

“2022. Año del Quincentenario de la Fundación de Toluca de Lerdo, Capital del Estado de México”.

# “Sistema de monitoreo en flota vehicular para el municipio de Huixquilucan”

## TESIS

**QUE PARA OBTENER EL GRADO DE:**

**MAESTRO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**PRESENTA:**

**ING. ALFREDO EMMANUEL CAMPOS PÉREZ**

**EN COLABORACIÓN DE:**

**ING. DANIEL SANTILLAN MARTINEZ**

**DIRECTOR DE TESIS**

**M. en TIC. JOSE LUIS CAMACHO CAMPERO**

**CUAUTITLÁN IZCALLI, EDO. DE MÉXICO**

**MARZO 2022**

“2022. Año del Quincentenario de la Fundación de Toluca de Lerdo, Capital del Estado de México”.

## Agradecimientos

A todos los que me apoyaron y motivaron para seguirme preparando y creciendo en conocimiento, a mi esposa por su comprensión, a Daniel por su muy provechosa colaboración y amistad, al director de Tesis por su eterna paciencia y consejos, a Dios por todo.

## **Resumen**

La presente tesis realiza un diseño y modelado para el desarrollo de un sistema de monitoreo en la flota vehicular para el municipio de Huixquilucan en el que se pretende tener un mejor control en las unidades, evitando el desvío de fondos y malas prácticas. Este diseño contiene un sistema de rastreo haciendo uso de un GPS y para el monitoreo de los datos vehiculares se hace uso del puerto OBD-II, estos datos son enviados vía mensaje de texto a un programa que capta y registra la información para cada unidad, con lo que quiere alcanzar una mejora en la organización, reducción de consumibles y promover las buenas prácticas en el uso de los vehículos.

Palabras clave: Sistema de monitoreo, OBD-II, GPS, SMS, flotillas vehiculares.

## **Abstract**

This thesis carries out a design and modeling for the development of a monitoring system in the vehicle fleet for the municipality of Huixquilucan in which it is intended to have better control in the units, avoiding the diversion of funds and bad practices. This design contains a tracking system using a GPS and for the monitoring of vehicle data, one is made from the OBD-II port, these data are sent via text message to a program that captures and records the information for each unit, with which it wants to achieve an improvement in the organization, reduction of consumables and promote good practices in the use of vehicles.

Keywords: Monitoring system, OBD-II, GPS, SMS, vehicle fleets.

## Contenido

### Contenido

Agradecimientos .....	1
Resumen.....	2
Abstract.....	2
Índice de Figuras.....	6
CAPÍTULO 1 .....	8
INTRODUCCIÓN .....	8
1. Introducción .....	9
1.2 Planteamiento del problema.....	10
1.3 Hipótesis .....	11
1.4 Justificación.....	11
1.5 Objetivos generales y específicos .....	12
General: .....	12
Específicos: .....	12
Alcances .....	12
1.6 Cronograma de actividades .....	13
CAPITULO 2 .....	14
MARCO TEÓRICO .....	14
2.1. Sistema de Posicionamiento Global.....	15
2.1.2 Arquitectura GPS .....	17
2.2 SISTEMA GSM .....	17
2.2.1 Funcionamiento de la red GSM .....	18
B) Estación base.....	19
C) Sistema de Red MTSO .....	20
2.2.2 Arquitectura del sistema GSM .....	20
2.2.3 Módulo SIM 808.....	21

2.3 Sistema OBD-II .....	22
2.4 Ingeniería de software.....	23
i) Ingeniería de requerimientos.....	24
ii) Metodologías Ágiles .....	25
A) Programación Extrema(PE) .....	25
B) Scrum.....	26
2.5 Lenguajes de Programación .....	30
2.5.1 Front End .....	31
2.5.2 Frameworks .....	33
2.5.3 Backend.....	35
A) Ruby .....	35
B) Java .....	35
2.5.4 Sistema Gestor de Base de Datos.....	36
a) SQL Server.....	37
b) Oracle.....	37
c) My SQL.....	38
2.5.4.1 Normalización de Base de Datos.....	39
2.5.5 Estándares de Calidad y Seguridad.....	39
2.5.6 Calidad de Software.....	41
a) Métricas de proceso de software .....	42
CAPITULO 3 .....	43
DESARROLLO.....	43
3.1 Diagramas para la obtención de las coordenadas del sistema de posicionamiento global.....	44
3.2 Diagrama para la obtención de datos del OBDII .....	47
3.3 Diagramas de casos de uso .....	49
3.4 Diagrama entidad-relación .....	52
3.5 Diagramas de clase.....	53
A) Clase Token Service.....	53
B) Clase Unidad Service .....	54

C)	Clase FacturaService .....	55
D)	Clase CirculacionService .....	56
E)	Clase SeguroService .....	57
F)	Clase Verificacion Service .....	58
G)	Clase TenenciaService .....	59
H)	Clase MantenimientoService .....	60
I)	Clase OdbService .....	61
J)	Clase RastreadorService .....	62
K)	Clase PosicionService .....	63
L)	Clase UsuarioService.....	64
M)	Clase ArchivoService.....	65
N)	Clase MarcaService.....	66
O)	Clase DepartamentoService .....	67
3.6	Maquetados .....	68
A)	Maquetado Token.....	68
B)	Maquetado Vista principal.....	69
C)	Maquetado Unidad .....	70
D)	Maquetado Factura.....	71
CAPITULO 4	.....	84
RESULTADOS	.....	84
4.1	Circuitos electrónicos .....	85
4.2	Configuración del Bluetooth .....	86
4.3	Envío de datos .....	88
4.4	Almacenamiento en plataforma digital .....	90
Conclusiones	.....	95
Referencias	.....	97
Bibliografía	.....	97

# Índice de Figuras

Figura 1 Cronograma de actividades - Desarrollo del proyecto (elaborado en Gantt Project) _	13
Figura 2 Cronograma de actividades- Desarrollo de Tesis (Elaborado en Gantt Project) ____	13
Figura 3 Arquitectura del sistema GSM (Tomasi,2003)_____	20
Figura 4 Diagrama de bloques obtención GPS (Autoría propia ) _____	44
Figura 5 Declaración de puertos de comunicación Arduino- SIM808, (autoría propia)_____	45
Figura 6 Comparación de arreglos en la petición recibida (autoría Propia) _____	46
Figura 7 Texto enviado por medio de la SIM808 (autoría propia)_____	46
Figura 8 Diagrama de bloques para la obtención de los datos obtenidos del protocolo OBDII (autoría propia) _____	47
Figura 9 Declaración de puertos de comunicación (Autoría propia) _____	48
Figura 10 comunicación HC-05 y ELM327 a 115200 baudios (autoría propia)_____	48
Figura 11 parámetros obtenidos del OBD II (autoría propia)_____	49
Figura 12 Diagramas de casos de uso_ Login (autoría propia)_____	50
Figura 13 Diagrama de casos de uso_ Gestión Catálogos Usuarios (autoría propia) _____	50
Figura 14 Diagrama de casos de uso_ Gestion de unidades (autoría propia) _____	51
Figura 15 Diagrama Entidad- Relación (autoría propia) _____	52
Figura 16 Diagrama de clases_ Token (autoría propia) _____	53
Figura 17Diagrama de clase _Unidad (autoría propia) _____	54
Figura 18 Diagrama de clase _ Factura (autoría propia)_____	55
Figura 19 Diagrama de clase _Circulación (autoría propia) _____	56
Figura 20 Diagrama de clase _ Seguro (autoría propia) _____	57
Figura 21 Diagrama de clase _ Verificación (autoría propia)_____	58
Figura 22 Diagrama de clase _ Tenencia (autoría propia) _____	59
Figura 23 Diagrama de clase _Mantenimiento (autoría propia) _____	60
Figura 24 Diagrama de clase _ OBD (autoría propia) _____	61
Figura 25 Diagrama de clase _ Rastreador (autoría propia) _____	62
Figura 26 Diagrama de clase _ Posición (autoría propia) _____	63
Figura 27 Diagrama de clase _ Usuario (autoría propia)_____	64
Figura 28 Diagrama de clase _ Archivo _____	65
Figura 29 Diagrama de clase _Marca (autoría propia) _____	66
Figura 30 Diagrama de clase _ Departamento (autoría propia) _____	67
Figura 31 Maquetado_ Token (autoría propia) _____	68
Figura 32 Maquetado_ Vista Principal (autoría propia) _____	69
Figura 33 Maquetado _ Unidad (autoría propia) _____	70
Figura 34 Maquetado _ Factura (autoría propia) _____	71
Figura 35 Maquetado _ Tarjeta de circulación (autoría propia)_____	72
Figura 36 Maquetado_ Seguros (autoría propia)_____	73
Figura 37 Maquetado_ Verificación (autoría propia) _____	74
Figura 38 Maquetado _ Tenencia (autoría propia) _____	75
Figura 39 Maquetado _ Mantenimiento (autoría propia) _____	76
Figura 40 Maquetado _ OBD-II (autoría propia) _____	77
Figura 41 Maquetado _ Rastreador (autoría propia) _____	78

<i>Figura 42 Maquetado_ Posición (autoría propia)</i>	79
<i>Figura 43 Maquetado_ Usuario (autoría propia)</i>	80
<i>Figura 44 Maquetado_ Marca (autoría propia)</i>	81
<i>Figura 45 Maquetado_ Departamento (autoría propia)</i>	82
<i>Figura 46 Maquetado_ Unidad Nueva (autoría propia)</i>	83
<i>Figura 47 Diagrama Esquemático del Circuito de comunicación OBD-II-GPS-SMS (autoría propia)</i>	85
<i>Figura 48 Componentes físicos conectados (fotografía por el autor)</i>	86
<i>Figura 49 Código para configurar el módulo BT-HC05</i>	87
<i>Figura 50 Instrucciones en la consola de Arduino para configurar el Bluetooth</i>	87
<i>Figura 51 Disposición del Sistema de comunicación (autoría propia)</i>	88
<i>Figura 52 Sistema de comunicación, envío de datos. (autoría propia)</i>	89
<i>Figura 53 Petición y adquisición de datos in-situ (fotografía por el autor)</i>	89
<i>Figura 54 Pantalla de Login ( propiedad del autor)</i>	90
<i>Figura 55 Listado de unidades ( propiedad del autor)</i>	91
<i>Figura 56 Listado de datos (propiedad del autor)</i>	92
<i>Figura 57 Listado seguros de unidad. (propiedad del autor)</i>	90
<i>Figura 58 Rastreador de la unidad. (propiedad del autor)</i>	91
<i>Figura 59 Edición de usuarios (propiedad del autor)</i>	92
<i>Figura 60 Histórico de ubicaciones geodésicas. (propiedad del autor)</i>	92



2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

# CAPÍTULO 1

# INTRODUCCIÓN

## 1. Introducción

Actualmente la implementación de las Tecnologías de la Información y Comunicación, desde finales del siglo pasado han logrado un impacto de escala mundial, en todos los ámbitos privados o públicos. El mundo globalizado de hoy requiere el uso de las TIC, estas mismas afectan en casi todos los aspectos de la vida económica. El aumento acelerado de la productividad y el crecimiento mercantil de los años 90 está estrechamente relacionado con el avance de las TIC en el sector de los programas informáticos y de tecnologías de control. El aumento de la productividad inducida por las TIC es una fuente de creación de empleo. El relanzamiento del crecimiento de la producción ha dado impulso al empleo como en Europa y en el continente americano. "Las TIC han demostrado ser uno de los sectores de mayor crecimiento de la economía mexicana, y aunque se encuentra rezagado en el desarrollo de las TICs, en contraste con algunas de las economías de América Latina, analistas de la industria prevén un repunte del 15% para los siguientes años" (Orozco,2014).

Recabar y distribuir información ya no es una tarea que corresponde solamente a países con tecnología de punta. Prueba de ello es que actualmente México ocupa el tercer lugar a nivel mundial en exportación del sector de Tecnologías de la Información y Comunicación, de acuerdo con datos de la Secretaría de Economía. (Orozco, 2014). México se ha logrado posicionar como uno de los más importantes exportadores en el mundo y en telecomunicaciones el volumen de servicios continúa en un constante crecimiento. Por otra parte una de las metas del gobierno del Estado de México consiste en avanzar hacia una plataforma digital que permita a los funcionarios públicos consolidar, en un conjunto de información único, los datos necesarios para brindar mejores servicios, así como poner a disposición de la población servicios por medios

electrónicos accesibles (portales transaccionales), con esto la función pública se vuelve más eficiente en términos de tiempo, servicio y capacidad de respuesta, al tiempo que mejora el acceso a la transparencia, elemento fundamental para la evaluación de los logros.

## **1.2 Planteamiento del problema**

En el municipio de Huixquilucan en el departamento de administración, actualmente se lleva un registro de unidades vehiculares de forma manual elaborado en una bitácora escrita en libreta, la cual está conformada por los siguientes campos: auto, modelo, año y placa. Por lo cual es información vulnerable, debido a que no hay control de las personas que tienen acceso a dicha información lo que sugiere que cualquier persona podría alterar dicha información. A demás, no se le da un seguimiento oportuno a las fechas de verificación u otros servicios que se les tiene que dar. Así mismo no se tiene un control del combustible que se usa, los kilómetros recorridos entre otros parámetros como el trazado de rutas realizadas por el personal a cargo de las unidades. La seguridad de la información es vital hoy en día y se debe tener en cuenta siempre los riesgos que se tienen al no tener la información bajo la custodia correspondiente.

Los automóviles son de carácter oficial también llamadas flotillas, actualmente no se tiene ningún sistema ni forma de gestionarlos, o de administrarlos, es por eso que se plantea el modelado de este sistema en donde se pueda tener la información disponible, ordenada e inventariada correctamente, contemplando de manera eficiente la seguridad de la información almacenada, por medio de roles y encriptación de la información.

### **1.3 Hipótesis**

Al modelar un sistema de rastreo haciendo uso de un GPS y un dispositivo que permita monitorear los datos vehiculares de la flotilla correspondiente al municipio de Huixquilucan, se mejorará la organización, se reducirán los costos de gasolina y tratar de evitar malas prácticas.

### **1.4 Justificación**

En el municipio de Huixquilucan considerado como uno de los pioneros en el uso de las tecnologías al implementar cámaras de seguridad vía satélite en todo el municipio y usar botones de pánico, programa conocido como Sistema Municipal de Tecnología Policiaca. Esta localidad es de los pioneros en digitalizar las licencias de funcionamiento SARE, lo que conlleva a las personas a ya no tener que presentarse a las oficinas públicas para tramitar su licencia, con esto se ponen a la vanguardia de la innovación y reducen los tiempos de entrega, no se debe olvidar que un punto importante es la reducción de la corrupción.

Dentro del mismo desarrollo tecnológico se propone un modelo de sistema que permita la gestión del parque vehicular del ayuntamiento. El modelo que se plantea ayudará por medio de un seguimiento en tiempo real de las unidades con esto se reducirá el mal uso de las unidades, ya que se planea el rastreo en todo momento de las mismas, una aplicación que permita llevar el control del uso de la gasolina y rendimiento de la unidad. Se gestionarán sus mantenimientos para evitar futuras fallas que conlleven un mayor gasto, se tendrá en cuenta las renovaciones de los seguros de los vehículos y el pago de las tenencias, con esto se busca reducir al menos en un 70% el mal uso de los recursos públicos. Entendiendo como mal uso, desde la forma de conducción hasta el robo de gasolina y/o dinero.

## 1.5 Objetivos generales y específicos

### **General:**

Diseñar y modelar un sistema de rastreo y comunicación de datos en el municipio de Huixquilucan que ayudará a tener un mejor control en las unidades, evitando el desvío de fondos y malas prácticas.

### **Específicos:**

1. Diseñar y programar el prototipo del circuito con el Sistema de Posicionamiento Global GPS.
2. Modelar el circuito para la adquisición de datos por OBD2
3. Realizar pruebas para validar el prototipo.

### **Alcances**

El sistema se comprenderá por el modelado y elaboración de un prototipo para que se puedan hacer pruebas y validación para la implementación futura de dicho sistema. De la forma que los entregables se comprenden de la siguiente manera:

1. Esquemático del circuito del GPS y del OBD II
2. Diseño del PCB para prototipo del GPS y OBD II
3. Tesis
4. Propuestas de mejora.

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

## 1.6 Cronograma de actividades

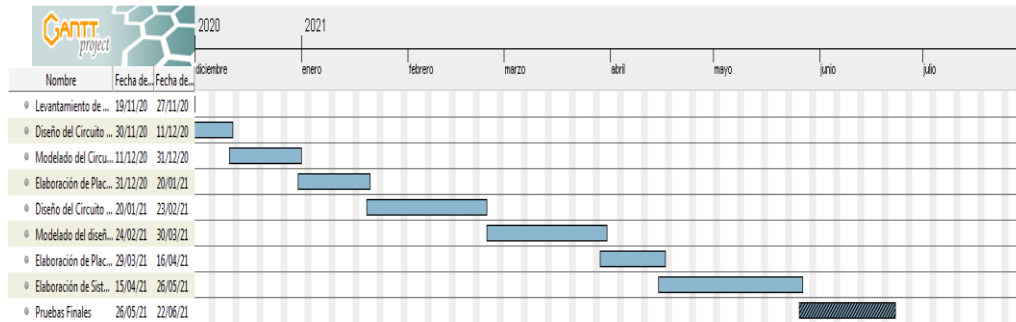


Figura 1 Cronograma de actividades - Desarrollo del proyecto (elaborado en Gantt Project)

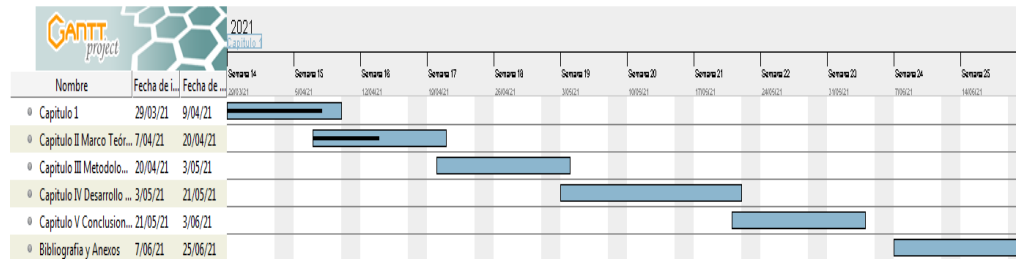


Figura 2 Cronograma de actividades- Desarrollo de Tesis (Elaborado en Gantt Project)

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

# **CAPITULO 2**

# **MARCO TEÓRICO**

## 2.1. Sistema de Posicionamiento Global

La geodesia o la determinación de puntos sobre la superficie terrestre comienza el 4 de Octubre de 1957 con el lanzamiento del primer satélite activo por la extinta Unión Soviética Rusa conocido popularmente como Sputnik I. Se pudo observar que determinando el corrimiento Doppler de las señales radiodifundidas por el Sputnik, desde estaciones de posición conocidas, era posible establecer la órbita del satélite. Esto permitió el planteo inverso, es decir, si la órbita era conocida previamente sería posible obtener la posición de un receptor en una ubicación cualquiera. Para ello habría que realizar observaciones durante varios pasos del satélite. Posteriormente en 1967 el sistema Transit se comenzó a utilizar en trabajos de tipo geodésico tales como mediciones de redes geodésicas extensas, determinación de parámetros entre sistemas geodésicos, y otras aplicaciones científicas y tecnológicas. (Huerta, E., Mangiaterra, A., & Noguera, G. (2005))

El Sistema de Posicionamiento Global (SPG) fue designado originalmente por el Departamento de Defensa de los Estados Unidos de América como "Sistema de Navegación mediante Tiempo y Distancia" (***Navigation Satellite Timing and Ranging***; NAVSTAR, por sus siglas en inglés) y fue diseñado para proveer capacidad de navegación a las fuerzas militares de los Estados Unidos en aire, mar y tierra bajo cualquier condición climática. Un beneficio colateral el Sistema ha sido su aplicación en el área civil; incluyendo recreación, control vehicular, navegación aérea y marítima y levantamientos topográficos. Aplicaciones más sofisticadas incluyen el monitoreo del movimiento de las placas de la corteza terrestre en áreas de alta sismicidad y la agricultura de precisión.



La operatividad del sistema no implica un compromiso legal del gobierno de Estados Unidos. Por lo tanto la Agencia Cartográfica del Departamento de Defensa, NIMA (National Imagery and Mapping Agency) puede modificar sin previo aviso su funcionamiento alterando, por ejemplo, el denominado mensaje de navegación (en el que está incluida información esencial para el cálculo como son las coordenadas de los satélites), limitando el acceso a uno o más componentes de la señal, alterando el estado de los relojes, degradando la precisión de las órbitas. En la actualidad, el uso civil de GPS ha sobrepasado largamente el uso militar, convirtiéndose de hecho en un servicio público de carácter mundial de enorme importancia y con innumerables aplicaciones. (Fallas, Jorge (2002)). Por lo anterior se puede conocer GPS proporciona el servicio de dos tipos de exactitud de servicios: normal y preciso. Los cuales se mencionan a continuación

#### A) Servicio de posicionamiento normal (SPS):

Éste es un servicio de posicionamiento disponible para todos los usuarios de GPS, como militares, privados y comerciales. El SPS proporciona una exactitud comprendida en 100 m horizontales, 156 m verticales y 185 m en 3 dimensiones, con una exactitud de transferencia de tiempo a la Red Mercator transversal universal (UTC, de Universal Transverse Mercator Grid) de menos de 340 nanosegundos. La exactitud del servicio SPS se puede degradar en épocas de emergencias nacionales. (Tomasi, 2003)

#### B) Servicio de posicionamiento preciso (PPS).

Éste es un servicio militar de gran exactitud para posicionamiento, velocidad y tiempo, disponible en forma mundial y continua a usuarios autorizados por el Departamento de Defensa de Estados Unidos. El equipo de usuario de Servicio de posicionamiento preciso (PPS) proporciona una exactitud predecible de 22 m horizontales, 27.7 m verticales y 34.5 m en 3 dimensiones (x,y,z), y una exactitud de transferencia de tiempo a UTC de menos de 200

nanosegundos. Sólo pueden recibir este servicio usuario autorizado con equipo y claves criptográficas, así como receptores con equipo especial. (Tomasi, 2003)

### **2.1.2 Arquitectura GPS**

El sistema GPS se compone en tres segmentos básicos: espacio, control y usuario. Los dos primeros de responsabilidad militar: segmento espacio, formado por 24 satélites GPS con una órbita de 26560 Km de radio y un periodo de 12 h.; segmento de control, que consta de cinco estaciones monitoras encargadas de mantener en órbita los satélites y supervisar su correcto funcionamiento, tres antenas terrestres que envían a los satélites las señales que deben transmitir y una estación experta de supervisión de todas las operaciones; y segmento usuario, formado por las antenas y los receptores pasivos situados en tierra. Los receptores, a partir de los mensajes que provienen de cada satélite visible, calculan distancias y proporcionan una estimación de posición y tiempo. (PozoRuz, 2016)

## **2.2 SISTEMA GSM**

A principios de la década de 1980, los sistemas telefónicos celulares analógicos tuvieron un periodo de rápido crecimiento en Europa Occidental, cada país desarrolló su propio sistema celular, incompatible con los de los demás, tanto en equipo como en el sistema de operación. La mayoría de los sistemas existentes funcionaba en distintas frecuencias, y todos eran analógicos. El principal requisito previo para un sistema de radio común, es el ancho de banda de radio. Esta condición había sido ya prevista unos pocos años antes, en 1978, cuando se decidió reservar la banda de frecuencia de  $900 \pm 25\text{MHz}$  para comunicaciones móviles. La estandarización en las telecomunicaciones es

fundamental. (Ruiz, 2018). En 1982, la Conferencia de Correos y Telégrafos Europeos (CEPT) formó un grupo de estudio, llamado "*Groupe Spécial Mobile*" (GSM) para estudiar y desarrollar un sistema telefónico móvil terrestre y público, paneuropeo. Su objetivo era el de estandarizar un único sistema de radiocomunicaciones a 900MHz.

En 1989 fue transferida la responsabilidad de la GSM al Instituto Europeo de Normas de Telecomunicación (ETSI, de ***European Telecommunication Standards Institute***) y en 1990 se publicó la fase I de las especificaciones GSM. Éste tuvo la ventaja de haberse diseñado desde cero, sin importar si era compatible con los sistemas telefónicos celulares analógicos existentes. (Tomasi, 2003). encuentro. En 1990, por requerimiento del Reino Unido, se añadió al grupo de estandarización la especificación de una versión de GSM a la banda de frecuencia de  $1800 \pm 75\text{MHz}$ . A esta variante se le llamó DCS1800 ("Digital Cellular System 1800"). El significado actual de las siglas GSM se ha cambiado y en la actualidad se hacen corresponder con "Global System for Mobile Communications". (Ruiz, 2018)

### **2.2.1 Funcionamiento de la red GSM**

La red GSM se divide en tres niveles principales los cuales se anuncian a continuación, la estación Móvil que está formada por el terminal telefónico, La estación base o transceptor y el sistema de red que permite el control de llamadas. A continuación, se describen cada uno de los niveles que componen la red GSM.

## **A) Estación Móvil**

La estación móvil esta formada por el terminal telefónico y una tarjeta inteligente SIM (Módulo de Identificación del Subscriptor), La tarjeta SIM almacena los datos del usuario, lo que permite a éste comunicarse independientemente del terminal que emplee (o teléfono celular).

## **B) Estación base**

Una estación base consiste en un radio transceptor de baja potencia, amplificadores de potencia, una unidad de control y otros componentes, que dependen de la configuración del sistema. La estación base está dividida en dos unidades la estación de transmisión y el controlador, que funcionan como control central para todos los usuarios dentro de una célula telefónica. La estación de transmisión se encarga de gestionar las comunicaciones por radio con las estaciones móviles. El controlador de la estación gestiona los recursos de radio de una o varias estaciones de transmisión, enlazándola al la Oficina de conmutación de telefonía móvil (MTSO) (Ruiz, 2018 ; Tomasi, 2003).

### C) Sistema de Red MTSO

La MTSO controla el procesamiento de llamadas, establecimiento de llamadas y terminación de llamadas, lo que incluye señalización, supervisión, conmutación y asignación de canales de radiofrecuencia (RF). La MTSO también proporciona un punto centralizado de administración y mantenimiento para toda la red, e interconecta con la red telefónica pública conmutada (PSTN, de *public switched telephone network*) por instalaciones de voz por cable, para cumplir con los servicios a suscriptores de teléfonos alámbricos convencionales. (Tomasi, 2003)

#### 2.2.2 Arquitectura del sistema GSM

La arquitectura del sistema GSM, mostrada en la Figura 3, consiste en tres subsistemas principales interconectados, que interaccionan entre sí y con los suscriptores a través de interfaces especificadas de red. Los tres subsistemas principales del GSM son subsistema de estación base

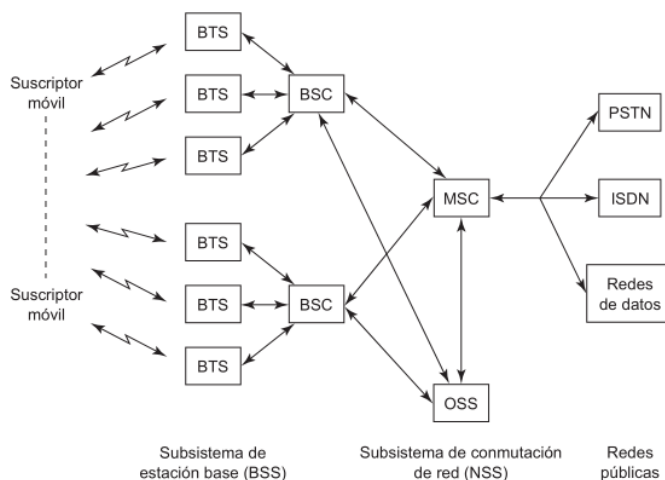


Figura 3 Arquitectura del sistema GSM (Tomasi,2003)

*Base Station Subsystem* (BSS), subsistema de red y conmutación, *Network and Switching Subsystem* (NSS) y subsistema de soporte operacional *Operational Support Subsystem* (OSS). Aunque la estación móvil es, técnicamente, otro subsistema, en general se le considera como parte de la estación base. Cada BSS consiste de muchos controladores de estación base *base station controller* (BSC) que se usan para conectar el MS con el NSS a través de uno o más MSC. El NSS administra las funciones de conmutación del sistema y permite que se comuniquen las MSC con otras redes telefónicas, como el servicio telefónico público y la ISDN. El OSS soporta la operación y el mantenimiento del sistema, y permite a los ingenieros vigilar, diagnosticar y localizar las fallas en cada aspecto de la red GSM. (Tomasi, 2003)

### **2.2.3 Módulo SIM 808**

El módulo SIM 808 es un módulo completo de Banda por Cuadratura GSM/GPRS que combina la tecnología de navegación satelital GPS. El diseño integra un GPRS y GPS en un paquete del tipo SMT el cual mejora el alcance de uso para diversas aplicaciones. Contiene una interfaz industrial estándar, así como un GPS universal, el cual permite rastrear activos variables sin problemas en cualquier ubicación y en cualquier momento con cobertura de señal. Abarca bandas de cuadratura en las siguientes frecuencias 850/900/1800/1900 MHz, lo que significa que es compatible con Chips o SIMS con tecnología 2G hasta 4G.

## 2.3 Sistema OBD-II

Es un sistema de diagnóstico integrado en la gestión del motor, ABS, etc. del vehículo, por lo tanto, es un programa instalado en las unidades de mando del motor. Su función es vigilar continuamente los componentes que intervienen en las emisiones de escape. En el momento en que se produce un fallo, el OBD lo detecta, se carga en la memoria y avisa al usuario mediante un testigo luminoso situado en el cuadro de instrumentos denominado *Malfunction Indicator Light* (MIL). El hecho de denominarse EOBD II es debido a que se trata de una adaptación para Europa del sistema implantado en Estados Unidos, además de tratarse de una segunda generación de sistemas de diagnóstico.

El OBD, por el hecho de vigilar continuamente las emisiones contaminantes, ha de tener bajo control no solo a los componentes, sino también el correcto desarrollo de las funciones existentes en el sistema de gestión del motor, por lo que se convierte en una excelente herramienta que debe facilitar la diagnosis de averías en los sistemas electrónicos del automóvil. La incorporación del sistema de diagnosis OBD viene impuesto por las directivas de la Unión Europea que pretenden minimizar y reducir la emisión de determinados gases de los automóviles y evitar la contaminación atmosférica para preservar el medio ambiente. Desde enero de 2000 que entró en vigor la Fase III se obliga al fabricante a incorporar un sistema de vigilancia de la contaminación provocada por el vehículo que informase al usuario de tal situación. Este sistema, encriptado, estandarizado para todos los fabricantes y que convive con el sistema de autodiagnosis propio de la marca, es el *European On Board Diagnosis* (EOBD) (Blasco, 2014)

## 2.4 Ingeniería de software

De acuerdo con (Sommerville,2011) la Ingeniería de software es una disciplina de la ingeniería la cual tiene como interés todos los aspectos de la producción de Software. No deberá confundirse con la ingeniería de sistemas la cual se dedica a todos los procesos del desarrollo de sistemas basados en computadoras, incluido el hardware, software e ingeniería de procesos. De acuerdo con el autor se estima que aproximadamente el 60% de los costos adjudicables al software son de desarrollo, mientras que el 40% es de prueba.

La ingeniería de software trata desde as primeras etapas de la especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación. El autor menciona dos conceptos importantes dentro de la ingeniería de software. La disciplina de ingeniería, que hace referencia a la teoría, métodos y herramientas adecuadas para lograr un objetivo. Y el segundo es todos los aspectos de la producción del software, en donde se manifiesta que no sólo los procesos técnicos del desarrollo son fundamentales en la ingeniería de software, sino también actividades como la administración del proyecto de software y el desarrollo de herramientas, así como métodos y teorías para apoyar la producción de software.



## ***i) Ingeniería de requerimientos***

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información. Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama ingeniería de requerimientos. El término «requerimiento» no se usa de manera continua en la industria del software.

En el otro extremo, consiste en una definición detallada y formal de una función del sistema. Algunos de los problemas que surgen durante el proceso de ingeniería de requerimientos son resultado del fracaso de hacer una separación clara entre esos diferentes niveles de descripción. En este texto se distinguen con el uso del término «requerimientos del usuario» para representar los requerimientos abstractos de alto nivel; y «requerimientos del sistema» para caracterizar la descripción detallada de lo que el sistema debe hacer.

Los requerimientos del usuario son enunciados, en un lenguaje natural junto con diagramas, acerca de qué servicios esperan los usuarios del sistema, y de las restricciones con las cuales éste debe operar. Los requerimientos del sistema son descripciones más detalladas de las funciones, los servicios y las restricciones operacionales del sistema de software (Sommerville,2011)

Frecuentemente, los requerimientos del sistema de software se clasifican como requerimientos funcionales o requerimientos no funcionales. Los requerimientos funcionales son aquellos que establecen los servicios que proveerá un sistema, las reacciones que deberá tener a ciertas entradas,

comportamientos específicos y en algunos casos explica lo que no deberá hacer el sistema. Por otro lado, los requerimientos no funcionales, Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema.

## ii) Metodologías Ágiles

Las metodologías ágiles se definen por ser flexibles, modificables a objetivos alcanzables. De acuerdo con (Pressman,2011) las metodologías ágiles se distinguen por tener una comunicación activa y continua entre los desarrolladores y los clientes. A demás hace mención a los 12 principios para lograr la agilidad, dentro de los cuales se encuentra, la entrega temprana y continua de software, adaptabilidad a cambios, en poco tiempo entregar avances que sean desarrollables, trabajo en conjunto del cliente con los desarrolladores, entre otros principios que son actualmente aplicados en las metodologías ágiles que se analizarán a continuación.

### **A) Programación Extrema(PE)**

De acuerdo con (Somerville, 2011) La programación extrema es una de las metodologías ágiles más utilizadas, es utilizada principalmente cuando no se tiene completamente claro el alcance del proyecto, por lo que se va adaptando según surjan las necesidades. En este tipo de programación las versiones actuales de un sistema pueden ser desarrolladas en un día por medio de la integración de diferentes programadores. El ciclo de liberación de la

programación extrema consta de seis pasos, primero se selecciona la historia de usuario para la liberación, después se desglosan las historias en tareas, posteriormente se planea la liberación, seguido se desarrolla, integra y se pone a prueba el software, después se hace la liberación de software y finalmente se evalúa el sistema (Somerville, 2011).

En la programación extrema, los requerimientos se expresan como escenarios que se implementan directamente como una serie de tareas. Los programadores trabajan en pares y antes de escribir el código desarrollan pruebas para cada tarea. Todas las pruebas deben ejecutarse con éxito una vez que el nuevo código se integre en el sistema. Entre las liberaciones del sistema existe un breve lapso. Este tipo de programación incluye algunas prácticas en las cuales se reflejan los fundamentos de la metodología ágil. Dichas practicas son la planeación incremental, liberaciones pequeñas, diseño simple, desarrollo de primera prueba, refactorización, programación en pares, propiedad colectiva, integración continua, ritmo sustentable y cliente en sitio es decir un representante que está de planta con los desarrolladores (Somerville, 2011).

## **B) Scrum**

Dicho nombre se deriva de una "jugada" dentro del deporte de Rugby que consiste en la recuperación rápida del juego posterior a una infracción menor. La primera vez que se hace uso de esta referencia de juego en el desarrollo de datos es en 1986, cuando Takeuchi y Nonaka utilizan el *Rugby Approach* con el cual definen un nuevo enfoque en el desarrollo de productos teniendo como objetivos el aumento de flexibilidad y rapidez, en conjunto con un equipo interdisciplinario y multifacético que son modificables. (Hirotaka Takeuchi, 2009)

El método de Scrum es un método ágil que ofrece un marco de referencia para la administración del proyecto. Se centra alrededor de un conjunto de sprints, que son periodos fijos cuando se desarrolla un incremento de sistema. La planeación se basa en priorizar un atraso de trabajo y seleccionar las tareas de importancia más alta para un sprint. (Somerville, 2011).

### **C) Desarrollo Adaptativo de Software (DAS)**

Esta metodología es propuesta por Jim Hightsmith como una técnica para construir software y sistemas complejos. Se basa en la colaboración humana y la organización propia del equipo. El DAS tiene un ciclo de vida que incorpora tres fases. La especulación en la cual se inicia el proyecto y se realiza la planeación de ciclo adaptativo, en donde se establecen fechas de entrega o descripciones del cliente. La colaboración se refiere al trabajo en equipo sin excluir la individualidad. Finalmente, el aprendizaje, los miembros de un equipo DAS desarrollan componentes integrantes de un ciclo adaptativo, y los equipos aprenden de tres maneras: Grupos enfocados, revisiones técnicas formales y post mortem. La filosofía del DAS es meritaria sin importar el modelo de proceso empleado.

### **D) Kanban**

Kanban se basa en una idea muy simple: el trabajo en curso (Work In Progress, WIP) ) implica que se genera una señal visual para indicar que hay nuevos bloques de trabajo que pueden ser comenzados porque el trabajo en curso actual no alcanza el máximo acordado. Kanban, por su naturaleza de Sistema Pull ("arrastre", "tirar"; la producción se realiza cuando el cliente retira un elemento terminado), también propicia diferir el compromiso tanto en la

priorización del trabajo nuevo como en la entrega del trabajo existente. Típicamente, los equipos llegarán a un acuerdo sobre la cadencia de reuniones de priorización con el bloque funcional que les precede en el proceso para decidir en qué deben trabajar a continuación. (David. J, 2010)

## **E) Agile Inception**

Agile inception (También conocida como Inception Deck o simplemente Inception) tiene como pilar un proceso de diez ejercicios, los cuales ayudan a los equipos a llegar rápidamente a un acuerdo en común. Para que una sesión de Inception sea exitosa, es muy importante que haya un facilitador neutral presente, que haga el papel de moderador para la conversación, sin tratar de influenciar las decisiones que el equipo haga. La obtención de requisitos se realiza a través de preguntas. Las respuestas están registradas y en artefactos. Estos artefactos luego se interpretan de diferentes maneras dependiendo de quién lea el documento, ya que estos artefactos están escritos en lenguaje natural.

Algunos enfoques han combinado la obtención tradicional con modelos de procesos comerciales (BP, por sus siglas en inglés) para mejorar la obtención de requisitos. Estos trabajos han evidenciado que un proceso de análisis de software se puede mejorar cuando el resultado final de la elicitación se representa mediante BP, ya que la representación y la interpretación de la información son únicas, eliminando así la ambigüedad del lenguaje natural. (Pastrana.M, 2018)

## **F) Design Spring (DS)**

Es un compendio de herramientas en el Design Thinking y destaca por ser un proceso que produce buenos resultados, tiene las mismas características que el Design Thinking, pero en un espacio de tiempo que oscila en una semana. Sus beneficios van más allá de que el producto se adhiera a las expectativas del usuario, pero hacen posible que el aprendizaje del equipo sobre el problema empresarial a resolver se profundice en poco tiempo, además de permitir que se produzcan menos cambios de requisitos durante el desarrollo del proyecto. La propuesta de DS es acelerar el aprendizaje en equipo para desarrollar la solución, produciendo así algo que estará muy cerca de lo que realmente quiere el usuario final y minimizando así los cambios que serían necesarios durante un proceso de desarrollo ágil. (Ferreira, V et al, 2019)

La metodología Scrum es utilizada en la elaboración del presente proyecto, debido a que esta metodología es un marco de trabajo diseñado para lograr la colaboración eficaz de equipos en proyectos, que emplea un conjunto de reglas e instrumentos, también define roles que generan la estructura necesaria para su correcto funcionamiento.

## 2.5 Lenguajes de Programación

En el siglo XIX Charles Babbage crea la máquina analítica que funcionaba con tarjetas perforadas, siendo esta la primera computadora e inicio de los lenguajes de programación. Se puede definir el lenguaje de programación como una notación o interfaz que permite comunicarle a una computadora lo que deseamos que haga, dicho de otra forma, se puede definir como un sistema de notación para describir computaciones en una forma legible tanto para la máquina como para el ser humano.

El lenguaje de programación se distingue por tener diferentes generaciones, en donde cada generación trata de resolver los problemas que se encontraban en cierta época. Así la primera generación pertenecía sólo al lenguaje máquina en secuencias de ceros y unos. La segunda generación al lenguaje ensamblador. C, FORTRAN o Java pertenecen a la 3ra generación. La cuarta generación son lenguajes con propósitos específicos como SQL, ABAP, etc. Finalmente, la quinta generación se enfoca en la Inteligencia Artificial para resolver problemas concretos con restricciones determinadas ejemplo son Prolog y Mercury. (Francisco Bellas, 2016)

Se puede definir al lenguaje de programación como al conjunto de instrucciones que se utilizan para construir un programa. Existe una amplia variedad de formas y tipos de lenguajes de programación. Desde el ensamblador hasta lenguajes de nivel alto (J.Bronson, 2007). Actualmente los lenguajes de programación también pueden ser clasificados en programación de fondo y programación visual, backend y frontend respectivamente.

### **2.5.1 Front End**

Se De acuerdo con el Glosario de Comunicación Estratégica, el termino *Front-End* o interfaz usuario se relaciona con todo lo que ven las personas, esto incluye el diseño y ciertos lenguajes. Se puede decir que el Front-end es la capa que se encuentra por encima del Back-end, ya que comprende menús desplegables, imágenes, íconos, colores, elementos gráficos, animaciones, aportando con ello una mejora a la experiencia de navegación en el lado delantero o frontal. Algunos de los programas más utilizados son:

**A) Hojas de Estilo en Cascada** (del inglés Cascading Style Sheets) o **CSS** es el formato recomendado para las páginas escritas con HTML, El uso de este estándar web permite el ahorro de tiempo. La sintaxis consta de dos partes: un selector y una declaración. Los estilos CSS fueron añadidos por el World Wide Web Consortium (W3C) en HTML4 para resolver los graves problemas que afectaban a la calidad de las páginas escritas en HTML. Las páginas que utilizan CSS además de ser más fáciles de escribir, también son más ligeras y aparecen rápidamente en el navegador (Durango A, 2015).



**B) HTML** (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) Este lenguaje no se considera propiamente como un lenguaje de programación sino más bien un lenguaje de marcado o marcación de hipertexto. El lenguaje de marcado hace uso de secuencias de comandos para realizar procesos, es decir, utiliza diferentes etiquetas para identificar el contenido. HTML es un lenguaje abstracto el cual usan las aplicaciones para representar documentos además este puede ser transmitido de manera relativamente sencilla por algún medio como el internet. (Jason.N , 2019)

**C) JavaScript.** HTML funciona como contenedor al bloque de instrucciones de JavaScript. Toda vez que se carga la página en HTML, el navegador ejecuta las instrucciones de JavaScript permitiendo de esta manera enriquecerla con nuevas funcionalidades (E. Gutierrez, 2009) . Aunque Javascript al igual que el lenguaje de programación Java son creación de Oracle, los dos lenguajes de programación tienen sintaxis, semántica y usos muy diferentes. Para crear un programa utilizando JavaScript se requiere sólo de un editor de textos o de html. La página puede ser creada con sólo el código JavaScript o también incorporando un programa en una página en lenguaje HTML. JavaScript es un lenguaje de programación interpretado, es decir que, en lugar de ser compilado, cada línea de programación es leída, traducida al lenguaje máquina y ejecutada (L.Fernandez, 2014).

En el presente proyecto se hace uso de JavaScript y de CSS en la parte de front-end debido a que ofrecen una gran flexibilidad para el manejo de distintos datos, esto se debe porque JavaScript y CSS han sido pensados para diseñadores de páginas web más que para programadores, por lo tanto, hay muchas

sentencias que permiten el ahorro de líneas de programación. JavaScript y CSS facilitan la programación frontend además de que cuenta con múltiples recursos para la depuración de código.

## 2.5.2 Frameworks

De acuerdo con (Miguel. T, 2016) Un framework se puede definir como un esquema que se establece para desarrollar y organizar un determinado software. Incluyendo módulos concretos cuyo objetivo es facilitar el trabajo del profesional. Algunos de los frameworks más utilizados son:

1. React;
2. Bootstrap;
3. AngularJS;

**React** permite crear interfaces de usuario interactivas de forma sencilla. Los componentes de React implementan un método llamado *render()* que recibe datos de entrada y retorna qué mostrar. Se puede acceder a los datos de entrada que se pasan al componente mediante *render()* a través de *this.props*. Además de obtener datos de entrada (a los que accedes a través de *this.props*), un componente puede tener datos en su estado interno (a los que accedes a través de *this.state*). Cuando los datos del estado de un componente cambian, se vuelve a invocar *render* con los nuevos valores en *this.state*. (React, Comunidad, 2021)

Haciendo uso de *props* y *state*, se puede integrar todo en una pequeña aplicación de tareas pendientes. Este ejemplo usa *state* para tener un control de la lista actual de objetos así como el texto que el usuario ha ingresado. A pesar de que los manejadores de eventos parecen ser renderizados en línea, serán recolectados e implementados usando delegación de eventos. Finalmente React

permite interactuar con otras bibliotecas y frameworks. Este ejemplo usa **remarkable**, una biblioteca externa de *Markdown*, que convierte los valores de <textarea> en tiempo real (React, Comunidad, 2021).

**SpringBoot** El framework utilizado en el proyecto es SpringBoot. Este framework está basado en *Spring Core*. La característica principal es que el desarrollador se enfoque más en la solución que en las configuraciones de framework por lo que esta parte se obvia (Oscar, 2018).

La configuración de este framework cuenta con un complejo módulo que auto configura todos los aspectos de la aplicación para poder ejecutar la aplicación, sin la necesidad de definir algo. La resolución de dependencias solo requiere de la determinación del tipo de proyecto que se estará utilizando y este framework se encargará de resolver todas las librerías/dependencias para que la aplicación funcione. En el despliegue se puede ejecutar como una aplicación Stand-alone, aunque es posible ejecutar aplicaciones web, ya que es posible desplegar las aplicaciones mediante un servidor web integrado, como es el caso de Tomcat, Jetty o Undertow.

Para las métricas se cuenta con servicios que permite consultar el estado de salud de la aplicación por defecto, accediendo saber si la aplicación está activada o desactivada, la memoria utilizada y disponible, número y detalle de los Bean's creado por la aplicación, controles para el prendido y apagado, entre otros. También este framework es extensible lo cual permite la creación de complementos, los cuales ayudan a que la comunidad de Software Libre cree nuevos módulos que faciliten aún más el desarrollo. (Blancarte, 2018)

### **2.5.3 Backend**

El backend también conocido como modo administrador se encarga de todos los procesos necesarios para que la web funcione de forma correcta. Es el área administrable de un sitio web en la que se crean y gestionan los contenidos. Solo es accesible para el administrador o los usuarios autorizados. El desarrollo de backend está a cargo de diseñadores web con experiencia en el desarrollo web, por ejemplo el lenguaje Ruby, PHP, CSS, HTML, Java entre otros.

#### **A) Ruby**

Lenguaje de programación con tipos dinámicos que es particularmente adecuado para programación de aplicaciones Web. Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre. (Ruby,2020)

#### **B) Java**

Java es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán, probablemente, a menos que tengan Java instalado y cada día se crean más. Java es rápido, seguro y fiable.

Desde ordenadores portátiles hasta centros de datos, desde consolas para juegos hasta computadoras avanzadas, desde teléfonos móviles hasta Internet, Java está en todas partes, si es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados (Oracle, 2021)

Siendo este último utilizado en el proyecto para que se pueda desarrollar la parte del backend. Se hace uso del lenguaje Java debido a la simplicidad de código, a la robustez en cuanto a seguridad, a que es escalable y también por la experiencia de trabajar con el mismo.

#### **2.5.4 Sistema Gestor de Base de Datos**

Una base de datos es un conjunto de datos relacionados y organizados con determinada estructura. De acuerdo a la organización se pueden distinguir diferentes modelos de bases de datos como el relacional, jerárquico o en red. El modelo con mayor extensión es el relacional. Con el cual para su manipulación y gestión surgen los llamados Sistemas Gestores de Bases de Datos (SGBD). Luego entonces el SGBD, se define como una aplicación o programa que permite a los usuarios definir, crear y mantener bases de datos, proporcionando acceso controlado a las mismas. El SGBD es una herramienta que sirve como interfaz entre el usuario y las bases de datos (Ibañez, 2015). Dentro de los sistemas gestores más utilizados se tienen:

## a) SQL Server

SQL Server es un sistema de gestión de base de datos relacional, desarrollado como un servidor que da servicio a otras aplicaciones de software que pueden funcionar ya sea en el mismo ordenador o en otro ordenador a través de una red. Los servidores SQL Server suelen presentar como principal característica una alta disponibilidad al permitir un gran tiempo de actividad y una conmutación más rápida. Todo esto sin sacrificar los recursos de memoria del sistema. Gracias a las funciones de memoria integradas directamente en los motores de base de datos SQL Server y de análisis, mejora la flexibilidad y se facilita el uso. Pero quizá su característica más destacada es que ofrece una solución robusta que se integra a la perfección con la familia de servidores Microsoft Server. (Parada, 2019)

## b) Oracle

Oracle es una herramienta para la gestión de bases de datos, usada principalmente por grandes empresas permitiéndoles controlar y gestionar una gran cantidad de contenidos desde un solo archivo, con ello permite la reducción de costos y minimiza el riesgo de pérdida de información. Usa un modelo relacional, es decir que los usuarios visualizan los datos en tablas con el formato filas/columnas. También es una herramienta de administración gráfica intuitiva y cómoda de utilizar, contiene un control de acceso con tecnología avanzada para vigilar la entrada a los datos. Usa lenguaje de diseño PL/SQL que permite implementar diseños "activos", que se pueden adaptar a las necesidades cambiantes de negocio. (Netec, 2021).

### c) My SQL

My SQL es un sistema gestor de bases de datos relacionales, multiplataforma y multi sistemas. My SQL dispone de dos programas principales, el servidor o motor y el cliente o monitor. El primero de ellos se encarga de estar a la espera de peticiones recibidas por el cliente, mientras que el segundo actúa como interfaz con el usuario. MySQL utiliza el lenguaje SQL o lenguaje de consulta estructurada. Es un sistema cliente/servidor permitiendo tabajar como servidor multiusuario y de subprocesamiento múltiple. (Ángel Cobo, 2015)

Dispone de un sistema sencillo de ayuda en línea, y de un monitor que permite realizar todas las operaciones desde la línea de comandos del sistema, sin la necesidad de ningún tipo de interface de usuario gráfica. Para utilizar My SQL es necesario arrancar el programa servidor, este estará funcionando en segundo plano como proceso o servicio. No tiene una interfaz y sólo está a espera de que alguien se conecte y envíe una solicitud (Ángel Cobo, 2015).

El sistema gestor utilizado en este proyecto es el MySQL, debido a que es multiplataforma, fácil de programar e interfaz amigable. A demás que no se tiene que estar pagando una renta como otros sistemas gestores. También se opta por utilizar MySQL debido a que no se tiene una base de datos muy grande.

### **2.5.4.1 Normalización de Base de Datos**

La normalización es el proceso de organización de datos como la creación de tablas, así como establecer relaciones entre dichas tablas de acuerdo con reglas diseñadas tanto para proteger los datos como para que la base de datos sea más flexible al eliminar la redundancia y la dependencia incoherente.

La normalización es el proceso de organización de datos en una base de datos. Esto incluye crear tablas y establecer relaciones entre dichas tablas de acuerdo con reglas diseñadas tanto para proteger los datos como para que la base de datos sea más flexible al eliminar la redundancia y la dependencia incoherente.

Los datos redundantes desperdician espacio en disco y crean problemas de mantenimiento. Si se deben cambiar los datos que existen en más de un lugar, los datos deben cambiarse exactamente del mismo modo en todas las ubicaciones. Un cambio de dirección de cliente es mucho más fácil de implementar si los datos se almacenan solo en la tabla Clientes y en ninguna otra parte de la base de datos.

### **2.5.5 Estándares de Calidad y Seguridad**

JSON Web Token, o JWT , es un estándar para pasar solicitud de forma segura en entornos con limitaciones de espacio. Ha encontrado su camino en los principales frameworks web3 web34. La simplicidad, la compacidad y la facilidad de uso son características clave de su arquitectura. Aunque todavía se utilizan



sistemas mucho más complejos, los JWT tienen una amplia gama de aplicaciones. (E. Peyrott, 2018)

El grupo JSON *Object Signing and Encryption* (JOSE) se formó en el año 2019. El objetivo del grupo era “estandarizar el mecanismo de protección de la integridad (firma y MAC) y el cifrado, así como el formato de claves e identificadores de algoritmos para respaldar la interoperabilidad de servicios de seguridad para protocolos que utilizan JSON”. Para el año 2013 estaban disponibles una serie de borradores, incluido un recetario con diferentes ejemplos del uso de las ideas producidas por el grupo. Estos borradores se convertirían más tarde en los RFC de JWT, JWS, JWE, JWK y JWA. A partir del año 2016, estos RFC se encuentran en el proceso de seguimiento de estándares y no se han encontrado erratas en ellos. (E. Peyrott, 2018)

Todos los JWT se construyen a partir de tres elementos diferentes: el encabezado, la carga útil y los datos de firma / cifrado. Los dos primeros elementos son objetos JSON de cierta estructura. El tercero depende del algoritmo utilizado para firmar o cifrar y, en el caso de JWT no cifrados, se omite. Los JWT se pueden codificar en una representación compacta conocida como serialización compacta JWS / JWE. Las especificaciones JWS y JWE definen un tercer formato de serialización conocido como serialización JSON, una representación no compacta que permite múltiples firmas o destinatarios en el mismo JWT.

La serialización compacta es una codificación segura para URL Base64 de los bytes UTF-8 de los dos primeros elementos JSON (el encabezado y la carga útil) y los datos, según sea necesario, para la firma o el cifrado (que no es un objeto JSON en sí). Estos datos también están codificados en Base64-URL. Estos tres elementos están separados por puntos (“.”). (Peyrott. E, 2018)

### **2.5.6 Calidad de Software**

Según Oscar et al.,( 2015), la calidad del software es un conjunto de cualidades que lo caracterizan y determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, precisión, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad.

La calidad del software se puede medir y varía de un sistema a otro o de un programa a otro. El software desarrollado para controlar las naves espaciales debería ser fiable y "perfecto". El software de un solo uso no reclama el mismo nivel de calidad. Los productos de software han existido durante mucho tiempo (10 años o más) pero necesitan confiabilidad, facilidad de mantenimiento y flexibilidad para reducir los costos de mantenimiento y mejorar a lo largo de su vida útil.

La calidad del software se puede medir una vez fabricado el producto. Sin embargo, cuando se detecta un problema causado por una falla de diseño, puede resultar muy costoso, por lo que es imperativo considerar tanto el logro de la calidad como su gestión en todas las etapas del ciclo de vida del software.

## a) Métricas de proceso de software

La eficacia de un proceso de software se mide indirectamente, para ello es necesario el uso de métricas basadas en los resultados que se derivan del proceso. Por ejemplo, medidas tomadas previas al lanzamiento del software, o los defectos detectados y reportados por los usuarios finales, entre otros. Dentro de estas métricas existen las métricas del proyecto las cuales tienen características tácticas esto es un gerente de proyecto y un equipo de software emplean estas métricas y los indicadores obtenidos de estas métricas sirven para para adaptar el flujo de trabajo del proyecto y las actividades técnicas. Las métricas del proyecto son consolidadas para crear métricas de proceso que a su vez puedan ser públicas para la organización de software como un todo. (Pressman, R, 2015)

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

# CAPITULO 3

# DESARROLLO

### 3.1 Diagramas para la obtención de las coordenadas del sistema de posicionamiento global

Un diagrama de bloques es la representación sistemática de un proceso dentro de un sistema, en este caso, un sistema electrónico, en cada bloque se representa una parte esencial del sistema, indicando el funcionamiento generalizado del mismo.

En la Figura 4 se muestra el diagrama de bloques que describe el proceso de comunicación para la obtención de las coordenadas del GPS. Primero se debe establecer la velocidad de comunicación por lo tanto se establece una velocidad de 9600 baudios para la consola que permite mostrar lo que está recibiendo y enviando el SIM808 con ayuda del Arduino como interfaz. Además se establece la velocidad de comunicación Arduino-SIM808 a 9600 baudios, esta velocidad es elegida ya que la tarjeta SIM808 la tiene por default, por lo tanto permite evitar conflictos en la comunicación.

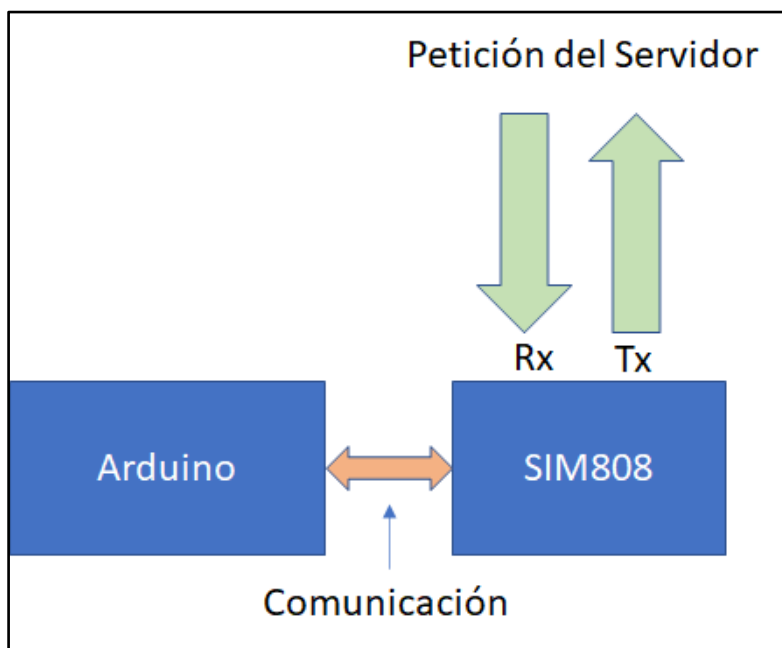


Figura 4 Diagrama de bloques obtención GPS (Autoría propia)

Toda vez que la codificación ha sido cargada correctamente al Arduino, es por medio de los pines 10 y 11 que tienen asignado funcionar como Transmisión y Recepción respectivamente, referente la comunicación que existe entre el Arduino y la SIM808. La SIM808 queda activa esperando una petición de localización para poder enviar los datos vía mensaje de texto. Esto es posible gracias a que la SIM808 además de contar con una antena GPS que funciona con una frecuencia UHF de 1575.42MHz, tiene también una SIM 3G/GSM la cual tiene la función de utilizar una red móvil disponible para poder realizar la comunicación de manera efectiva. Cuando el módulo SIM808 recibe la petición, es a través del pin 10 que le indica al microprocesador Arduino que tiene una petición, este microprocesador compara un arreglo de letras o palabras que deben coincidir al 100%, si la petición contiene el arreglo esperado entonces se le responde por medio del pin 11 que mande la información del posicionamiento al número que realizó la petición, de lo contrario, sólo ignora la petición y vuelve a su estado inicial esperando una nueva petición.

```

gpetm1 Arduino 1.8.17 Hourly Build 2021/09/06 02:33
Archivo Editor Programa Herramientas Ayuda
gpetm1$
13 char phone[16]; // se declara una variable tipo char para el número telefonico que hace la petición
14 char dateTime[24]; // se declara una variable tipo char para la fecha de recepción de petición
15
16 #define PIN_TX 10 //pin de salida de transmisión del arduino
17 #define PIN_RX 11 // pin de entrada del arduino
18 SoftwareSerial mySerial(PIN_TX, PIN_RX);
19 DFRobot_SIM808 sim808(mySerial);//Conexión RX, TX, PWR,
20 SoftwareSerial mySerialEIM(2, 3); // RX, TX
21 EIM327 myEIM327;
22 void getGPS();
23
24 void setup()
25 {
26 // mySerialEIM.begin(115200);
27 //myEIM327.begin(mySerialEIM, true, 2000);
28 mySerial.begin(9600);
29 Serial.begin(9600);
30
31 //***** Inicializa el modulo SIM808 *****
32 while (!sim808.init())
33 {
34 Serial.print("SIM808 Error en la conexión\r\n");
35 delay(1000);
36 }
37 delay(3000);
38
39 Serial.println("SIM Init success");
40
41 Serial.println("Conexión Exitosa, en espera de un mensaje");
42
43 /*Enciende el GPS*/
44 if (sim808.attachGPS())
45 Serial.println("GPS funcionando ");

```

Figura 5 Declaración de puertos de comunicación Arduino- SIM808, (autoría propia)

En esta Figura 5 se muestran los puertos declarados para la comunicación entre el Arduino y la SIM808.

```

58 sim808.readSMS(messageIndex, message, MESSAGE_LENGTH, phone, datetime);
59
60 //*****In order not to full SIM Memory, is better to delete it*****
61 sim808.deleteSMS(messageIndex);
62 Serial.print("From number: ");
63 Serial.println(phone);
64 Serial.print("Datetime: ");
65 Serial.println(datetime);
66 Serial.print("Received Message: ");
67 Serial.println(message);
68
69 String stringOne = String(message); // convertimos el mensaje char en un arreglo
70 String stringTwo = String("datos"); // declaramos la palabra datos en un arreglo
71 //usamos una condición para comparar los valores de entrada ( mensaje que llega, con la palabra declarada)
72 if (stringOne.equals(stringTwo)) {
73     getGPS();
74
75     Serial.println("Enviando datos ...");
76     delay(1000);
77
78     Serial.print(lat);
79     Serial.print(",");
80     Serial.print(lon);
81     Serial.println(phone);
82
83     sim808.sendSMS(phone, MESSAGE);
84     // sim808.sendSMS(phone,lon);
85 }
86 Serial.println("En espera de petición");
87 }
88 //***** Detecting unread SMS *****
89 // messageIndex = sim808.isSMSunread();
90

```

Figura 6 Comparación de arreglos en la petición recibida (autoría Propia)

En esta parte del código se puede observar que es necesario que la palabra coincida a la perfección sin discriminar mayúsculas, minúsculas o acentos, para que pueda responder enviando datos (ver Figura 6).

```

102 Serial.print("/");
103 Serial.print(sim808.GPSdata.month);
104 Serial.print("/");
105 Serial.print(sim808.GPSdata.day);
106 Serial.print(" ");
107 Serial.print(sim808.GPSdata.hour);
108 Serial.print(":");
109 Serial.print(sim808.GPSdata.minute);
110 Serial.print(":");
111 Serial.print(sim808.GPSdata.second);
112 Serial.print(".");
113 Serial.println(sim808.GPSdata.centisecond);
114 Serial.print("latitude :");
115 Serial.println(sim808.GPSdata.lat);
116 Serial.print("longitude :");
117 Serial.println(sim808.GPSdata.lon);
118 Serial.print("speed_kph :");
119 Serial.println(sim808.GPSdata.speed_kph);
120 Serial.print("heading :");
121 Serial.println(sim808.GPSdata.heading);
122 Serial.println();
123
124 float la = sim808.GPSdata.lat;
125 float lo = sim808.GPSdata.lon;
126 float ws = sim808.GPSdata.speed_kph;
127
128 dtostrf(la, 4, 6, lat); //put float value of la into char array of lat. 4 = number of digits before decimal sign. 6 = number of digits after the decimal sign.
129 dtostrf(lo, 4, 6, lon); //put float value of lo into char array of lon
130 dtostrf(ws, 6, 2, wspeed); //put float value of ws into char array of wspeed
131
132 sprintf(MESSAGE, "https://www.google.com/maps?q=%s,%s\nVelocidad : %s kph\n. ", lat, lon, wspeed);
133
134

```

Figura 7 Texto enviado por medio de la SIM808 (autoría propia)

Finalmente, si hay coincidencia entre la petición que dispone de una mensaje tipo cadena de caracteres y el arreglo esperado, el cual está definido por la palabra 'datos' entonces el mensaje será enviado con la localización así como un url que proporciona la ubicación directamente en Google Maps (ver Figura 7).

### 3.2 Diagrama para la obtención de datos del OBDII

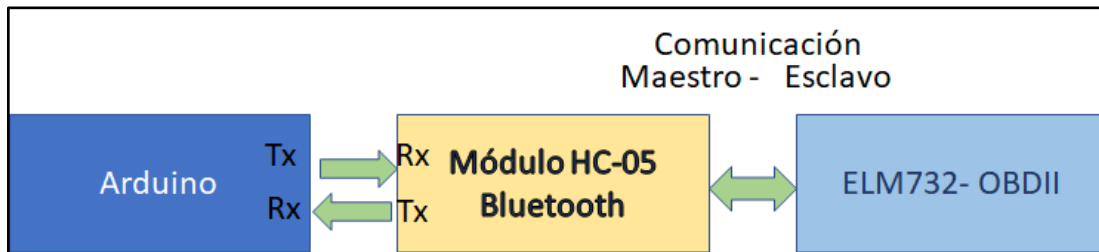


Figura 8 Diagrama de bloques para la obtención de los datos obtenidos del protocolo OBDII (autoría propia)

En la Figura 8 se muestra el diagrama de bloques para la obtención de datos que provienen directamente del automóvil por medio del protocolo OBDII. Primeramente es necesario configurar el módulo Bluetooth para que esté configurado como maestro y puede tener una conexión exitosa con el módulo ELM327- OBDII el cual extrae todos los parámetros del puerto CAN del automóvil y los convierte a datos legibles. Es importante mencionar que la velocidad de comunicación del módulo HC-05 Bluetooth con el ELM327-OBDII se ajusta a 115200 baudios, ya que funciona mejor y sin problemas de conectividad a velocidades superiores. Toda vez que el módulo bluetooth está configurado y con una conexión estable, por medio del Arduino en el cual se configuran los puertos 2 y 3 como transmisor y receptor, se le manda la petición al módulo ELM327 por medio de comandos PID los cuales sirven para que se les otorgue



un parámetro específico del OBDII. Una vez que estos parámetros son recibidos por el ELM327, son regresados al Arduino en donde se pueden leer por medio de la consola y a demás pueden ser enviados por mensaje de texto al solicitante.

```

gpselm
Archivo Editar Programa Herramientas Ayuda

gpselm
10 char lon[12]; //se declara una variable tipo char para la longitud
11 char vsped[12]; // se declara una variable tipo char para la velocidad
12
13 char phone[16]; // se declara una variable tipo char para el número telefonico que hace la petición
14 char datecine[24]; // se declara una variable tipo char para la fecha de recepción de petición
15
16 #define PIN_TX 10 //pin de salida de transmisión del arduino
17 #define PIN_RX 11 // pin de entrada del arduino
18 SoftwareSerial mySerial(PIN_TX, PIN_RX);
19 #ifndef SIM808 SIM808(mySerial) //Conexión RX, TX, PWR.
20 SoftwareSerial mySerialELM(2, 3); // RX, TX
21 ELM327 myELM327;
22 void getGPS();
23
24 void setup()
25 {
26 // mySerialELM.begin(115200);
27 //myELM327.begin(mySerialELM, true, 2000);
28 mySerial.begin(9600);
29 Serial.begin(9600);
30
31 //***** Inicializa el modulo SIM808 *****
32 while (!sim808.init())
33 {
34 Serial.print("Sim808 Error en la conexión\r\n");
35 delay(1000);
36 }
37 delay(3000);
38 Serial.println("SIM Init success");
39
40 Serial.println("Conexión Exitosa, en espera de un mensaje");
41
42

```

Figura 9 Declaración de puertos de comunicación (Autoría propia)

Los puertos 2 y 3 son declarados para que exista una comunicación entre el Bluetooth y el Arduino (ver Figura 9).

```

gpselm$
16 #define PIN_TX 10 //pin de salida de transmisión del arduino
17 #define PIN_RX 11 // pin de entrada del arduino
18 SoftwareSerial mySerial(PIN_TX, PIN_RX);
19 #ifndef SIM808 SIM808(mySerial) //Conexión RX, TX, PWR.
20 SoftwareSerial mySerialELM(2, 3); // RX, TX
21 ELM327 myELM327;
22 void getGPS();
23
24 void setup()
25 {
26 mySerialELM.begin(115200);
27 myELM327.begin(mySerialELM, true, 2000);
28 mySerial.begin(9600);
29 Serial.begin(9600);
30
31 //***** Inicializa el modulo SIM808 *****
32 while (!sim808.init())
33 {
34 Serial.print("Sim808 Error en la conexión\r\n");
35 delay(1000);
36 }
37 delay(3000);
38 Serial.println("SIM Init success");
39
40 Serial.println("Conexión Exitosa, en espera de un mensaje");
41
42
43 /*Enciende el GPS*/
44 if (! sim808.attachGPS())
45 Serial.println("GPS funcionando ");
46 else
47 Serial.println("Falla en el GPS revise conexión");
48 }

```

Figura 10 comunicación HC-05 y ELM327 a 115200 baudios (autoría propia)

Para una correcta comunicación es necesario que la velocidad entre el Bluetooth y el ELM327 este a 115200 baudios (ver Figura10).

```

43 /*Enciende el GPS*/
44 if ( sim008.attachGPS())
45   Serial.println("GPS funcionando ");
46 else
47   Serial.println("Falla en el GPS revise conexión");
48 }
49 void loop()
50 {
51   float tempRPM = myELM327.rpm();
52   float fuelRate = myELM327.fuelRate();
53   uint32_t kph = myELM327.kph();
54   uint16_t kmr1 = myELM327.distanceCodesCleared();
55
56   if (myELM327.status == ELM_SUCCESS)
57   {
58     km = (uint32_t)kph;
59     kmr = (uint16_t)kmr1;
60     Serial.print("RPM: "); Serial.println(tempRPM);
61     Serial.print("RENDIMIENTO: "); Serial.println(fuelRate);
62     Serial.print("KILOMETROS POR HORA: "); Serial.println(km);
63     Serial.print("KILOMETROS TOTALES REC: "); Serial.println(kmr);
64     delay(1000);
65   }
66   messageIndex = sim008.isSMSunread();
67   Serial.print("messageIndex: ");
68   Serial.println(messageIndex);
69
70   //***** At least, there is one UNREAD SMS *****
71   if (messageIndex > 0) {
72     sim008.readSMS(messageIndex, message, MESSAGE_LENGTH, phone, datetime);
73
74     //*****In order not to full SIM Memory, is better to delete it*****
75     sim008.deleteSMS(messageIndex);
  
```

Figura 11 parámetros obtenidos del OBD II (autoría propia)

Los parámetros obtenidos del OBDII son las revoluciones por minuto RPM, Rendimiento de Gasolina, Kilómetros por hora instantáneo, Kilómetros totales recorridos, Todos estos datos son enviados junto con los datos de ubicación por mensaje de texto (ver Figura 11).

### 3.3 Diagramas de casos de uso

Los diagramas mostrados a continuación son una forma de diagrama de comportamiento basado en el lenguaje de modelado unificado, con la cual son representados los diferentes procesos dentro del sistema de gestión vehicular. Esta parte del modelado es la que se encuentra en la adquisición de datos hacia la base de datos, la cual estará generando un histórico por cada unidad de interés.

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

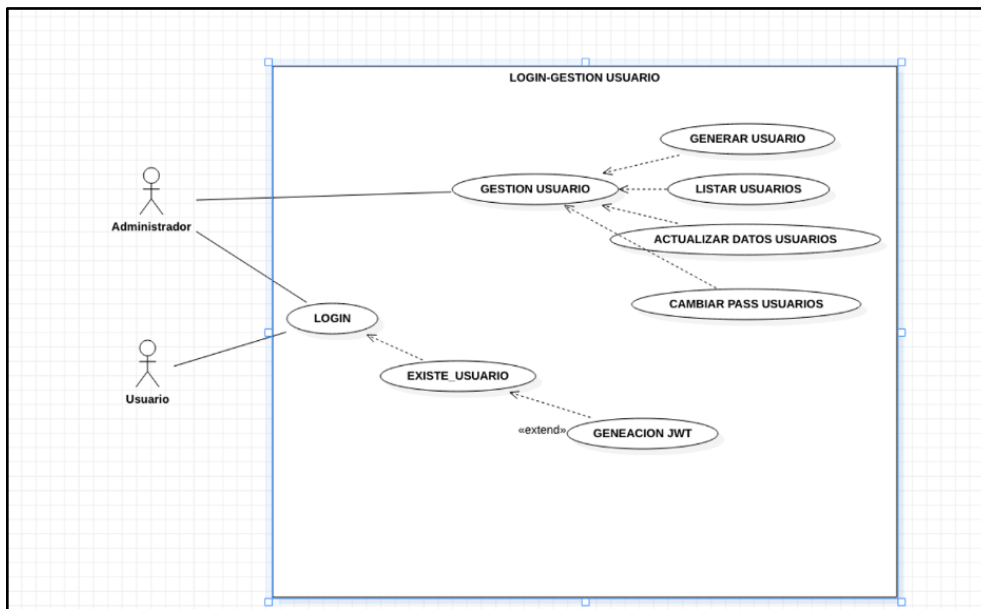


Figura 12 Diagramas de casos de uso\_ Login (autoría propia)

En la Figura 12 Diagramas de casos de uso\_ Login (autoría propia) se muestra el diagrama de casos de uso para el inicio de sesión, en donde se puede observar que existen dos roles, el de administrador y el de usuario.

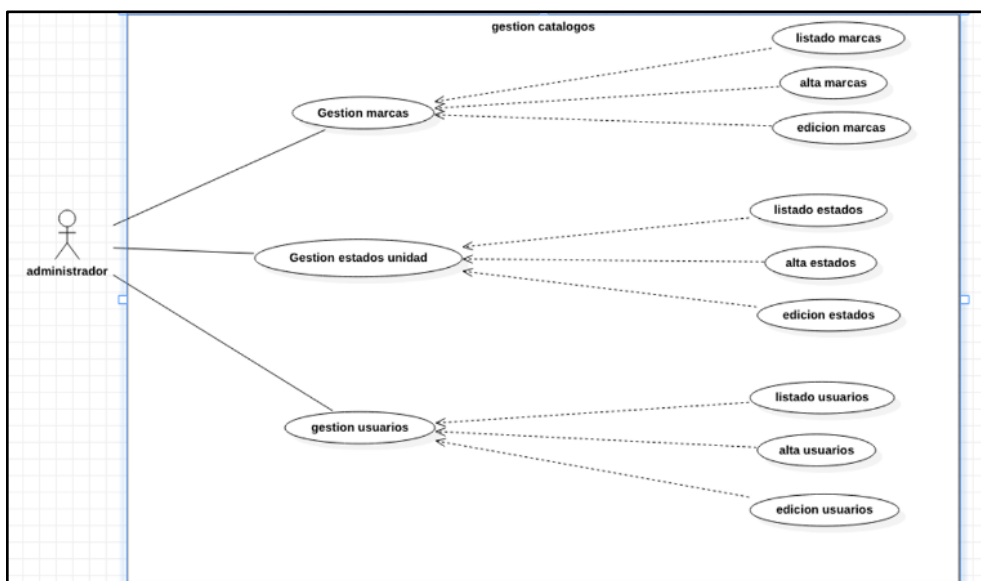


Figura 13 Diagrama de casos de uso\_ Gestión Catálogos Usuarios (autoría propia)

Como administrador se dispone la gestión de diferentes catálogos, como son la gestión de marcas, estados de unidad y de usuarios, en cada una de las gestiones se puede ver, crear y editar sus componentes (ver Figura 13).

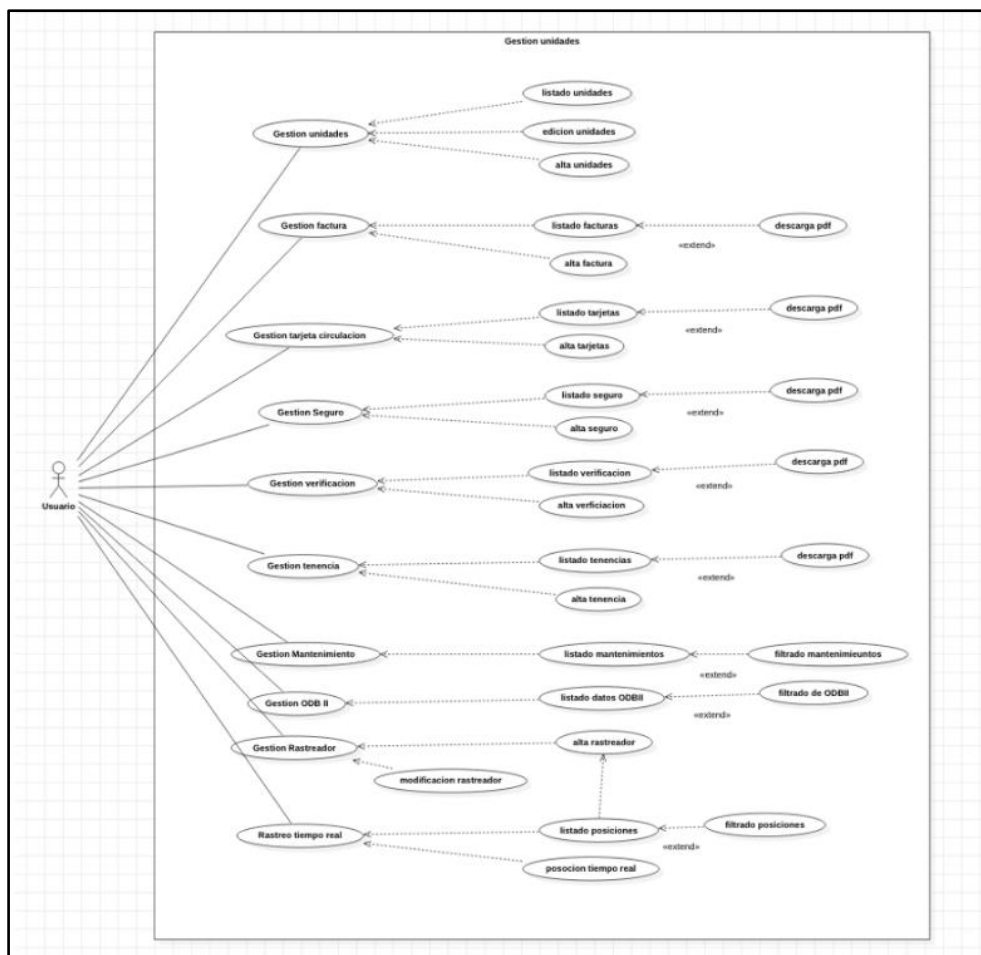


Figura 14 Diagrama de casos de uso\_ Gestion de unidades (autoría propia)

El usuario por su parte dispone del seguimiento que se le esta dando a la unidad, por ejemplo el estado del seguro, mantenimiento, km recorridos, alertas del OBD-II, etc. Con el fin de que el usuario esté al pendiente de su unidad y hacer uso de buenas practicas (ver Figura 14).

### 3.4 Diagrama entidad-relación

En la Figura 15 se muestra el diagrama de entidad relación, en donde se puede observar las tablas que conforman la base de datos, las llaves primarias, llaves foráneas, tipo de datos, longitudes permitidas y como es que se relacionan las tablas dentro de la base de datos.

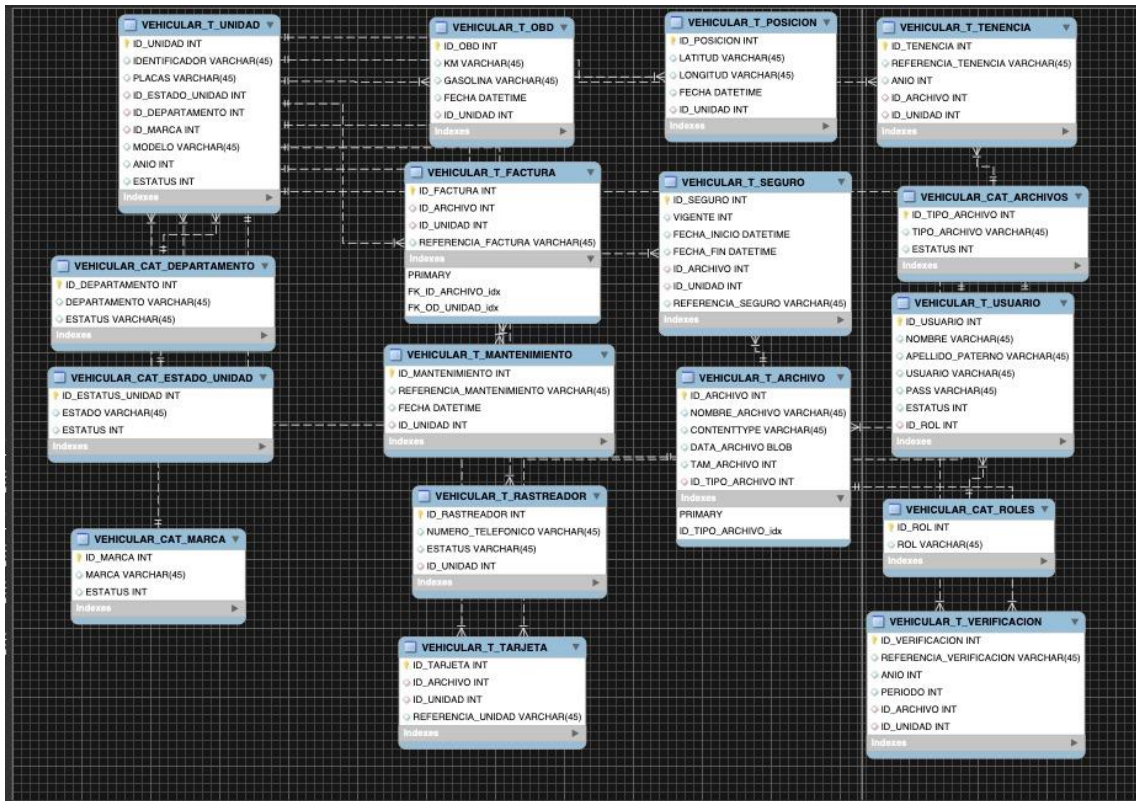


Figura 15 Diagrama Entidad- Relación (autoría propia)

### 3.5 Diagramas de clase

#### A) Clase Token Service

Esta Clase ayuda a generar un token JWT mismo que será necesario para cualquier petición hacia nuestro back, tomando en cuenta el token se decidirá que tipo de usuario admirador u operativo, el menú de opciones es dinámico tomando en cuenta el rol (ver Figura 16).

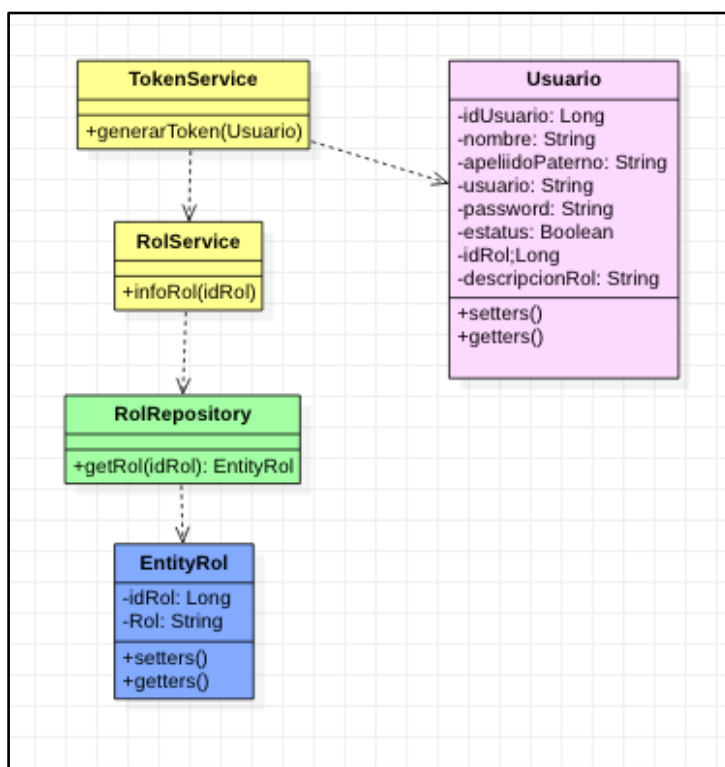


Figura 16 Diagrama de clases\_ Token (autoría propia)

## B) Clase Unidad Service

Esta Clase permite generar una lista de las unidades disponibles, generar unidades nuevas y modificar unidades (ver Figura 17).

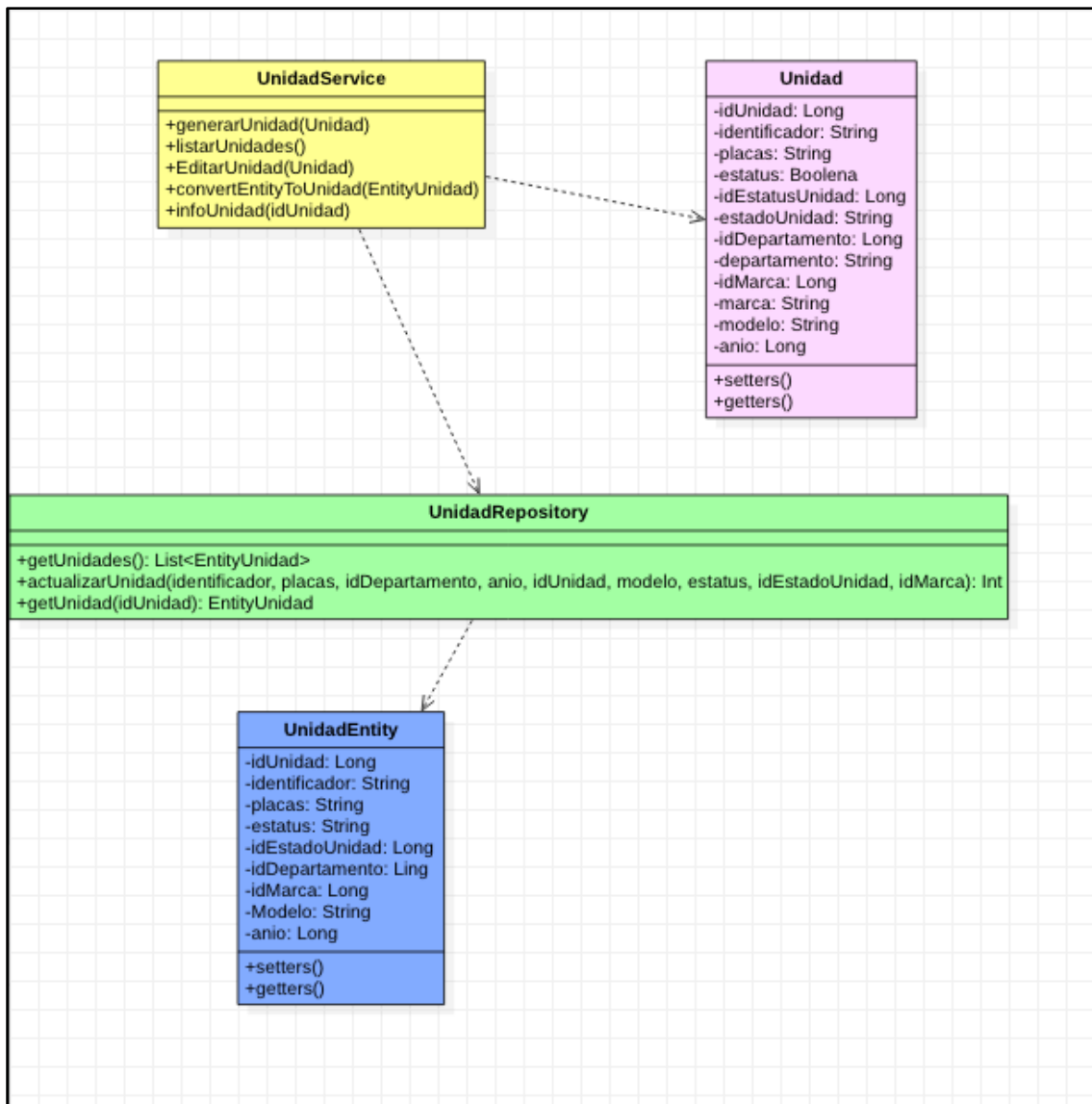


Figura 17 Diagrama de clase \_Unidad (autoría propia)

### C) Clase FacturaService

Con esta clase se obtiene el listado de la factura o las factura de unidad, también cargar nuevas facturas y descargar el archivo PDF del mismo (ver Figura 18)

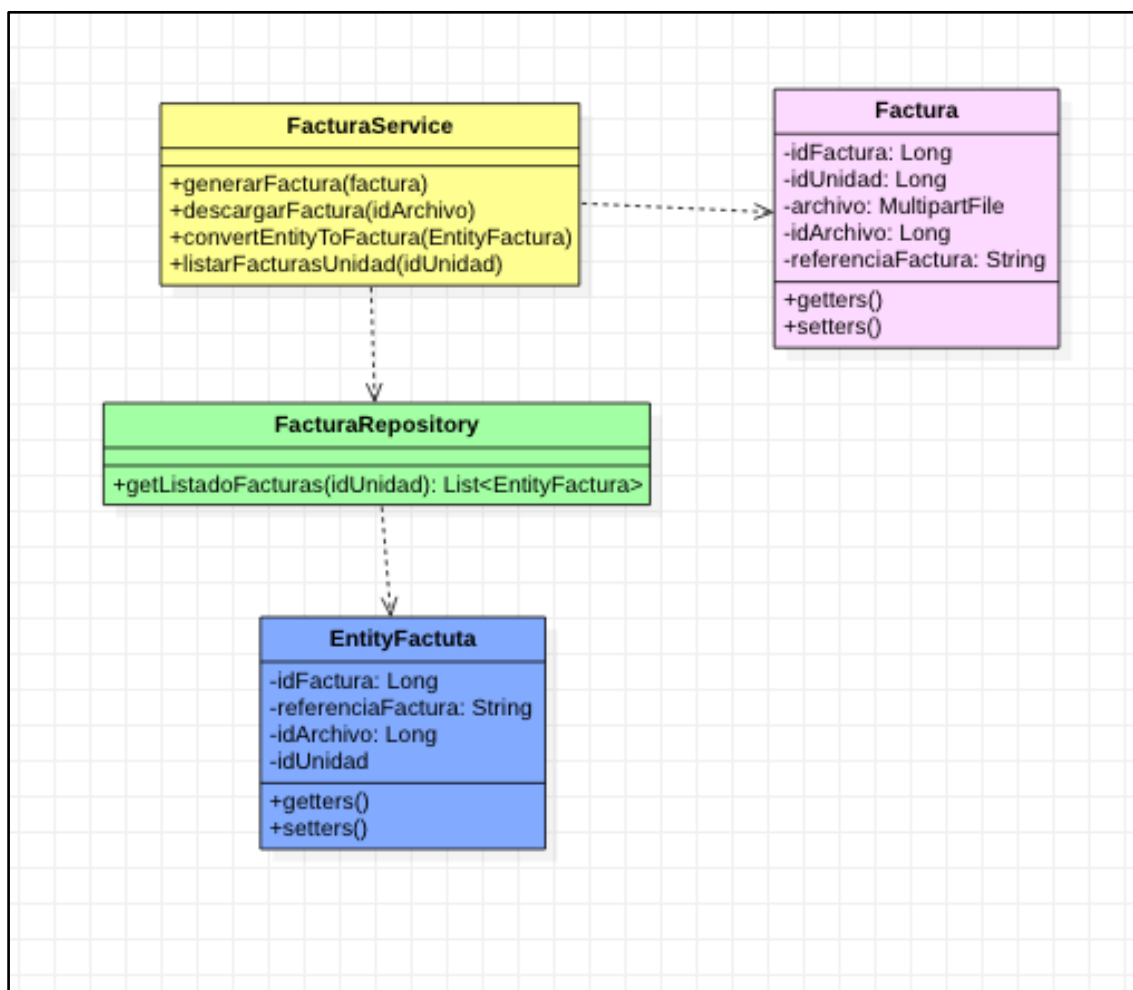


Figura 18 Diagrama de clase \_ Factura (autoría propia)



## D) Clase CirculacionService

Esta clase ayuda a obtener el listado de las tarjetas de circulación de unidad, cargar nuevas y descargar el archivo PDF del mismo (ver Figura 19)

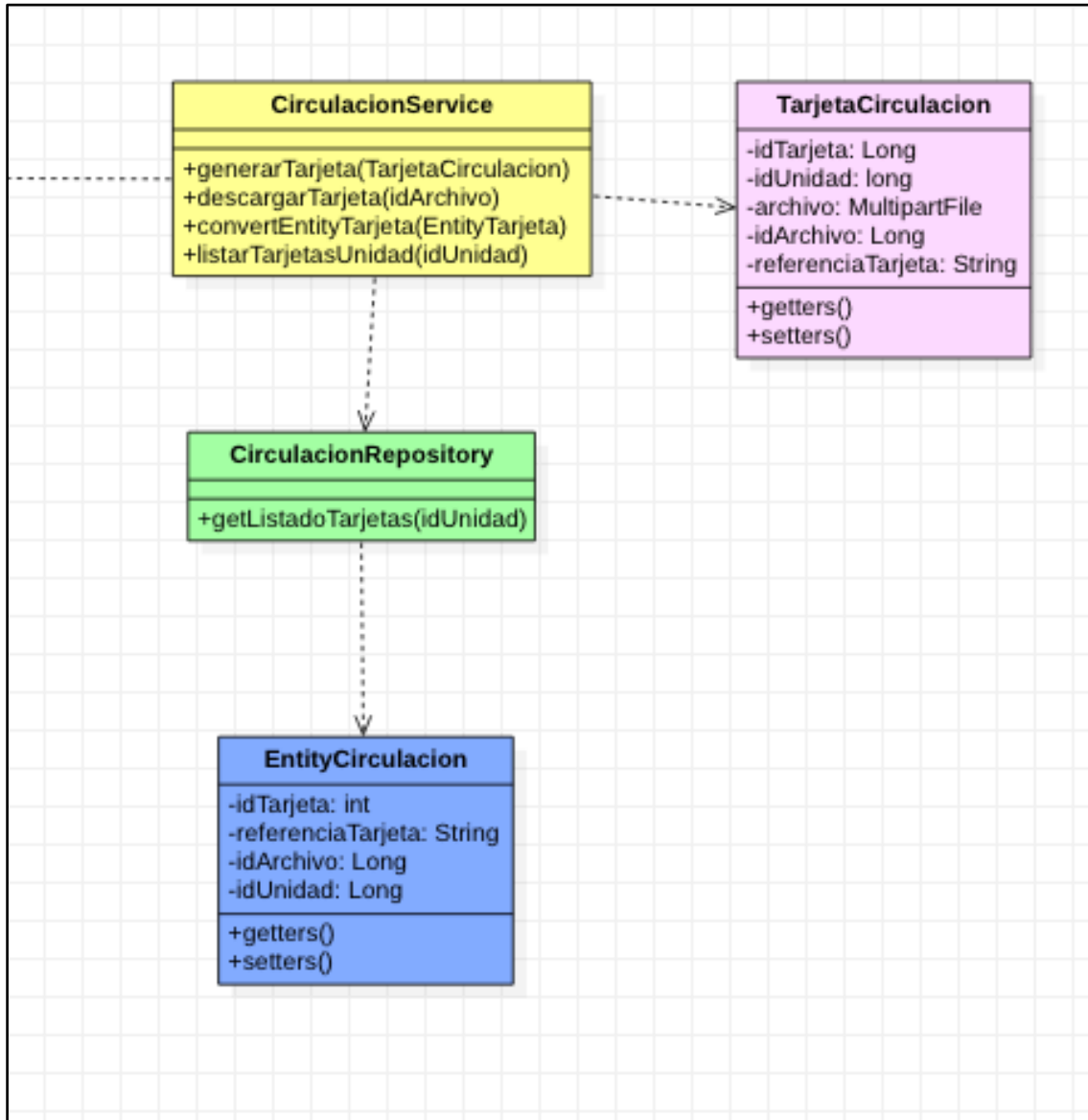


Figura 19 Diagrama de clase \_Circulación (autoría propia)

## E) Clase SeguroService

Esta clase ayuda para obtener el listado de los seguros de la unidad, cargar nuevos y descargar el archivo PDF del mismo (ver Figura 20)

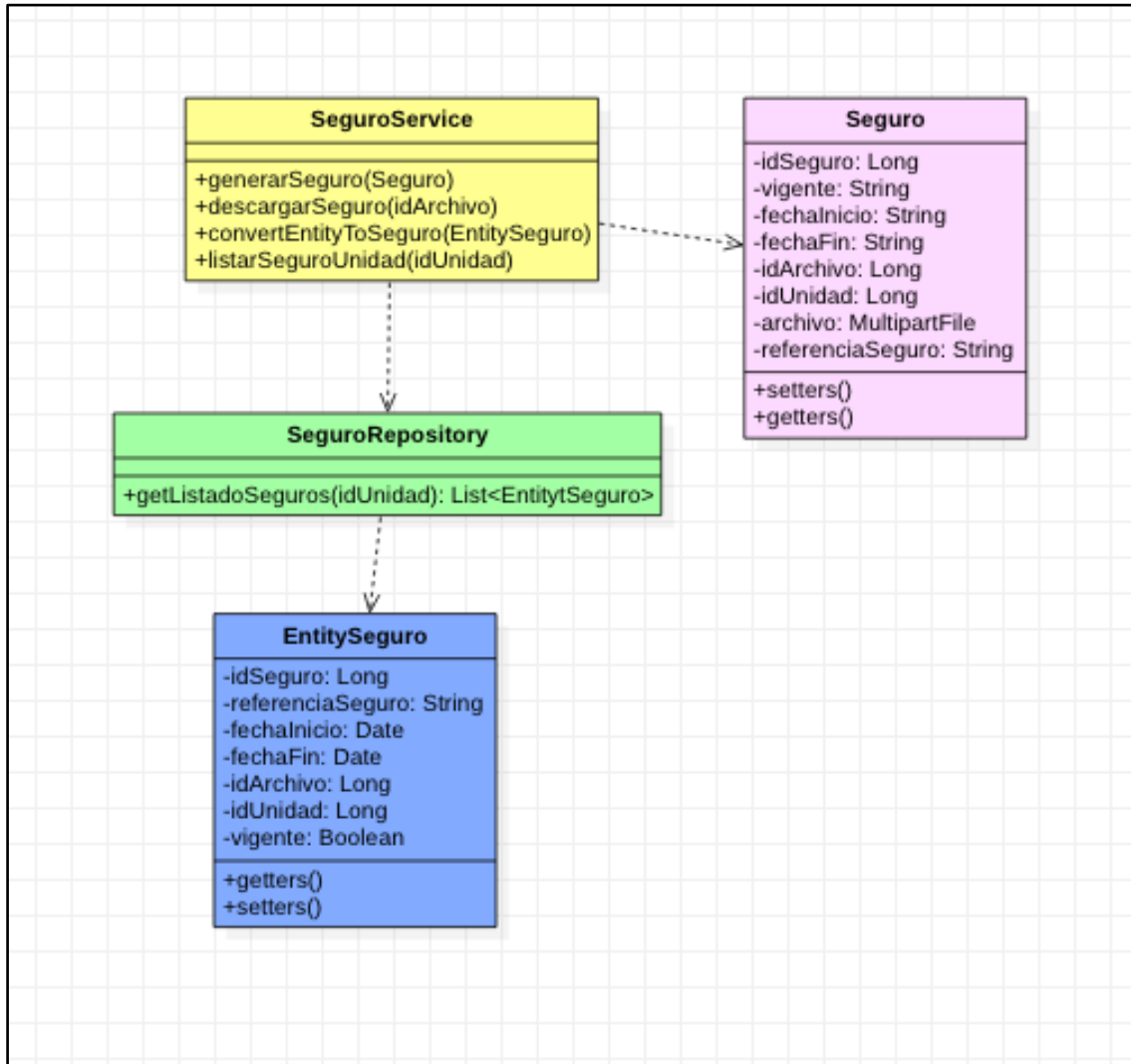


Figura 20 Diagrama de clase \_ Seguro (autoría propia)

## F) Clase Verificacion Service

Clase que ayudara a obtener el listado de las verificaciones de la unidad, cargar nuevos y descargar el archivo PDF del mismo (ver Figura 21) .

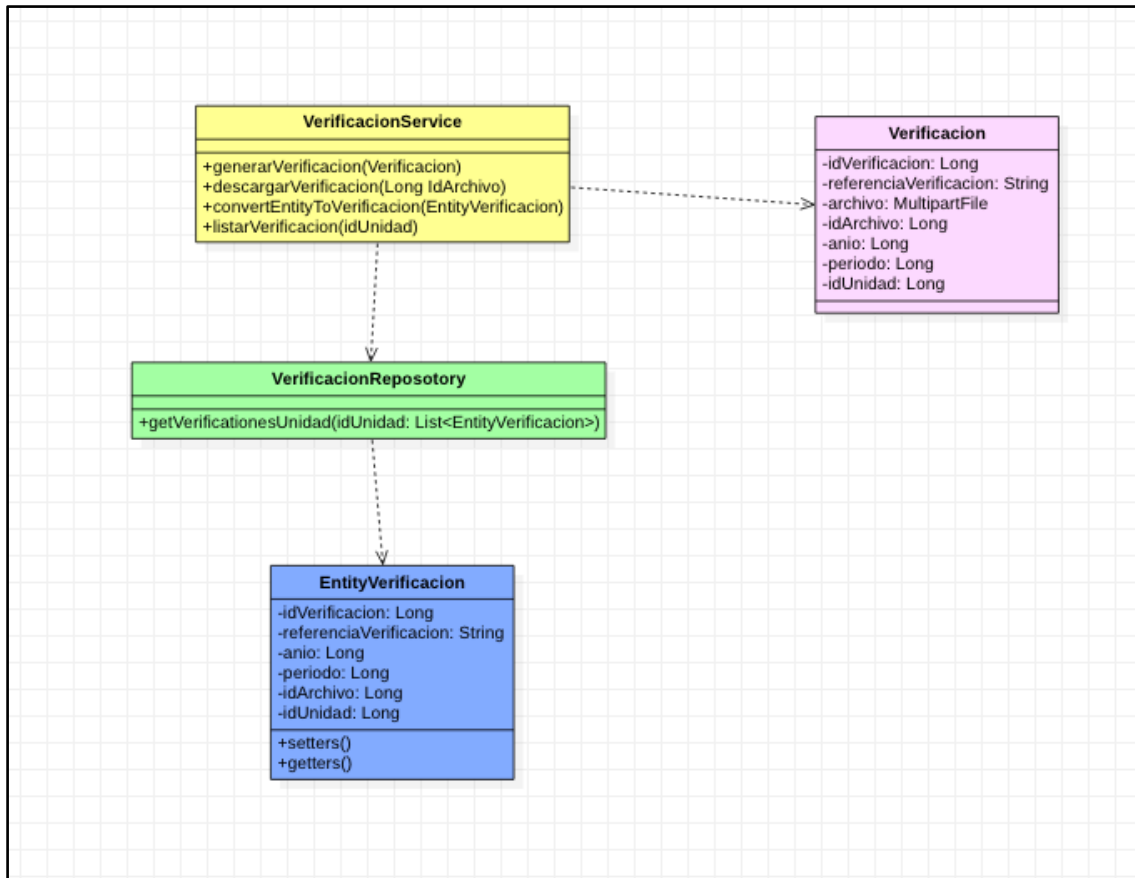


Figura 21 Diagrama de clase \_ Verificación (autoría propia)

## G) Clase TenenciaService

Clase que nos ayudara obtener el listado de las Tenencias de la unidad, cargar nuevos y descargar el archivo PDF del mismo (ver Figura 22)

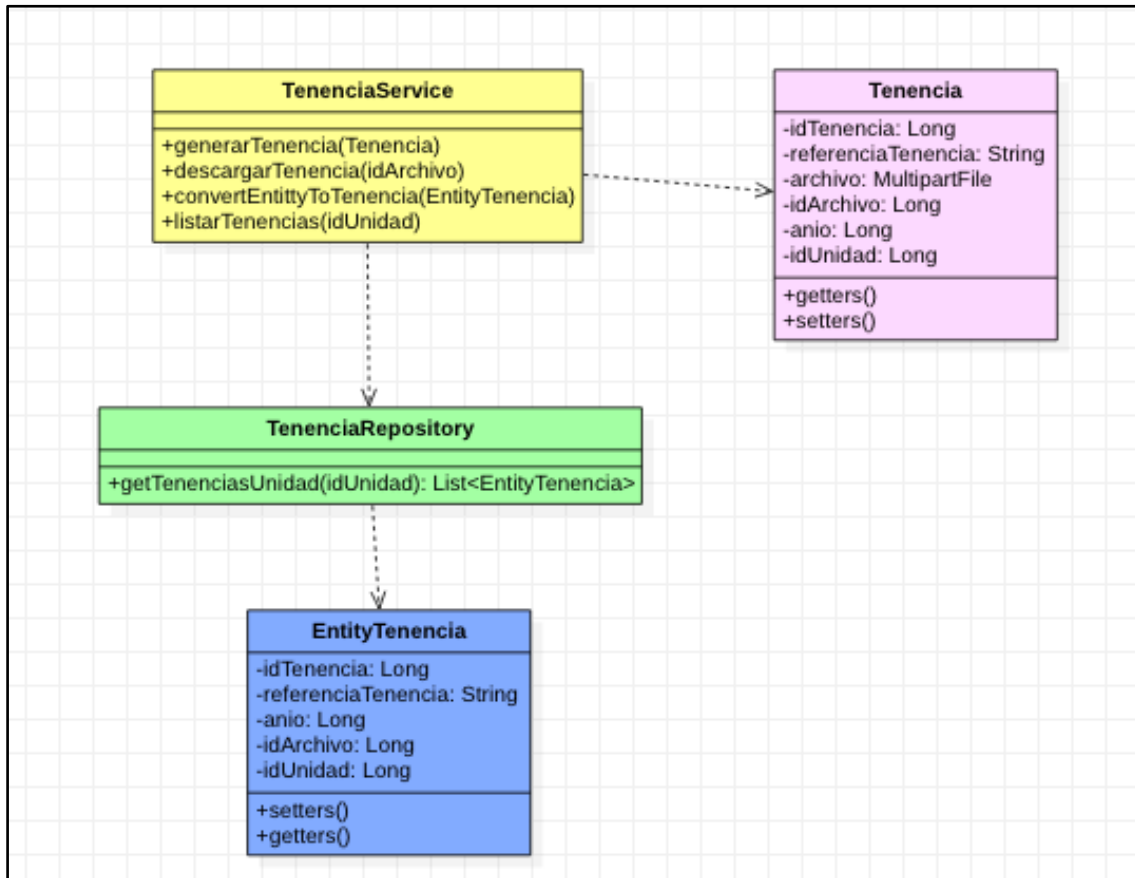


Figura 22 Diagrama de clase \_ Tenencia (autoría propia)

## H) Clase MantenimientoService

Clase que nos ayudara obtener el listado de los mantenimientos de la unidad, cargar nuevos (ver Figura 23)

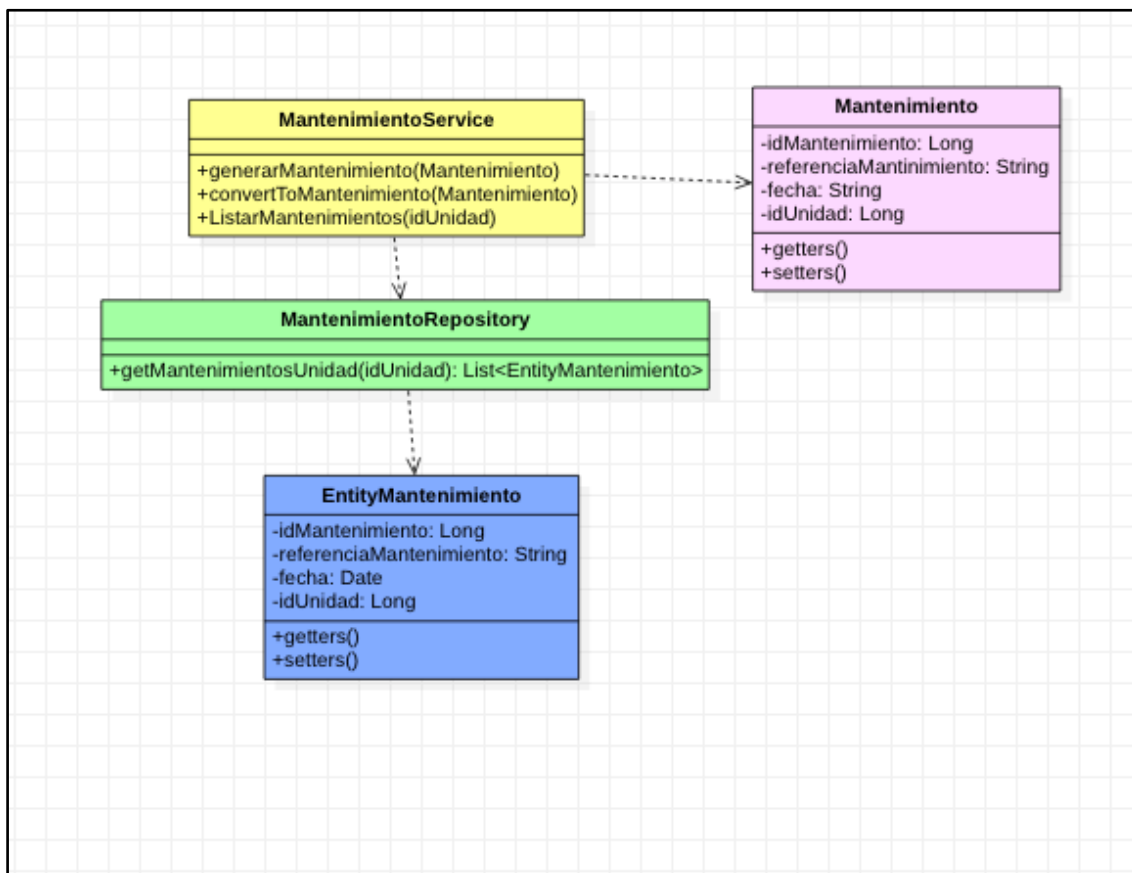


Figura 23 Diagrama de clase \_Mantenimiento (autoría propia)

## I) Clase OdbService

Clase que nos ayudara obtener el listado de los registros referentes al ODB de la unidad, cargar nuevos registros con ayuda del microprocesador (ver Figura 24).

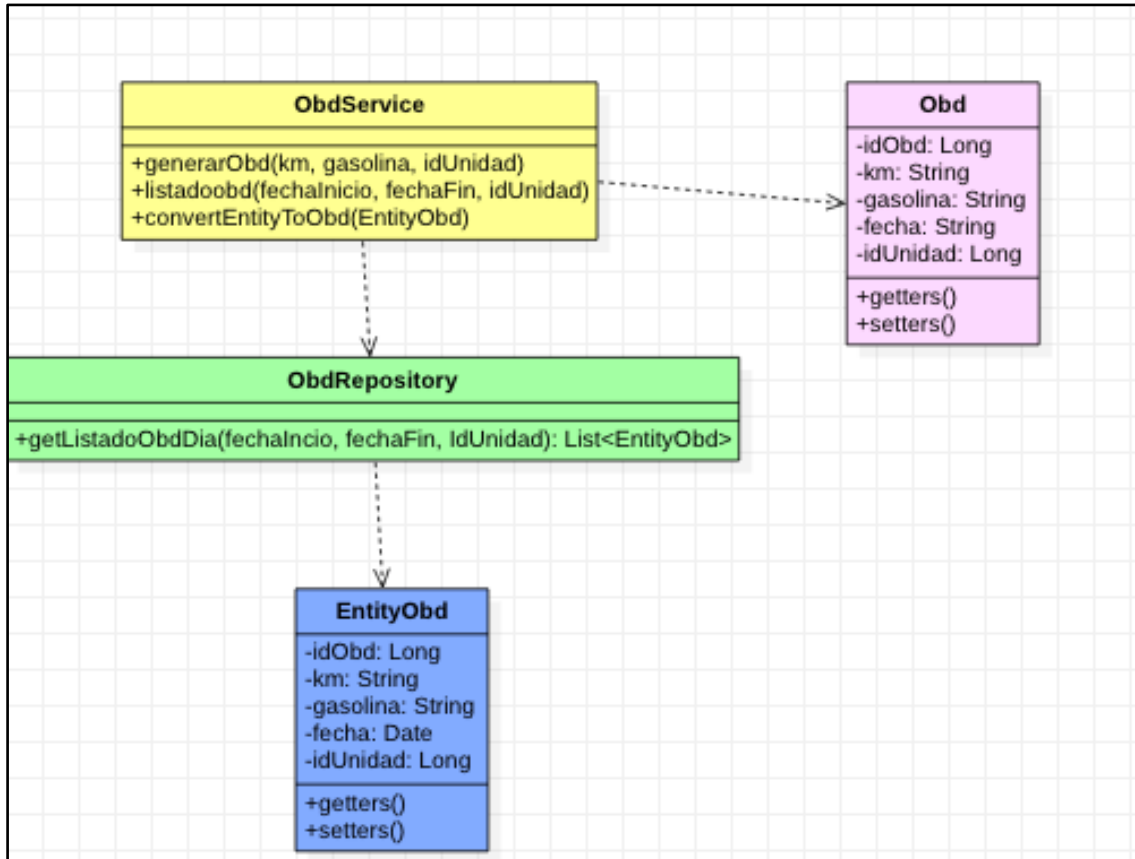


Figura 24 Diagrama de clase \_ OBD (autoría propia)

## J) Clase RastreadorService

Clase que nos ayudara obtener el rastreador de la unidad, así como cambiar la configuración del mismo, ya que con esta información se le harán peticiones al microcontrolador usando SMS (ver Figura 25).

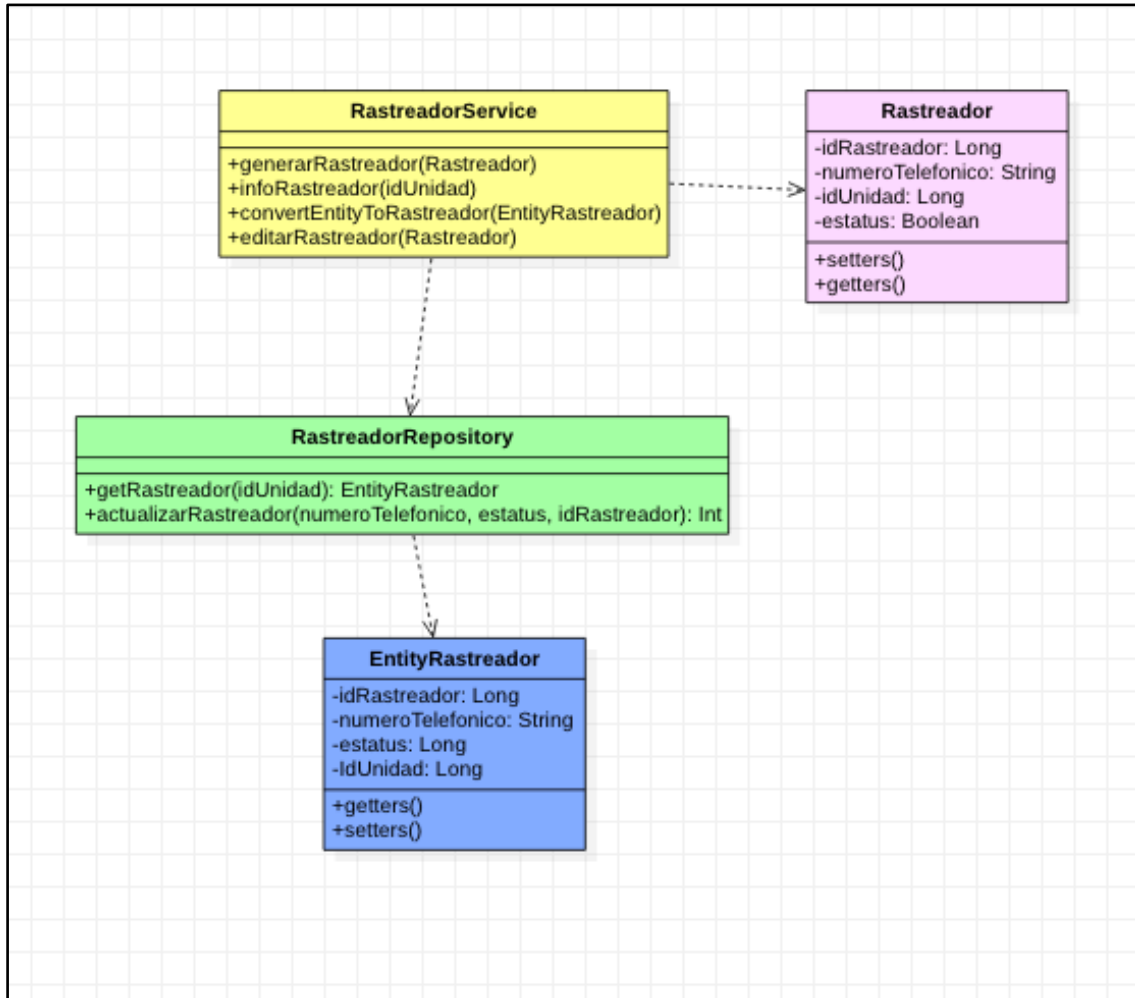


Figura 25 Diagrama de clase \_ Rastreador (autoría propia)

### K) Clase PosicionService

Clase que nos ayudara obtener el rastreador de la unidad, así como cambiar la configuración del mismo, ya que con esta información se le harán peticiones al microcontrolador usando SMS (ver Figura 26).

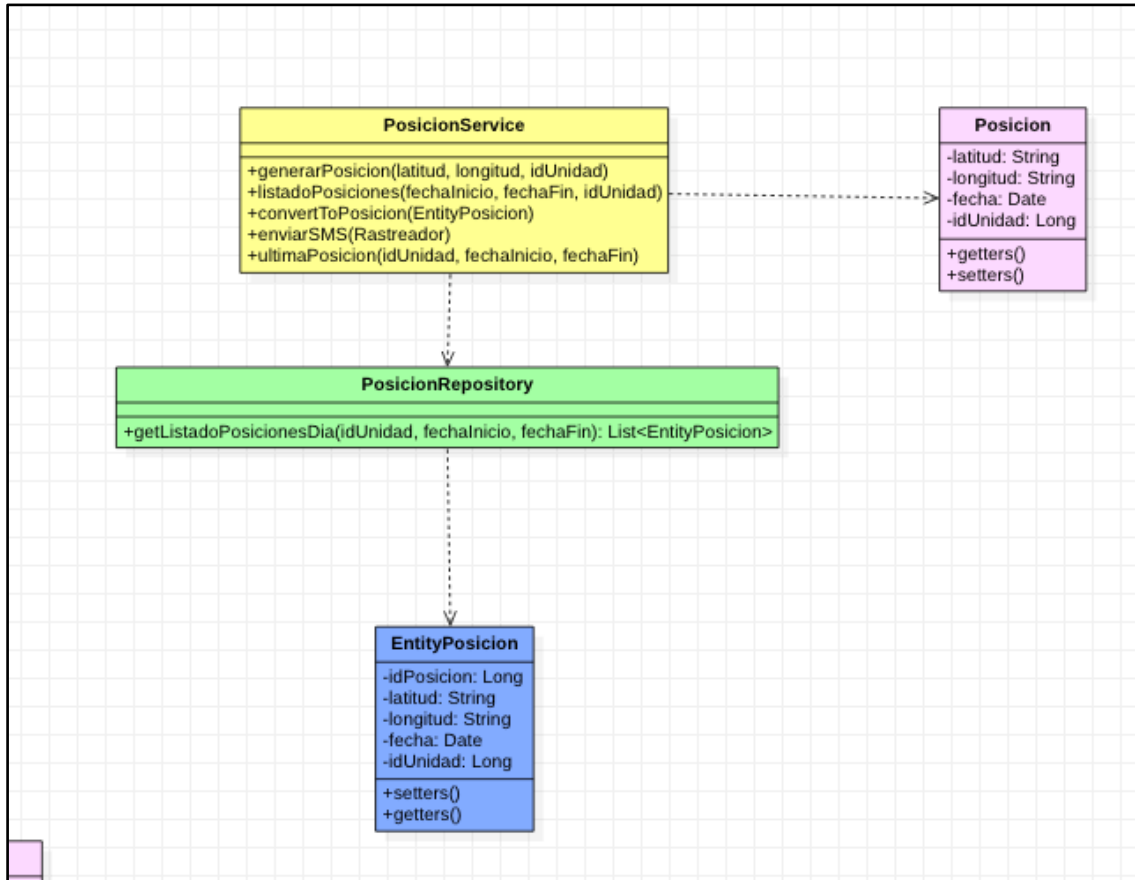


Figura 26 Diagrama de clase \_ Posición (autoría propia)



## L) Clase UsuarioService

Clase que ayuda a obtener la información de los usuarios, modificarla y crear nuevos usuarios esta clase se comparte con el rol de usuario y el rol de administrador, el rol de usuario solo puede cambiar su contraseña, mientras que en el rol del administrador puede cambiar las contraseñas de todos los usuarios (ver Figura 27 )

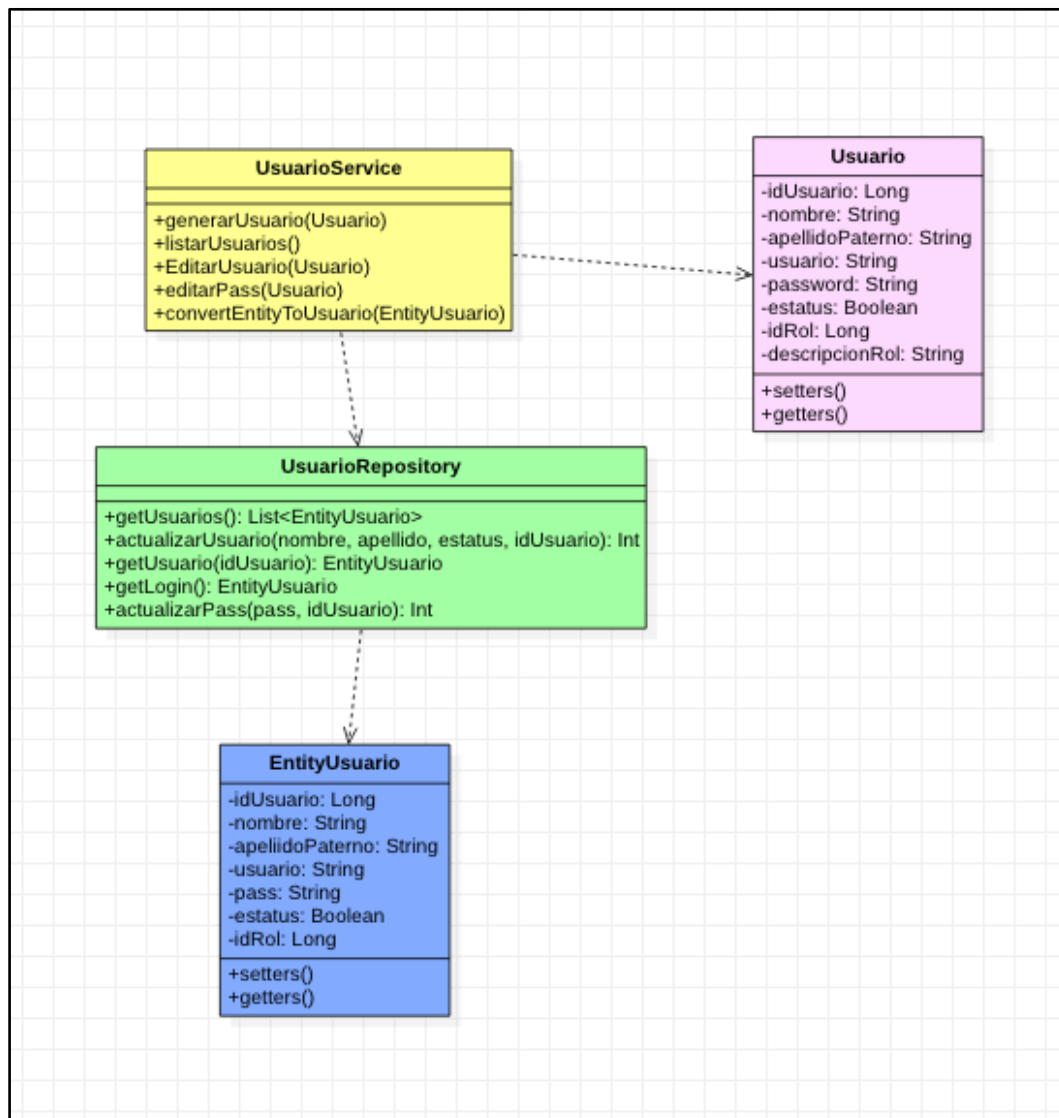


Figura 27 Diagrama de clase \_ Usuario (autoría propia)

### M) Clase ArchivoService

Clase que ayuda a obtener un archivo cargado y así como cargar diferentes archivos en el flujo del sistema, como pudiera ser tenencia, seguro, verificación, etc. (ver Figura 28 )

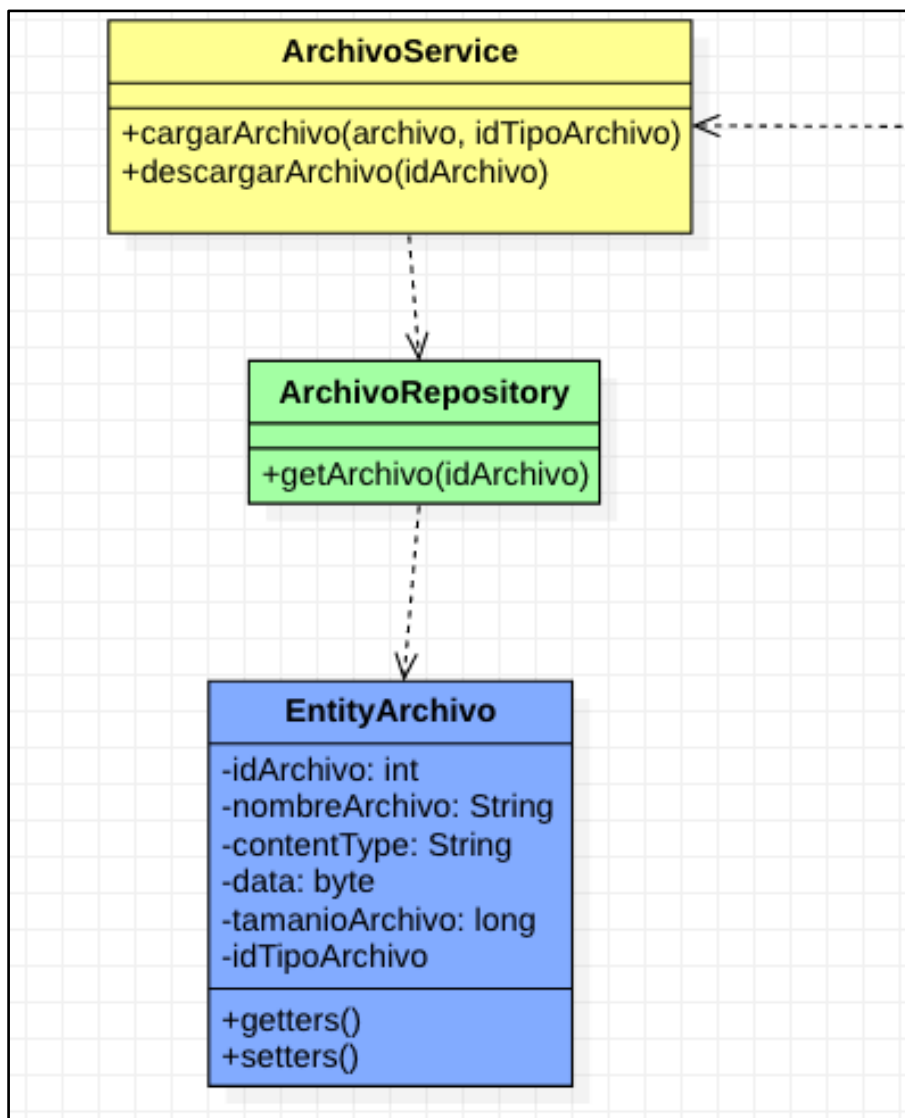


Figura 28 Diagrama de clase \_ Archivo

## N) Clase MarcaService

Clase que nos ayudara obtener el listada de las marcas de automoviles, cargar nuevas y editar las marcas (ver Figura 29 )

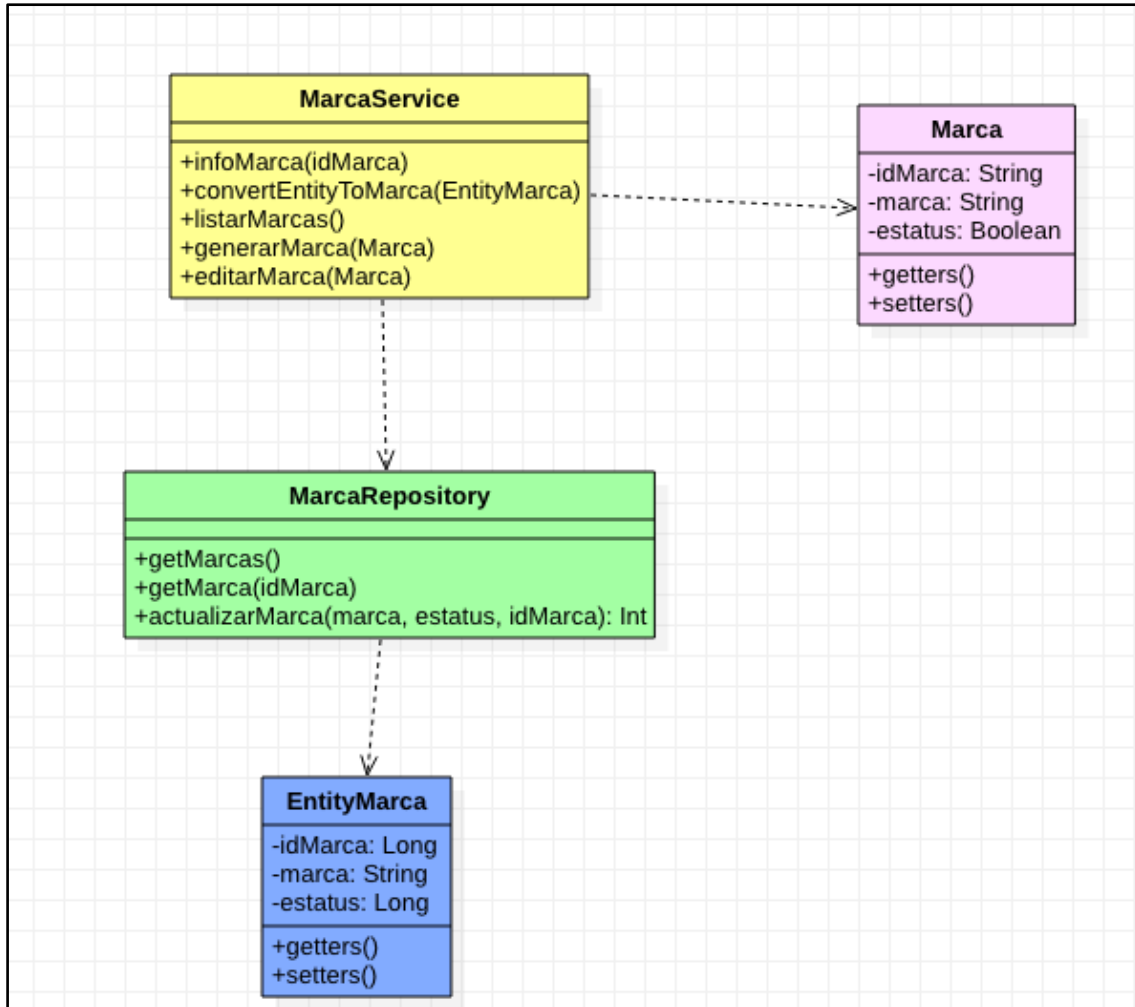


Figura 29 Diagrama de clase \_Marca (autoría propia)

### O) Clase DepartamentoService

Clase que ayuda a obtener el listado de las marcas de los departamentos en los cuales pertenece cada unidad, crear nuevos departamentos y editar (ver Figura 30).

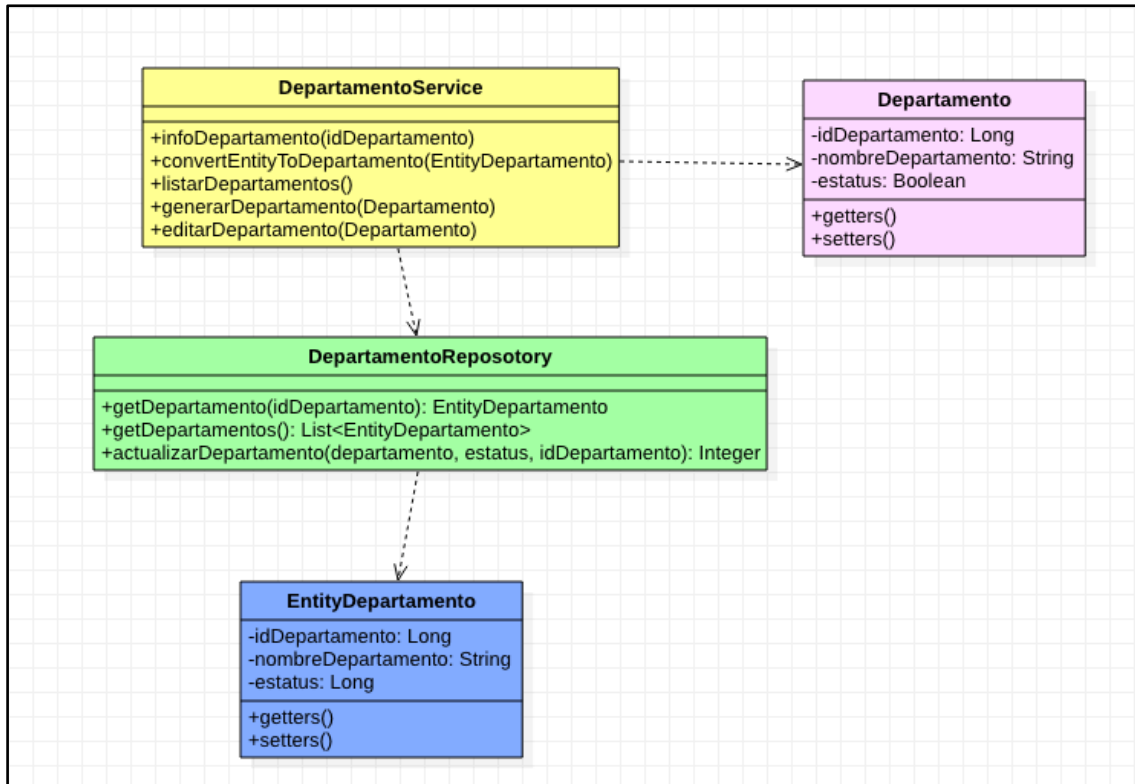


Figura 30 Diagrama de clase \_ Departamento (autoría propia)

### 3.6 Maquetados

En este apartado se muestran los maquetados de cada uno de los servicios, una aproximación de lo que se espera obtener como interfaz del sistema de gestión vehicular.

#### A) Maquetado Token

En la Figura 31 se muestra el maquetado de la interface del login o inicio de sesión en donde se genera el Token de seguridad.



Maquetado de la interfaz de login para el Gestor Vehicular. La pantalla muestra el logo "GOBIERNO DIGITAL Muixquilucan" y el título "GESTOR VEHICULAR". Hay campos de entrada para "Usuario" y "Contraseña", y un botón "Acceder".

Figura 31 Maquetado\_Token (autoría propia)

## B) Maquetado Vista principal

Al ingresar como administrador en la vista principal se podrá realizar la gestión de las unidades (ver Figura 32).

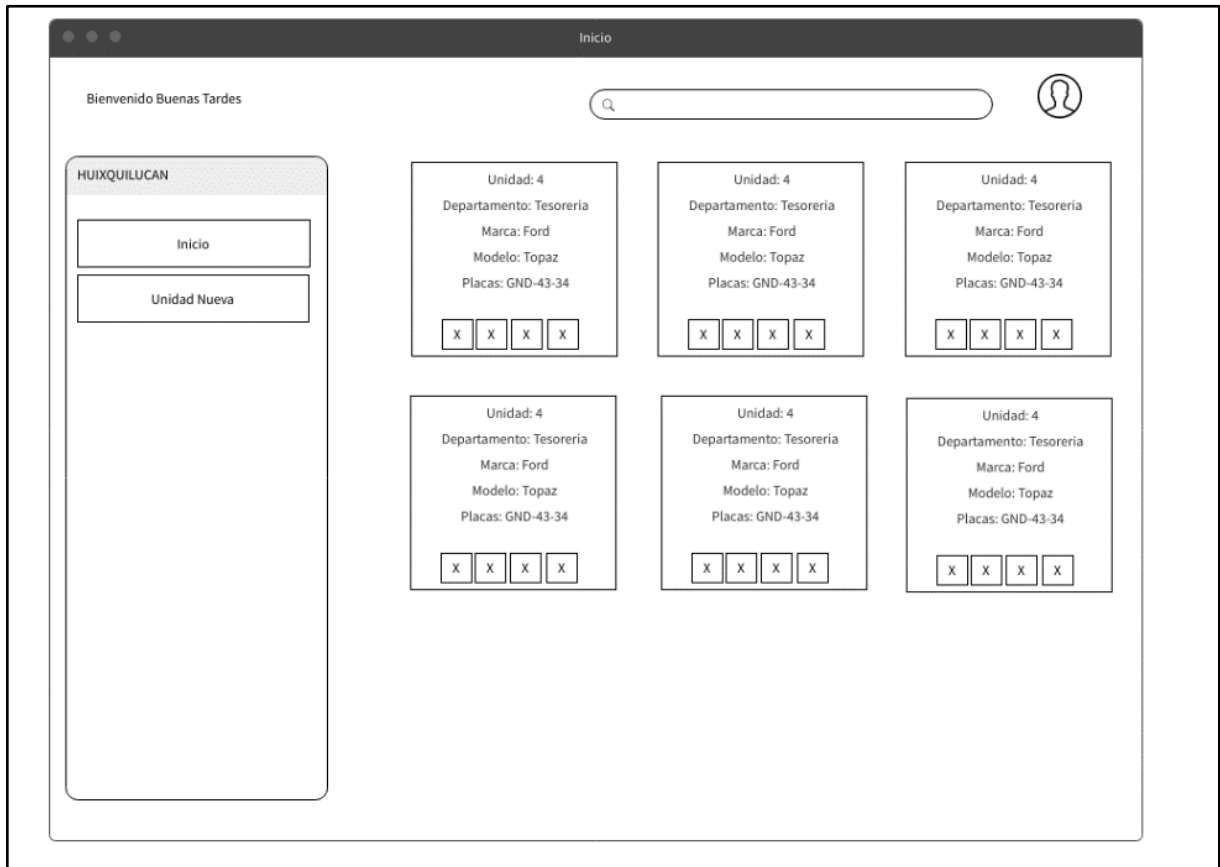


Figura 32 Maquetado\_Vista Principal (autoría propia)

### C) Maquetado Unidad

La vista para agregar una unidad nueva estará dispuesta como se muestra a continuación (ver Figura 33).

Unidada Nueva

Bienvenido Buenas Tardes

HUIXQUILUCAN

Inicio

Unidad Nueva

Panel

Identificador Unidad

Placas

Departamento

Marca

Modelo

Año

Factura

Tarjeta circulacion

Generar Unidad

Figura 33 Maquetado \_ Unidad (autoría propia)

## D) Maquetado Factura

Al ingresar como usuario es posible tener acceso al listado de facturas para un seguimiento oportuno. (ver Figura 34).

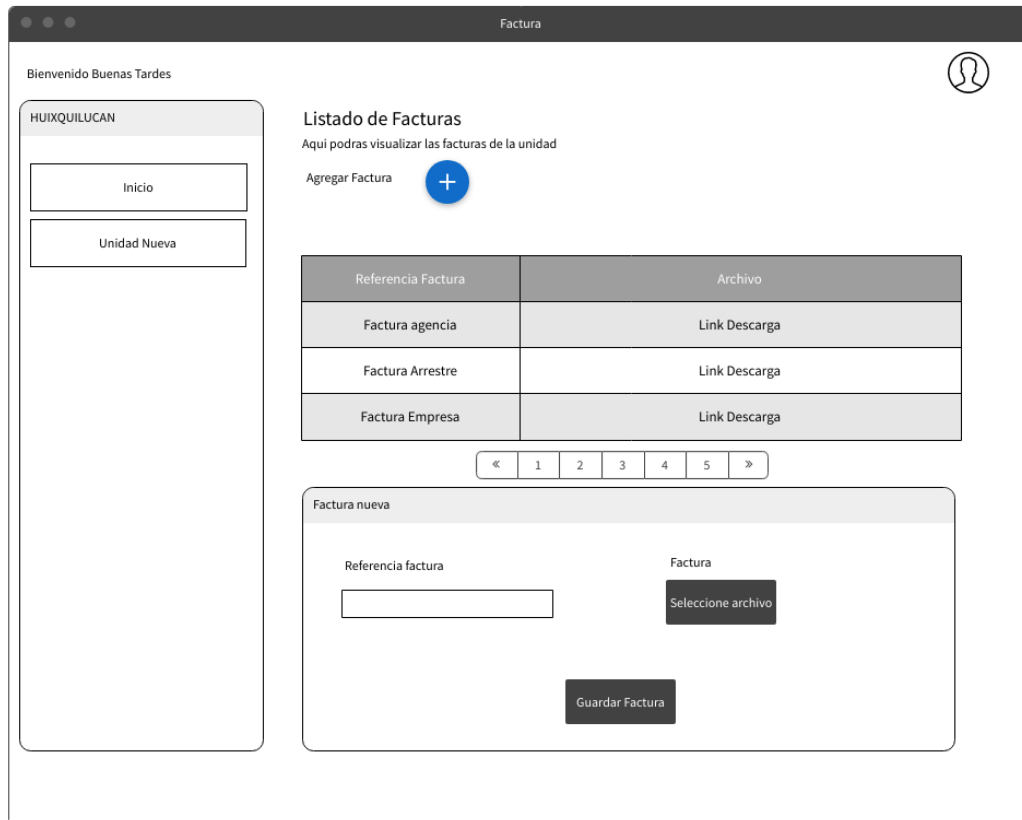


Figura 34 Maquetado \_ Factura (autoría propia)



## E) Maquetado Tarjeta de Circulación

En la siguiente figura (Figura 35) se puede ver el seguimiento de las tarjetas de circulación así como vigencias y renovaciones.

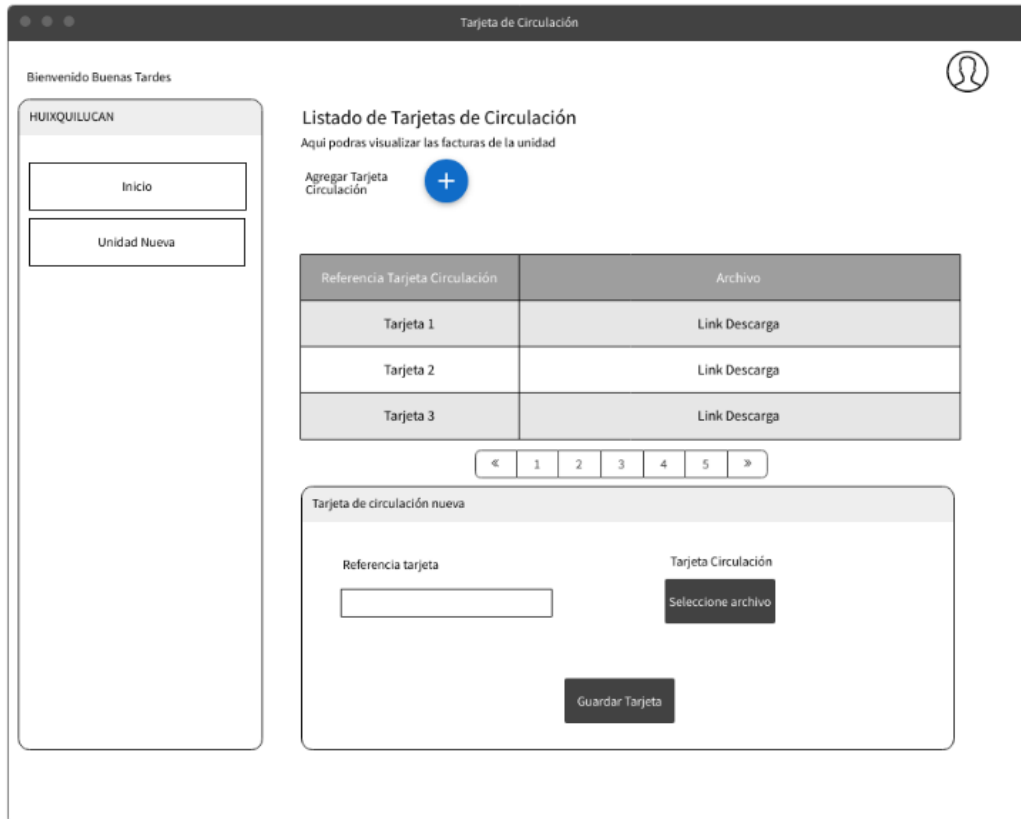


Figura 35 Maquetado \_ Tarjeta de circulación (autoría propia)

## F) Maquetado Seguros

En la Figura 36 se muestra la interfaz correspondiente al listado de seguros, en donde también es posible agregar o editar los mismos.

Bienvenido Buenas Tardes

HUIXQUILUCAN

Inicio

Unidad Nueva

Listado de Tarjetas de Seguros

Aquí podras visualizar los seguros de la unidad

Agregar seguro +

Referencia Seguro	Vigencia Inicio	Vigencia fin	Activo	Archivo
Seguro 1	23-02-2019	23-02-2020	activo	Link Descarga
Seguro 2	23-02-2019	23-02-2020	activo	Link Descarga
Seguro 3	23-02-2019	23-02-2020	activo	Link Descarga

< 1 2 3 4 5 >

Seguro nuevo

Referencia Seguro

Vigencia Inicio 12 May 2016

Vigencia Fin 12 May 2016

Archivo seguro

Seleccione archivo

Guardar Seguro

Figura 36 Maquetado\_ Seguros (autoría propia)

## G) Maquetado Verificación

En esta figura se muestra la interfaz del listado de verificaciones por periodos en lapsos de tiempo, y por unidad seleccionada (ver Figura 37).

Bienvenido Buenas Tardes

HUIXQUILUCAN

Inicio

Unidad Nueva

### Listado de Tarjetas de verificaciones

Aquí podras visualizar las verificaciones de la unidad

Agregar verificación

Referencia Verificación	Año	Periodo	Archivo
Verificación 1	2020	Periodo 1	Link Descarga
Verificación 2	2020	Periodo 2	Link Descarga
Verificación 3	2021	Periodo 1	Link Descarga
Verificación 4	2021	Periodo 2	Link Descarga

<< 1 2 3 4 5 >>

### Verificación nueva

Referencia verificación

Año

Periodo

Archivo verificación

Seleccione archivo

Figura 37 Maquetado\_ Verificación (autoría propia)

## H) Maquetado Tenencia

El listado de pagos de tenencia y/o refrendos se muestran en el siguiente maquetado ( ver Figura 38).

Bienvenido Buenas Tardes

HUIXQUILUCAN

Inicio

Unidad Nueva

### Listado de Tarjetas de tenencias

Aquí podras visualizar las tenencias de la unidad

Agregar tenencia

Referencia Tenencia	Año	Archivo
Tenencia 1	2019	Link Descarga
Tenencia 2	2020	Link Descarga
Tenencia 3	2021	Link Descarga

< 1 2 3 4 5 >

#### Tenencia nueva

Referencia Tenencia

Año

Archivo verificación

Figura 38 Maquetado \_ Tenencia (autoría propia)

## I) Maquetado Mantenimiento

El listado de mantenimientos programados ya sea por kilometraje u ordinarios semestrales o anuales se verán registrados y el usuario puede tener acceso para visualizar los mismos (ver Figura 39)

Bienvenido Buenas Tardes

HUIXQUILUCAN

Inicio

Unidad Nueva

Listado de Mantenimientos

Aquí podrás visualizar los mantenimientos de la unidad

Agregar mantenimiento +

Referencia Mantenimiento	Fecha Mantenimiento
Mantenimiento 1	23-05-2019
Mantenimiento 2	23-05-2020
Mantenimiento 3	23-05-2021

« 1 2 3 4 5 »

Mantenimiento nuevo

Referencia Mantenimiento

Fecha Mantenimiento

Guardar Mantenimiento

Figura 39 Maquetado \_ Mantenimiento (autoría propia)

## J) Maquetado OBD-II

En la adquisición de datos del OBD-II se podrán ir registrando datos, como km recorridos, rendimiento de gasolina, alertas (ver Figura 40).

OBD

Bienvenido Buenas Tardes  
Bienvenido Buenas Tardes

HUIXQUILUCAN

Inicio

Unidad Nueva

Listado de datos OBD  
Aquí podrás visualizar los datos OBD de la unidad

Filtrado

Desde: 12 May 2016

Hasta: 12 May 2016

Buscar

Km	Gasolina	Fecha
100	20	23-05-2019
150	19	24-05-2019
200	16	25-05-2019

« 1 2 3 4 5 »

Figura 40 Maquetado \_ OBD-II (autoría propia)

## K) Maquetado Rastreador

En esta parte, se manda un mensaje de petición a la unidad registrada para obtener datos de la ubicación (ver Figura 41)

La imagen muestra una maquetación de una interfaz de usuario web para un sistema de rastreo. El encabezado de la página indica 'Rastreador' y 'Bienvenido Buenas Tardes'. A la izquierda, hay un menú con el título 'HUIXQUILUCAN' y dos opciones: 'Inicio' y 'Unidad Nueva'. El contenido principal está dividido en dos secciones. La superior, titulada 'Rastreador', incluye el subtítulo 'Aquí podrás visualizar el rastreador de la unidad', un botón 'Agregar rastreador' con un signo más (+) y un formulario para 'Rastreador nuevo/Edición Rastreador'. Este formulario contiene un campo de texto para 'Numero telefonico', un grupo de botones de opción para 'Selecciona una opción' con 'Activo' (seleccionado) e 'Inactivo', y un botón 'Guardar Rastreador'.

Figura 41 Maquetado \_ Rastreador (autoría propia)

## L) Maquetado Posición

La posición geodésica es mostrada en este apartado, también es posible tener un historial de los recorridos realizados (ver Figura 42).

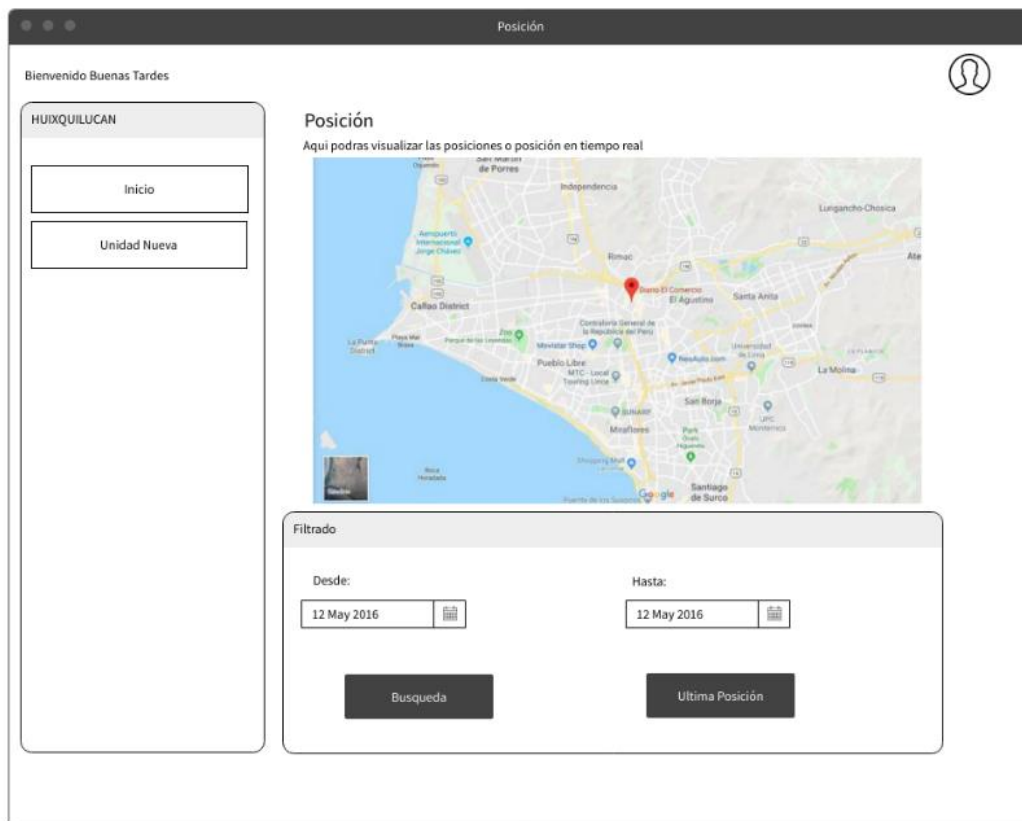


Figura 42 Maquetado \_ Posición (autoría propia)



## M) Maquetado Usuario

Esta es la interfaz que se muestra cuando el usuario desea cambiar de contraseña, con la condicional de que se utilicen caracteres especiales y al menos 10 dígitos (ver Figura 43).

La imagen muestra una maquetación de una interfaz de usuario (UI) para un sistema de usuario. El título de la ventana es "Usuario".

En la parte superior izquierda, se muestra un mensaje de bienvenida: "Bienvenido Buenas Tardes".

El contenido principal está dividido en dos secciones:

- HUIXQUILUCAN:** Una sección lateral que contiene dos botones: "Inicio" y "Unidad Nueva".
- Cambio de password:** Una sección central que contiene:
  - Un campo de entrada etiquetado "Password" con el texto de ayuda: "Mínimo diez caracteres, al menos una letra, un número y un carácter especial".
  - Un segundo campo de entrada etiquetado "Password" con el texto de ayuda: "Confirme el password".
  - Un botón "Cambiar Password" ubicado debajo de los campos de entrada.

En la parte superior derecha, hay un ícono de perfil de usuario.

Figura 43 Maquetado\_ Usuario (autoría propia)

## N) Maquetado Marca

Se pueden gestionar las unidades por marcas, para una administración eficiente (ver Figura 44).

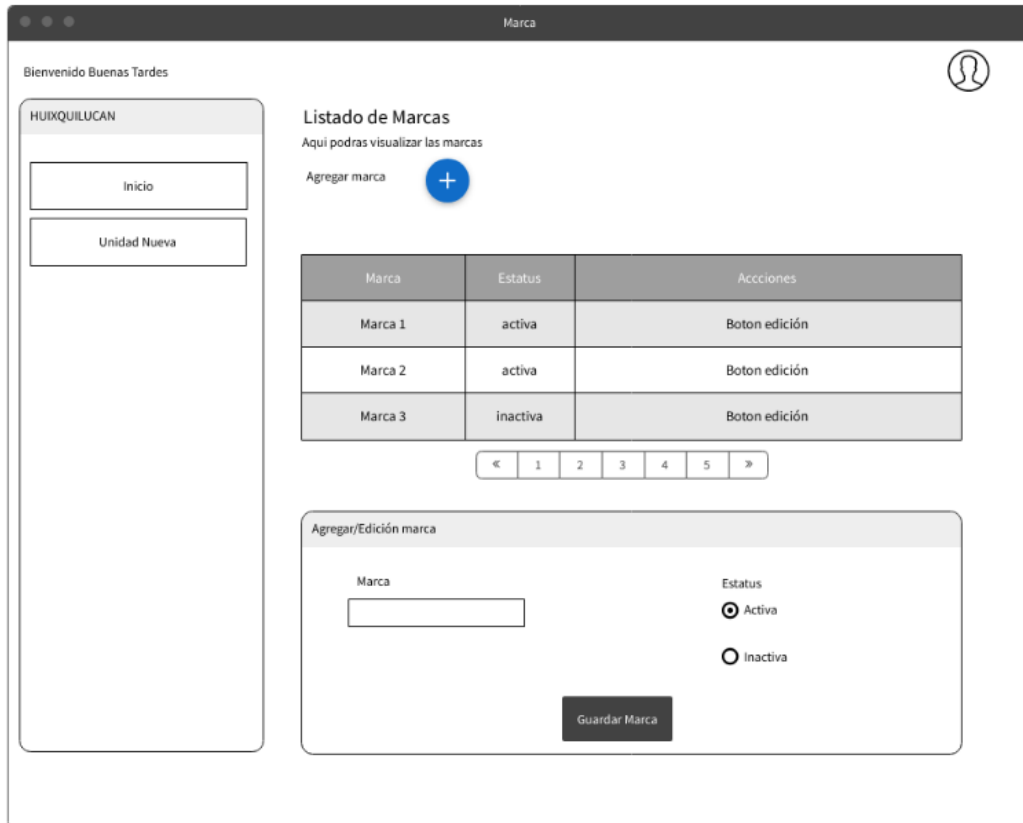


Figura 44 Maquetado \_ Marca (autoría propia)

## O) Maquetado Departamento

En esta interfaz se visualizan los departamentos en los cuales pertenecen las unidades, para que se tenga una gestión eficiente (ver Figura 45).

Bienvenido Buenas Tardes

HUIXQUILUCAN

Inicio

Unidad Nueva

Listado de Departamentos

Aquí podrás visualizar los departamentos

Agregar departamento

Departamento	Estatus	Acciones
Departamento 1	activa	Boton edición
Departamento 2	activa	Boton edición
Departamento 3	inactiva	Boton edición

< 1 2 3 4 5 >

Agregar/Edición departamento

Departamento

Estatus

Activa

Inactiva

Guardar

Figura 45 Maquetado\_ Departamento (autoría propia)

## P) Maquetado Unidad Nueva

Este es el maquetado para el registro de nuevas unidades en donde se podrán dar de alta con los diferentes atributos. (ver Figura 46)

Unidat Nueva

Bienvenido Buenas Tardes

HUIXQUILUCAN

Inicio

Unidad Nueva

Panel

Identificador Unidad

Placas

Departamento

Marca

Modelo

Año

Factura

Tarjeta circulacion

Figura 46 Maquetado \_ Unidad Nueva (autoría propia)

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

# **CAPITULO 4**

# **RESULTADOS**

#### 4.1 Circuitos electrónicos

En la Figura 47 se muestra el esquemático electrónico, dicho diagrama está diseñado en el software Proteus 8.5, en el mismo se observa la conectividad de los dispositivos, como componente principal se tiene el microcontrolador ATMEL de Arduino Uno desde el cual se mandan instrucciones (ver Capítulo 3) para solicitar y enviar datos. La etapa de adquisición de datos se encuentra integrada por el módulo GPS así como el BT-HC-05 el cual esta comunicado de forma permanente con la interfaz del OBD-II que a su vez está conectado al puerto del automóvil.

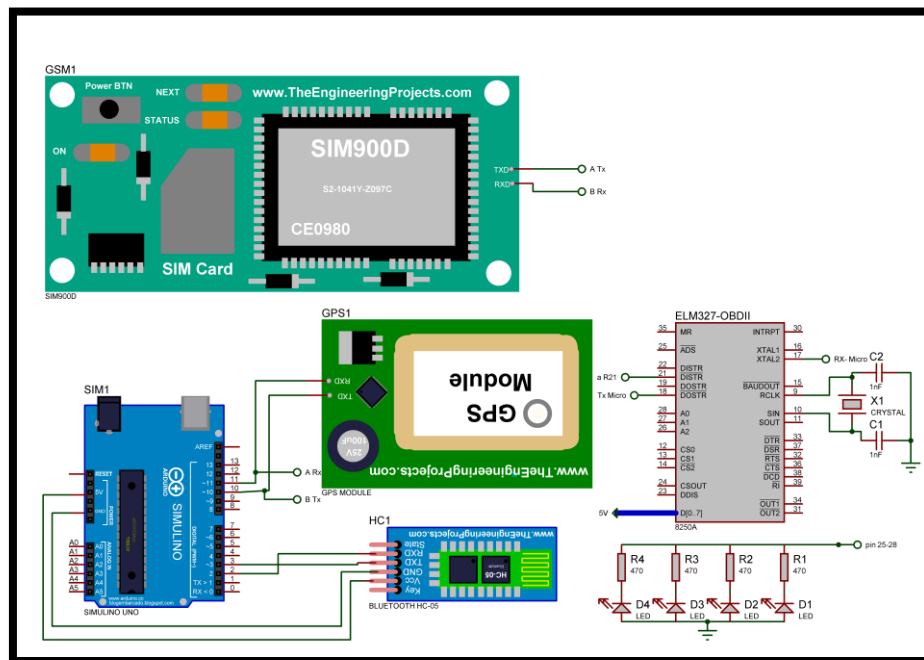


Figura 47 Diagrama Esquemático del Circuito de comunicación OBD-II-GPS-SMS (autoría propia)

Los componentes conectados de forma física son mostrados en la Figura 48, se puede observar la placa Arduino en color naranja en la parte superior izquierda, en donde están conectados los puertos de entrada y de comunicación.

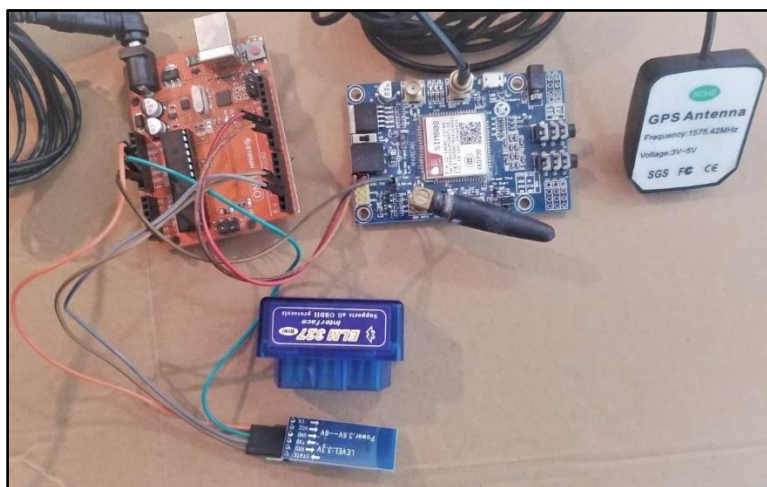


Figura 48 Componentes físicos conectados (fotografía por el autor)

## 4.2 Configuración del Bluetooth

La configuración del módulo de Bluetooth es importante para la adquisición de datos, ya que la comunicación tiene la parte de transmisión y recepción de datos, la adaptación al dispositivo con el cual entablará la comunicación es importante, para ello es importante contar con un módulo Bluetooth que permita ser configurado en modo Maestro- Esclavo, es decir, que permita la comunicación sólo con un dispositivo. En este proyecto se optó por utilizar el módulo HC-05, el cual permite editar la configuración para que pueda ser Maestro de un dispositivo en este caso la interfaz del OBD-II.

Para poder realizar la configuración del modulo BT HC-05 se debe activar los comandos AT, estos comandos sirven para configurar y parametrizar diversos dispositivos. A demás el módulo BT debe entrar en modo configuración esto se hace oprimiendo el botón EN del inglés "enable" en caso de no contar con un botón, se induce un corto circuito en pin de "WakeUp". Toda vez activado el modo editor del módulo se ejecuta un código en arduino code( Figura 49 ).

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

```

bluetooth Arduino 1.8.17 Hourly Build 2021/09/06 02:33
Archivo Editar Programa Herramientas Ayuda
bluetooth
1 #include <SoftwareSerial.h>
2
3 SoftwareSerial BT1(5, 6); // TX de modulo | RX de modulo
4
5 void setup() {
6   pinMode(8, OUTPUT);
7   pinMode(9, OUTPUT);
8   digitalWrite(9, HIGH); //Al poner en HIGH forzaremos al moduloa entrar en modo configuracion.
9   delay(500); // Espera antes de encender el modulo
10  Serial.begin(9600);
11  Serial.println("Levantando el modulo HC-05");
12  digitalWrite(8, HIGH); //Enciende el modulo
13  Serial.println("Listo para recibir comandos AT");
14  BT1.begin(38400); //Si no responde con un OK cambiar la velocidad por una de las siguientes 9600, 19200, 57600 o 115200
15 }
16
17 void loop() {
18   if (BT1.available())
19     Serial.write(BT1.read());
20   if (Serial.available())
21     BT1.write(Serial.read());
22 }
    
```

Figura 49 Código para configurar el modulo BT-HC05

Finalmente se siguen las instrucciones en la consola del Arduino para para que el dispositivo quede configurado en modo Maestro, terminado ese proceso, cada vez que se encienda el módulo BT buscará exclusivamente la dirección MAC para realizar la conexión de manera efectiva (ver Figura 50).

No	Comando AT	operación	descripción
1	AT+RESET	N/A	resetea valores
2	AT+ORGL	N/A	restaura valores de fabrica
3	AT+ROLE=	1	se coloca en modo Maestro
4	AT+UART=	115200,0,0	muestreo en baudios
5	AT+CMODE=	0	Se conecta específicamente a una dirección
6	AT+BIND=	1234,56,789c72	dirección MAC de la interfaz
7	AT+INIT	N/A	Inicia la comunicación
8	AT+PAIR=	1234,56,789c72,20	Realiza el pareo con la interfaz OBD-II, lo realiza en 20 segundos
9	AT+LINK=	1234,56,789c72	Realiza el enlace permanente con la interfaz OBD-II

Figura 50 Instrucciones en la consola de Arduino para configurar el Bluetooth



### 4.3 Envío de datos

En la siguiente figura se muestra la disposición de los componentes dentro del vehículo, debajo del volante se encuentra el puerto CAN que permite la conexión con el OBD-II, para comunicación se hace uso de una interfaz que permite la lectura de datos mediante el módulo Bluetooth BT-HC05. El sistema de comunicación podrá ser montado ya sea sobre el tablero o debajo de la guantera.



Figura 51 Disposición del Sistema de comunicación (autoría propia)

En la Figura 52 se muestra la representación del sistema funcionando de manera completa, al interior de vehículo se encuentran los componentes para el envío de datos i.e, OBD-2, GPS, Microprocesador, SIM808. El usuario o administrador entra al sistema para estar monitoreando las unidades que desee, por su parte el sistema estará registrando de manera continua datos para los históricos, para que estos puedan ser analizados posteriormente.

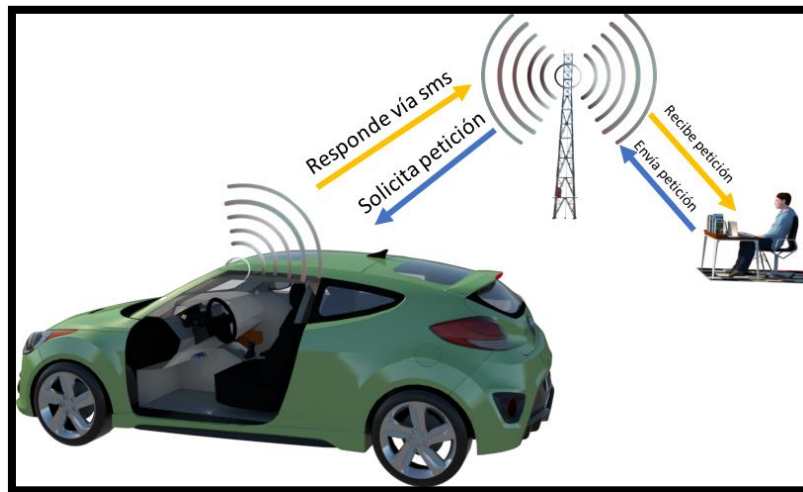


Figura 52 Sistema de comunicación, envío de datos. (autoría propia)

. En la Figura 53 se muestra el sistema en operación montado en un automóvil, en donde se comprueba el correcto funcionamiento de la adquisición y envío de información. Se realiza una petición remota a la SIM 3G, el microprocesador recibe la petición con la cadena de caracteres correcta y envía una respuesta al Arduino, así como una serie de datos que son enviadas vía SMS al emisor remoto.



Figura 53 Petición y adquisición de datos in-situ (fotografía por el autor)

#### 4.4 Almacenamiento en plataforma digital

En la Figura 54. se muestran la pantalla de inicio en donde se puede acceder con rol de administrativo o rol de usuario, a demás se asigna un token para evitar accesos no permitidos.

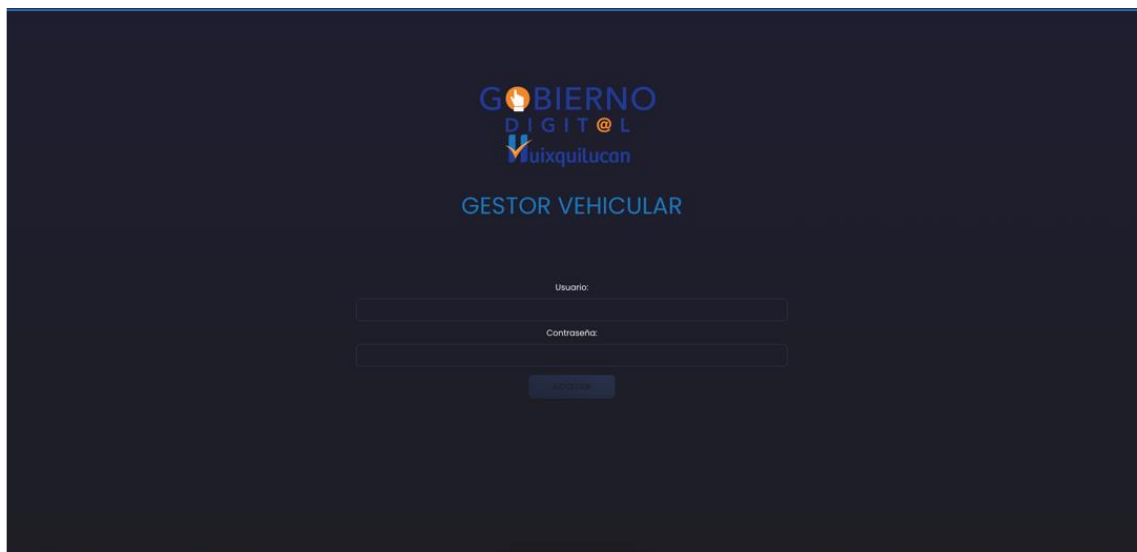


Figura 54 Pantalla de Login ( propiedad del autor)

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

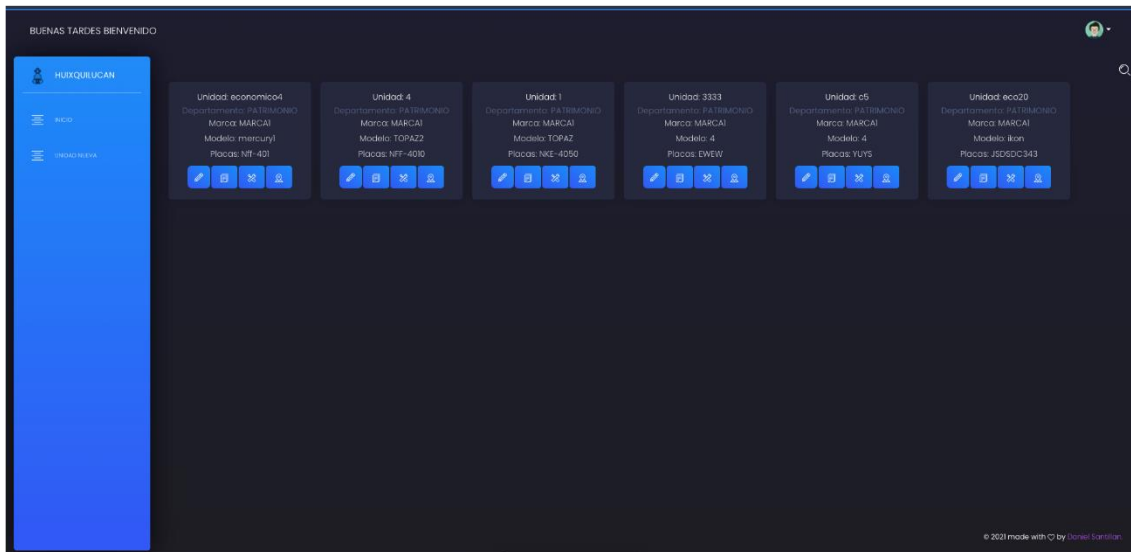


Figura 55 Listado de unidades ( propiedad del autor)

En la Figura 55 se muestran las unidades que han sido registradas o dadas de alta, dentro de las mismas se pueden editar o generar unidades nuevas.

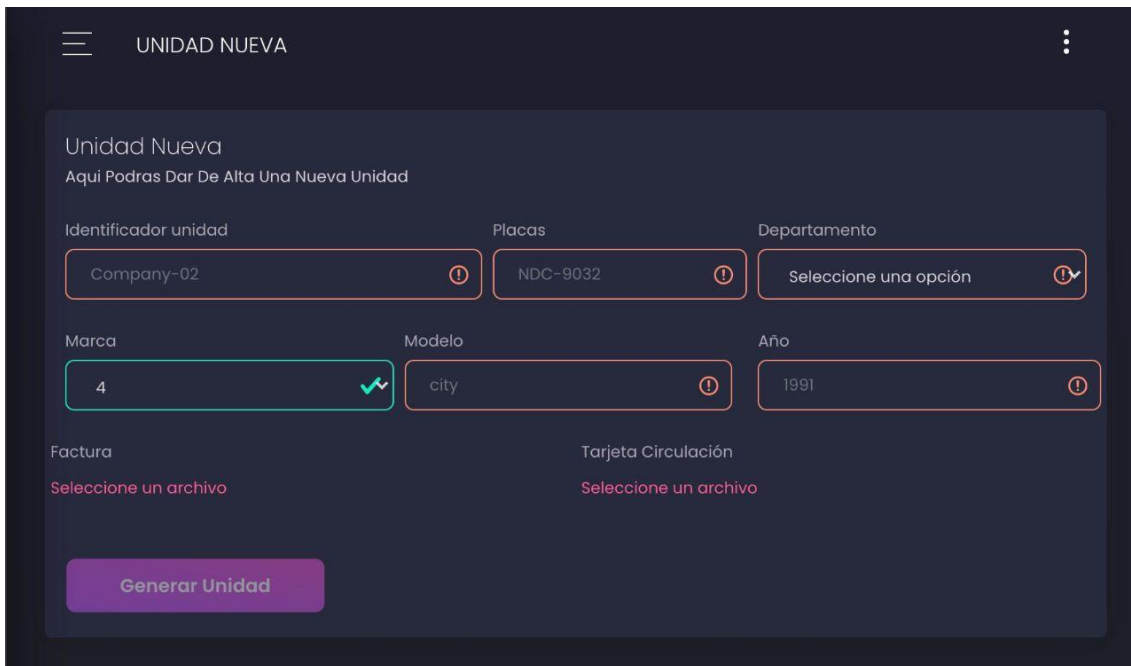


Figura 56 Crear unidad nueva ( propiedad del autor)

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

En la Figura 56 Se muestra el caso específico donde nos ayuda crear nuevas unidades.

BUENAS TARDES BIENVENIDO

HUIQUILUCAN

Listado de datos ODB  
Aquí Podrás Visualizar Los datos ODB de La Unidad

Filtrado

Desde: 02/02/2021 Hasta: 28/04/2021

Buscar

KM	GASOLINA	FECHA
100	1	06-02-2021
100	1	06-02-2021

PAGINA 1 DE 1 PAGINAS

© 2021 made with by Daniel Sanfilippo

Figura 57 Listado de datos (propiedad del autor)

En la Figura 57 se muestra el listado de datos que provienen del OBD-II, además permite cargar nuevos registros por medio del microprocesador.

BUENAS TARDES BIENVENIDO

HUIQUILUCAN

Listado de Seguros  
Aquí Podrás Visualizar Los Seguros de La Unidad

+ Agregar Seguro

FACTURA	VIGENCIA INICIO	VIGENCIA FIN	ACTIVO	ARCHIVO
	23-02-2021	24-02-2021	X	
	23-02-2021	24-02-2021	✓	
	10-02-2021	16-02-2021	X	
holi	01-02-2021	02-02-2021	✓	
3333334343	24-02-2021	28-02-2021	✓	

PAGINA 1 DE 1 PAGINAS

© 2021 made with by Daniel Sanfilippo

Figura 58 Listado seguros de unidad. (propiedad del autor)

En la Figura 57 se muestra la pantalla que permite la obtención del listado de los seguros por unidad, además permite el cargar nuevas unidades y descargar el listado en formato pdf.

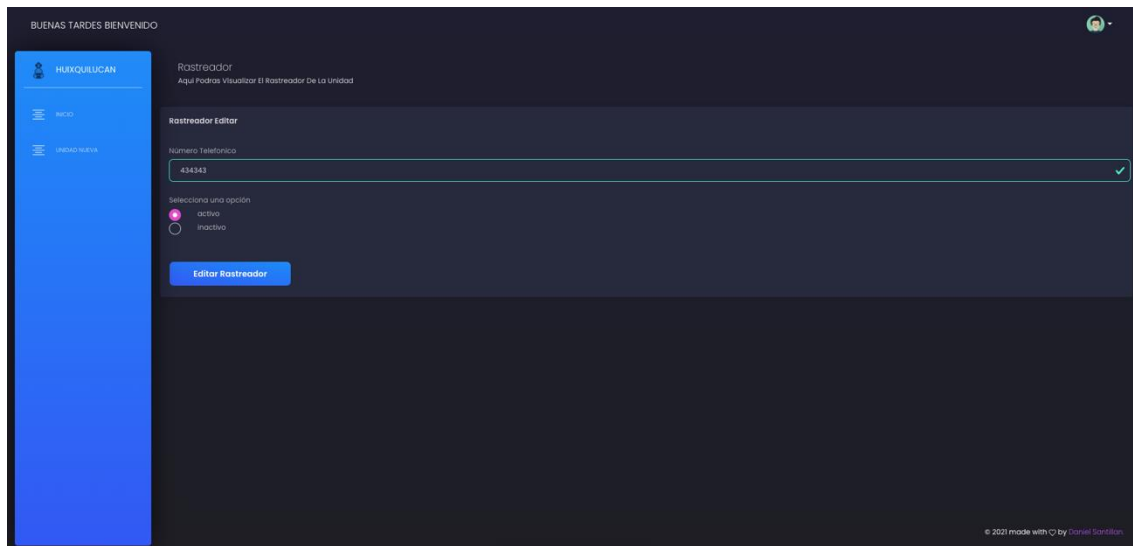


Figura 598 Rastreador de la unidad. (propiedad del autor)

En la Figura 58 se muestra la pantalla que permite configurar el rastreador de la unidad, así como cambiar la configuración de este, con esta información se le harán peticiones al microcontrolador mediante un envío de mensaje de texto por SMS.

2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

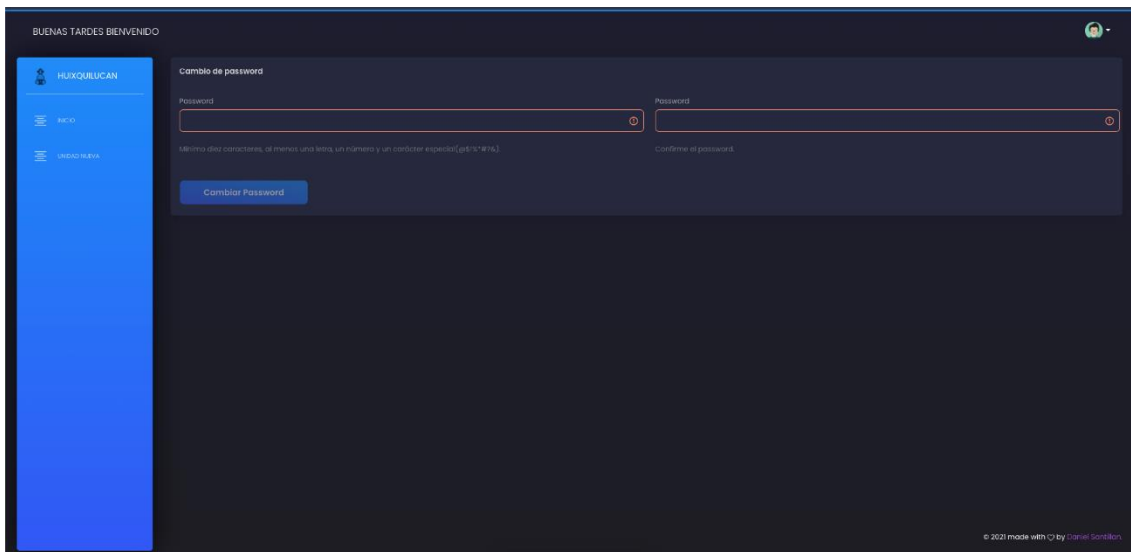


Figura 609 Edición de usuarios (propiedad del autor)

La Figura 59 muestra la pantalla la que permite obtener la información de los usuarios, modificarla y crear nuevos usuarios esta clase se comparte con el rol de usuario y el rol de administrador, el rol de usuario solo puede cambiar su contraseña, mientras que en el rol del administrador puede cambiar las contraseñas de todos los usuarios. La Figura 59 muestra el histórico de las ubicaciones actualizadas o posiciones del vehículo en determinado intervalo de tiempo.

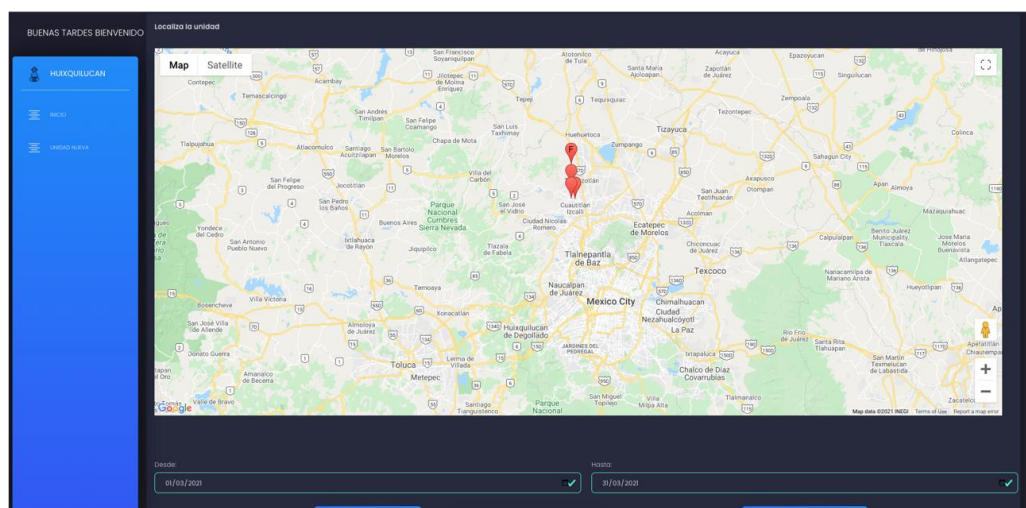


Figura 60 Histórico de ubicaciones geodésicas. (propiedad del autor)

## Conclusiones

El diseño y modelado del sistema fue concluido de manera satisfactoria, cumpliendo con ello el objetivo principal "Diseñar y modelar un sistema de rastreo y comunicación de datos en el municipio de Huixquilucan que ayudará a tener un mejor control en las unidades, evitando el desvío de fondos y malas prácticas." Se pudo adquirir parámetros del automóvil mediante el OBD-II, como el km total, velocidad, revoluciones por minuto, rendimiento y kilómetros por hora. A demás por medio del GPS se obtienen las coordenadas en tiempo real que permite estar al tanto de la unidad en todo momento. Se le pueden agregar mejoras como un sistema de alarma en caso de que esté mas de cierto tiempo detenido, o si rebasa cierta velocidad, esto para tener un monitoreo con mayor eficiencia, sin embargo, este sistema es funcional y con un costo accesible, teniendo con ello una ventaja en el mercado. Con un Sistema web para el manejo de la flotilla municipal, se logró tener un control preciso de los documentos de cada unidad, así en el caso de necesitarlos se tienen ordenados y de fácil acceso, uno de los propósitos del actual ayuntamiento es la reducción del uso del papel, al tener todo en digital es posible compartir el expediente entero de una unidad de manera rápida y ecológica, el proceso de la contratación del seguro del parque vehicular se volvió una tarea fácil de realizar reduciendo el número de unidades a cero sin seguro vigente, el rastreo de las unidades en tiempo real y la obtención de la relación del kilometraje recorrido ayudo en gran medida a determinar que unidades presentaban un rendimiento bajo y son candidatos directos a un mantenimiento correctivo y preventivo, de igual modo se detectó unidades estaban siendo usadas de manera inadecuada.



2020. "Año de Laura Méndez de Cuenca; emblema de la mujer Mexiquense".

Haciendo el uso correcto de las herramientas brindadas durante nuestra formación se logró la implementación de este proyecto, el cual es en beneficio en la sociedad, el proyecto puede ser implementado en cualquier entidad que tenga una flotilla vehicular, ya sea de gobierno o publica, este proyecto puede ser tomado para complementarse con una solución del travelling Salesman Problem.

## Referencias

### Bibliografía

- Alliance, S. (marzo de 2018). *Scrum*. Obtenido de What is Scrum:  
[http://www.scrumalliance.org/pages/what\\_is\\_scrum](http://www.scrumalliance.org/pages/what_is_scrum)
- Ángel Cobo, P. G. (2015). *PHP y MySQL*. España: Diaz de Santos.
- Bautista Q. (2012). *Programación Extrema XP*. Bolivia : Union Bolivariana.
- Beck, K. (1999). *Extreme Programming Explained: Embrace*. Stoughton:  
Addison Wesley.
- Blancarte, O. (17 de Julio de 2018). *Que es Spring Boot*. Obtenido de  
<https://www.oscarblancarteblog.com/2018/07/17/spring-boot-relacion-los-microservicios/>
- Blasco, V. (2014). *Sistema de Diagnostico OBDII*. Obtenido de  
[http://electronicar.net/IMG/Articulo\\_OBDII.pdf](http://electronicar.net/IMG/Articulo_OBDII.pdf)
- Colaborativo, Redacción Power. (9 de mayo de 2019). *Power Data*. Obtenido de  
<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-un-gestor-de-datos-y-para-que-sirve>
- Durango A, A. (2015). *Diseño Web cons CSS: 2a Edición*. EU: campus academy.
- Francisco Bellas, R. M. (2016). Lenguajes de Programación y procesadores. En R. M. Francisco Bellas. Madrid: Editorial Universitaria Ramon Areces.
- Francisco Cortazar, R. M. (2016). *Lenguajes de programación y procesadores*. Madrid: UNED.
- Gutierrez, E. (2009). *JavaScript Conceptos básicos y avanzados*. Barcelona: Ramon Maita.
- Hirota Takeuchi, I. N. (Enero-Febrero de 2009). The new new product development game. *Stop running the relay race and take up rugby*. Harvard, EU: Harvard Bussines.
- Ibañez, L. H. (2015). *Administración de Sistemas Gestores de Bases de Datos*. Barcelona: Ra-Ma.

- J.Bronson, G. (2007). *C++ Para ingeniería y Ciencias*. México: Thomson.
- JSMS, J. S. (2020). *HTML desde cero*. México: JSMS Productions.
- L.Fernandez, L. (2014). *Java Script Ejercicios practicos*.
- MDN, C. (10 de mayo de 2021). *Mozilla.org*. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>
- Netec. (2021). *que es oracle*. Obtenido de <https://www.netec.com/que-es-oracle>
- Orozco, N. M. (17 de Junio de 2014). *Forbes*. Obtenido de Forbes: <https://www.forbes.com.mx/tics-oportunidad-de-desarrollo-para-mexico/>
- Parada, M. (23 de Noviembre de 2019). *Open Webinars*. Obtenido de <https://openwebinars.net/blog/que-es-sql-server/>
- PozoRuz, A. R.-A. (2016). *Sistema de posicionamiento global (GPS) ): Descripción, análisis de errores, aplicaciones y futuro*. Malaga.
- React, Comunidad. (Abril de 2021). *React*. Obtenido de <https://es.reactjs.org/>
- Rodriguez, E. (27 de Enero de 2020). *Seoestudios*. Obtenido de Qué es backend y por qué es tan importante para tu web: <https://www.seoestudios.es/blog/que-es-backend-web/>
- Ruiz, E. d. (2018). *Sistemas de Telefonía fija y móvil*. Madrid: Paraninfo.
- Tomasi, W. (2003). *Sistemas de Comunicaciones Electrónicas*. En W. Tomasi, *Sistemas de Comunicaciones Electrónicas* (pág. 976). Arizona: Prentice Hall.