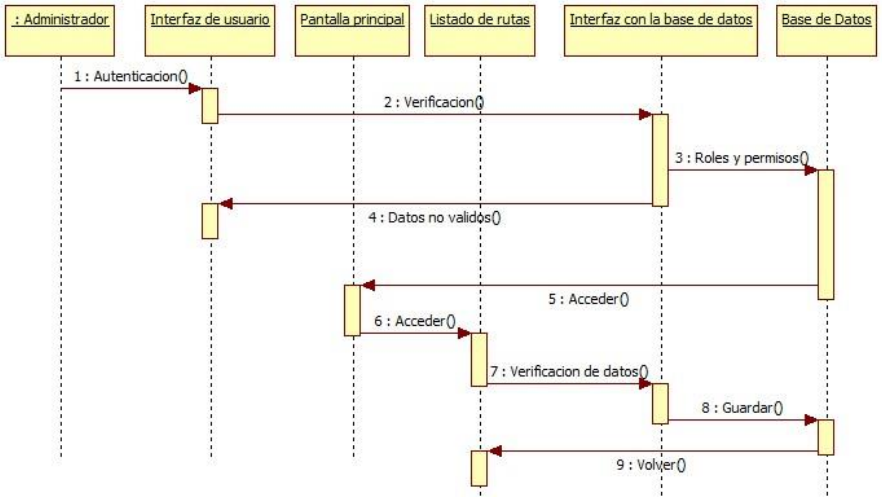

**Fuente:** (Elaboración propia)

**DIAGRAMA DE SECUENCIAS:** REGISTRO DE RUTAS

En este proceso se describe que los usuarios se realizarán la autenticación correspondiente para luego llevarlos a la pantalla principal, cabe aclarar, que la base de datos devolverá permisos y roles para verificar las operaciones que el usuario puede realizar.

Por otro lado ya autentificado en el sistema en la pantalla de listado de rutas estarán todas las operaciones de agregación, eliminación o modificación, todos los datos son verificados para su validez y posteriormente se almacena en la base de datos. Todo este proceso es de suma importancia ya que estas rutas están compuestas por micro rutas las cuales perteneces a un distrito en particular e eran enlazadas a los vehículos correspondientes para la geo localización.

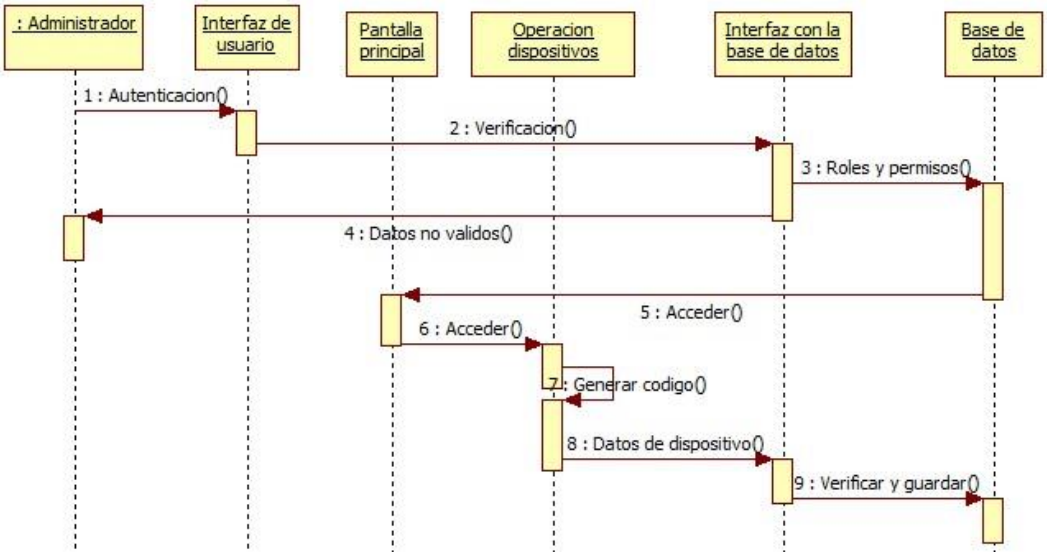
**Figura Nº 3.14:** Diagrama de secuencias: Registro de rutas




**Fuente:** (Elaboración propia)

**DIAGRAMA DE SECUENCIAS:** REGISTRO DE DISPOSITIVO

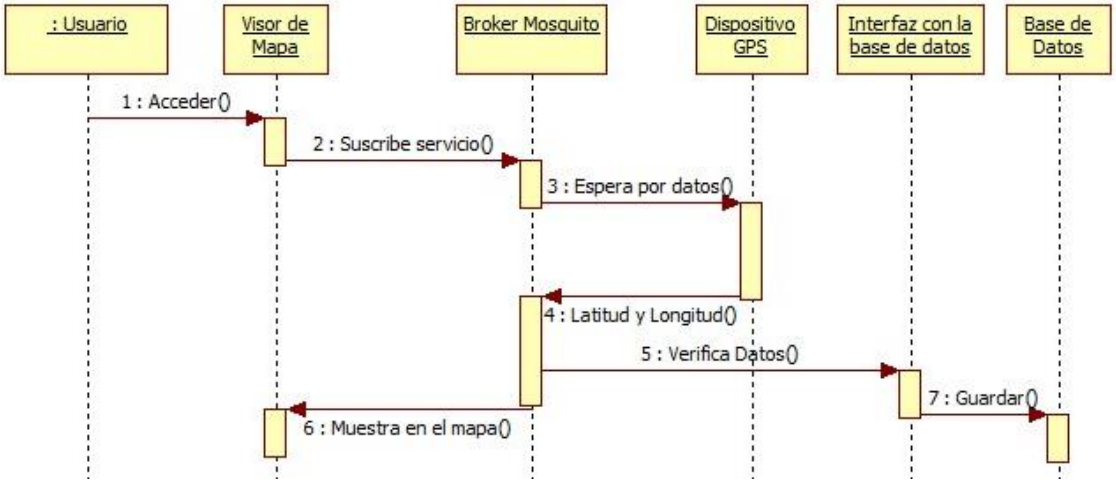
**Figura Nº 3.15:** Diagrama de secuencia: Registro de dispositivo




**Fuente:** (Elaboración propia)

**DIAGRAMA DE SECUENCIAS:** GEO LOCALIZACION DEL VEHICULO RECOLECTOR DE BASURA

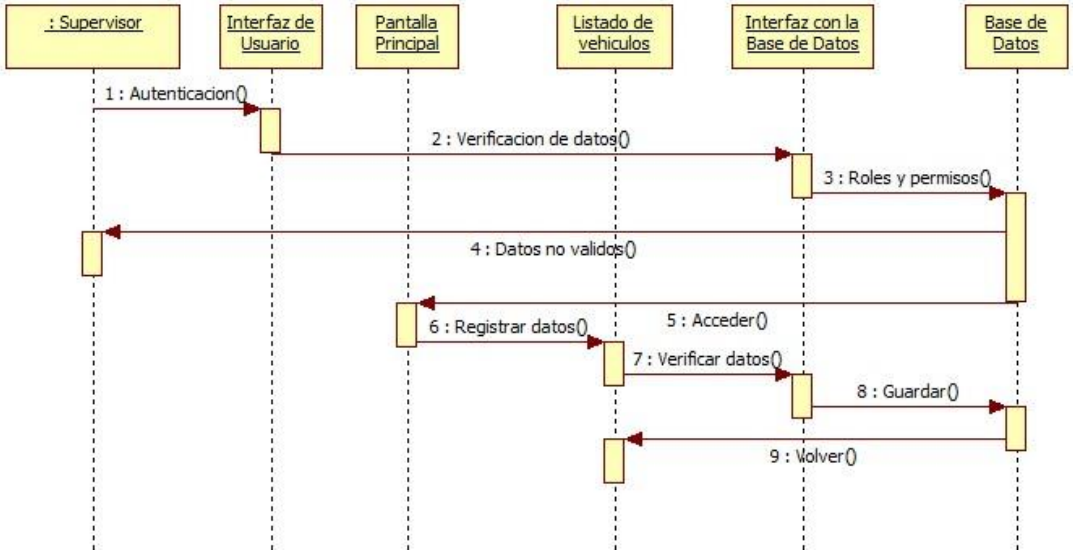
**Figura Nº 3.16:** Diagrama de secuencia: Geo Localización del vehículo




**Fuente:** (Elaboración propia)

**DIAGRAMA DE SECUENCIAS:** REGISTRO DE DATOS DE VEHICULO

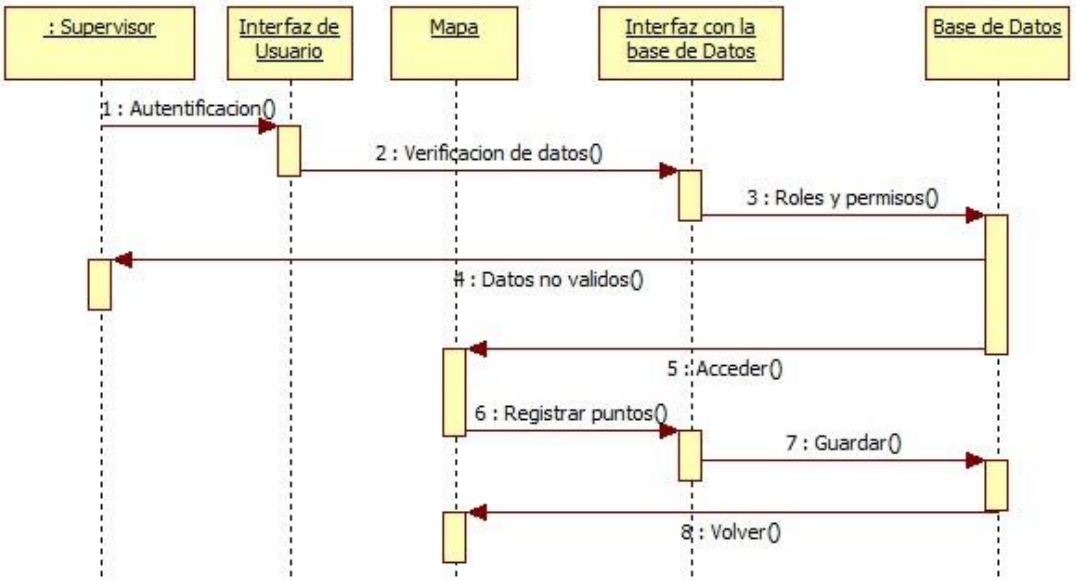
**Figura Nº 3.17:** Diagrama de secuencia: Registro de datos de vehículo




**Fuente:** (Elaboración propia)

**DIAGRAMA DE SECUENCIAS:** REGISTRO DE PUNTOS DE ACOPIO

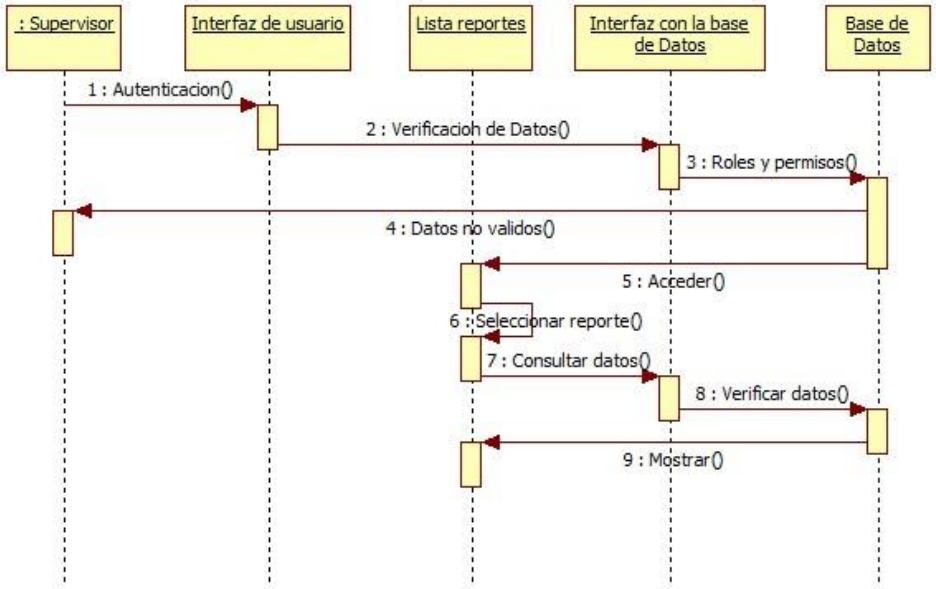
**Figura Nº 3.18:** Diagrama de secuencia: Registro de puntos de acopio




**Fuente:** (Elaboración propia)

**DIAGRAMA DE SECUENCIAS:** GENERACION DE REPORTES

**Figura Nº 3.19:** Diagrama de secuencia: Generación de reportes

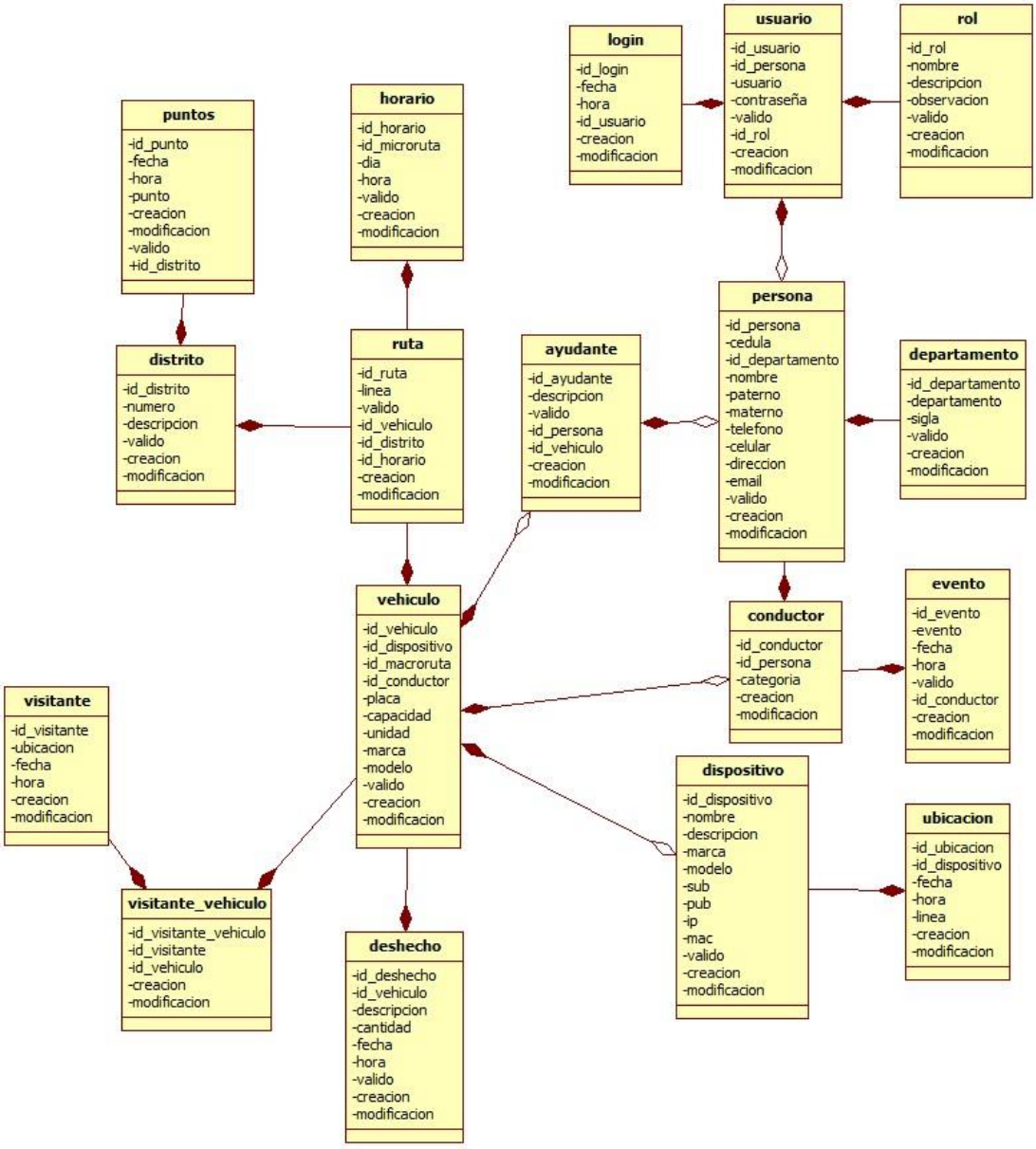



**Fuente:** (Elaboración propia)

### 3.2.2.1.2. Diseño del sistema

#### a) Modelo conceptual (Diagrama de clases)

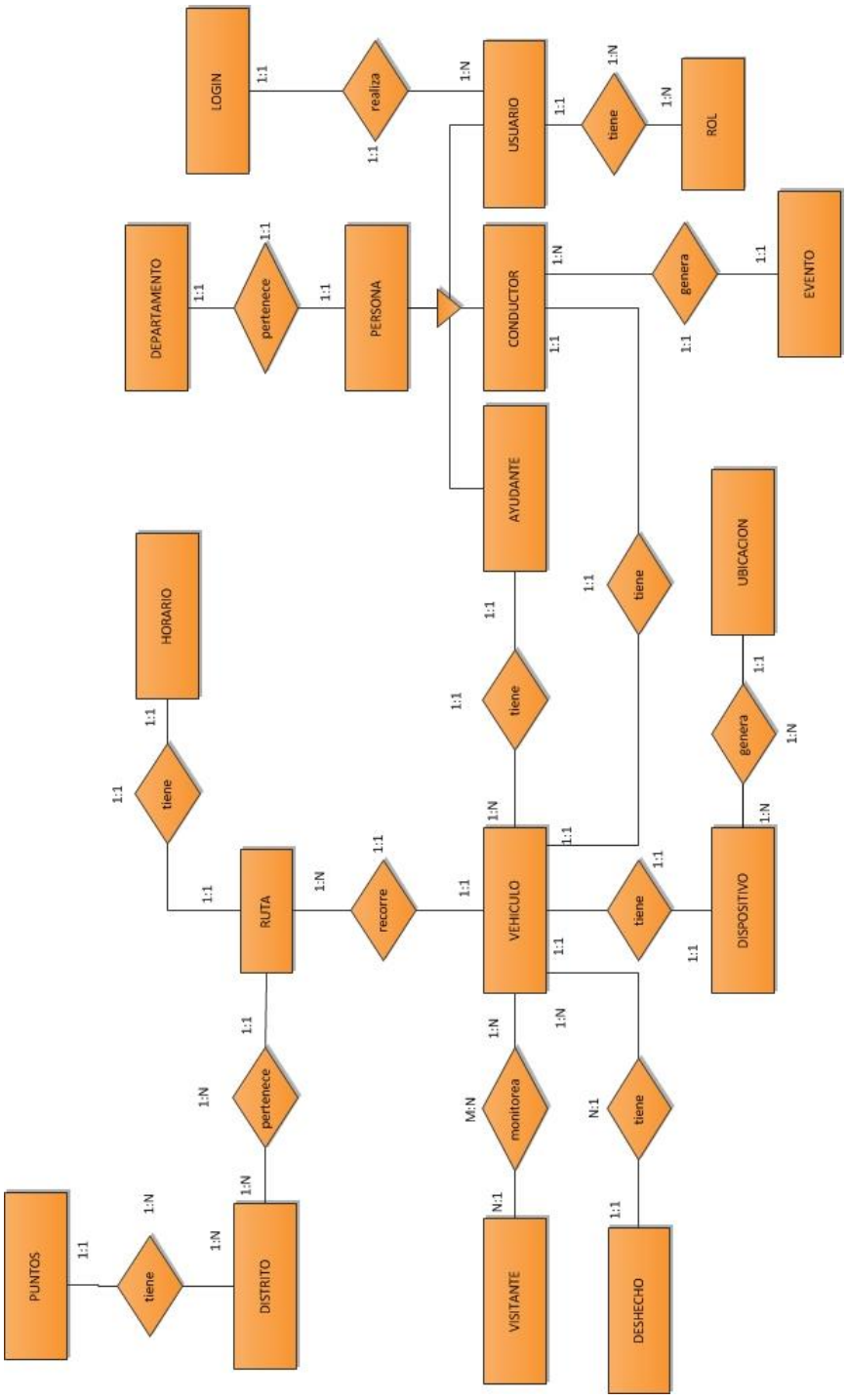
**Figura Nº 3.20:** Diagrama de clases del sistema




**Fuente:** (Elaboración propia)

**b) Modelo Entidad – Relación**

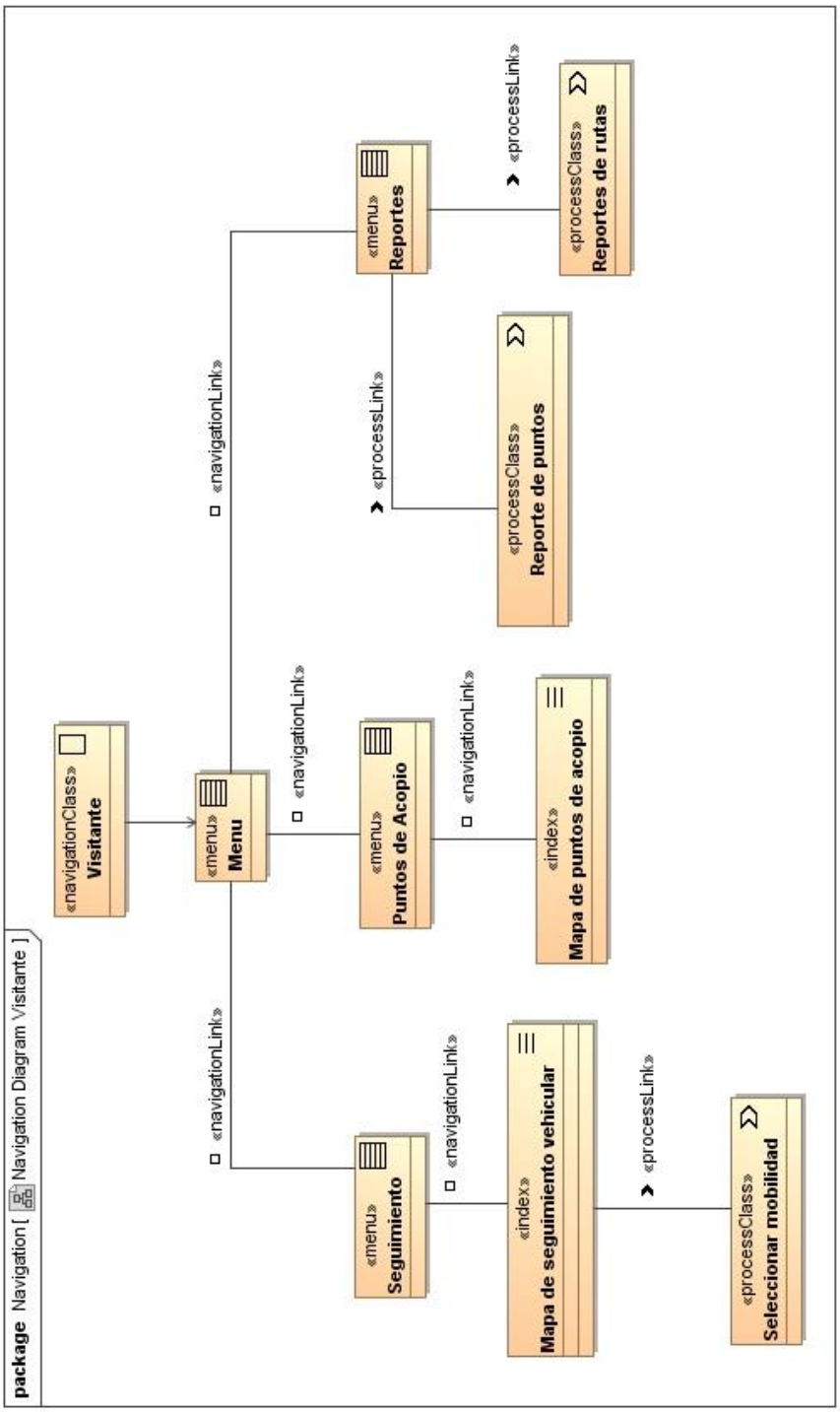
**Figura Nº 3.21:** Modelo entidad – relación




**Fuente:** (Elaboración propia)

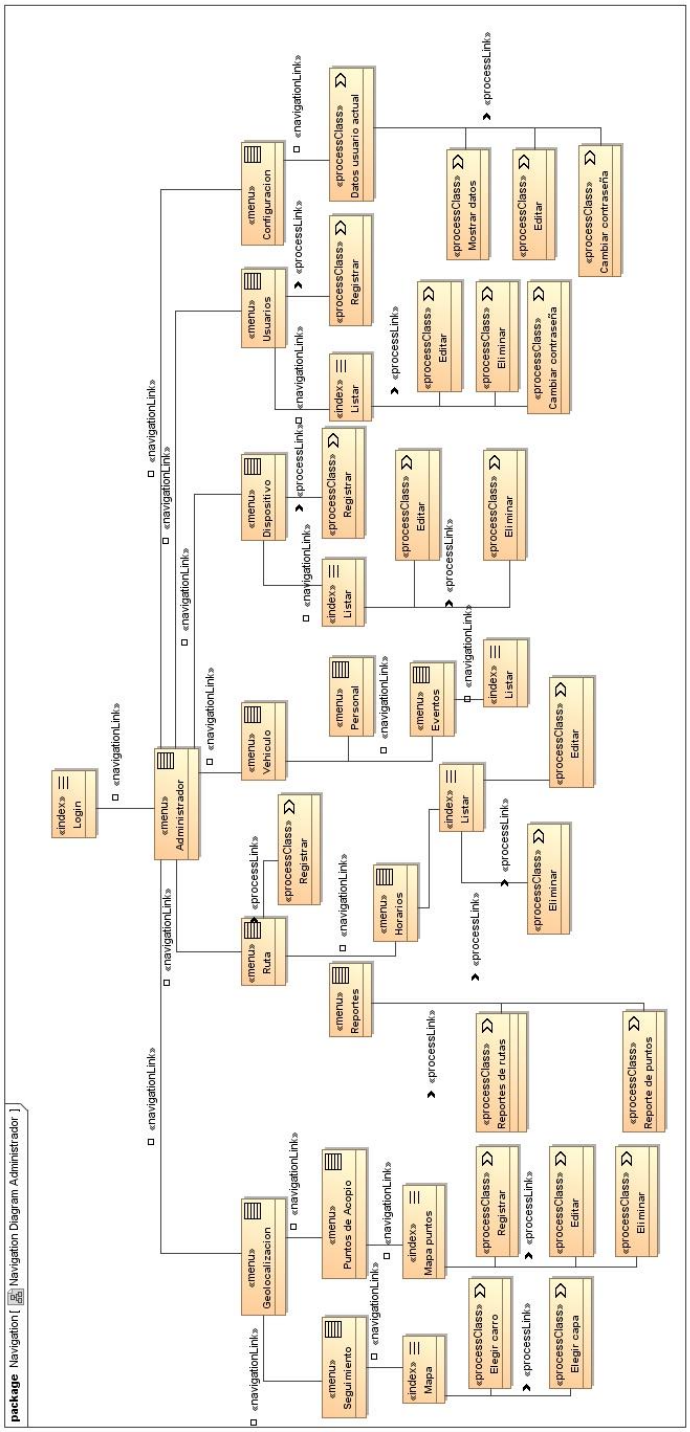
## c) Diagrama navegacional

**Figura Nº 3.22:** Diagrama navegacional del Visitante



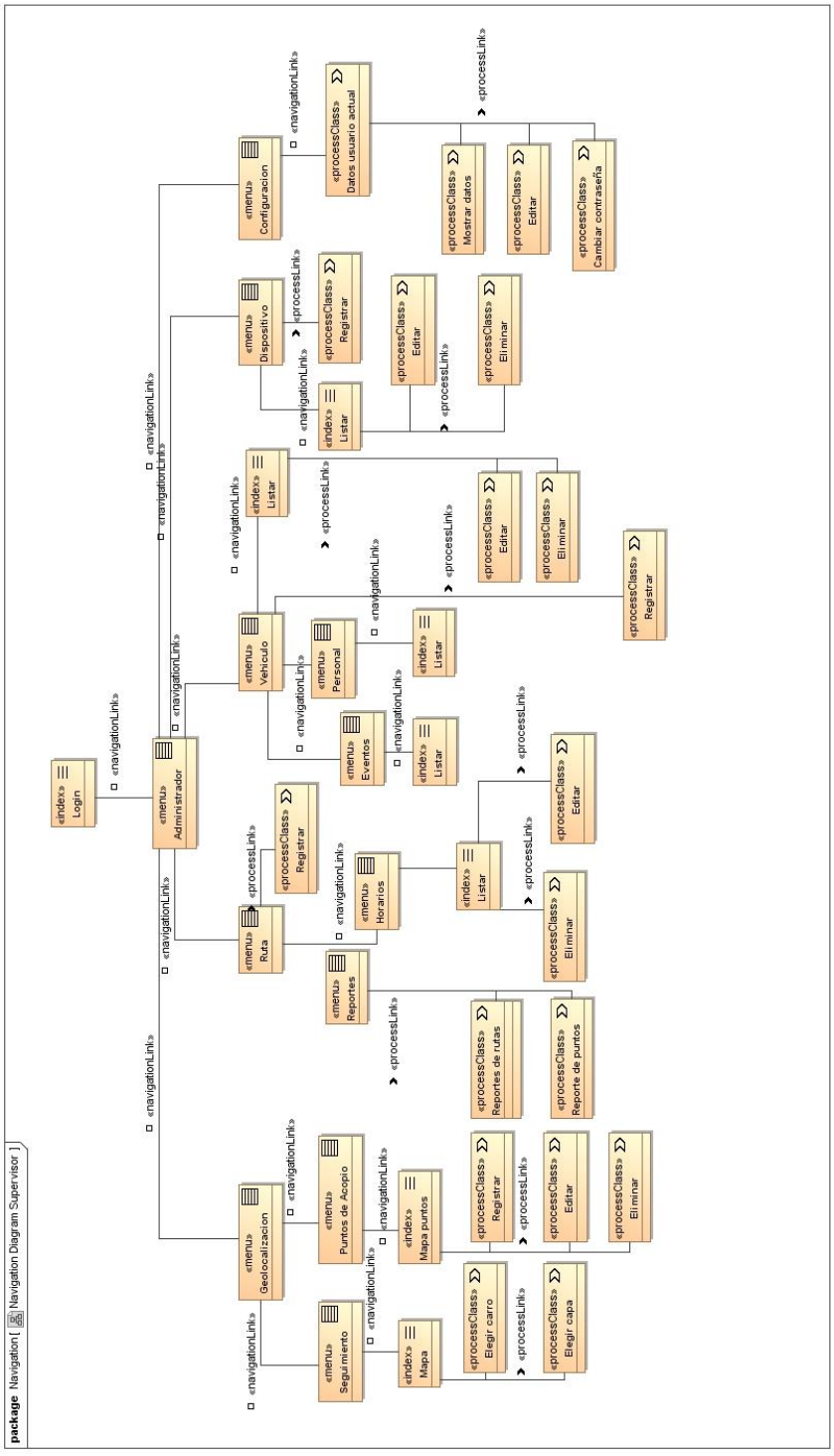

**Fuente:** (Elaboración propia)

**Figura Nº 3.23:** Diagrama navegacional del Administrador




**Fuente:** (Elaboración propia)

**Figura Nº 3.24:** Diagrama navegacional del Supervisor

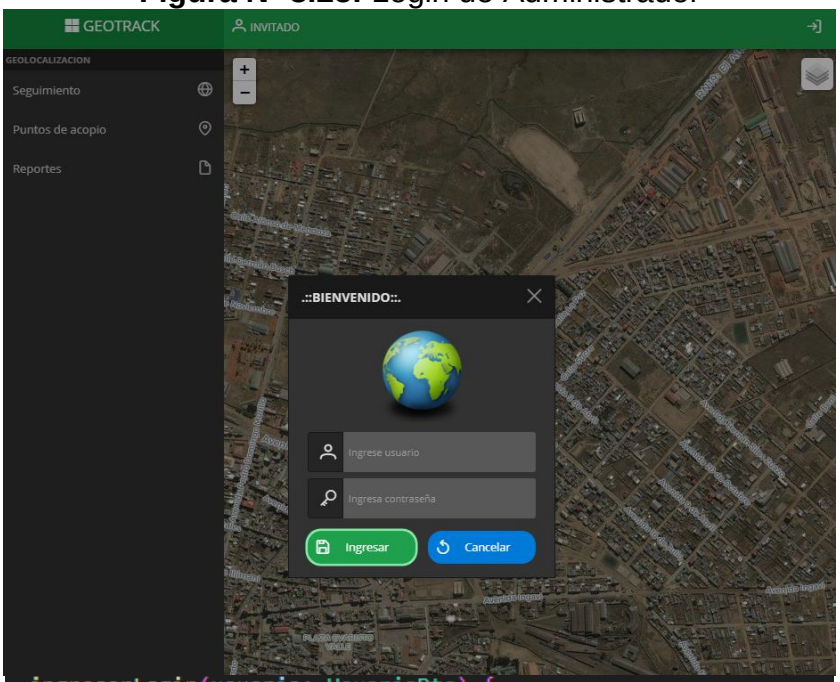



**Fuente:** (Elaboración propia)

### 3.2.3. Fase de construcción

#### 3.2.3.1. Diseño de la interfaz web

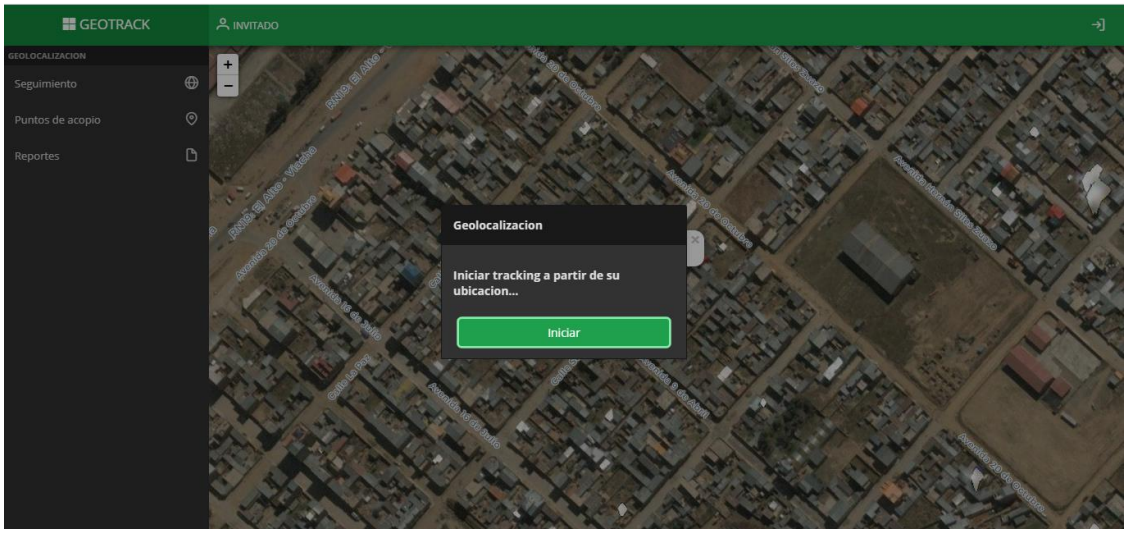
**Figura Nº 3.25:** Login de Administrador




```
ingresarLogin(usuario: UsuarioDto) {
  this.authService.login(usuario).subscribe(
    result => {
      console.log(result);
      if (result) {
        this.clean();
        this.displayDialog = false;
        this.isLoged = true;
        this.actual = this.authService.getCurrentUser();
        if (this.actual) {
          this.nombre = this.actual.usuario;
          this.helper.setRol(this.actual.rol);
        }
      } else {
        this.addSingle('info', 'Usuario no encontrado');
      }
    },
    error => {
      const errorMensaje = error;
    }
  );
}
```

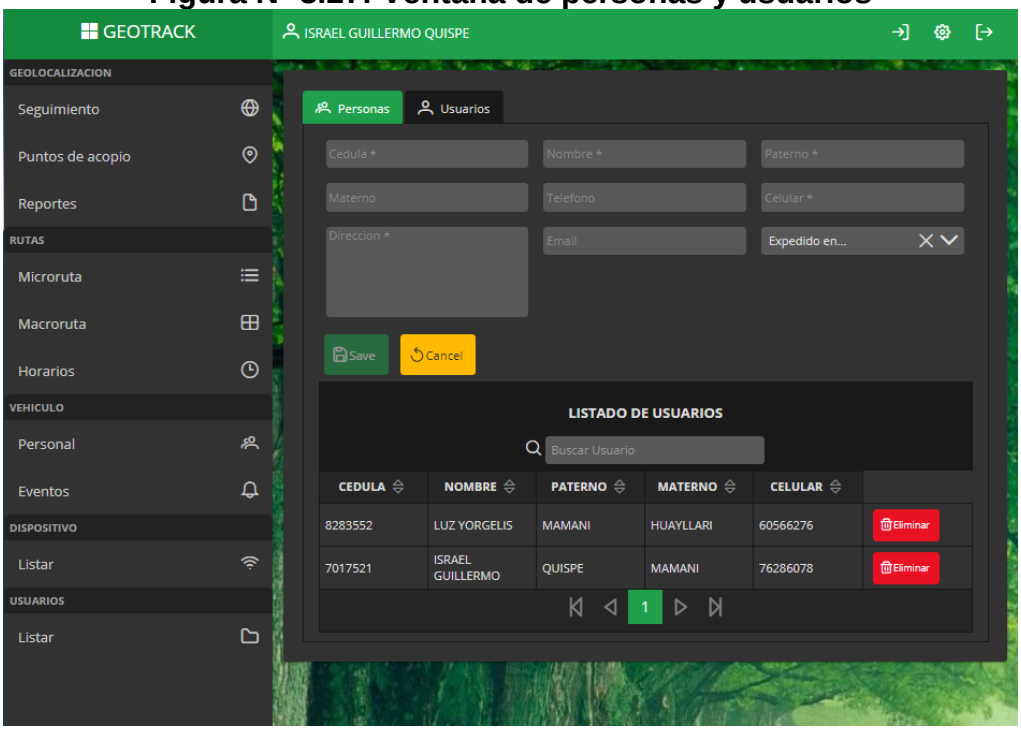
**Fuente:** (Elaboración propia)

**Figura Nº 3.26:** Inicio de sistema



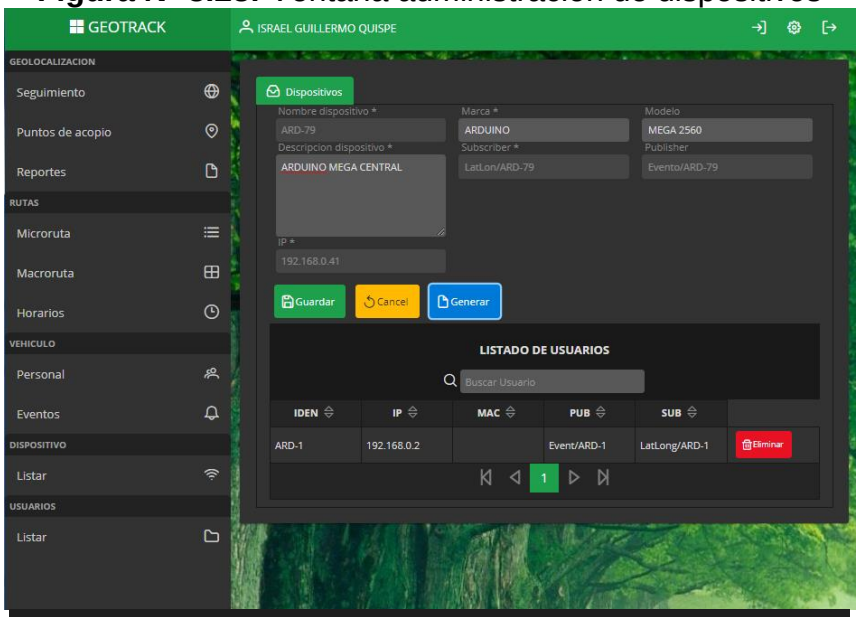

**Fuente:** (Elaboración propia)

**Figura Nº 3.27: Ventana de personas y usuarios**




**Fuente:** (Elaboración propia)

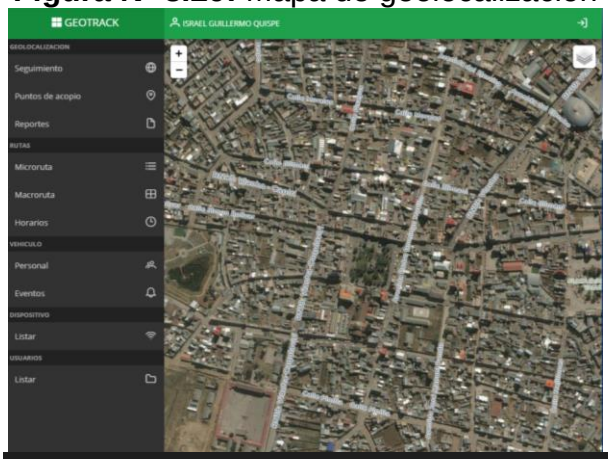
**Figura Nº 3.28:** Ventana administración de dispositivos




```
generar(value: any) {
  let ip = '';
  let nombre = '';
  let sub = '';
  let pub = '';
  this.dispositivo = new DispositivoDto();
  this.dispositivo.marca = value.marca;
  this.dispositivo.modelo = value.modelo;
  this.dispositivo.descripcion = value.descripcion;
  nombre = this.dispositivo.marca.substr(0, 3);
  nombre += '-';
  nombre += Math.floor(Math.random() * (99 - 10)) + 10;
  this.dispositivo.nombre = nombre;
  sub += 'LatLon/' + nombre;
  pub += 'Evento/' + nombre;
  this.dispositivo.sub = sub;
  this.dispositivo.pub = pub;
  ip = '192.168.0.' + Math.floor(Math.random() * 20) + 1;
  this.dispositivo.ip = ip;
  this.dispositivoForm.controls.nombreDisp.setValue(this.
  dispositivo.nombre);
  this.dispositivoForm.controls.subscriber.setValue(this.
  dispositivo.sub);
  this.dispositivoForm.controls.publisher.setValue(this.
  dispositivo.pub);
  this.dispositivoForm.controls.ip.setValue(this.dispositivo.ip);
  console.log(this.dispositivo);
}
```

**Fuente: (**Elaboración propia)

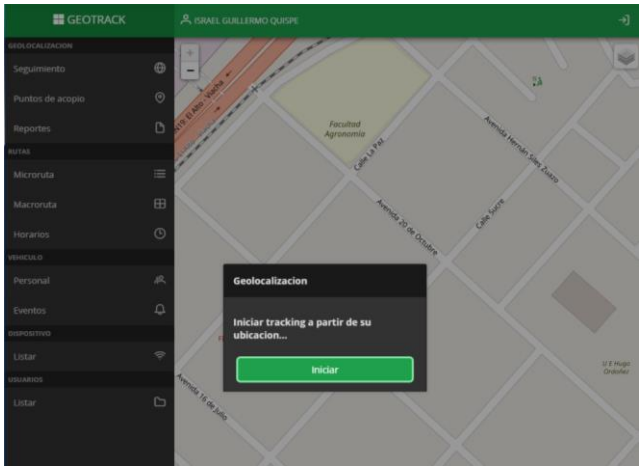
**Figura Nº 3.29:** Mapa de geolocalizacion




```
private initMap(): void {
  const options = {
    bingMapsKey: 'AnSERau4L-wtJSyhhTXBE7WmVMOft_yhaeAvftc9Vjo-ijbhH59sJmDeHa_aMS8v',
    imagerySet: 'Aerial',
    maxZoom: 18,
  };
  const bing = new bingTileLayer(options);

  const grafical = L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
    maxZoom: 18,
  });

  const carto = L.tileLayer('https://{s}.basemaps.cartocdn.com/light_only_labels/{z}/{x}/{y}{r}.png', {
    subdomains: 'abcd',
    maxZoom: 18,
  });

  this.map = L.map('mapa', {
    center: [this.latitud, this.longitud],
    zoom: 18,
    layers: [bing, grafical],
  });

  const baseMaps = {
    'Planos': bing,
    'Satelite': grafical,
  };

  const overlayMaps = {
    'Cites': carto,
  }
```

**Fuente:** (Elaboración propia)

**Figura Nº 3.30: Inicio de tracking**




**Fuente:** (Elaboración propia)

## 3.3. DISEÑO Y CONFIGURACIÓN DE PROTOTIPO

En esta etapa del proyecto se realizó la instalación en un sistema operativo Debian 9, los servicios necesarios para la comunicación entre el Arduino y nuestro servicio web. Se irán revisando esquemas de circuito del prototipo, y la correspondiente configuración del código fuente de control y también de los servicios involucrados.

### 3.3.1. Instalación del Broker Mosquito

A continuación, se describe la instalación, del bróker mosquito para el envío de datos. Cabe aclarar que toda la configuración se la realizar bajo el sistema operativo Debian en su versión 9, ya que es ampliamente extendido en diferentes tipos de proyectos tecnológicos como sistema operativo.

Iniciamos ejecutando los siguientes dos comandos para actualizar los repositorios de Debian:

```
$ sudo apt update
$ sudo apt upgrade
```

Posteriormente realizamos la instalación de Mosquito Brocker y del software necesario para los clientes que consumirán el servicio:

```
$ sudo apt-get install mosquito
$ sudo apt-get install mosquito-clients
```

De esta manera realizamos la instalación básica del brocker Mosquito, a manera de aclaración será necesario ingresar datos de usuario y contraseña al realizar la instalación de los paquetes ya que los mismos comandos modificaran el sistema de archivos existente en el sistema operativo.

### 3.3.2. Configuración del Broker Mosquito

En esta sección se pretende describir la configuración básica del Brocker Mosquitto, así también la configuración básica de la seguridad para un servicio más confiable.

Para la configuración necesaria del brocker Mosquito editaremos el archivo mosquitto.conf, la misma se encuentra ubicada en la carpeta de instalación del Brocker, en el caso de Debian 9 se encuentra en la dirección /etc/mosquitto/mosquitto.conf.

En primer lugar configuraremos los archivos de logs donde se almacenarán todos los sucesos ocurridos en el brocker Mosquitto, para tal efecto agregamos los siguientes comandos en el archivo mosquitto.conf.

```
#Guarda todos los sucesos en archivo
log_dest file /var/log/mosquitto/mosquitto.log
log_type all
log_timestamp true
```

Posteriormente reiniciamos el servicio del Brocker Mosquitto:

```
$ sudo systemctl restart mosquitto
```

El siguiente paso que nos queda es la configuración de la autenticación y autorización para el servicio de esta manera dotándole de confiabilidad y seguridad.

Configuremos Mosquitto para usar contraseñas. Mosquitto incluye una utilidad para generar un archivo de contraseña especial llamado `mosquitto_passwd`. Este comando le pedirá que ingrese una contraseña para el nombre de usuario especificado y coloque los resultados en `/etc/mosquitto/passwd`.

```
$ sudo mosquitto_passwd -c /etc/mosquitto/passwd geo
```

Al ejecutar el comando anterior en consola nos pedirá ingresar la contraseña que nosotros elijamos y de esta manera el usuario y la contraseña

Ahora abriremos un nuevo archivo de configuración para Mosquitto y le diremos que use este archivo de contraseñas para requerir inicios de sesión para todas las conexiones:

```
$ sudo nano /etc/mosquitto/conf.d/default.conf
```

Y escribir en el fichero:

```
password_file  /etc/mosquitto/passwd

allow_anonymous false
```

El comando allow_anonymous false deshabilitará todas las conexiones no autenticadas, y la línea del archivo password_file le indica a Mosquitto dónde buscar información de usuario y contraseña.

Posteriormente reiniciamos el servicio del Brocker Mosquitto:

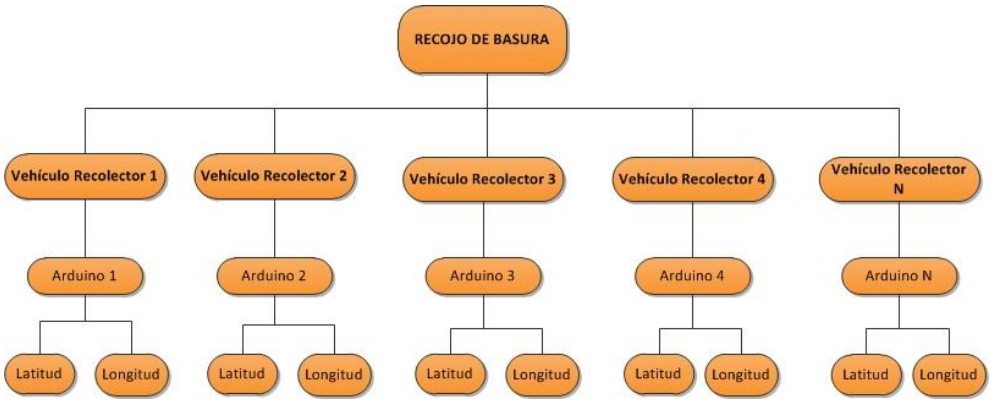
```
$ sudo systemctl restart mosquitto
```

Para publicar y suscribirse con usuario y contraseña usar:

```
$ mosquitto_pub -d -t “test” -m “hola_mundo” -u “geoUser” -P “password”

$ mosquitto_sub -d -t “test” -u “geoUser” -P “password”
```

El servicio del Brocker se estructurará de la siguiente manera:

**Figura Nº 3.31:** Estructura de servicio MQTT




**Fuente:** (Elaboración propia)

### 3.3.3. Arquitectura del prototipo

A continuación se realiza una descripción grafica de la Arquitectura del prototipo de geo localización.

**Figura Nº 3.32:** Arquitectura del prototipo


**Fuente:** (Elaboración propia)

### 3.3.4. Materiales para el desarrollo del prototipo

Para la elaboración del prototipo se hará uso de los siguientes materiales:

**Tabla Nº 3.14:** Materiales para el prototipo

| CANTIDAD | MATERIAL | DESCRIPCION |
|---|---|---|
| 1 | Arduino Mega | Sera la parte central del prototipo, procesara todo lo necesario de los datos. |
| 1 | Módulo GSM/GPRS SIM 900 | Encargado de la comunicación con brocker MQTT |
| 1 | Módulo GPS NEO 6M | Encargado de generar los datos de latitud y longitud para enviarlo al Arduino. |
| 4 | Leds | Iluminacion para la señalización del dispositivo |
| 1 | Display LCD 16x2 | Encargado de visualizar avisos, mensajes, etc en el prototipo. |

**Fuente:** (Elaboración propia)

### 3.3.5. Construcción del Prototipo

Para la construcción del prototipo la misma se dividirá en dos partes, por un lado haremos referencia al desarrollo del prototipo como tal, esto significa que se realizara el montaje y la configuración necesaria del hardware. Por otro lado,

se desarrollara el software de control, la cual gestionara los módulos y el dispositivo central como es el Arduino para la interactuación con el servidor.

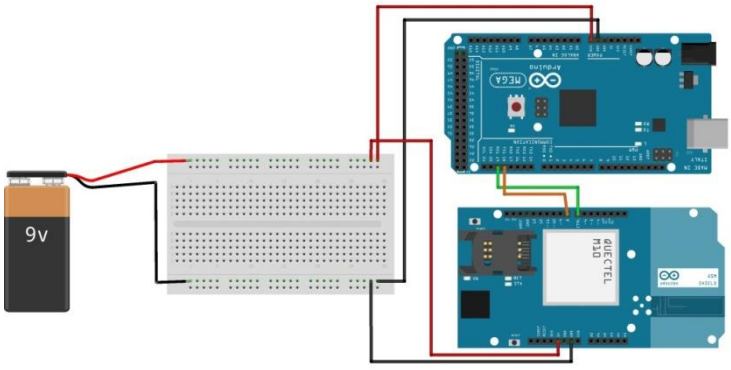
### 3.3.5.1. Desarrollo del prototipo

a) En primera instancia describiremos la conexión del Arduino y el módulo GSM/GPRS SIM 900.

   La alimentación en general que usaran ambos dispositivos es de 9V, por lo tanto, el arduino se alimentara por un pin GND para la parte negativa, y por el pin Vin para la parte positiva. Así también para la alimentación del módulo GSM/GPRS se la realizara conectando el pin GND a negativo y el pin Vin a positivo de la placa descrita.

   Para la transmisión de datos se la realizará de manera serial, por lo tanto, se usara el puerto serial 1 del arduino con el puerto serial del módulo GSM/GPRS, para tal efecto se hace la conexión del pin 7 y 8 del módulo GSM/GPRS al pin 19 y 18 del Arduino respectivamente

**Figura Nº 3.33:** Conexión Arduino y Modulo GSM/GPRS SIM900

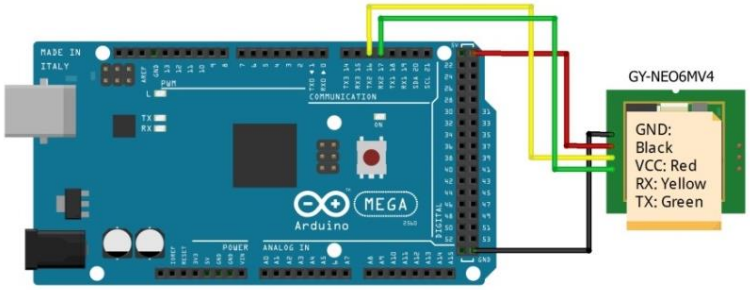



**Fuente:** (Elaboración propia)

b) En esta segunda parte se describirá la conexión del Arduino y el modulo GPS NEO 6M. Para la alimentación se realiza la conexión del ping GND Y 5V del Arduino al pin GND y Vcc respectivamente del módulo GPS. y para el manejo de los datos se realiza la conexión de Tx y Rx del módulo

GPS a los pines 16 y 17 respectivamente del Arduino los cuales hacen al puerto serial 2.

**Figura Nº 3.34:** Conexión Arduino módulo GPS NEO 6M

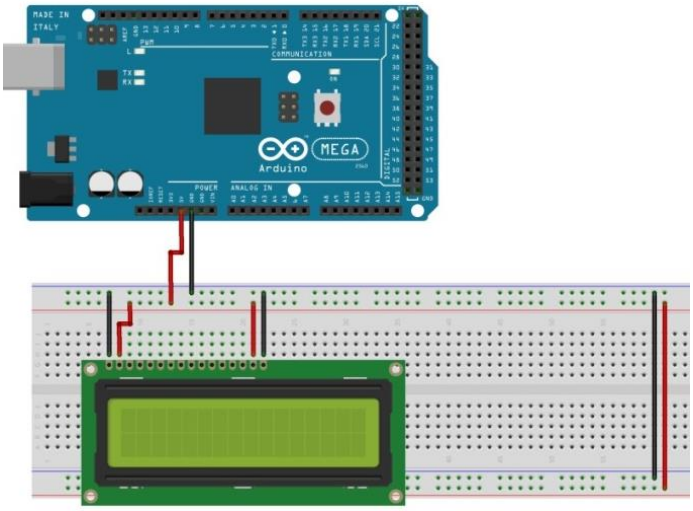



**Fuente:** (Elaboración propia)

c) En esta etapa se describe la conexión de componentes extras para la visualización de datos y señalización de las mismas junto a los pulsadores necesarios.

Para tal efecto primero hacemos la conexión de VSS y K irán a Gnd o negativo, así también realizamos la conexión de VDD y A, a Vcc o positivo, las mismas que serán la alimentación del componente Display LCD 16x2.

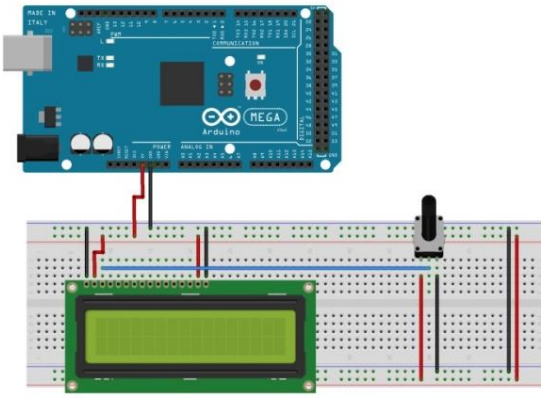
**Figura Nº 3.35:** Conexión inicial de la alimentación Display LCD 16x2




**Fuente:** (Elaboración propia)

El siguiente paso es conexión de un potenciómetro de 10K que permite el ajuste del contraste, los extremos del potenciómetro van al positivo y negativo de la alimentación, mientras que el cursor (centro) va al pin VO del módulo LCD.

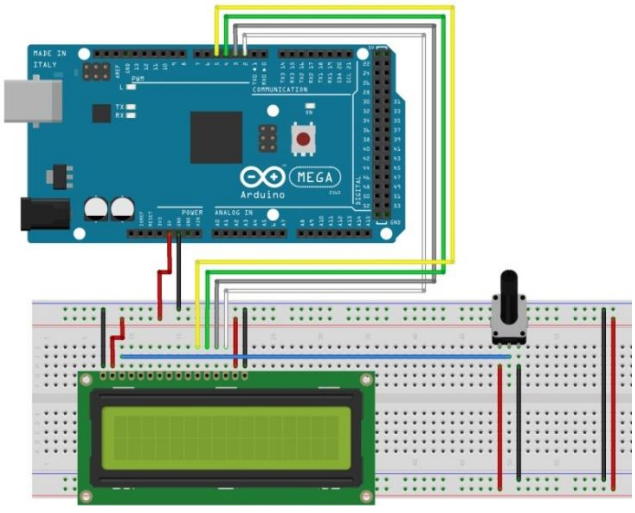
**Figura Nº 3.36:** Conexión para regulación de contraste de display



**Fuente:** (Elaboración propia)

A continuación conectamos los pines de datos D4, D5, D6, D7. Utilizaremos solamente estos 4 pines ya que operaremos la pantalla en Modo de 4 bits.

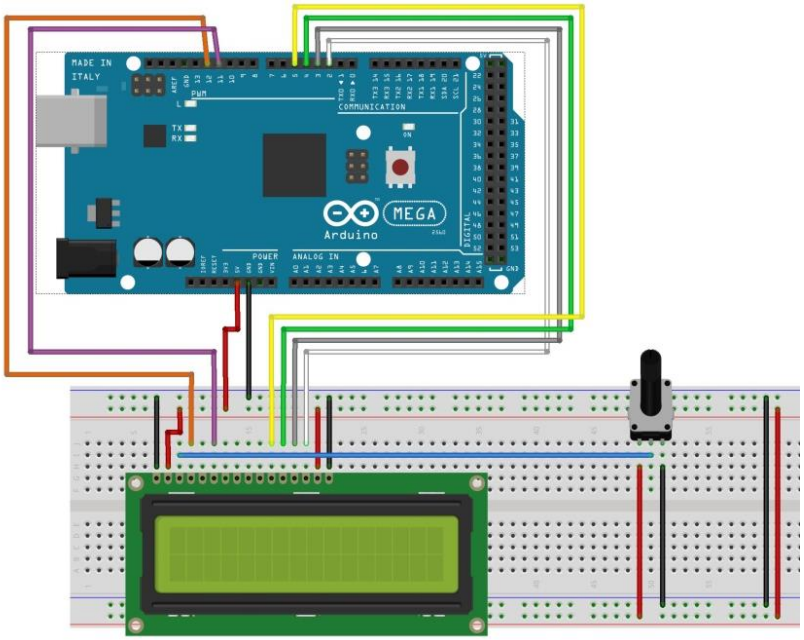
**Figura Nº 3.37:** Conexión de pines de datos para el display



**Fuente:** (Elaboración propia)

Finalmente realizamos la conexión de los pines de control RS y E. El pin RW es opcional y en este caso no lo usamos para ahorrar un pin y mantener todo más simple.

**Figura Nº 3.38:** Conexión de pines de configuración de display




**Fuente:** (Elaboración propia)

### 3.3.5.2. Desarrollo del software de control

Para el desarrollo de software de control primero generaremos un algoritmo la cual describirá los pasos para realizar todos los procesos necesarios, ya sea como el envio de datos, recepción de datos, conexión con el servidor y visualización de datos. Posteriormente tendremos el siguiente código.

```
#define TINY_GSM_MODEM_SIM900
#define SerialMon Serial
#define SerialAT Serial2
#define TINY_GSM_DEBUG SerialMon
#define GSM_PIN ""
String ARDUINO_ID = "ARD-1";
String MQTT_USER = "guillermo";
String MQTT_PASS = "Kuillerlearsi10+";
```

```
#define MQTT_BROKER "68.183.101.117"
#define MQTT_PORT 1883
const char apn[] = "internet.nuevatel.com";
const char gprsUser[] = "";
const char gprsPass[] = "";
const char* broker = "68.183.101.117";
const char* topicGps = "LatLon";
#include <TinyGsmClient.h>
#include <PubSubClient.h>
#include <TinyGPS.h>
#include <LiquidCrystal.h>
TinyGsm modem(SerialAT);
TinyGPS GPS;
TinyGsmClient client(modem,0);
PubSubClient mqtt(client);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  inicializarLcd();
  inicializarMqtt();
}
void inicializarLcd(){
  lcd.begin(16,2);
  Serial1.begin(9600);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(ARDUINO_ID);
  lcd.setCursor(0,1);
  lcd.print("GPS + LCD");
  delay(5000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" INICIANDO GPS");
  lcd.setCursor(0,1);
  lcd.print("----------------");
}
void inicializarMqtt(){
  SerialMon.begin(19200);
  delay(10);
  SerialMon.println("Espere...");
```

```
SerialAT.begin(19200);
delay(6000);
SerialMon.println("Iniciando modem...");
modem.restart();
//modem.init();
String modemInfo = modem.getModemInfo();
SerialMon.print("Modem Info: ");
SerialMon.println(modemInfo);
if ( GSM_PIN && modem.getSimStatus() != 3 ) {
  modem.simUnlock(GSM_PIN);
}
SerialMon.print("Esperando red...");
if (!modem.waitForNetwork()) {
  SerialMon.println(" Error de red");
  delay(10000);
  return;
}
SerialMon.println(" Red correcta");

if (modem.isNetworkConnected()) {
  SerialMon.println("Conexion de Red GPRS Realizada");
}

SerialMon.print(F("Conectando a "));
SerialMon.print(apn);
if (!modem.gprsConnect(apn, gprsUser, gprsPass)) {
  SerialMon.println(" Error");
  delay(10000);
  return;
}
SerialMon.println(" Exito");

if (modem.isGprsConnected()) {
  SerialMon.println("GPRS Conectado");
}

mqtt.setServer(MQTT_BROKER, MQTT_PORT);
mqtt.setCallback(mqttCallback);
reconnect();
```

```
}
void loop() {
  float latitude, longitude, velocidad;
  mqtt.loop();

  if(!mqtt.connected()){
    reconnect();
  }
  if (Serial1.available()) {
    int c = Serial1.read();
    if(GPS.encode(c)){
      GPS.f_get_position(&latitude, &longitude);
      velocidad = GPS.f_speed_kmph();
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Lat= ");
      lcd.setCursor(7,0);
      lcd.print(latitude, 6);
      lcd.setCursor(0,1);
      lcd.print("Lon= ");
      lcd.setCursor(6,1);
      lcd.print(longitude,6);

      //SerialMon.println(payloadGps(latitude,longitude));
      mqtt.publish(topicGps,payloadGps(latitude, longitude, velocidad).c_str());
      delay(3000);
    }

  }


}

void mqttCallback(char* topic, byte* payload, unsigned int len){
  SerialMon.print("Message arrived [");
  SerialMon.print(topic);
  SerialMon.print("]: ");
  SerialMon.write(payload, len);
  SerialMon.println();
```

```
  if (String(topic) == topicGps) {
    mqtt.subscribe(topicGps);
  }
}
void reconnect(){
  while(!mqtt.connected()){
    if(mqtt.connect(ARDUINO_ID.c_str(),MQTT_USER.c_str(),MQTT_PASS.c_str())){
      SerialMon.println(" Conexion al Brocker correcta");
      mqtt.subscribe(topicGps);
      mqtt.connected();
    }else{
      SerialMon.print("Failed, rc=");
      SerialMon.print(mqtt.state());
      SerialMon.println("Proximo intento en 2 segundos");
      delay(2000);
    }
  }
}
String payloadGps(float latitude, float longitude, float velocidad){
  String dataString = "";
  String la = String(latitude,6);
  String lo = String(longitude,6);

  dataString = "{\"sensorID\":\""+ARDUINO_ID;
  dataString  = dataString + "\",\"latitud\":\""  + la;
  dataString  = dataString + "\",\"longitud\":\""  + lo;
  dataString = dataString + "\",\"velocidad\":\"" + velocidad;
  dataString  = dataString + "\"}";
  return dataString;
}
```

## 3.4. MÉTRICAS DE CALIDAD DE SOFTWARE

Para medir la calidad del software es necesario identificar y evaluar aspectos particulares del sistema que contribuirá a la calidad global de la misma, para tal efecto se hará uso de los factores de calidad que nos provee la ISO 9126.

### 3.4.1. Factores de calidad ISO 9126

A continuación revisaremos los factores que definen la calidad de software en la norma ISO 9126 para la evaluación de la misma.

#### 3.4.1.1. Funcionalidad

La funcionalidad nos describirá las funciones que puede proveer el sistema al usuario a partir de ciertas características y capacidades de la misma para tal efecto debemos identificar las siguientes características:

- Número de entradas de usuario.
- Número de salidas de usuario.
- Número de petición de usuario.
- Número de archivos.
- Número de interfaces externas.

En función a las características identificadas dentro del sistema, que no es nada más que el conteo de las mismas, se aplicará a la tabla de factor de ponderación como se lo describe a continuación.

**Número de entradas de usuario,** se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

**Tabla Nº 3.15:** Número de entradas de usuario

| Entradas del usuario | | |
|---|---|---|
| **1** | Módulo de geolocalización | 2 |
| **2** | Módulo de rutas | 18 |
| **3** | Módulo de vehículos | 11 |
| **4** | Módulo de dispositivo | 7 |
| **5** | Módulo de usuarios | 20 |
| **TOTAL** | | 58 |

**Fuente:** (Elaboración propia)

**Número de salidas de usuario,** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se

refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.

**Tabla Nº 3.16:** Número de salidas de usuario

| | Salidas de usuario | |
|---|---|---|
| **1** | Módulo de geo localización | 3 |
| **2** | Módulo de rutas | 9 |
| **3** | Módulo de vehículos | 7 |
| **4** | Módulo de dispositivo | 3 |
| **5** | Módulo de usuarios | 4 |
| | **TOTAL** | 26 |

**Fuente:** (Elaboración propia)

**Número de peticiones de usuario,** una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediatamente. Se cuenta cada petición por separado.

**Tabla Nº 3.17:** Número de peticiones de usuario

| | Peticiones de usuario | |
|---|---|---|
| **1** | Módulo de geo localización | 2 |
| **2** | Módulo de rutas | 9 |
| **3** | Módulo de vehículos | 10 |
| **4** | Módulo de dispositivo | 3 |
| **5** | Módulo de usuarios | 8 |
| | **TOTAL** | 32 |

**Fuente:** (Elaboración propia)

**Número de archivos,** se cuenta archivo maestro lógico.

**Tabla Nº 3.18:** Número de archivos

| | Número de archivos | |
|---|---|---|
| **1** | Todos los archivos | 19 |
| | **TOTAL** | 19 |

**Fuente:** (Elaboración propia)

**Número de interfaces externas,** se cuenta todas las interfaces legibles por la máquina

**Tabla Nº 3.19:** Número de interfaces externas

| Interfaces externas | | | |
|---|---|---|---|
| **1** | Internet | | 3 |
| | | **TOTAL** | 3 |

**Fuente:** (Elaboración propia)

Para calcular los puntos de función se tiene:

**Tabla Nº 3.20:** Cálculo de punto de función no ajustado

| Parámetros de Medición | Cuentas | Factor de Ponderación | | | Total |
|---|---|---|---|---|---|
| | | Simple | Medio | Complejo | |
| **Nº De entradas de Usuario** | 58 | **3** | 4 | 6 | 174 |
| **Nº de Salidas de Usuario** | 26 | **4** | 5 | 7 | 104 |
| **Nº de Peticiones de Usuario** | 32 | **3** | 4 | 6 | 96 |
| **Nº de Archivos de Operación** | 19 | **7** | 10 | 15 | 133 |
| **Nº de Interfaces Externos** | 3 | **5** | 7 | 10 | 15 |
| | | | | **TOTAL** | 522 |

**Fuente:** (Elaboración propia)

Para calcular el punto de función se utiliza la siguiente ecuación:

$$\boldsymbol{PF = Cuenta\ Total} * (\mathbf{0,65} + \mathbf{0.1} * \sum \boldsymbol{F_i})$$

Dónde:

**Cuenta Total:** es la suma de todas las entradas obtenidas en Nº de Entradas, Nº de Salidas, Nº de Peticiones, Nº de Archivos, Nº de interfaces externas.

**0,65:** nivel de confiabilidad del sistema

**0.01:** nivel de error de confiabilidad del sistema

$\sum F_i$**:** son los valores de ajuste de complejidad según las respuestas a las preguntas destacadas en la siguiente tabla.

**Tabla Nº 3.21:** Valores de ajuste de complejidad

| Nº | Factor de complejidad | Sin influencia | Incidental | Moderada | Medio | Significativo | Esencial | $F_i$ |
|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | |
| 1 | ¿Requiere el sistema copias de seguridad y de recuperación fiables? | | | | | | X | 5 |
| 2 | ¿Se requiere comunicación de datos? | | | | | X | | 4 |
| 3 | ¿Existen funciones de procesamiento distribuido? | | | | X | | | 3 |
| 4 | ¿Es crítico el rendimiento? | | | | | X | | 4 |
| 5 | ¿Se ejecuta el sistema en un entorno operativo, existente y fuertemente utilizado? | | | | | | X | 5 |
| 6 | ¿Requiere el sistema entrada de datos interactiva? | | | | | | X | 5 |
| 7 | Facilidad Operativa | | | | | | X | 5 |
| 8 | ¿Se actualizan los archivos maestros de forma interactiva? | | | | | | X | 5 |
| 9 | ¿Son complejos las entradas, las salidas, los archivos o las peticiones? | | | | X | | | 3 |
| 10 | ¿Es complejo el procesamiento interno? | | | | | X | | 4 |
| 11 | ¿Se ha diseñado el código para ser reutilizable? | | | | | X | | 4 |
| 12 | Facilidad de instalación | | | | X | | | 3 |
| 13 | ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones? | | | | | | X | 5 |
| 14 | Facilidad de cambio | | | | | | X | 5 |
| | **FACTOR DE COMPLEJIDAD TOTAL (FCT)** | | | | | | | **60** |

**Fuente:** (Elaboración propia)

Una vez obtenido el valor de ajuste y reemplazando a la ecuación tiene:

$$\boldsymbol{PF = Cuenta\ Total} * (\mathbf{0,65 + 0,01} * \sum \boldsymbol{F_i})$$

$$\boldsymbol{PF = 522} * (\mathbf{0,65 + 0,01} * \mathbf{60})$$

$$\boldsymbol{PF = 652,5}$$

Ajustando a la curva normal:

$$Funcionalidad = \frac{PF}{PF_{Máxima}}$$

Dónde:

$PF_{Máxima} = Cuenta\ Total * (0{,}65 + 0{,}01 * \sum F_i)$, se toma el valor máximo de $\sum F_i$ ; $[i = 1\ a\ 14], [0 \leq \sum F_i \leq 70]$.

Reemplazando se tiene:

$$PF_{Máxima} = Cuenta\ Total * (0{,}65 + 0{,}01 * \sum F_I)$$

$$PF_{Máxima} = 522 * (0{,}65 + 0{,}01 * 70)$$

$$\boldsymbol{PF_{Máxima} = 704,7}$$

Por tanto:

$$Funcionalidad = \left(\frac{PF}{PF_{Máxima}}\right) * 100$$

$$Funcionalidad = \frac{652{,}5}{704{,}7} * 100$$

$$\boldsymbol{Funcionalidad = \ 92,59\%}$$

Entonces la funcionalidad del sistema es un 92,59% esto quiere decir que el sistema tiene un 92,59% que funcione sin riesgos de fallo y operatividad constante y un 7,41% de colapso del sistema.

### 3.4.1.2. Confiabilidad

Para determinar la confiabilidad se toma en cuenta las fallas que se producen en el sistema en un tiempo determinado, también es el grado en que el sistema responde bajo las condiciones definidas durante un intervalo de tiempo dado.

$$F(t) = f * e^{(-\frac{\lambda}{10} * t)}$$

Dónde:

f: es la funcionalidad del sistema ya calculada: 92,59%

λ: es la probabilidad de error que puede tener el sistema 0,03 (3%)

t: tiempo que dura una gestión en el sistema: 12 meses

A continuación se realiza el cálculo de la probabilidad de que el sistema tenga fallas:

$$F(t) = 92{,}59 * e^{(-\frac{3}{10}*12)}$$

$$F(t) = 92{,}59 * 0{,}0273$$

$$\boldsymbol{F(t) = 2{,}53\%}$$

La probabilidad de que ocurra fallas en el sistema es de 2,4%, entonces la probabilidad de que el sistema esté libre de fallos es de:

$$P(T \geq t) = 1 - F(t)$$

$$P(T \geq t) = (1 - 0{,}0253) * 100\%$$

$$\boldsymbol{P(T \geq t) = 97{,}47\%}$$

Por lo tanto se puede decir que el sistema tiene un grado de confiabilidad de $F(12) = 97{,}47\%$ y seguirá funcionando en un año (12 meses).

#### 3.4.1.3. Usabilidad

En la siguiente tabla se observa estos criterios en niveles de porcentajes a lo que llego el sistema en cuanto a su comprensibilidad, para el usuario, y posteriormente se da el porcentaje final de usabilidad del sistema. Se realizó una encuesta al usuario final sobre el manejo, la comprensión y la facilidad de aprender el sistema para medir la usabilidad según la siguiente tabla.

Cabe aclarar que la encuesta se la realizo a 10 usuarios para tomarlo como base da calculo.

**Tabla Nº 3.22:** Encuesta de la usabilidad del sistema

| Nº | PREGUNTA | RESPUESTAS | | % |
|---|---|---|---|---|
| | | SI(1-10) | NO(1-10) | |
| **1** | ¿El acceso al sistema es complicado? | 0 | 10 | 100 |
| **2** | ¿Las respuestas del sistema son de su comprensibilidad? | 9 | 1 | 90 |
| **3** | ¿Tiene alguna dificultad en realizar los procesos del sistema? | 2 | 8 | 80 |
| **4** | ¿La interfaz de la aplicación es amigable y entendible a su parecer? | 1 | 9 | 90 |
| **5** | ¿Son comprensibles y satisfactorios los datos que se muestran en el sistema? | 1 | 9 | 90 |
| **6** | ¿El sistema es de fácil uso bajo su criterio? | 2 | 8 | 80 |
| **7** | ¿El sistema coadyuva en el mejoramiento del servicio de recojo de basura? | 2 | 8 | 80 |
| **8** | ¿Durante el uso del sistema se produjo errores? | 1 | 9 | 90 |
| | TOTAL | | | **87,5%** |

**Fuente:** (Elaboración propia)

De acuerdo a los datos obtenidos en la tabla de usabilidad, se concluye que el sistema tiene una usabilidad del **87,5%.** Esto indica que le usuario de la aplicación tiene una conformidad hacia el sistema y que este podrá coadyuvar de alguna manera en solucionar y mejorar los problemas en el servicio de recojo de basura.

### 3.4.1.4. Mantenimiento

Es el conjunto de atributos relacionados con la facilidad de entender, modificar o corregir en un sistema software, en este sentido se usara el índice de madurez.

$$IMS = \frac{[M_T - (F_a + F_c + F_d)]}{M_T}$$

**Tabla Nº 3.23:** Facilidad de mantenimiento

| TIPO | DESCRIPCION | VALOR |
|---|---|---|
| $M_T$ | Número de módulos de la versión actual | 5 |
| $F_C$ | Número de módulos en la versión actual que se han cambiado | 0 |
| $F_a$ | Número de módulos en la versión actual que se han añadido | 1 |
| $F_d$ | Número de módulos de la versión anterior que se ha borrado en la versión actual | 0 |

**Fuente:** (Elaboración propia)

Reemplazando los datos en la ecuación obtendremos:

$$IMS = \frac{[5 - (1 + 0 + 0)]}{5}$$

$$IMS = 0{,}8 * 100\%$$

$$\boldsymbol{IMS = 80\%}$$

Con lo que podemos concluir que el nuevo sistema tiene una estabilidad de 80% la cual es la facilidad de mantenimiento y el 20% restante es el margen de error correspondiente a los cambios y modificaciones efectuados desde el prototipo de la versión actual

Puesto que es un sistema diseñado con los requerimientos actuales con el tiempo surgirán nuevos requerimientos los cuales cambiara el valor del índice de madurez del software.

El mantenimiento del software está dado por cuatro clases:

- **Correctivo:** tiene por objetivo localizar y eliminar los posibles defectos de los programas. Un efecto es una característica del sistema con el potencial de causar falla.
- **Adaptativo:** consiste en la modificación de un programa debido a cambios en el entorno (hardware y software) en el cual se ejecuta. Estos cambios pueden afectar al sistema operativo, a la arquitectura física del sistema o al entorno de desarrollo del software.
- **Perfectivo:** es el conjunto de actitudes para mejorar o añadir nuevas funcionalidades requeridas por el usuario.
- **Preventivo:** consiste en la modificación del software para mejorar sus propiedades (calidad y/o matenibilidad) sin alterar sus especificaciones funcionales.

### 3.4.1.5. Portabilidad

El sistema por estar diseñado en un entorno de acceso via web, mide la portabilidad en: lado del servidor, el lado del cliente. La portabilidad del software se enfoca en tres aspectos:

- Hardware del Servidor
- Sistema Operativo del Servidor
- Software del Servidor

Por lo mencionado anteriormente del sistema, es portable en sus diferentes entornos tanto en hardware y software por lo que puede considerar una portabilidad de 100%.

### 3.4.1.6. Calidad global

En función a la evaluación realizada de las características principales en función a la norma ISO 9126 se obtuvieron los siguientes datos.

**Tabla Nº 3.24:** Evaluación de la calidad global

| Nº | CARACTERISTICA | % |
|---|---|---|
| **1** | Funcionalidad | 92,59% |
| **2** | Confiabilidad | 97,47% |
| **3** | Usabilidad | 87,5% |
| **4** | Mantenimiento | 80% |
| **5** | Portabilidad | 100% |
| | TOTAL | **91,51%** |

**Fuente:** (Elaboración propia)

En resumen, se puede observar en función a las características evaluadas que el sistema tiene una calidad global del **91,51%,** obteniendo una muy buena puntuación y garantizando el funcionamiento eficaz y eficiente coadyuvando de la mejor manera el mejoramiento del servicio de recojo de basura.

## 3.5. ESTIMACIÓN DE COSTOS

En este apartado la tarea principal es el de medir los costos y los recursos humanos necesarios para el desarrollo completo del proyecto, así también se trata de calcular con certeza los tiempos necesarios para la realización de la misma. En este caso utilizaremos el Modelo Constructivo de Costos para el cálculo de tales características dentro del desarrollo del proyecto.

### 3.5.1.1. Costo de hardware

Para el cálculo del costo del prototipo hardware se enumera los materiales utilizados tomando en cuenta la cantidad y los costos por unidad en el actual mercado. Así también se toman en cuenta los servicios de alojamiento del sistema web ya que la institución como tal no cuenta con uno disponible.

En la siguiente tabla se puede observar todos los costos por conceptos de construcción del prototipo y el servicio de alojamiento del sistema que es muy importante para el funcionamiento total.

**Tabla Nº 3.25:** Costos de prototipo y servicio broker

| Nº | DESCRIPCION | C/U * | CANTIDAD | TOTAL * |
|---|---|---|---|---|
| | **COSTO MATERIALES PARA CONSTRUCCION DE PROTOTIPO** | | | |
| **1** | Arduino Mega | 120 | 1 | 120 |
| **2** | Módulo GSM/GPRS SIM 900 | 230 | 1 | 230 |
| **3** | Módulo GPS NEO 6M | 70 | 1 | 70 |
| **4** | Display LCD 16x2 | 30 | 1 | 30 |
| | | | TOTAL MATERIAL | **450** |
| | **COSTOS DE ALOJAMIENTO WEB PARA BROCKER MOSQUITTO** | | | |
| **1** | Servicio de alojamiento en la nube DIGITAL OCEAN | 140/mes | 6 MESES | 840 |
| | | TOTAL ALOJAMIENTO SERVICIO | | **840** |

* **costo en bolivianos**

**Fuente:** (Elaboración propia)

Por lo tanto el costo del prototipo como tal es de 450Bs equivalente en dólares al tipo de cambio de 6.96 es de **65$** más el costo del servicio de alojamiento

web para el bróker que es de 840Bs por 6 meses equivalente en dólares a **120$.**

### 3.5.1.2. COCOMO II (Modelo Constructivo de Costos)

En primer lugar calcularemos el esfuerzo de desarrollo, para lo cual necesitamos hallar la variable KLDC (Kilo-líneas de código). La misma nos dará un valor acerca del tamaño del software implementado.

**Estimar el tamaño del software:**

$$\boldsymbol{PFA = PF * [0,65 + 0,01 * \sum F_i]}$$

Para obtener el punto de función Sin Ajustar y el Ajuste de Complejidad hacemos referencia a la tabla 3.20 y 3.21 donde los valores son:

**Cuenta Total = 522 y $\boldsymbol{\sum F_i = 60}$**

Por lo tanto el multiplicador será:

$$Multiplicador = 0{,}65 + 0{,}01 * \sum F_i$$

$$Multiplicador = 0{,}65 + 0{,}01 * 60$$

$$\boldsymbol{Multiplicador = 1,25}$$

Reemplazando en la ecuación:

$$PFA = Cuenta\ Total * Multiplicador$$

$$PFA = 552 * 1{,}25$$

$$\boldsymbol{PFA = 690}$$

**Conversión de los puntos de fusión a KDLC**

Ahora convertimos los PFA a miles de líneas de código. Para ello veremos la siguiente tabla:

**Tabla Nº 3.26:** Conversión de puntos de función a KLDC

| LENGUAJE | FACTOR LDC/PF |
|---|---|
| Java | 53 |
| **JavaScript** | **47** |
| Visual Basic | 46 |
| ASP | 36 |
| Visual C++ | 34 |
| PHP | 12 |
| Ensamblador | 320 |
| C | 150 |

**Fuente:** (Elaboración propia)

Calculamos KLDC:

$$KLDC = \frac{FactorLCD * PFA}{1000}$$

$$KLDC = \frac{47 * 690}{1000}$$

$$\boldsymbol{KLDC = 32{,}43}$$

Dónde:

**KLDC:** número estimado de líneas de código distribuidas (en miles).

**Determinar factor exponencial de escala (B)**

$$\boldsymbol{B = 0{,}91 + 0{,}01 * \sum_{i=1}^{5} SF_i}$$

Dónde:

0,91: Valor confiable

0,01: error

$\sum_{i=1}^{5} SF_i$ : sumatoria de factores de escala.

**Tabla Nº 3.27:** Factores de escala

| i | Factor de escala SF | Muy bajo | Bajo | Normal | Alto | Muy alto | Extra |
|---|---|---|---|---|---|---|---|
| 1 | Precedencia **PEC** | Totalmente diferente | Muy diferente | Aspectos novedosos | Bastante parecido | Muy parecido | Idéntico a precios |
| | | 6,2 | **4,96** | 3,72 | 2,48 | 1,24 | 0 |
| 2 | Flexibilidad en el desarrollo **FLEX** | Rigurosa | Flexibilidad ocasional | Cierta flexibilidad | Acuerdo general | Cierto acuerdo | Metas generales |
| | | 5,06 | 4,05 | **3,04** | 2,03 | 1,01 | 0 |
| 3 | Arquitectura / Resolución de riesgo **RESL** | Poca -20% | Alguna -40% | Siempre -60% | Generalmente 75% | Principalmente 90% | Completo 100% |
| | | 7,07 | 5,65 | **4,24** | 2,83 | 1,41 | 0 |
| 4 | Interacciones difíciles **TEAM** | Interacciones difíciles | Interacciones con alguna dificultad | Interacciones básicamente cooperativas | Ampliamente cooperativas | Altamente cooperativas | Interacciones sin fisuras |
| | | 5,48 | 4,38 | 3,29 | 2,19 | 1,1 | **0** |
| 5 | Madurez del proceso **PMAT** | SW/CMM Nivel 1 Inferior | SW/CMM Nivel 1 Superior | SW/CMM Nivel 2 | SW/CMM Nivel 3 | SW/CMM Nivel 4 | SW/CMM Nivel 5 |
| | | 7,8 | 6,24 | **4,68** | 3,12 | 1,56 | 0 |

**Fuente:** (Boehm, 2000)

Reemplazando en la ecuación se tiene:

$$\sum_{i=1}^{5} SF_i = PREC + FLEX + RESL + TEAM + PMAT$$

$$\sum_{i=1}^{5} SF_i = 4{,}96 + 3{,}04 + 4{,}24 + 0{,}0 + 4{,}68 = 16{,}92$$

Reemplazando en la ecuación:

$$\boldsymbol{B} = 0{,}91 + 0{,}01 * 16{,}92 = \mathbf{1{,}0792}$$

**Determinar el esfuerzo nominal (E)**

Para calcular los factores de escala se aplicará la siguiente ecuación:

$$E = A * (Tamaño)^B * \prod_{i=1}^{17} EM_i$$

Dónde:

**E:** esfuerzo nominal de medid en (hombres/mes)

**A:** es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño (A=2,94)

**Tamaño:** es el tamaño del software a desarrollar expresado en miles de líneas de código fuente (KLDC=32,43)

**B:** es el factor exponencial de escala, ya calculada anteriormente (**1,0792**)

$\boldsymbol{EM_i}$**:** Factores de Esfuerzo Compuesto, corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo.

**Determinar factor de esfuerzo compuesto** $(\boldsymbol{EM_i})$

$$\prod_{i=1}^{17} EM_i = RELY * DATA * CPLX * RUSE * DOCU * TIME * STOR * PVOL * ACAP$$
$$* PCAP * AEXP * PEXP * LTEX * PCON * TOOL * SITE * SCED$$

**Tabla Nº 3.28:** Multiplicadores de esfuerzo

| FACTOR | Abrev. | MUY BAJO | BAJO | MEDIO | ALTO | MUY ALTO | MEXTR. ALTO |
|---|---|---|---|---|---|---|---|
| Producto | **RELY** | 0,75 | 0,88 | 1,00 | 1,15 | 1,39 | - |
| | **DATA** | - | 0,93 | 1,00 | 1,09 | 1,19 | - |
| | **CPLX** | 0,73 | 0,88 | 1,00 | 1,15 | 1,30 | 1,66 |
| | **RUSE** | - | 0,91 | 1,00 | 1,14 | 1,29 | 1,49 |
| | **DOCU** | - | 0,95 | 1,00 | 1,06 | 1,13 | - |
| Plataforma | **TIME** | - | - | 1,00 | 1,11 | 1,31 | 1,67 |
| | **STOR** | - | - | 1,00 | 1,06 | 1,21 | 1,57 |
| | **PVOL** | - | 0,87 | 1,00 | 1,15 | 1,30 | - |
| Personal | **ACAP** | 1,50 | 1,22 | 1,0 | 0,83 | 0,67 | - |
| | **PCAP** | 1,37 | 1,16 | 1,0 | 0,87 | 0,74 | - |
| | **AEXP** | 1,22 | 1,10 | 1,0 | 0,89 | 0,81 | - |
| | **PEXP** | 1,25 | 1,12 | 1,0 | 0,88 | 0,81 | - |
| | **LTEX** | 1,22 | 1,10 | 1,0 | 0,91 | 0,84 | - |
| | **PCON** | 1,24 | 1,10 | 1,0 | 0,92 | 0,84 | - |
| Proyecto | **TOOL** | - | 1,12 | 1,00 | 0,86 | - | - |
| | **SITE** | 1,25 | 1,10 | 1,00 | 0,92 | - | 0,78 |
| | **SCED** | 1,29 | 1,10 | 1,00 | 1,00 | - | - |

**Fuente:** (Boehm, 2000)

$$\prod_{i=1}^{17} \boldsymbol{EM_i} = 1{,}00 * 1{,}09 * 1{,}00 * 1{,}00 * 1{,}00 * 1{,}00 * 1{,}00 * 0{,}87 * 1{,}00 * 1{,}00 * 1{,}00 * 1{,}00 * 1{,}00 * 1{,}00 * 1{,}00 * 1{,}10 * 1{,}00 = \mathbf{1{,}04313}$$

Reemplazando en la ecuación:

$$E = A * (Tamaño)^B * \prod_{i=1}^{17} EM_i$$

$$E = 2{,}94 * (32{,}43)^{1{,}0792} * 1{,}04313$$

$$\boldsymbol{E = 131{,}0086}$$

**Determinar el tiempo de desarrollo (TDES)**

$$TDES = 3{,}67 * (E)^{[0{,}28+0{,}2*(B-1{,}01)]}$$

$$TDES = 3{,}67 * (131{,}0086)^{[0{,}28+0{,}2*(1{,}0792-1{,}01)]}$$

$$TDES = 3{,}67 * 131{,}0086^{0.2938}$$

$$TDES = 3{,}67 * 4{,}1885$$

$$\boldsymbol{TDES = 15{,}37\ (meses) = 15\ meses}$$

**Determinar la cantidad de personas (CH)**

$$CH = \frac{E}{TDES}$$

$$CH = \frac{131{,}0086}{15{,}37}$$

$$CH = 8{,}52\ personas = \mathbf{9\ personas}$$

**Costo del software**

El salario promedio de un programador es de 2500 Bs. O su equivalente en dólares con la tasa de cambio de 6,96 es de 359,1954 $us.

$$(costo\ total) = salario\ programador * Tiempo$$

$$(costo\ mes) = P * salario\ programador$$

Por lo tanto:

$$\boldsymbol{Costo\ mensual = 359,1954 * 15 = 5.387,93\ \$us.}$$

$$\boldsymbol{Costo\ total = 5.387,93 * 9 = 48.491,37\ \$us}$$

En resumen, se requiere 9 personas estimando un trabajo de 15 meses y con un costo total de 48.491,37 $us.

# CAPITULO IV

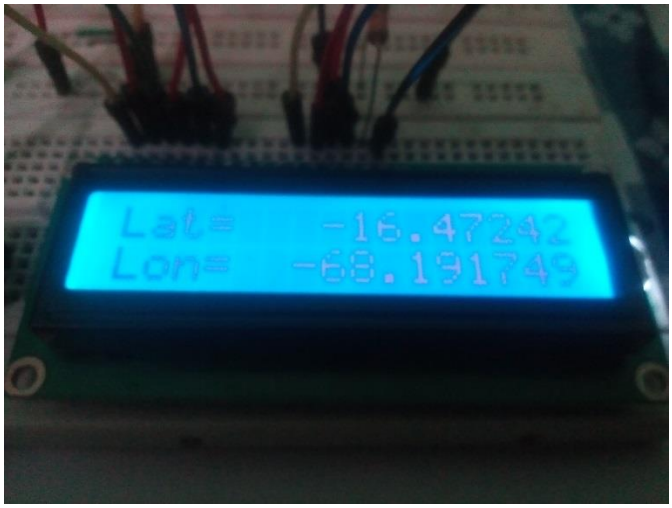
# PRUEBAS Y RESULTADOS

## 4.1. DESARROLLO DE PRUEBAS Y RESULTADOS

A continuación veremos las pruebas y resultados del sistema implementado mostrando la interactuación del prototipo, con el sistema y el brocker que es la parte central para la comunicación de datos.

Primeramente iniciamos el dispositivo de geo localización:

**Figura Nº 4.1:** Pantalla de información dispositivo de geo localización




**Fuente:** (Elaboración propia)

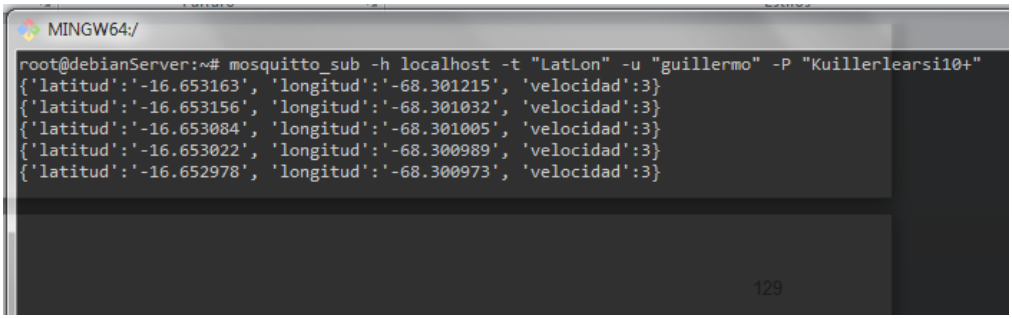
En la anterior imagen podemos ver el display informativo que tiene el prototipo de geo localización y se puede observar que al iniciarlo, en primera instancia se realiza la configuración inicial automática y posteriormente comienza a mandar la ubicación en función el modulo GPS NEO 6M a través del módulo GSM/GPRS SIM 900.

La misma es enviado al servidor al broker Mosquitto previamente configurado la cual recibe los datos y los transmite a todos aquellos que se haya suscrito el broker.

Cabe aclarar que Arduino se publica los mensajes sobre el broker mosquitto mandando los datos de usuario y contraseña, caso contrario no sería posible publicar la ubicación que se obtiene, como parte de la seguridad el broker está configurado con el respectivo usuario y contraseña para el acceso a la mism.

A continuación podemos observar como es la recepción de los datos en la consola del bróker.

**Figura Nº 4.2:** Verificación de broker Mosquitto

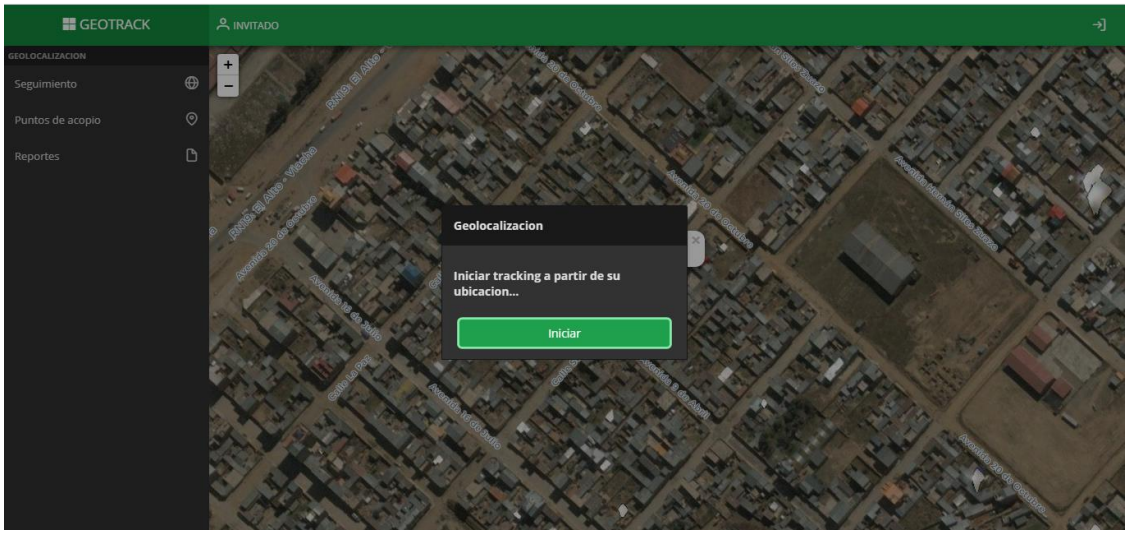



**Fuente:** (Elaboración propia)

El broker está a la escucha de los mensajes que envían los Publisher en este caso el Publisher que sería el Arduino está mandando la ubicación cada tres segundos, y broker automáticamente va mandando estos datos a todos los clientes suscritos al broker.

El cliente que se suscribe a los datos de ubicación lo realiza de la siguiente manera:

**Figura Nº 4.3:** Pantalla de inicio de seguimiento y geo localización

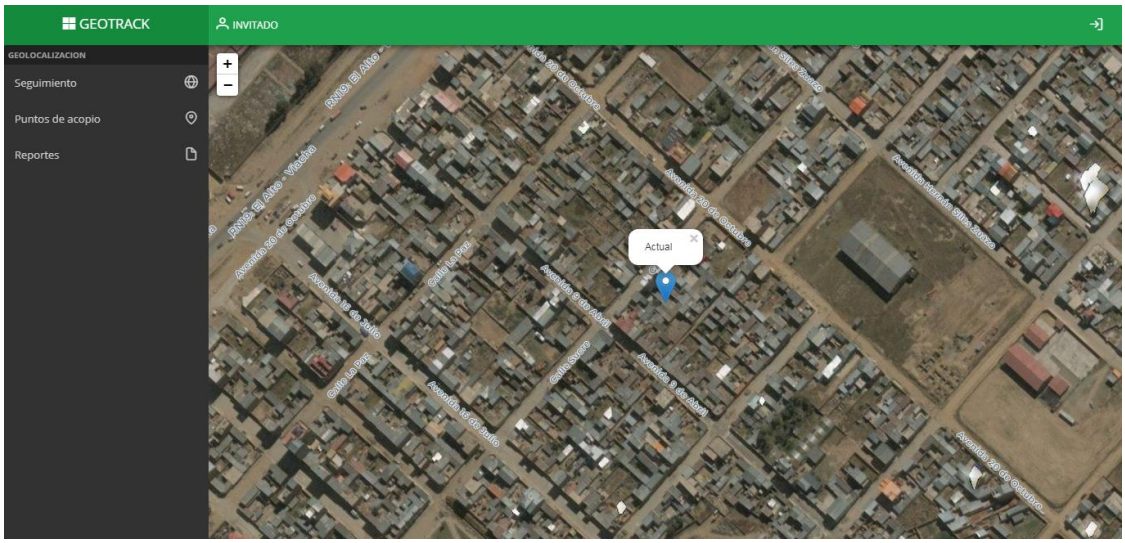


**Fuente:** (Elaboración propia)

En el navegador al entrar a la aplicación nos pregunta si queremos iniciar la el tracking correspondiente, esto quiere decir que al iniciar lo que estamos haciendo es suscribirnos al broker mosquito y estamos a la escucha de todos los datos que vayan enviando el prototipo de geo localización.

Una vez aceptado obtenemos lo siguiente, a medida que vayan llegando los datos de la ubicación a través del broker la aplicación los recibe los procesa y los marca en el mapa, en tiempo real de esta manera pudiendo hacer la supervisión correspondiente a los vehículos recolectores de basura.

**Figura Nº 4.4:** Seguimiento y geo localización de dispositivo



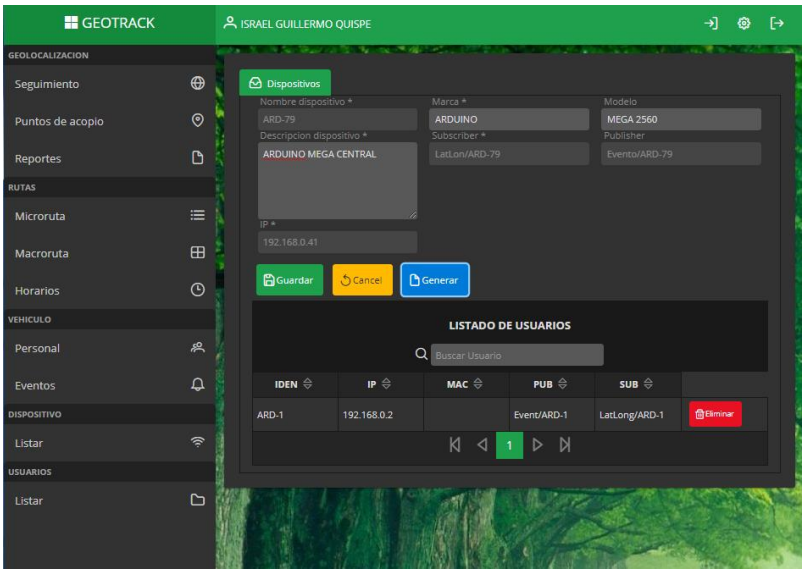

**Fuente:** (Elaboración propia)

Cabe aclarar que por seguridad los dispositivos que van a realizar el tracking correspondiente deben estar registrados en el sistema, ya que internamente la misma se la valida, a continuación podemos observar el módulo de registro de dispositivo de geo localización.

En esta parte del sistema, para el registro de dispositivo solo debe ingresarse tres datos marca, modelo y descripción del dispositivo, y en función a esos datos se genera otros datos muy a partes que servirán como identificación para

la misma, y a su vez todo esto generará el código correspondiente para el control del prototipo de geo localización con los parámetros ya definidos listo para el funcionamiento.

**Figura Nº 4.5:** Administración de dispositivos de geo localización




**Fuente:** (Elaboración propia)

# CAPITULO V

# CONCLUSIONES Y RECOMENDACIONES

## 5.1. CONCLUSIONES

Una vez finalizado el desarrollo e implementación del “SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS – CASO: GOBIERNO AUTONOMO MUNICIPAL DE VIACHA”, se ha logrado alcanzar el objetivo principal planteado, los requerimientos de la institución.

Tomando en cuenta los objetivos planteados se llega a las siguientes conclusiones:

- Se realizó el diagnostico correspondiente del funcionamiento actual del servicio de recojo de basura, recabando datos muy importantes para el desarrollo e implementación del presente sistema.
- Se aplicó la metodología UWE para el análisis y diseños del presente sistema en función a los requerimientos definidos por la institución.
- Se generó el presupuesto correspondiente para la implementación del prototipo y sistema del presente trabajo.
- Se implementó el prototipo de geo localización para los vehículos en función a los requerimientos de la institución.
- Se aplicó conceptos del internet de las cosas, otorgando conectividad a objetos de uso cotidiano, en este caso, los vehículos recolectores de basura, junto al brocker Mosquito que es el corazón de la comunicación en el internet de las cosas.
- Se generó los reportes necesarios de las rutas realizadas por los vehículos encargados del recojo de basura en formato pdf.

## 5.2. RECOMENDACIONES

Finalmente se describen algunas estrategias, además de trabajos futuros a partir del proyecto desarrollado:

- Se recomienda realizar el uso diferentes tecnologías de hardware abierto como por ejemplo Raspberry PI, las cuales ofrecen mayores prestaciones de rendimiento.
- Se recomienda el uso de sensores para el mejor reconocimiento del entorno en la cual está en funcionamiento el prototipo de geo localización.
- Se recomienda buscar alternativas más eficientes para la alimentación eléctrica del prototipo.
- Se recomienda explorar nuevas tecnologías, plataformas para le mejor desarrollo del sistema en general.

# BIBLIOGRAFIA

BBVA Open4U. (5 de Marzo de 2020). *bbvaopen4u.com*. Recuperado el 29 de Mayo de 2019, de https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos

Académicos Intercontinentales. (10 de Enero de 2019). *eumed.net*. Recuperado el 21 de Junio de 2019, de http://www.eumed.net/libros-gratis/2008a/351/Calidad%20de%20Software.htm

Agafonkin, V. (10 de Enero de 2020). *Leaflet*. Recuperado el 10 de Abril de 2019, de https://leafletjs.com/

Apaza, S. Z. (2016). *Prospección geoeléctrica de agua subterránea mediante dispositivos móviles y arduino.* La Paz.

Arduino. (21 de Enero de 2020). *arduino.cc*. Recuperado el 18 de Julio de 2019, de https://www.arduino.cc/en/Guide/Introduction

Arduino. (20 de Mayo de 2020). *store.arduino.cc*. Recuperado el 10 de Marzo de 2019, de https://store.arduino.cc/usa/mega-2560-r3

Artero, Ó. T. (2013). *Arduino curso práctico de formación.* Mexico DF, Mexico: Alfaomega.

Astudillo, J. P., & Delgado, E. G. (2012). *Sistema de localizacion monitoreo y control vehicular basado en los protocolos GPS/GSM/GPRS.* Cuenca.

Auth0. (23 de Enero de 2020). *jwt.io*. Recuperado el 20 de Mayo de 2019, de https://jwt.io/introduction/

Biundo, A. M. (2008). *Diseño de un sistema de seguridad y monitoreo de vehiculos.* Sertanejas.

Blanco, S. D. (20 de Mayo de 2020). *metodologiauwe.wordpress.com*. Recuperado el 23 de Abril de 2019, de https://metodologiauwe.wordpress.com/2015/06/25/hello-world/

Boehm, B. (2000). *Software Cost Estimation with Cocomo II* (Primera ed.). Prentice Hall.

Choque, M. K. (2016). *Sistema Domótico de Seguridad perimetral basado en Arduino.* La Paz.

ConceptoDefinicion. (1 de Marzo de 2020). *conceptodefinicion.de*. Recuperado el 1 de Marzo de 2020, de https://conceptodefinicion.de/monitoreo/

CriarWeb S.L. (10 de Abril de 2020). *desarrolloweb.com*. Obtenido de https://desarrolloweb.com/home/angular

Danielle, D. (16 de Febrero de 2020). *http://arquitecturasomos4.blogspot.com/*. Recuperado el 29 de Mayo de 2019, de http://arquitecturasomos4.blogspot.com/2014/12/publish-subscriber.html

De León, J. F. (2013). *Control de navegación con Arduino de un modelo de barco.* Valencia.

Definicion.DE. (10 de Febrero de 2020). *Definicion.DE*. Recuperado el 10 de Febrero de 2020, de https://definicion.de/conectividad/

Delfín, M. (2012). *Diseño del servicio de aseo del municipio de Viacha - La Paz.* Gobierno Autonomo Municipal de Viacha. Viacha: Propio.

DesarrolloWeb. (21 de Enero de 2020). *desarrolloweb.com*. Recuperado el 29 de Mayo de 2019, de https://desarrolloweb.com/articulos/que-es-mvc.html

Dev Code. (10 de Abril de 2019). *Dev Code*. Recuperado el 10 de Abril de 2019, de https://devcode.la/blog/que-es-nodejs/

DIGITAL55. (10 de Abril de 2020). *digital55.com*. Obtenido de https://www.digital55.com/desarrollo-tecnologia/apificacion-modelos-nestjs/

Eclipse Foundation. (21 de Marzo de 2020). *https://mosquitto.org/*. Recuperado el 29 de Mayo de 2019, de https://mosquitto.org/

EcuRed. (1 de Marzo de 2020). *EcuRed*. Recuperado el 1 de Marzo de 2020, de https://www.ecured.cu/Circuito_electr%C3%B3nico

EDteam. (10 de Abril de 2019). *ed.team*. Recuperado el 10 de Abril de 2019, de https://ed.team/blog/agiliza-tu-desarrollo-en-nodejs-con-el-orm-sequelize

Escuela de Organizacion Industrial. (6 de Mayo de 2012). Recuperado el 20 de Febrero de 2020, de https://www.eoi.es/blogs/cesaraparicio/2012/05/06/el-modelo-cocomo-para-estimar-costes-en-un-proyecto-de-software/

Fandom. (5 de Febrero de 2020). *Fandom, Inc*. Recuperado el 24 de Junio de 2019, de https://modelos-de-evaluacion-de-los-red-grupo-8-udes.fandom.com/es/wiki/Modelo_de_Calidad_McCall

Freeman, A. (2018). *Pro Angular 6.* London: Apress.

Garcia, E. (12 de Marzo de 2020). *emmanuelgs.blogspot*. Recuperado el 21 de Junio de 2019, de https://emmanuelgs.blogspot.com/2011/04/poo-clase-9-diseno-de-pruebas-unitarias.html

Garcia, J. M. (2019). *Teoría y ejercicios prácticos de Dinamica de Sistemas* (Segunda ed.). Barcelona.

Geek Factory. (10 de Marzo de 2020). *geekfactory.mx*. Recuperado el 20 de Febrero de 2019, de https://www.geekfactory.mx/tienda/radiofrecuencia/sistema-minimo-sim900-modulo-simcom/

Geek Factory. (10 de Marzo de 2020). *geekfactory.mx*. Recuperado el 20 de Febrero de 2019, de https://www.geekfactory.mx/tienda/radiofrecuencia/ublox-neo-6m-modulo-gps-economico/

Geoinnova Formación SIG y Medio Ambiente. (10 de Febrero de 2020). *geoinnova*. Recuperado el 10 de Febrero de 2020, de https://geoinnova.org/cursos/que-son-los-sistemas-de-informacion-geografica-sig/

Gómez, A., Migani, S., & Otazú, A. (2008). *COCOMO - Un modelo de estimación de proyectos de software.*

If.Deiby. (2 de Mayo de 2020). *mccallisos.blogspot.com*. Recuperado el 24 de Junio de 2019, de https://mccallisos.blogspot.com/

Johansen, O. (1993). *Teoria General de Sistemas.* Mexico: Grupo Noriega Editores.

Kendall, K. (2005). *Análisis y Diseño de Sistemas.* Mexico: Pearson.

*La Teoría General de Sistemas.* (1995). Madrid: Gráficas Marte.

Lampkin, V. (2014). *Responsive Mobile User Experience Using MQTT and IBM MessageSight.*

Limachi, O. A. (2016). *Administración remota para el monitoreo de sensores en base a raidiofrecuencia aplicando Arduino.* La Paz.

Lombardi, A. (2015). *WebSocket Lightweight Client-Server Communications.* California: O'Reilly.

Monografias.com. (1 de Marzo de 2020). *monografias.com*. Recuperado el 1 de Marzo de 2020, de https://www.monografias.com/trabajos5/andi/andi.shtml

Mysliwiec, K., & Staron, J. (10 de Abril de 2019). *https://docs.nestjs.com/*. Recuperado el 10 de Abril de 2019, de https://docs.nestjs.com/

Naylamp Mechatronics. (25 de Enero de 2020). *https://naylampmechatronics.com/*. Recuperado el 10 de Diciembre de 2018, de https://naylampmechatronics.com/sensores-posicion-inerciales-gps/106-modulo-gps.html

Olaya, V. (2014). *Sistemas de Información Geográfica.*

OpenStreetMap Wiki Contributos. (10 de Enero de 2020). *wiki.openstreetmap.org*. Obtenido de https://wiki.openstreetmap.org/wiki/ES:Leaflet

Pressman, R. S. (2003). *Ingeniería de Software - Un enfoque práctico.* Madrid, España: Fareso.

Pressman, R. S. (2010). *Ingenieria del Software - Un enfoque práctico* (Séptima ed.). Mexico D.F.: McGraw-Hill.

Prometec. (20 de Junio de 2020). *prometec.net*. Recuperado el 14 de Abril de 2019, de https://www.prometec.net/gprs-llamar-enviar-sms/

Punto Flotante. (28 de Marzo de 2020). *Punto Flotante S.A.* Recuperado el 20 de Enero de 2019, de http://www.puntoflotante.net/TUTORIAL-MODEM-GSM-GPRS.htm

QGIS. (3 de Julio de 2020). *qgis.org*. Recuperado el 20 de Febrero de 2020, de https://www.qgis.org/es/site/about/index.html

Quispe, I. (2016). *Internet de las Cosas, Control y Seguimiento de un Automóvil.* La Paz.

RIP Tutorial. (10 de Febrero de 2020). *riptutorial.com*. Obtenido de https://riptutorial.com/es/sequelize-js

Robles, V. (8 de Abril de 2019). *victorroblesweb.es/*. Recuperado el 8 de Abril de 2019, de https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/

Sancho, V. (20 de Febrero de 2020). *trellat.es*. Recuperado el 20 de Mayo de 2019, de https://trellat.es/los-comandos-at/

Schmuller, J. (2010). *Aprendiendo UML en 24 Horas.* Pearson Educación Latinoamérica.

SIMCom. (2015). *SIM900 AT Commands Manual.* Shanghai: SIM Technology Building.

Sommerville, I. (2011). *Ingeniería de Software* (Novena edición ed.). Mexico: Pearson.

The PostgreSQL Global Development Group. (10 de Enero de 2020). *postgresql.org*. Recuperado el 10 de Abril de 2019, de https://www.postgresql.org/docs/9.6/intro-whatis.html

The PostgreSQL Global Develpoment Group. (26 de Febrero de 2020). *postgresql.org*. Obtenido de https://www.postgresql.org/about/

Tutorials Point. (3 de Mayo de 2020). *tutorialspoint.com*. Obtenido de https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm

Von Bertalanffy, L. (1989). *Teoria General de los Sistemas.* Mexico: Fondo de Cultura Económica.

Web Engineering Group. (10 de Mayo de 2020). *uwe.pst.ifi.lmu.de*. Recuperado el 23 de Abril de 2019, de http://uwe.pst.ifi.lmu.de/

Web profit. (10 de Enero de 2020). *Gestopolis*. Recuperado el 9 de Diciembre de 2018, de https://www.gestiopolis.com/sistema-control-gestion-conceptos-basicos-diseno/

WorldTimeZone. (5 de Mayo de 2020). *WorldTimeZone.com*. Recuperado el 25 de Enero de 2019, de http://www.worldtimezone.com/

www018. (s.f.). Recuperado el 23 de Abril de 2019, de http://uwe.pst.ifi.lmu.de/teachingTutorialNavigationSpanish.html

www019. (s.f.). Recuperado el 23 de Abril de 2019, de http://uwe.pst.ifi.lmu.de/teachingTutorialProcessSpanish.html

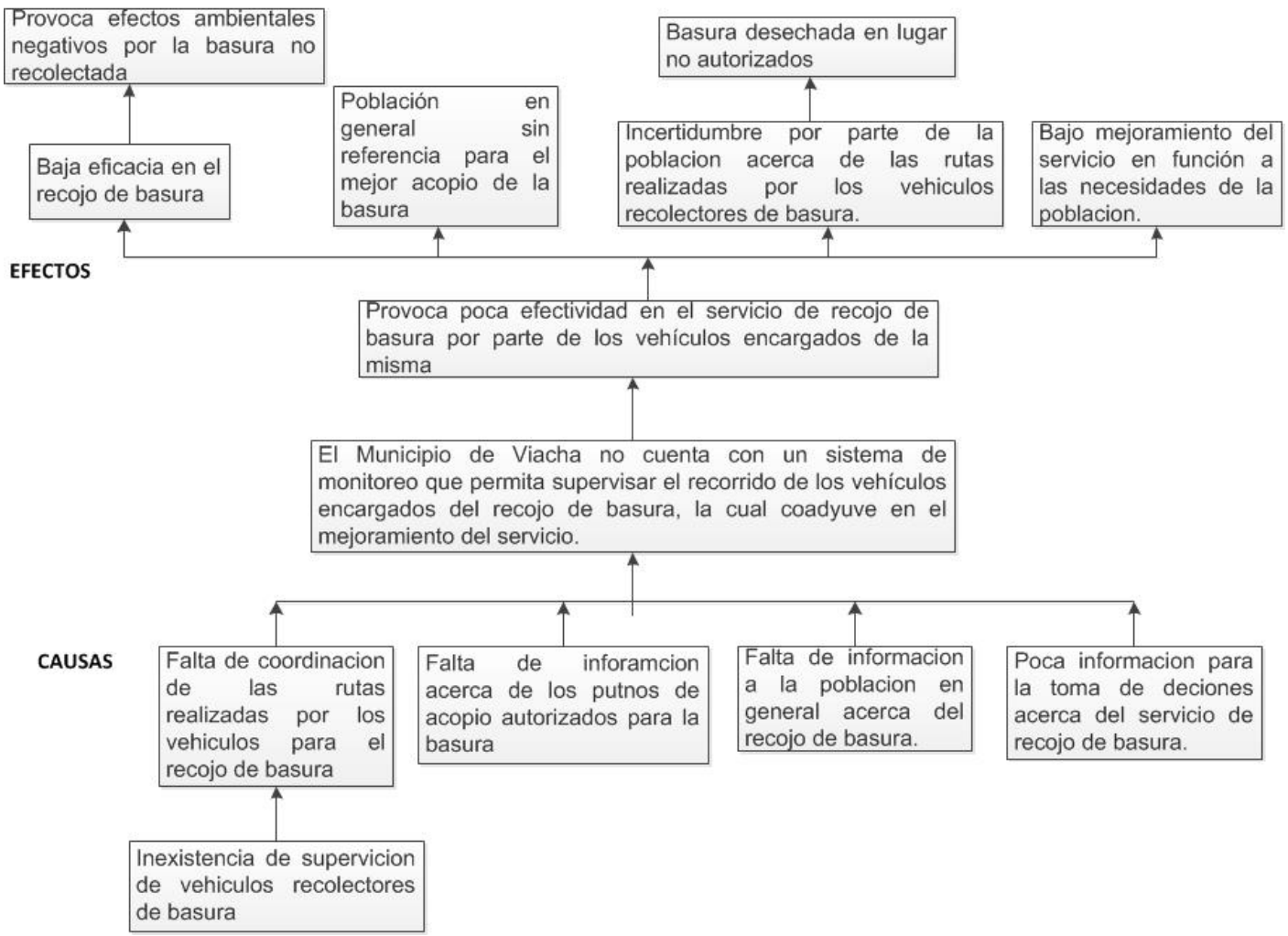
www020. (s.f.). Recuperado el 23 de Abril de 2019, de http://uwe.pst.ifi.lmu.de/teachingTutorialPresentationSpanish.html

www030. (s.f.). Recuperado el 24 de Junio de 2019, de https://trabajo-modulo-evaluacion-red.fandom.com/wiki/MODELO_DE_EVALUACI%C3%93N_McCAL

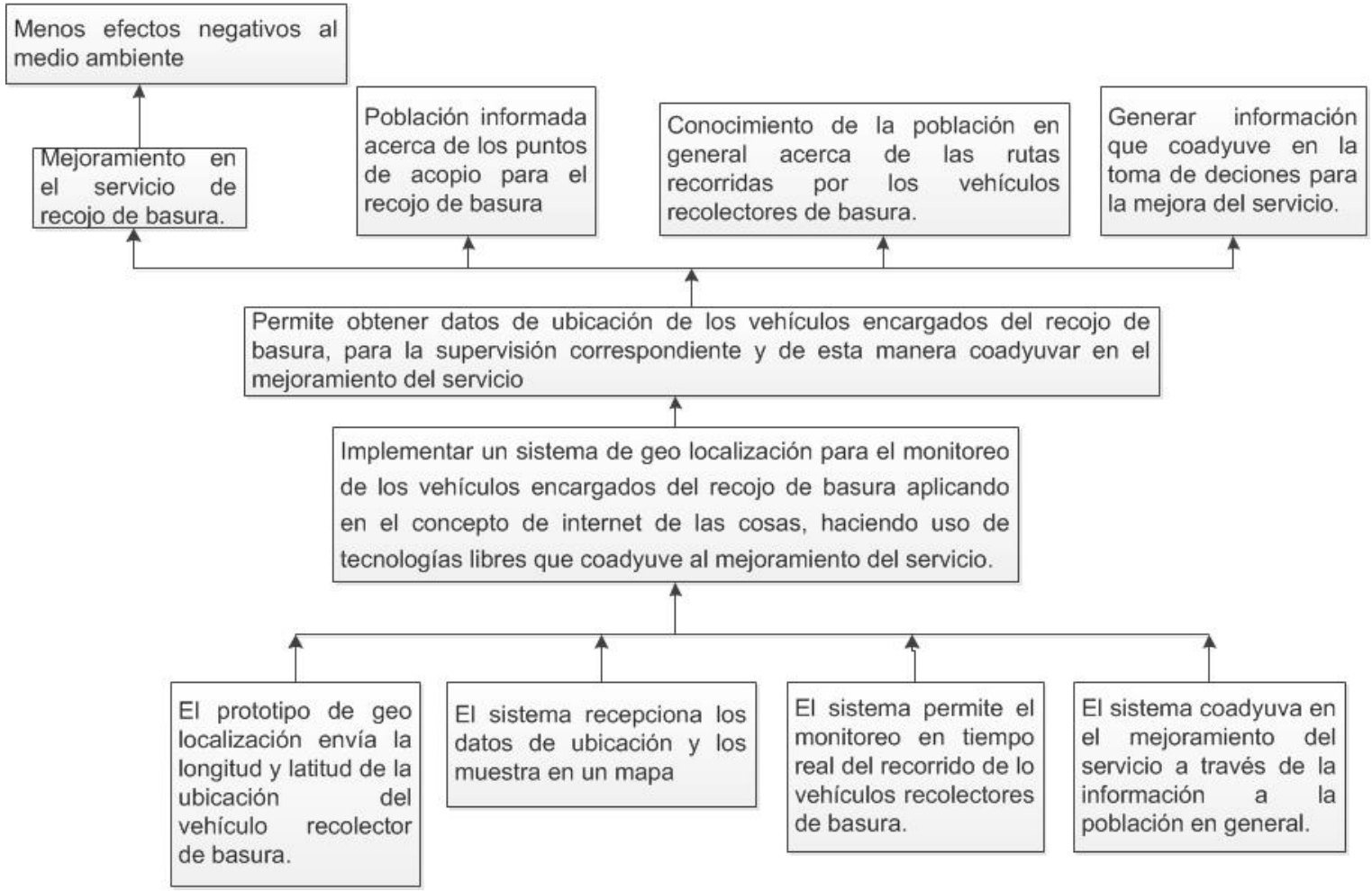
www032. (s.f.). Recuperado el 20 de Febrero de 2020, de http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/

# ANEXOS

## ÁRBOL DE PROBLEMAS



## ÁRBOL DE MEDIOS Y FINES



**Entrevista para Ingeniería de Requerimientos.**

1) ¿En qué consiste el servicio de recojo de basura del municipio?

……

……...

2) ¿Qué problemas se tiene al momento de ofrecer el servicio de recojo de basura?

……

……...

3) ¿Existe algún tipo de supervisión a los vehículos recolectores de basura por parte de la población?

……

……...

4) ¿Cómo se informa a la población acerca de los puntos de acopio para la basura?

……

……...

5) ¿Existe efectos ambientales causados por el mal manejo de la basura en el Municipio?

……

……...

6) ¿Qué información recaban acerca del recojo de basura en el municipio?

……

……...

7) ¿Existe algún sistema implementado de supervisión de los vehículos de recojo de basura para el cumplimiento fiel de las rutas a realizar?

……

……...

8) ¿Cómo están definidos las rutas y horarios para el servicio de recojo de basura?

……

……...

9) ¿Qué diagnostico le merece al actual servicio de recojo de basura?

……

……...

## Cuestionario

**1. ¿Tiene alguna dificultad en realizar los procesos del sistema?**

SI NO

**2. ¿Las respuestas del sistema son de su comprensibilidad?**

SI NO

**3. ¿Tiene alguna dificultad en realizar los procesos del sistema?**

SI NO

**4. ¿La interfaz de la aplicación es amigable y entendible a su parecer?**

SI NO

**5. ¿Son comprensibles y satisfactorios los datos que se muestran en el sistema?**

SI NO

**6. ¿El sistema es de fácil uso bajo su criterio?**

SI NO

**7. ¿El sistema coadyuva en el mejoramiento del servicio de recojo de basura?**

SI NO

**8. ¿Durante el uso del sistema se produjo errores?**

SI NO

# AVALES DE CONFORMIDAD

El Alto, 9 de Julio de 2020

Señor:

M.Sc. Ing. Enrique Flores Baltazar

**TUTOR METODOLÓGICO - TALLER II**

**CARRERA DE INGENIERIA DE SISTEMAS**

**UNIVERSIDAD PÚBLICA DE EL ALTO**

**Presente.-**

**Ref.: Aval de Conformidad**

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS" Caso: Gobierno Autónomo Municipal de Viacha, que propone el postulante: Israel Guillermo Quispe Mamani, con C.I.: 7017521 L.P. para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales

Atentamente,

**Lic. Freddy Salgueiro Trujillo**
**TUTOR ESPECIALISTA**

El Alto, 10 de Julio de 2020

Señor:

M.Sc. Enrique Flores Baltazar

**TUTOR METODOLÓGICO - TALLER II**

**CARRERA DE INGENIERIA DE SISTEMAS**

**UNIVERSIDAD PÚBLICA DE EL ALTO**

**Presente.-**

**Ref.: Aval de Conformidad**

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS" Caso: Gobierno Autónomo Municipal de Viacha, que propone el postulante: Israel Guillermo Quispe Mamani, con C.I.: 7017521 L.P., para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales

Atentamente,

……..

**Ing. Rosa Verastegui Ontiveros**

TUTOR REVISOR

El Alto, Julio de 2020

Señor:

Ing. David Carlos Mamani

**DIRECTOR**

**CARRERA DE INGENIERIA DE SISTEMAS**

**UNIVERSIDAD PÚBLICA DE EL ALTO**

**Presente.-**

**Ref.: Aval de Conformidad**

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS" Caso: Gobierno Autónomo Municipal de Viacha, que propone el postulante: Israel Guillermo Quispe Mamani, con C.I.: 7017521 L.P., para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales

Atentamente,

.....

**MSc. Enrique Flores Baltazar**

TUTOR METODOLOGICO TALLER II

# GOBIERNO AUTÓNOMO MUNICIPAL DE VIACHA

**Primera Sección Capital de la Provincia Ingavi**
**Viacha - La Paz - Bolivia**

Viacha, 6 de Septiembre, de 2019

Señor:

Ing. Enrique Flores Baltazar

**TUTOR METODOLOGICO TALLER II**

**CARRERA INGENIERIA DE SISTEMAS**

**UNIVERSIDAD PÚBLICA DE EL ALTO**

Presente.-

**Ref. AVAL DE CONFORMIDAD DEL PROYECTO DE GRADO "SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS – CASO: GOBIERNO AUTONOMO MUNICIPAL DE VIACHA"**

Distinguido Ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado denominado "SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS – CASO: GOBIERNO AUTONOMO MUNICIPAL DE VIACHA", realizado por el postulante universitario: Israel Guillermo Quispe Mamani, con cedula de identidad: 7017521 L.P., mencionado proyecto fue monitoreado y verificado por el director y responsable de unidad de medio ambiente, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales

Atentamente

Ing. Rene L. Rojas Gutierrez
DIRECTOR DE GESTION
AMBIENTAL Y RIESGOS a.i.
Gobierno Autonomo Municipal de Viacha

GOBIERNO AUTONOMO MUNICIPAL DE VIACHA
DIRECCION DE GESTION AMBIENTAL Y RIESGOS
D.G.A.
Viacha - Bolivia

Dirección: Plaza Mariscal José Ballivián Nº 102 • Teléfonos: 2800006 - 2800508 - 2800759 • Fax. 2800130
Página Web: www.viacha.gob.bo • Correo Electrónico: info@viacha.gob.bo

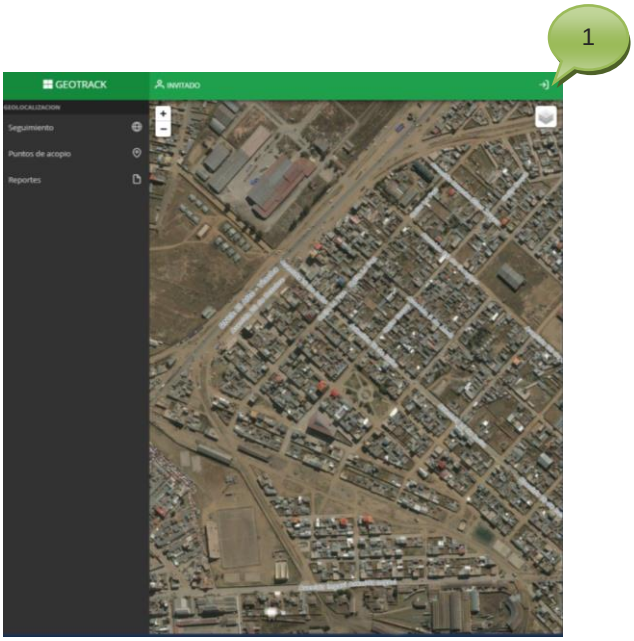
# MANUAL DE ADMINISTRADOR GEOTRACK

SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS.

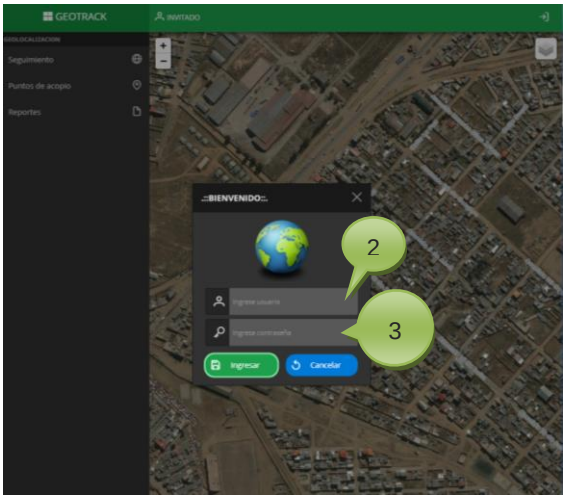
# MANUAL DE USUARIO ADMINISTRADOR - SISTEMA

MODULO DE LOGIN

Para tener acceso al sistema como administrador se debe realizar el login correspondiente.




1. Clic para desplegar la pantalla de login.

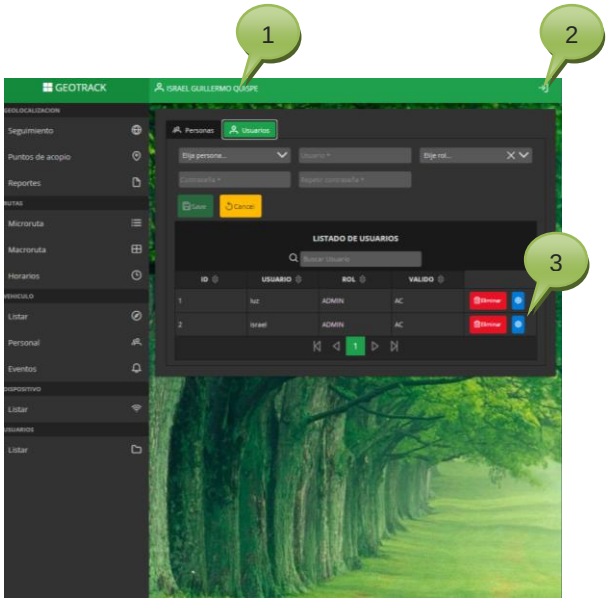



2. Ingresar usuario

3. Ingresar password

4. Clic en Ingresar

Una vez que el administrador ingrese su nombre de usuario y contraseña podrá acceder al sistema, en el cual tendrá el control total de la misma.



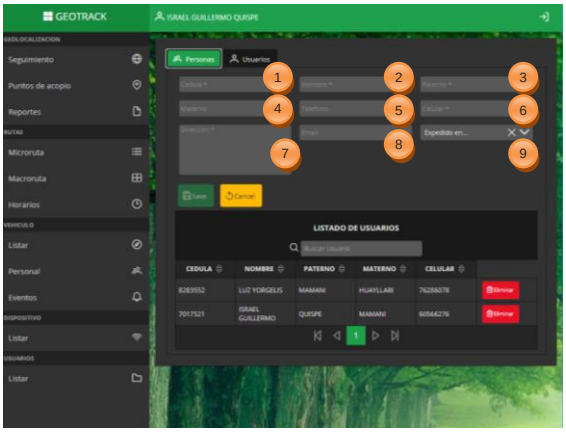

1. Usuario actual logeado

2. Salir del sistema

3. Listado de usuarios

**MODULO DE REGISTRO DE PERSONA**

En este módulo se realiza el registro correspondientes a las personas intervinientes en el sistema.




Los datos requeridos para realizar el registro de una nueva persona son:

1. Ingresar el número de cedula de identidad

2. Ingresar nombres

3. Ingresar apellido paterno

4. Ingresar apellido materno (no obligatorio)

5. Ingresar teléfono (no obligatorio)

6. Ingresar celular

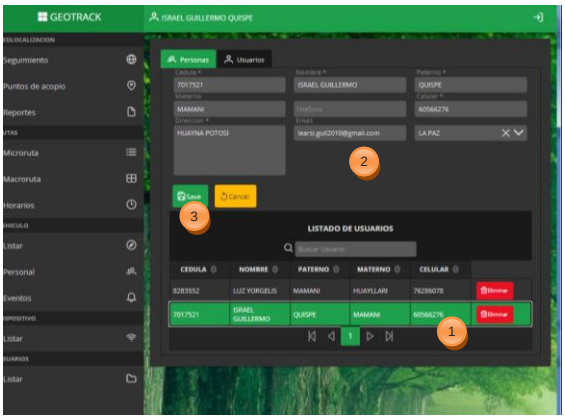
7. Ingresar dirección

8. Ingresar email

9. Seleccionar el departamento donde se expidió el ci.

Así también en el listado de personas tenemos las opciones de eliminar y modificar los registros.

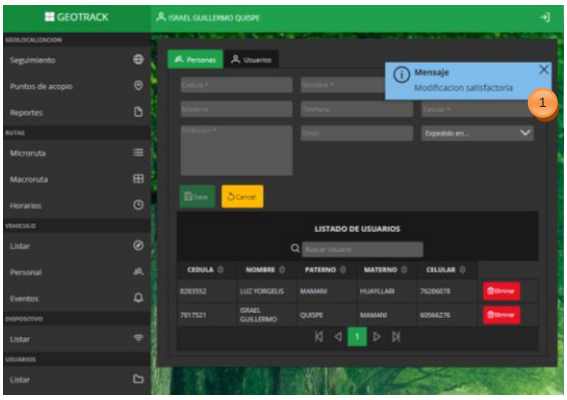
Para modificar:




1. Se realiza la selección del registro en el listado haciendo clic

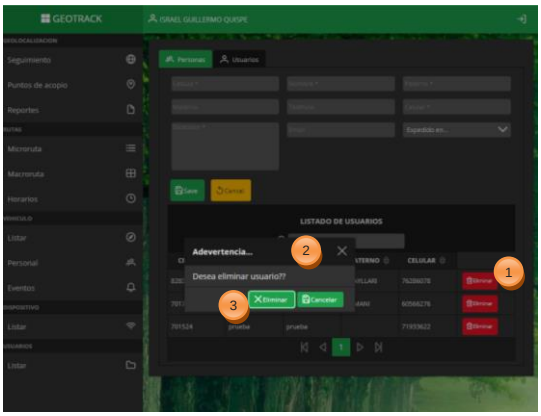
2. Se modifican los datos necesarios

3. Clic en save




1. Modificación satisfactoria

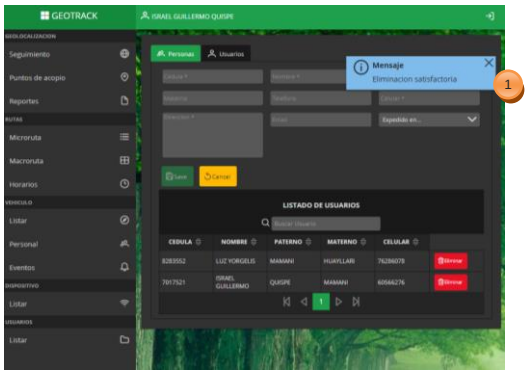
Para eliminar




1. Hacemos clic en eliminar del registro correspondiente.

2. Nos saldrá una ventana de confirmación.

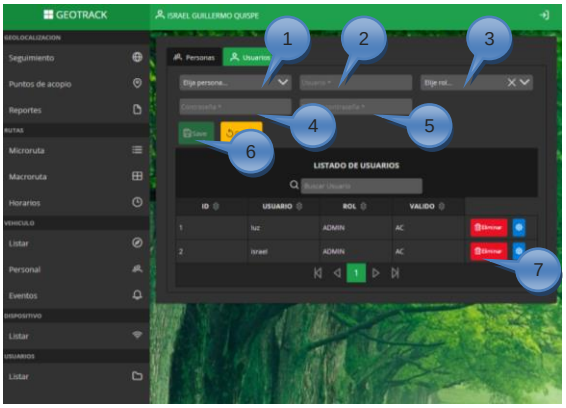
3. Hacemos clic en Eliminar.




1. Eliminación satisfactoria

## MODULO REGISTRO DE USUARIO

Para el registro de un nuevo usuario realizaremos lo siguiente:




1. Elegir persona ya registrada por su ci

2. Ingresar usuario

3. Elegir rol

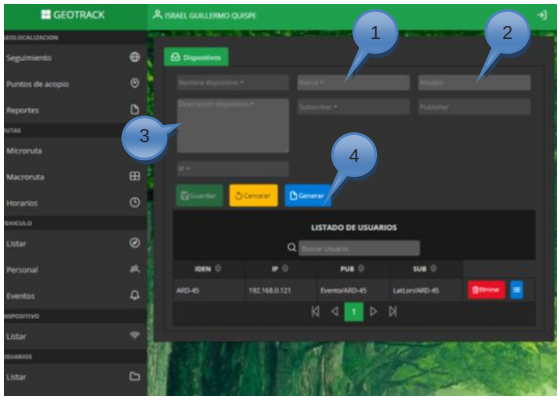
4. Ingresar contraseña

5. Ingresar contraseña nuevamente para verificación

6. Hacer clic en Save.

7. Opción para eliminación de usuario.

## MODULO DE REGISTRO DE DISPOSITIVO DE GEO LOCALIZACION



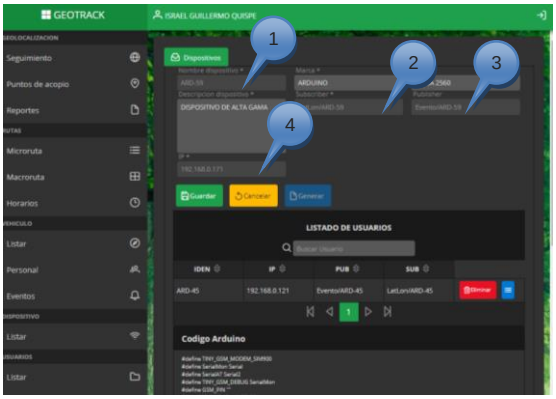

1. Ingresar marca del dispositivo (Ejemplo: Arduino)

2. Ingresar modelo del dispositivo (Ejemplo: MEGA 2560)

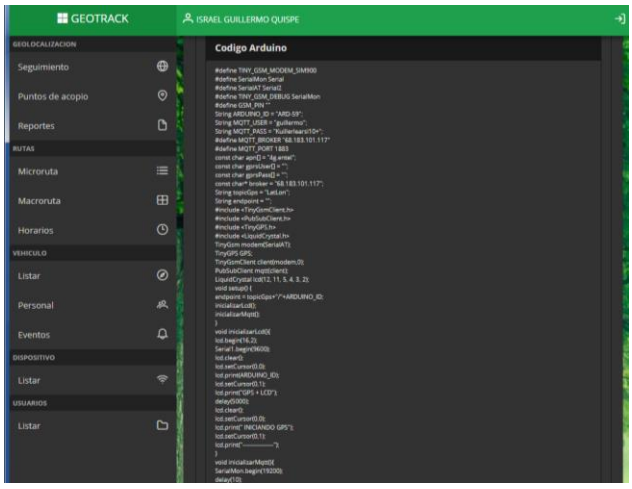
3. Ingresar descripción.

4. Hacemos clic en generar

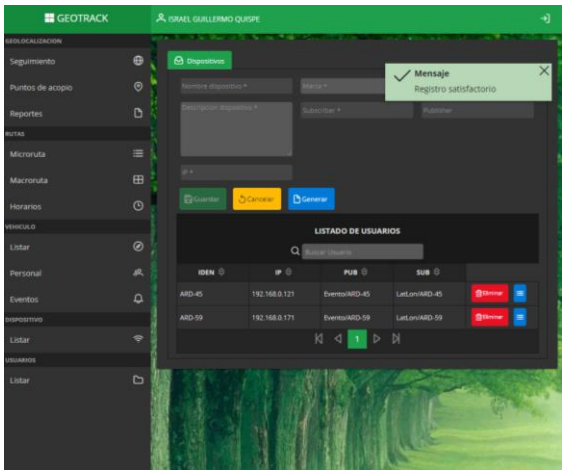
Al hacer clic en generar el sistema genera datos necesarios para antes de registro y así también el código de control para el arduino, la cual se lo debe grabar en el arduino con el IDE correspondiente.




Se puede observar que los campos 1,2,3,4 se generaron automáticamente y juntamente a ello se generó el siguiente código de control para el Arduino:




Posteriormente de dar clic en generar damos clic en guardar

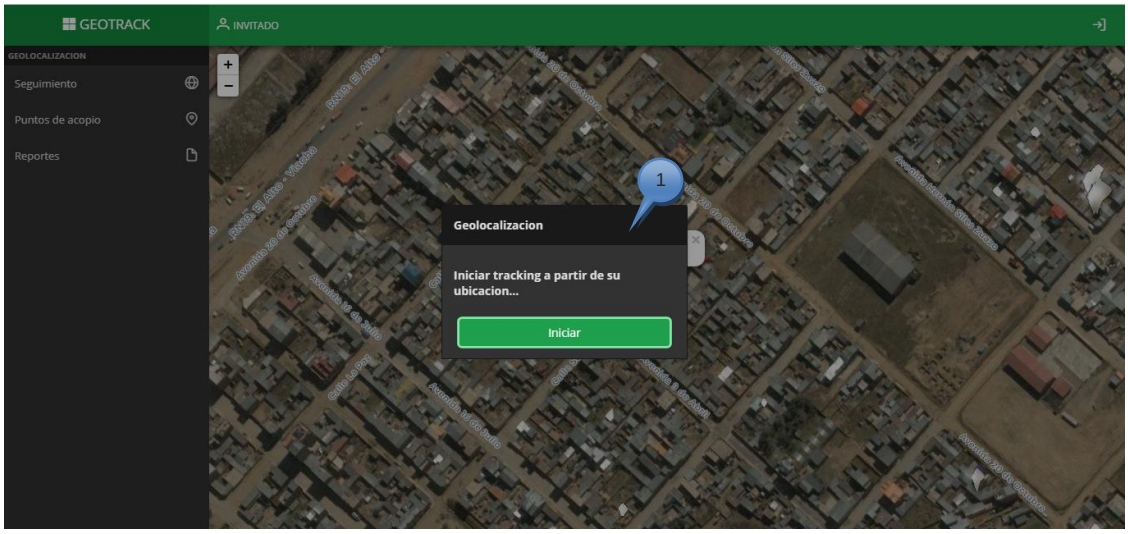



Se puede observar que se realiza un registro exitoso, y el dispositivo se lista automáticamente para otras operaciones como eliminación o generación de código de control nuevamente.

# MANUAL DE USUARIO VISITANTE GEOTRACK

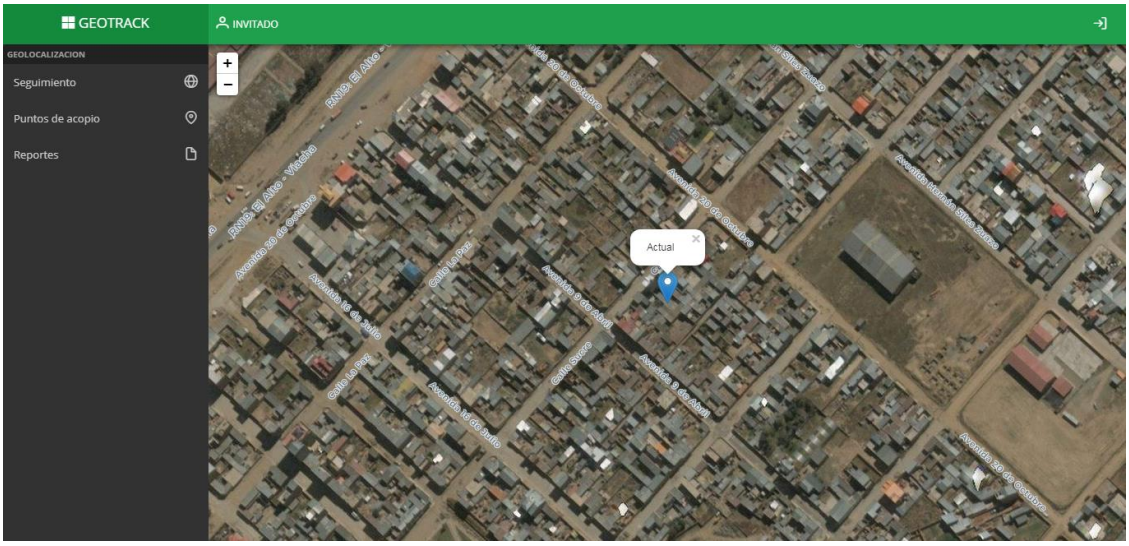
SISTEMA DE GEO LOCALIZACION DE VEHICULOS RECOLECTORES DE BASURA APLICANDO INTERNET DE LAS COSAS.

Para el usuario visitante a la aplicación simplemente se accede a la misma y nos muestra como pagina principal el siguiente mapa:

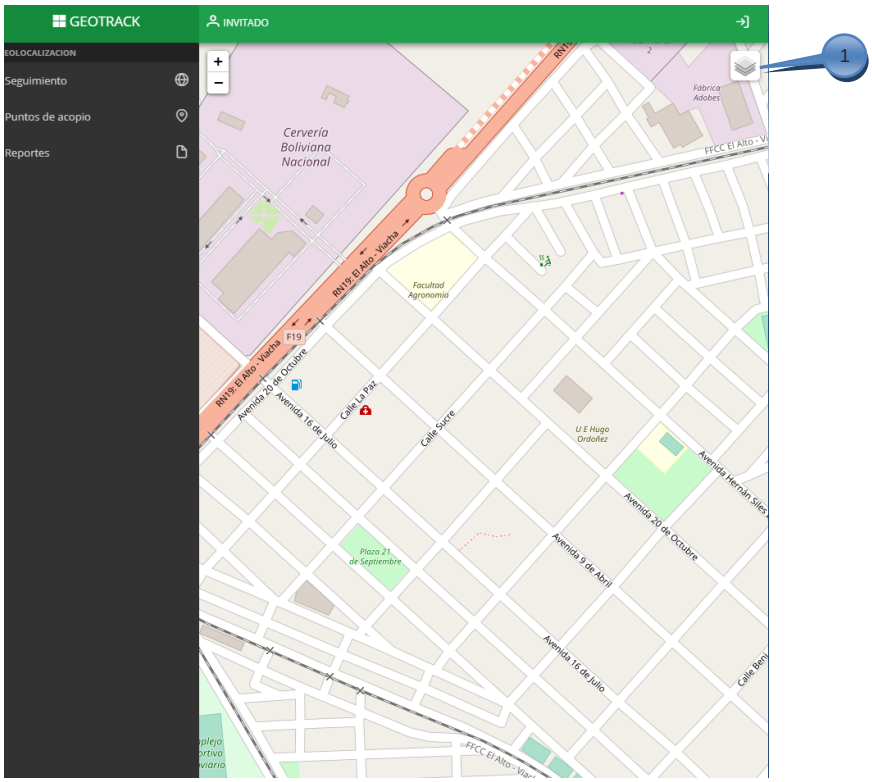



Nos aparece una ventana emergente para comenzar con el seguimiento correspondiente al dispositivo de geo localización.

1. Hacemos clic en Iniciar, y automáticamente el mapa se ubicara en el dispositivo para realizar la geo localización.



Como se puede observar el dispositivo a medida que va en movimiento la misma se va mostrando en el mapa realizando el seguimiento correspondiente




Así también, se tiene dos tipos de controles.

1. Elección de capas: La misma nos da las opciones de elegir capas en planos o de vista satelital, así también, una capa adicional para ver el nombre de las calles.