

UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Presentado por:

ELIZABETH ALEJANDRINA QUINTANA OLARTE

**DESARROLLO DE UN SISTEMA DE GEOLOCALIZACIÓN DE
ALERTA DE RECOJO DE RESIDUOS SÓLIDOS EN EL DISTRITO
DE SAN JERÓNIMO, 2018.**

Asesor:

MSc. ING. CARLOS YINMEL CASTRO BULEJE

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE
SISTEMAS**

ANDAHUAYLAS – APURÍMAC – PERÚ

2018



APROBACIÓN DEL ASESOR

Quién suscribe:

MSc. Carlos Yinmel Castro Buleje, por la presente:

CERTIFICA,

Que, la Bachiller en Ingeniería de Sistemas Elizabeth Alejandrina Quintana Olarte ha culminado satisfactoriamente el informe final de tesis intitulado: "DESARROLLO DE UN SISTEMA DE GEOLOCALIZACIÓN DE ALERTA DE RECOJO DE RESIDUOS SÓLIDOS EN EL DISTRITO DE SAN JERÓNIMO, 2018" para optar el Título Profesional de Ingeniero de Sistemas.

Andahuaylas, 01 de Diciembre del 2018

MSc. Carlos Yinmel Castro Buleje
Asesor

Bach. Elizabeth Alejandrina Quintana Olarte
Tesista



APROBACIÓN DEL JURADO DICTAMINADOR

LA TESIS: Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018; para optar el Título Profesional de Ingeniero de Sistemas, ha sido evaluada por el Jurado Dictaminador conformado por:

PRESIDENTE: ING. JUAN JOSÉ ORÉ CERRÓN

PRIMER MIEMBRO: Dr. JULIO CESAR HUANCA MARÍN

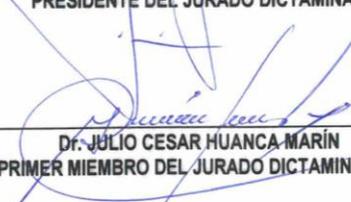
SEGUNDO MIEMBRO: MSc. MAGALY ROXANA ARANGÜENA YLLANES

Habiendo sido aprobado por UNANIMIDAD, en la ciudad de Andahuaylas el día 19 del mes de noviembre de 2018

Andahuaylas, 01 de Diciembre de 2018.



ING. JUAN JOSÉ ORÉ CERRÓN
PRESIDENTE DEL JURADO DICTAMINADOR



Dr. JULIO CESAR HUANCA MARÍN
PRIMER MIEMBRO DEL JURADO DICTAMINADOR



MSc. MAGALY ROXANA ARANGÜENA YLLANES
SEGUNDO MIEMBRO DEL JURADO DICTAMINADOR



FACULTAD DE INGENIERÍA

ACTA DE SUSTENTACIÓN DE TESIS

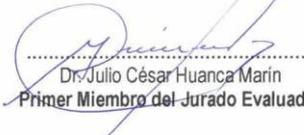
En la Av. José María Arguedas del Local Académico SL01 (Ccoyahuacho) en el auditorio de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional José María Arguedas ubicado en el distrito de San Jerónimo de la Provincia de Andahuaylas, siendo las 11:00 horas del día 19 de noviembre del año 2018, se reunieron los docentes: Ing. Juan José Oré Cerrón, Dr. Julio César Huanca Marín, MSc. Magaly Roxana Arangüena Yllanes, en condición de integrantes del Jurado Evaluador del Informe Final de Tesis intitulado: "DESARROLLO DE UN SISTEMA DE GEOLOCALIZACIÓN DE ALERTA DE RECOJO DE RESIDUOS SÓLIDOS EN EL DISTRITO DE SAN JERÓNIMO, 2018", cuyo autor es el Bachiller en Ingeniería de Sistemas ELIZABETH ALEJANDRINA QUINTANA OLARTE, el asesor MSc. Carlos Yimmel Castro Buleje, con el propósito de proceder a la sustentación y defensa de dicha tesis.

Luego de la sustentación y defensa de la tesis, el Jurado Evaluador ACORDÓ: APROBAR por UNANIMIDAD al Bachiller en Ingeniería de Sistemas ELIZABETH ALEJANDRINA QUINTANA OLARTE, obteniendo la siguiente calificación y mención:

Nota escala vigesimal		Mención
Números	Letras	
17	Deciseiete	Muy Buena

En señal de conformidad, se procedió a la firma de la presente acta en 03 ejemplares.


Ing. Juan José Oré Cerrón
Presidente del Jurado Evaluador


Dr. Julio César Huanca Marín
Primer Miembro del Jurado Evaluador


MSc. Magaly Roxana Arangüena Yllanes
Segundo Miembro del Jurado Evaluador

EDICATORIA

A mis padres por el esfuerzo, dedicación y amor incondicional, por inculcarnos la unión familiar.

A mis hermanos y hermanas por ser un ejemplo a seguir, por el apoyo incondicional, por sus palabras de aliento y por guiarme en los buenos y malos momentos, por enseñarme que todo es posible si se tiene voluntad y ganas de salir adelante y de quienes estaré siempre agradecida.

AGRADECIMIENTOS

A la persona que más valoro mi madre Alejandra Olarte por el esfuerzo incondicional, por apoyarme siempre y de quien siento gran admiración.

A mi padre Celso Quintana quien desde la gloria de Dios guía mi camino y el de mi familia, así mismo agradezco a mis familiares y amigos que estuvieron siempre apoyándome.

Al MSc. Edwin Roque Tito y MSc. Carlos Yinmel Castro Buleje por su apoyo como asesores en el desarrollo de mi tesis.

Agradezco a los profesores de la escuela profesional de ingeniería de sistemas por su labor incansable, dedicación y esfuerzo denodado de transmitir sus conocimientos para formar profesionales de éxito.

ÍNDICE

EDICATORIA	v
AGRADECIMIENTOS	vi
RESUMEN.....	xiii
ABSTRACT	xiv
CHUMASQA	xv
CAPITULO 1: INTRODUCCIÓN.....	1
1.1. DATOS GENERALES	2
1.1.1. TÍTULO DEL PROYECTO.....	2
1.1.2. AUTOR DEL PROYECTO	2
1.1.3. ASESOR DEL PROYECTO.....	2
1.1.4. LÍNEA DE INVESTIGACIÓN	2
1.1.5. ÁREA PRIORIZADA DEL PROYECTO	2
1.1.6. INSTITUCIÓN Y LUGAR DE EJECUCIÓN DEL PROYECTO	2
1.2. PLANTEAMIENTO DEL PROBLEMA.....	3
1.2.1. REALIDAD PROBLEMÁTICA.....	3
1.3. FORMULACIÓN DEL PROBLEMA.....	5
1.4. OBJETIVOS	5
1.4.1. OBJETIVO GENERAL.....	5
1.4.2. OBJETIVOS ESPECÍFICOS.....	5
1.5. JUSTIFICACIÓN	5
1.6. VIABILIDAD DE LA INVESTIGACIÓN.....	6
1.6.1. VIABILIDAD TÉCNICA:.....	6
1.6.2. VIABILIDAD ECONÓMICA.....	7
1.6.3. VIABILIDAD LEGAL	7
1.6.4. VIABILIDAD OPERATIVA.....	7
1.7. LIMITACIÓN DEL ESTUDIO	8
CAPITULO 2: MARCO TEÓRICO.....	9
2.1. ANTECEDENTES.....	9
2.1.1. ANTECEDENTES INTERNACIONALES	9
2.1.2. ANTECEDENTES NACIONALES.....	10
2.1.3. ANTECEDENTES LOCALES	11
2.2. MARCO CONCEPTUAL.....	11
2.2.1. RASPBERRY PI	11
2.2.2. HARDWARE RASPBERRY PI.....	15

2.2.3.	SOFTWARE PARA RASPBERRY PI	17
2.2.4.	DISPOSITIVOS MÓVILES.....	17
2.2.5.	APLICACIÓN MÓVILES	19
2.2.6.	SISTEMAS OPERATIVOS	19
2.2.7.	SISTEMA OPERATIVO MÓVILES	19
2.2.8.	BASE DE DATOS FIREBASE	22
2.2.9.	IONIC	22
2.2.10.	GEOLOCALIZACIÓN	23
2.2.11.	CONTAMINACIÓN	23
2.2.12.	RESIDUOS SÓLIDOS.....	24
2.2.13.	GESTIÓN AMBIENTAL	24
2.2.14.	METODOLOGÍA V.....	25
CAPITULO 3: MATERIALES Y METODOS		27
3.1.	ADMINISTRACIÓN Y GESTIÓN DEL PROYECTO BASADO EN PMBOK... 27	27
3.1.1.	PLAN DE GESTIÓN DE LA INTEGRACIÓN.....	27
3.1.2.	PLAN DE GESTIÓN DE ALCANCE.....	30
3.1.3.	PLAN DE GESTIÓN DEL CRONOGRAMA	31
3.1.4.	PLAN DE GESTIÓN DE LOS COSTOS	32
3.1.5.	PLAN DE GESTIÓN DE LA CALIDAD.....	33
3.1.6.	PLAN DE GESTIÓN DE LOS RECURSOS	34
3.1.7.	PLAN DE GESTIÓN DE LAS COMUNICACIONES.....	35
3.1.8.	PLAN DE GESTIÓN DE RIESGO.....	36
3.1.9.	PLAN DE GESTIÓN DE LAS ADQUISICIONES.....	36
3.1.10.	PLAN DE GESTIÓN DE LOS INTERESADOS	37
3.2.	METODOLOGÍA DE DESARROLLO	38
3.3.1.	NIVEL 1 DESARROLLO: FASE DE ESPECIFICACIONES	38
3.3.2.	NIVEL 2 DESARROLLO: DISEÑO DE ALTO NIVEL	41
3.3.3.	NIVEL 3 DESARROLLO: DISEÑO EN DETALLE	44
3.3.4.	NIVEL 4 DESARROLLO: IMPLEMENTACIÓN	47
CAPÍTULO 4: RESULTADOS Y DISCUSIÓN		71
4.1.	TEST UNITARIO	71
4.1.1.	PRUEBAS DE HARDWARE	71
4.1.2.	PRUEBAS DE SOFTWARE	71
4.2.	TEST DE INTEGRACIÓN.....	73
CONCLUSIONES.....		79
RECOMENDACIONES.....		80

REFERENCIAS BIBLIOGRÁFICAS.....	81
ANEXOS	83
ANEXO 1: GLOSARIO DE ACRÓNIMOS	83
ANEXO 2: CÓDIGO FUENTE	84
MANUAL DE USUARIO	96
MANUAL DEL ADMINISTRADOR.....	99

Lista de Tablas

Tabla 1: Sistemas Operativos en Perú	22
Tabla 2: Plan de Gestión de la Integración	28
Tabla 3: Plan de Gestión de Alcance	30
Tabla 4: Plan de Gestión de cronograma	31
Tabla 5: Plan de Gestión de los Costos.....	32
Tabla 6: Plan de Gestión de la Calidad	33
Tabla 7: Plan de Gestión de los Recursos.....	34
Tabla 8: Plan de Gestión de las Comunicaciones	35
Tabla 9: Plan de Gestión de Riesgo	36
Tabla 10: Plan de Gestión de las Adquisiciones.....	37
Tabla 11: Plan de Gestión de los Interesados	38
Tabla 12: Características Arduino y Raspberry Pi	41
Tabla 13: Casos de uso	46

Lista de figuras

Figura 1: Raspberry Pi.....	12
Figura 2: Raspberry Pi Modelo A.....	14
Figura 3: Raspberry pi modelo b.....	14
Figura 4: Puerto GPIO.....	16
Figura 5: Modelo en V.....	25
Figura 6: Áreas del conocimiento PMBOK.....	27
Figura 7: Diagrama caso de uso general.....	42
Figura 8: Diagrama de sub caso de uso 1.....	43
Figura 9: Diagrama de sub caso de uso 2.....	43
Figura 10: Diagrama de Despliegue.....	44
Figura 11: Diagrama de actividad mostrar vehículo.....	45
Figura 12: Diagrama de actividad mi ubicación.....	45
Figura 13: Diagrama de actividad reporte.....	46
Figura 14: NOOBS.....	48
Figura 15: Descarga NOOBS.....	48
Figura 16: Partes de la Raspberry.....	49
Figura 17: Instalación de Raspbian.....	49
Figura 18: Vista Principal.....	50
Figura 19: Configuración Raspberry Pi.....	50
Figura 20: GPS U-Blox NEO.....	51
Figura 21: Diagrama de conexión.....	51
Figura 22: Transmisión de datos GPS.....	53
Figura 23: Transmisión GPS.....	54
Figura 24: Script envio.py.....	54
Figura 25: Servidor Initial State.....	55
Figura 26: Gráfica Initial State.....	55
Figura 27: Package.json.....	56
Figura 28: Cordova - Plugins.....	57
Figura 29: Directorio del Proyecto "Localizador".....	58
Figura 30: Instalar Angularfire2 Firebase.....	60
Figura 31: Nuevo Proyecto Firebase.....	60
Figura 32: Script Firebase.....	61
Figura 33: Script de configuración de Firebase.....	61
Figura 34: Archivo de configuración Firebase "src/app/module.ts".....	61
Figura 35: Localizador Database.....	62
Figura 36: rootPage: Login.....	63
Figura 37: login.ts.....	63
Figura 38: loggedin.html.....	64
Figura 39: loggedin.ts.....	64
Figura 40: acceso a loggedin.....	65
Figura 41: librería vercamion.ts.....	65
Figura 42: vercamion.ts.....	66
Figura 43: Librerías ubicación.....	66
Figura 44: ubicacion.ts.....	67

Figura 45: Reportar html	68
Figura 46: Importar librerías	68
Figura 47: modal-detalle-sitio.ts	69
Figura 48: notificación.ts	69
Figura 49: src/page/modal-detalle-sitio.ts.....	70
Figura 50: src/pages/reporte.ts	70
Figura 51: app.module.ts	72
Figura 52: Diagrama de conexión en Fritzing.....	73
Figura 53: scripts transmisión base de datos	74
Figura 54: Transmisión a Firebase.....	74
Figura 55: prueba del módulo mostrar vehículo	75
Figura 56: Verificación del módulo mi ubicación	75
Figura 57: Reporte	76
Figura 58: Lista de reportes	76
Figura 59: Notificaciones	77
Figura 60: Eliminar Reporte	77
Figura 61: Pruebas	78

RESUMEN

El servicio de recolección de residuos sólidos es realizado por la municipalidad a través de vehículos recolectores, sin embargo, este servicio no es del todo adecuado puesto que los vehículos pasan en horarios establecidos, y los usuarios no tienen la seguridad de dejar sus residuos en dichos horarios, por lo que se hace necesario mejorar esta forma de recolección de residuos. El presente trabajo de investigación tecnológica se tituló: "Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018", se desarrolló con la finalidad de mantener informado en tiempo real a los ciudadanos mediante notificación y geolocalización, se desarrolló sirviéndose de un sistema de posicionamiento global mediante satélites que permite visualizar en el mapa el trayecto del vehículo, así mismo mediante el módulo reporte de residuos permita al ciudadano realizar reportes de acumulación de residuos en la calle; para así disminuir la contaminación ambiental. Se ha construido un sistema formado por un dispositivo con la capacidad de realizar la geolocalización obteniendo coordenadas exactas de latitud, longitud y comunicarlás a una aplicación móvil siendo este como canal de comunicación entre el sistema y el usuario.

La aplicación del sistema de geolocalización puede llevarse a cabo en diferentes entornos, ya que se puede incrementar más módulos como cámara, display, así mismo no solo se puede utilizar para localizar vehículos sino también para localizar objetos, personas, mascotas.

En el desarrollo del proyecto se empleó la metodología V (verificación y validación) que permite la implementación de Hardware y Software. Se emplearon una serie de tecnologías y herramientas como el uso del firebase, Raspberry Pi, Framework Ionic 3.0 con el fin de lograr una aplicación funcional para teléfonos móviles Smartphone de interfaz amigable.

De esta manera se pudo concluir con el desarrollo de la aplicación, obteniendo como resultado un sistema de fácil uso para el usuario con una aplicación de interfaz amigable y un dispositivo que solo necesita de conectar a una fuente de alimentación de 5v para que funcione.

Palabras clave:

Geolocalización, Raspberry Pi, Framework Ionic, Smartphone, metodología V.

ABSTRACT

The solid waste collection service is carried out by the municipality through collection vehicles, however, this service is not entirely adequate since the vehicles pass at scheduled times, and users do not have the security of leaving their waste in said schedules, so it is necessary to improve this form of waste collection. The present work of technological research was titled: "Development of a geolocation system for the collection of solid waste in the district of San Jerónimo, 2018", was developed with the purpose of keeping citizens informed in real time by means of notification and geolocalization, was developed using a global positioning system through satellites that allows to visualize on the map the path of the vehicle, likewise through the waste reporting module allows the citizen to make reports of waste accumulation in the street; to reduce environmental pollution. A system has been built consisting of a device with the ability to perform geolocation obtaining exact latitude, longitude coordinates and communicate them to a mobile application, this being the communication channel between the system and the user.

The application of the geolocation system can be carried out in different environments, since it can increase more modules such as camera, display, likewise not only can be used to locate vehicles but also to locate objects, people, pets.

In the development of the project the methodology V (verification and validation) that allows the implementation of Hardware and Software was used. A series of technologies and tools were used, such as the use of the firebase, Raspberry Pi, Ionic 3.0 Framework in order to achieve a functional application for Smartphone mobile phones with a friendly interface.

In this way it was possible to conclude with the development of the application, obtaining as a result a user friendly system with a friendly interface application and a device that only needs to connect to a 5v power supply to work.

Key words:

Geolocation, Raspberry Pi, Ionic Framework, Smartphone, methodology V.

CHUMASQA

Kay maytu qillqa ruwasqaypa sutimmi “Tupay q’upa”, chaymi ruwakunlli, runakuna yachananpaq, maypin kachkan wischusqa qanrakuna quñusqa, chaymi willana kanqa chillikuwan. Hinallantaqsi yachasun maypin takyarun chay qanrakuna.

Chaymanta willawasun chay satélite nisqan maypi kachkan carro qancrakuna apaq hinallamantaq yachana kanqa lliw puririnapi wischusqa q’upa warnu kasqanta. Mana llunpay unquy rikurinanpaq kay pachamamanchisma.

Chamikay kay sistema de geolocalizacion willawachik maypi ima sayaytaq kachkan willanapaqpacha chilikuwan qayanapaqpacha runa kunaman.

Kay sistema de geolocalizacion niqan maypipas funcionaman allinta chiqichasqaqa, manan carrukunata tarinallapaqchu, kanmansi tukuy imakunata, allquchakunata, tarinmansi. Lliwchanchis puririnchisqa.

Kay llankaypiqa puririnachim achka programakunata(Hardware y Software) mana sasachakun kanampaq. Llapanchik sumaqta llaqtanchikta purinichinanpaq.

CAPITULO 1: INTRODUCCIÓN

La importancia del cuidado del medio ambiente es hoy en día innegable es por ello que el programa de las Naciones Unidas para el medio ambiente(PNUMA) mantiene en acuerdo a los países para contribuir con leyes que protejan el ambiente, así también motivar a las personas para que se conviertan en agentes activos del desarrollo sostenible y equilibrado. En el Perú el ministerio del ambiente (MINAM) establece como objetivo prioritario a la gestión integral de residuos sólidos a nivel nacional, ya que cuando se trata del cuidado del medio ambiente y de la limpieza pública es ahí donde la recolección de residuos sólidos se convierte en un tema muy importante ya que una buena planificación y logística de esta actividad podría evitar agentes que derivan en la contaminación.

El presente trabajo de investigación tiene como objetivo elaborar un sistema de geolocalización de alerta de recojo de residuos sólidos urbanos con funcionalidad acorde a las tecnologías. Por ello el sistema consta de un módulo gps y un ordenador de placa reducida mediante el cual se obtiene los datos(coordenadas) y envía mediante notificación al usuario: todo ello aprovechando las funcionalidades del teléfono inteligente.

El presente trabajo de investigación consta de 5 capítulos que a continuación se menciona:

CAPITULO I Introducción: datos generales; donde se encuentra el título del proyecto, el autor, asesor y línea de investigación. Planteamiento del problema; Se describe y fundamenta el problema identificado, los objetivos para dar una posible solución, la justificación, viabilidad de la investigación y algunas limitaciones del estudio.

CAPITULO II marco teórico: consta de la descripción de los antecedentes acerca del tema de estudio y de los fundamentos teóricos.

CAPITULO III materiales y métodos: propuesta de solución; se describe la gestión del proyecto y la ejecución del desarrollo del proyecto de acuerdo a la metodología empleada.

CAPITULO IV resultados y discusiones: resultados de la investigación, conclusión y recomendaciones; se establecen las conclusiones de acuerdo a los objetivos planteados, se describe las recomendaciones.

1.1. DATOS GENERALES

1.1.1. TÍTULO DEL PROYECTO

Desarrollo de un sistema de Geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.

1.1.2. AUTOR DEL PROYECTO

Nombres y apellidos : Elizabeth Alejandrina Quintana Olarte

Escuela Profesional : Ingeniería de Sistemas

E-mail : alejandrinae012@gmail.com

1.1.3. Asesor del proyecto

Nombres y apellidos : Msc. Carlos Yinmel Castro Buleje

Departamento Académico : Ingeniería y Tecnología Informática

Categoría docente : Auxiliar

Modalidad : Tiempo completo

E-mail : ccastro@unajma.edu.pe

1.1.4. Línea de investigación

Línea 4: Ingeniería de Software.

1.1.5. Área priorizada del proyecto

0403 0202 Desarrollo y aplicación en sistemas de comunicaciones (Hardware y software).

1.1.6. Institución y lugar de ejecución del proyecto

Municipalidad distrital de San Jerónimo.

1.2. PLANTEAMIENTO DEL PROBLEMA

1.2.1. Realidad problemática

La creciente cantidad de residuos representan un problema medioambiental, social, tecnológico, económico y de salud para la sociedad y para las administraciones locales, según el Ministerio del Ambiente en el Perú un 30% de los residuos quedan en la calle y más de un 50% no terminan en lugares donde se procesan y traten de manera adecuada (plantas de tratamiento de residuos sólidos), debido a que no se implementan políticas medioambientales, sociales y tecnológicas adecuadas que corrijan o ayuden a solucionar estos problemas por parte de las autoridades y también por desinformación de la población conllevando esto a la contaminación ambiental, donde los principales recursos naturales afectados son el agua, suelo y aire.

Para algunos países los residuos evocan imágenes negativas, sin embargo, para otros representan una oportunidad como es el caso de los países europeos que aprovechan los recursos y las tecnologías disponibles y tienen la voluntad de gestionar los residuos de una forma ecológica generando puestos de trabajo y posibilidades de negocio llegando algunos a comprar residuos a otros países como por ejemplo Suecia, así mismo, los países que conforman la Unión Europea generan un alto nivel de crecimiento en los sectores de residuos y reciclado facturando así un estimado de 100 000 millones de euros colaborando así al cuidado del medio ambiente. Acuri & et-al (2007).

Después de 15 años de promulgada la ley general de residuos sólidos, el Perú sufre aún de graves problemas de limpieza pública, cada día somos más habitantes urbanos (ahora 75% de los peruanos vivimos en las ciudades) y cada día en las ciudades el Peruano produce más basura (en promedio un peruano genera más de medio kilo al día). El volumen de basura producido en el Perú está aumentando; hace 10 años era de 13 mil T/día, hoy alcanza las 18 mil T. el 50% de estos residuos no se disponen adecuadamente: tenemos ciudades sucias, calles, ríos, playas y quebradas sucias, etc. Ministerio del ambiente (2015).

El ministerio del ambiente establece que los municipios son las entidades encargadas de dar el servicio de recolección, transporte y disposición final de los residuos sólidos, como señala el capítulo 21 de la agenda 21, así mismo establece que el rol de los vecinos es poner la basura en su lugar, no tirarla en cualquier sitio y almacenarla en un lugar seguro, sin embargo esto no se ve reflejado en muchos departamentos como el caso del departamento de Apurímac, donde por ejemplo uno de los males es el destino final de los residuos recolectados es en botaderos a cielo abierto.

Generando esto un malestar en la población ya que los botaderos de residuos a cielo abierto son fauna nociva transmisora de múltiples enfermedades que representan un peligro para la salud y la seguridad de los pobladores de la zona.

Los sistemas de recolección de residuos sólidos actualmente implantados en el Distrito de San Jerónimo de la provincia de Andahuaylas no optan por estrategias eficientes que optimicen sus recursos ni implantan políticas tecnológicas (TIC) que ayuden a mejorar el proceso de recolección para así colaborar con el servicio de recolección de residuos, así también con el cuidado del medio ambiente.

Por otro lado, tanto los municipios como los propios ciudadanos tienen la responsabilidad de hacer este servicio de recolección de residuos sea más eficiente por lo que se requiere un trabajo en conjunto, de manera que el ciudadano tenga una participación activa. Pues actualmente los ciudadanos por las múltiples actividades que realizan en las horas de la mañana olvidan el hecho de sacar a la calle en los días y horarios preestablecidos, olvidando también estar pendientes del vehículo recolector de residuos, ya que solo se sabe que pasa en días alternados (lunes, miércoles y viernes) y en horas de la mañana de 9:00 am a 10:00 am, así mismo actualmente la Municipalidad consta con un vehículo recolector de residuos, y dos vehículos menores los cuales transitan por las arterias de la ciudad haciendo el recojo diario de los residuos sólidos, sin embargo, estos no son llevados en su totalidad al

botadero a cielo abierto ubicado en el cerro San José, a este problema contribuyen las personas que por falta de cultura arrojan la basura en las calles y también por falta de información del recorrido del camión recolector de residuos y por desconocimiento del tiempo que emplean en su recorrido, esto conlleva a sacar la basura a destiempo (dejando botado en las calles, esquinas o vertiendo al río Chumbao) o no la sacan generando así focos infecciosos, mal aspecto de las calles.

1.3. FORMULACIÓN DEL PROBLEMA

Deficiente gestión de recolección de residuos sólidos municipales en el distrito de San Jerónimo.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un sistema de geolocalización para determinar la ubicación del vehículo recolector de residuos sólidos en el Distrito de San Jerónimo, 2018.

1.4.2. OBJETIVOS ESPECÍFICOS

- ✓ Analizar la arquitectura Raspberry Pi 3
- ✓ Diseñar y desarrollar el módulo reporte de residuos sólidos para el ciudadano
- ✓ Diseñar y desarrollar el módulo mostrar vehículo
- ✓ Realizar las pruebas del funcionamiento de la geolocalización

1.5. JUSTIFICACIÓN

Gestionar los residuos sólidos de una manera integral significa manipularlos correctamente desde la generación hasta la disposición, utilizando las tecnologías emergentes, dándoles un destino final ambientalmente seguro. De esta manera la presente Investigación, geolocalización de alerta de recojo de residuos busca que el ciudadano tenga una participación activa colaborando con la extracción de los residuos en el momento indicado para que no tengan la necesidad de sacar los residuos antes y que estos estén

demasiado tiempo en la vía pública a la espera de ser retirados (generando esto muchas veces que los animales que están por la zona puedan esparcirlo).

Por otra parte, el sistema a desarrollarse será capaz de mostrar en un mapa el recorrido en tiempo real del recolector de residuo, como también contará con la funcionalidad de notificar cuando el vehículo este acercándose a la vivienda del ciudadano que esté utilizando la aplicación, logrando así evitar que el ciudadano sea quien tenga que ocuparse de ingresar periódicamente a la aplicación para ver el recorrido del vehículo.

Los beneficios de este sistema es colaborar con el cuidado del medio ambiente y preservar la salud de los ciudadanos del distrito de San Jerónimo quienes son los beneficiarios directos del sistema.

1.6. VIABILIDAD DE LA INVESTIGACIÓN

1.6.1. VIABILIDAD TÉCNICA:

Es Viable técnicamente, ya que se cuenta con Hardware, software y recurso Humano para el desarrollo del sistema, tales como:

HARDWARE:

- Una Laptop Toshiba Core I5, sistema operativo Windows 7, con procesador de 2.30 GHz.
- Un equipo móvil con sistema operativo Android versión 4.4.4
- Raspberry Pi 3 modelo B
- Gps ublox neo6m
- Tarjeta Sd clase 10 de 16Gb

SOFTWARE:

- Rasbian jessie versión 2.8.2
- Cordova-Apache
- Node.Js v8.0.0

- Framework Ionic 3.20.0
- Firebase

1.6.2. VIABILIDAD ECONÓMICA

El presente proyecto se desarrollará con software libre, pero la adquisición adicional de los equipos son de mediano costo los cuales están accesibles para comprar ya que se cuenta con recursos económicos.

Con estas consideraciones el costo de ejecución del presente proyecto es de S/. 11.705 y será financiado por la tesista.

1.6.3. VIABILIDAD LEGAL

En este proyecto de investigación se hará uso de software Open Source los cuales no cuentan con restricciones legales por ejemplo:

- Raspberry Pi: Fundación Raspberry Pi, lanzamiento 2012, con software open source denominada Raspbian jessie.
- Firebase y cordova: Libre y de código abierto bajo la licencia Apache versión 2.0
- Framework Ionic y Node.Js: de código abierto bajo la licencia de MIT (Instituto de tecnología de Massachusetts).

1.6.4. VIABILIDAD OPERATIVA

Es viable operativamente porque la población hace uso de dispositivos móviles los cuales tiene el sistema operativo y además tienen conocimientos de usuario básico y por ende manejan aplicaciones variadas en el entorno, así mismo la aplicación será de fácil uso y amigable al usuario.

1.7. LIMITACIÓN DEL ESTUDIO

En la investigación se encontraron las posibles limitaciones:

- La aplicación funcionará solo en los Smartphone que tengan el sistema Android a partir de la versión 4.4 en adelante.
- El sistema dependerá de la tecnología Raspberry Pi 3 B.
- Para el usuario final podría suponer una limitación el acceso a internet ya que el aplicativo gestiona los datos de la nube en tiempo real.

CAPITULO 2: MARCO TEÓRICO

2.1. ANTECEDENTES

2.1.1. ANTECEDENTES INTERNACIONALES

Sancho (2017) En su tesis “Desarrollo de un sistema de localización y aplicación móvil para vehículos en aparcamientos” en la Universidad Politécnica de Valencia concluye que: Se ha diseñado un dispositivo capaz de realizar la geolocalización y gestionar esta información de la forma que se ha considerado más adecuada para garantizar su correcto funcionamiento. Se ha logrado la correcta integración de todos los elementos electrónicos en un solo producto en el que se ha cuidado tanto la funcionalidad como la estética. Con la inclusión de la batería se ha conseguido una autonomía elevada del dispositivo resultado útil para otros usos a parte de la localización de vehículos, como pueda ser el control de objetos personales como mochilas o maletas. Relacionado con el punto anterior, cabe destacar la mejora añadida posteriormente de la funcionalidad de comunicación a distancia que permite al usuario saber dónde se encuentra el dispositivo y como llegar hasta él incluso estando fuera del rango de conexión de este.

Diana & Germán (2015) Realizaron una investigación “Prototipo de sistema de vigilancia basado en la internet de las cosas con aplicativo para dispositivos móviles” en la Universidad distrital Francisco José de Caldas ciudad concluye que: El prototipo de sistema de vigilancia es una red de comunicación que captura, procesa y transfiere la información (imágenes) al usuario; esta red de comunicación se estableció a través de la conexión de un subsistema de captura y un subsistema de sincronización. La comunicación entre el subsistema de captura y el dispositivo móvil se estableció a través de la creación de una aplicación móvil para el sistema operativo Android con conexión directa con los servidores de almacenamiento en la nube Dropbox, Google Drive y Amazon S3. La velocidad de sincronización de las imágenes dentro del prototipo depende de la velocidad de la conexión a internet utilizada y de la calidad de las imágenes capturadas.

Reyes et al. (2015) En el artículo “Sistema de monitoreo del LOBOBUS” del Instituto tecnológico de Pachuca benemérita Universidad Autónoma de Puebla, concluye que: La aplicación permite tener un seguimiento del movimiento de cada autobús que incluya el sistema, lo cual permite conocer su localización vía GPS dentro de la ruta asignada. Esto junto con el microcontrolador con Wifi Electric Imp permitió que los datos se muestren en páginas web, así como el desarrollo de una App en Android que permita conocer a partir de la geolocalización en cuanto tiempo llegará la siguiente unidad del transporte LOBOBUS a ubicación de la parada. A diferencia de otras alternativas donde se hace uso de envío de datos vía celular y que implica un costo, la propuesta sólo requiere que durante todo el trayecto de la ruta se tenga cobertura de Internet, con lo cual no existe un costo adicional por el uso del sistema. Como parte del trabajo futuro se mejorará la interface del sistema y se desarrollará una aplicación para dispositivos móviles (tabletas, teléfonos inteligentes, etc.) donde se pueda disponer de la información del teléfono (ubicación), para integrar un icono con la finalidad de mostrar la posición del dispositivo y como llegar a las paradas oficiales del LOBOBUS.

2.1.2. ANTECEDENTES NACIONALES

Guerra (2013) En su tesis “Diseño de un sistema de control domótico y video vigilancia supervisado por un teléfono móvil” en la Pontificia Universidad Católica del Perú concluye que: El diseño del sistema en que consiste esta tesis es factible en el ámbito de una red local, debido a que sus requerimientos de ancho de banda son muy bajos, además no requiere un servidor con grandes capacidades de procesamiento para su funcionamiento ni la instalación de muchos componentes complejos.

Debido al reducido tamaño del microcontrolador de dispositivos, que funciona también como servidor web y central telefónica, la instalación del sistema será sencilla y puede darse en cualquier lugar de la casa. Debido al bajo costo del sistema y a utilizarse hardware y software de libre distribución, se presenta como una muy buena propuesta de negocio. Gracias a las alertas instantáneas que genera el sistema puede ayudar al usuario a prevenir que se complete un robo.

Esquivel (2016) En su tesis “Sistema de geo localización vía web y móvil para mejorar la búsqueda de personas en desastres naturales en la ciudad de Trujillo 2016”, concluye que: Con la implantación del sistema de geolocalización vía web y móvil se logró mejorar la búsqueda de personas desaparecidas en posibles desastres naturales y a la vez prevenirlos. Con los resultados obtenidos se disminuye el tiempo de búsqueda de personas desaparecidas o damnificadas, a través del sistema de geolocalización implantado, en un 49.33%.

2.1.3. ANTECEDENTES LOCALES

Huaraca & et-al (2017) En su proyecto de investigación “Sistema de Geolocalización en tiempo real, para el servicio de transporte universitario” en la Universidad Nacional José María Arguedas concluye que: La función del sistema principalmente es captar las coordenadas exactas de los buses y para ello se les instala un raspberry con su respectivo gps, y a si la raspberry logra enviar las coordenadas a firebase un servicio de Google que te ayuda almacenados datos y actualizándose a tiempo real, y desde la base de datos firebase captaríamos los datos desde nuestro terminal Android a si logrando observar la ubicación actual y exacta de cada uno de los buses en nuestro Smartphone.

2.2. MARCO CONCEPTUAL

2.2.1. RASPBERRY PI

Según Gonzáles (2015) “la Raspberry Pi es una pequeña placa de tamaño similar al de una tarjeta de crédito a la que se puede conectar ratón, teclado y pantalla y tener multitud de usos como ordenador convencional dado que dispone hasta de conexión a internet.”



Figura 1: Raspberry Pi

Fuente: Raspberry Guía del usuario-Upton & Halfacree

Upton & Halfacree (2014) El nombre de Raspberry proviene de la tradición de asignar el nombre de una fruta a un nuevo sistema informático (ejemplo: Acorn, Tangerine, Apple o BlackBerry) acompañado de Pi, que hace referencia al sistema al lenguaje de programación en el que está basado, Python.

Características Generales

Guerra (2013) raspberry pi cuenta con varias funcionalidades y prestaciones que lo hace una opción muy atractiva como controlador de dispositivos:

- Cuenta con una unidad central de procesamiento que funciona a 700MHz, pero que puede ser sobrecorrida hasta 1GHz sin perder su garantía.
- Cuenta con una memoria RAM de 512 MB, que si tenemos en cuenta las aplicaciones en las cuales se utiliza el raspberry pi, es la apropiada para su correcto funcionamiento.
- Cuenta con dos puertos USB, es compatible con la mayoría de periféricos con los cuales es compatible una PC, pueden usarse para conectar todo tipo de periféricos de entrada y de salida.

- Tiene un puerto Ethernet para la conexión a internet, sin embargo también puede darse la conexión usando un adaptador USB para su uso en una red inalámbrica.
- Permite todas las ventajas y facilidades del almacenamiento externo pues cuenta con un puerto para tarjetas SD, SDHC y SDXC, es decir permite un almacenamiento de hasta 2TB.
- Permite una variedad de sistemas operativos de distribución Linux.
- Tiene un consumo energético bastante reducido en 3.5W.

Diana & Germán (2015) “Este microcomputador no incluye un disco duro o unidad de almacenamiento, solo es necesario una tarjeta SD o una MicroSD con un adaptador para reemplazar el disco de estado sólido para el almacenamiento del sistema operativo y los archivos esenciales para su funcionamiento; su diseño y funcionalidad permiten que sea conectado a un monitor de computador o televisor y manejado a través de un teclado y ratón estándar.”

2.2.1.1. TIPOS DE RASPBERRY PI

Upton & Halfacree (2014) indica que “Raspberry Pi se encuentra actualmente disponible en dos modelos diferentes, conocidos como el Modelo A y el Modelo B. Aunque existen diferencias (con el Modelo A sacrificando algo de funcionalidad en razón de disminuir su costo y los requerimientos de energía) ambos comparten un montón de similitudes.” Para este proyecto se utiliza el Raspberry Pi 3 B por su compatibilidad y manejo dinámico.

- **Modelo A**

Upton & Halfacree (2014) “es la más económica de las Raspberrys, el Modelo A está diseñado para ser asequible pero aún flexible. Aparte de ser económica si se compara al Modelo B, el Modelo A consume menos energía y es una buena elección para proyectos que utilizan la energía solar, eólica o de batería. Aunque el SoC BCM2835 del Modelo A es tan poderoso como el del Modelo B, este viene sólo con 256MB de memoria. Esto es una consideración

importante al momento de decidir qué modelo comprar, ya que puede ocasionar que las aplicaciones más complejas se ejecuten lentamente”.

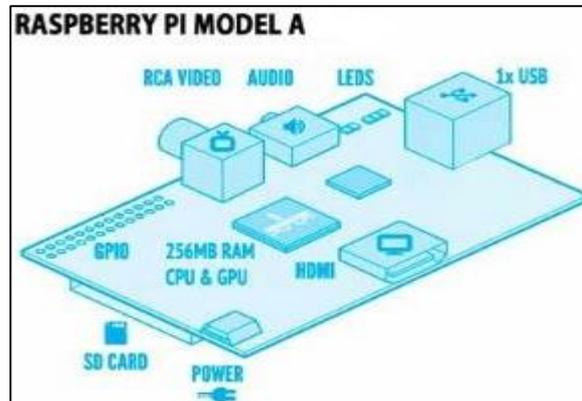


Figura 2: Raspberry Pi Modelo A
Fuente: <https://bit.ly/2sPGQBt>

- **Modelo B**

Upton & Halfacree (2014) “es más costosa que la Modelo A, pero viene con algunas ventajas considerables. En su interior incluye el doble de memoria (512MB), mientras que el exterior hay puertos adicionales que no están disponibles en el modelo económico. Para muchos usuarios, el Modelo B es una inversión que vale la pena; sólo aquellos con requisitos particulares de peso, espacio o consumo energético deben considerar adquirir el Modelo A para su uso general.”

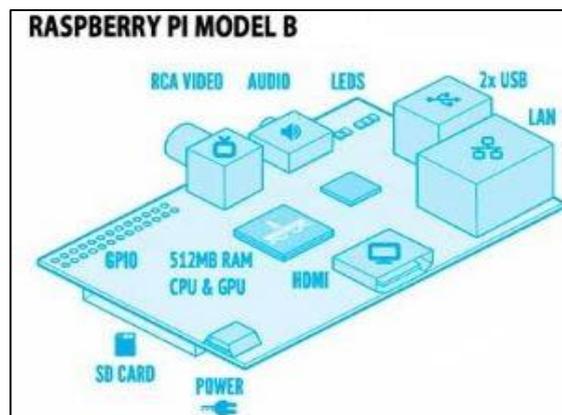


Figura 3: Raspberry pi modelo b
Fuente: <https://bit.ly/2spggbt>

2.2.2. HARDWARE RASPBERRY PI

- **Procesador**

(Diana & Germán , 2015) La Raspberry Pi cuenta con un chip integrado Broadcom BMC 2835 que contiene un procesador ARM11, un procesador gráfico VideoCore IV y diferentes cantidades de memoria de acceso aleatorio RAM. Este chip se encarga de los procesos de audio, video, procesamiento gráfico y comunicación entre las interfaces del teclado y el ratón.

- **Entrada de video**

La Raspberry Pi posee un puerto CSI (Camera Interface Serie) que permite instalar un módulo de cámara y transmitir los datos de pixeles al procesador.

- **Salidas de video**

Las salidas de video compatibles con la Raspberry Pi son: video compuesto, video HDMI(High-Definition Multimedia Interface) y video DSI (Display Serial Interface). El video

compuesto y HDMI son de fácil acceso al usuario y no requieren de un software especial para su manejo. El video compuesto está diseñado para conectar la Raspberry Pi a televisores a través del conector RCA (Radio Corporation America) mientras que el video HDMI permite conectar dispositivos modernos y transmite una calidad de imagen mejor...

- **Salidas de audio**

El audio en la Raspberry Pi puede ser transmitido a una pantalla a través del puerto HDMI. Cuando este puerto está configurado adecuadamente transporta las señales de audio y de video simultáneamente.

- **Puertos USB**

La Raspberry Pi posee puertos USB (Universal Serial Bus) que permiten la conexión de elementos externos como el teclado y el ratón. Una Raspberry Pi puede tener 2 o 4 puertos USB según su clasificación.

- **Puerto Ethernet**

La Raspberry Pi dispone de un puerto Ethernet 10/100 para conectarse a internet cuando no se disponga de una conexión inalámbrica.

- **Fuente de alimentación**

La Raspberry Pi es alimentada por medio de un conector micro USB y requiere de por lo menos 700mA para funcionar correctamente.

- **Puerto GPIO**

El puerto GPIO (General Purpose Input/Output) es el puerto de entrada y salida de propósito general de la Raspberry Pi. Este puerto posee 26 o 40 pines dependiendo del modelo de la placa y permite la interacción de la Raspberry con otros componentes y circuitos electrónicos externos.

Raspberry Pi B+ J8 Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power	Red	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	Blue	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	Blue	Ground	06
07	GPIO04 (GPIO_GCLK)	Green	(TXD0) GPIO14	08
09	Ground	Black	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	Green	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Green	Ground	14
15	GPIO22 (GPIO_GEN3)	Green	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	Red	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Purple	Ground	20
21	GPIO09 (SPI_MISO)	Purple	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	Purple	(SPI_CE0_N) GPIO08	24
25	Ground	Black	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	Yellow	(I2C ID EEPROM) ID_SC	28
29	GPIO05	Green	Ground	30
31	GPIO06	Green	GPIO12	32
33	GPIO13	Green	Ground	34
35	GPIO19	Green	GPIO16	36
37	GPIO26	Green	GPIO20	38
39	Ground	Black	GPIO21	40

Figura 4: Puerto GPIO

Fuente: (González & Palomo, 2015) pdf memoria tfg

Como se puede observar en la Figura 5 la Raspberry Pi B cuenta con dos filas de 20 pines cada una. Estos pines sirven de interfaz

entre la Raspberry Pi y el exterior. GPIO (General Purpose Input/Output) es un sistema de Entrada/Salida de propósito general que te permite dar un uso más amplio y realizar multitud de proyectos. Con estos pines se pueden controlar multitud de dispositivos como luces leds, motores o relés.

2.2.3. SOFTWARE PARA RASPBERRY PI

Gonzales & Marin (2015) “El Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse.

El chip ARM11 en el corazón del Pi (modelos de primera generación) se basa en la versión 6 de la ARM . Las versiones actuales de varias versiones populares de Linux, incluyendo Ubuntu , no se ejecutarán en el ARM11. No es posible ejecutar Windows en el Raspberry Pi original, aunque en el nuevo Raspberry Pi 2 es capaz de ejecutar Windows 10 . El Raspberry Pi 2 actualmente sólo es compatible con Ubuntu Snappy Core , Raspbian , OpenELEC y RISC OS.”

2.2.4. DISPOSITIVOS MÓVILES

De acuerdo a Alonso & et al (2011) Un dispositivo móvil se puede definir como un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los teléfonos móviles, los PDAs o los Tablet PCs.

A. TIPOS DE DISPOSITIVOS MOVILES

Según Chiliquinga (2010) para competir en el mercado, cada fabricante equipa a sus dispositivos con diversas características esperando atraer al mayor número de usuarios disponibles. El dispositivo más común actualmente es el teléfono móvil, seguido de los conocidos PDAs (Personal Data Assistant). Todos los dispositivos disponen en mayor o menor medida de posibilidades de desarrollo.

Los fabricantes permiten que terceras partes puedan desarrollar aplicaciones que los usuarios puedan instalar y disfrutar en los dispositivos. La aplicación más común encontrada en estos dispositivos suele ser un navegador Web, adaptado por supuesto al tamaño y forma del aparato en cuestión. Entre los tipos de dispositivos móviles tenemos:

- Notebooks
- PDas
- Smartphones
- Telefonos celulares

B. PDA

Martínez (2011, pág. 8) Una PDA (del inglés Personal Digital Assistant (asistente digital personal)), es una computadora de mano originalmente diseñada como agenda electrónica con un sistema de reconocimiento de escritura. Hoy día (2011) estos dispositivos pueden realizar muchas de las funciones de una computadora de escritorio pero con la ventaja de ser portátil.

C. SMARTPHONE O TELEFONOS INTELIGENTES:

Alonso & et al (2011, pág. 3) Un “smartphone” (teléfono inteligente en español) es un dispositivo electrónico que funciona como un teléfono móvil con características similares a las de un ordenador personal. Es un elemento a medio camino entre un teléfono móvil clásico y una PDA ya que permite hacer llamadas y enviar mensajes de texto como un móvil convencional pero además incluye características cercanas a las

de un ordenador personal. Una característica importante de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero.

CARACTERÍSTICAS DE LOS DISPOSITIVOS MÓVILES:

Chiliquina (2010, pág. 24) Algunas de las características que hacen que estos dispositivos sean diferentes de los ordenadores de sobremesa son los siguientes:

- Funcionalidad limitada.
- No necesariamente extensible y actualizable.
- Menos complicado en su manejo.
- Fácil de aprender su operación.

2.2.5. APLICACIÓN MÓVILES

De acuerdo a Gamboa Safla (2015) define una aplicación móvil como “un software desarrollado específicamente para dispositivos móviles que realiza una cierta tarea específica como Facebook, Twitter, WhatsApp, Messenger, juegos, calendario, reproductor de música, etc.”

2.2.6. SISTEMAS OPERATIVOS

Wolf & et-al (2014) Manifiesta que “el sistema operativo es el principal programa que se ejecuta en toda computadora de propósito general. Los hay de todo tipo, desde muy simples hasta complejos, y entre más casos de uso hay para el cómputo en la vida diaria, más variedad habrá en ellos”.

2.2.7. SISTEMA OPERATIVO MÓVILES

Un sistema operativo móvil es un sistema que controla un dispositivo móvil al igual que las computadoras más grandes utilizan Windows, Linux o Mac OS entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica,

los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos

Pedrozo (2012) indica que los componentes de un sistema operativo móvil son los siguientes:

- **Capas:** Al igual que los sistemas operativos presentes en los componentes informáticos más grandes, un sistema operativo móvil también se encuentra compuesto por varias capas.
- **Kernel:** Una de las más importantes y esenciales piezas que componen cualquier sistema operativo, sea el de nuestro móvil, o el de la PC, es el denominado núcleo o Kernel, el cual es la capa de software que permite el acceso a los diferentes elementos de hardware que conforman nuestro móvil.

También es el encargado de brindar diferentes servicios a las capas superiores como los controladores de hardware, gestión de procesos, sistemas de archivos, además del acceso y administración de la memoria del sistema.

- **Middleware:** Esta capa es el conjunto de módulos que permite que las aplicaciones diseñadas y escritas para tales plataformas puedan ser ejecutadas.

Su funcionamiento es totalmente transparente para el usuario, no debiendo realizar ninguna acción ni configurar alguna para su correcto desenvolvimiento.

El Middleware brinda la posibilidad de ejecutar servicios muy importantes para que otras aplicaciones, en capas superiores de la jerarquía, puedan ejecutarse.

- **Entorno de ejecución de aplicaciones:** Esta capa provee de todos los elementos necesarios para la creación y desarrollo de software es decir contiene elementos que serán de gran ayuda a los mismos, en el momento de escribir aplicaciones compatibles con ese sistema operativo.
- **Interfaz de Usuario:** La interfaz de usuario es el elemento del teléfono que usualmente utilizamos para interactuar con el aparato.

Sin esta capa no sería posible utilizar nuestro dispositivo, ya que presenta todos los elementos necesarios para facilitar cualquier tipo de tarea que deseemos realizar en nuestra terminal. Además incluye todos los elementos gráficos que harán posible el uso cómodo y sencillo del móvil: botones, menús, pantallas y listas, entre otros.

A. TIPOS DE SISTEMAS OPERATIVOS MÓVILES

Se cuentan con una gran cantidad de sistemas operativos móviles y para el presente proyecto usaremos el SO android, los sistemas operativos más importantes son:

- **Android:** Pedrozo (2012) Es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google. Este sistema por lo general maneja aplicaciones como Google Play. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.
- **iOS:** Wolf & et-al (2014) El sistema operativo de Apple, y diseñado exclusivamente para el hardware producido por dicha compañía. Fue el primero en implementar la interfaz usuario multitouch y, en buena medida, se puede ver como el responsable de la explosión y universalización en el uso de dispositivos móviles. Al igual que el sistema operativo que emplean para sus equipos de escritorio, MacOS X, iOS está basado en el núcleo Darwin, derivado de FreeBSD, un sistema libre tipo Unix.
- **Windows Phone:** Gómez (2013) es un sistema operativo desarrollado por la compañía Microsoft, como sucesor de la plataforma Windows Mobile. Está enfocado en el mercado de

consumo generalista, en lugar del mercado empresarial al que estaba destinado el Windows Mobile. Windows pone y Windows Mobile no son compatibles, con lo que todas las aplicaciones que existían de Windows Mobile se quedaron obsoletas con el paso del tiempo.

Los sistemas operativos móviles más usados en Perú son:

Tabla 1: Sistemas Operativos en Perú

Sistema Operativo	Usuario
Android	72,6%
Apple iOS	8,0%
Windows Phone	6,3%
Otros	13,1%

Fuente: ComScore, CNN Expansión

2.2.8. BASE DE DATOS FIREBASE

Firebase (2016) “Es una plataforma que facilita el desarrollo de aplicaciones, proporcionando a sus usuarios diversas funcionalidades. En este proyecto interviene únicamente el servicio de bases de datos. Se trata una base de datos NoSQL en la nube que almacena los datos en formato JSON, accesible mediante diversas APIs.”

- **noSQL:** Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, su funcionalidad y el rendimiento a escala. Usan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en memoria y búsqueda. Amazon (2018).

2.2.9. IONIC

Ionic Framework es un SDK de código abierto que permite a los desarrolladores crear aplicaciones móviles de gran calidad y rendimiento utilizando tecnologías web conocidas (HTML, CSS y JavaScript). Ionic se centra principalmente en la apariencia y la interacción de la interfaz de usuario de una aplicación. Esto significa que no es un reemplazo para

Cordova o su marco de JavaScript favorito. En su lugar, Ionic encaja bien con estos proyectos, para simplificar una gran parte de su proceso de desarrollo de aplicaciones: el front-end. Ionic(2018).

2.2.10. GEOLOCALIZACIÓN

Martínez (2011) indica que “El GPS (Global Positioning System) o NAVSTAR-GPS es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo, la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos de América.”

2.2.11. CONTAMINACIÓN

Según Bermúdez (2010) “la contaminación es la presencia o incorporación al ambiente de sustancias o elementos tóxicos que son perjudiciales para el hombre o los ecosistemas (seres vivos). Existen diferentes tipos de contaminación, Los tipos de contaminación más importantes son los que afectan a los recursos naturales básicos: el aire, los suelos y el agua. Existen diferentes tipos de contaminación que dependen de determinados factores y que afectan distintamente a cada ambiente”.

Tipos de contaminación: Bermúdez (2010) en su investigación define los principales tipos de contaminación son:

- **Contaminación del agua:** Se entiende por contaminación del medio hídrico a la acción o al efecto de introducir materiales o inducir condiciones sobre el agua que, de modo directo o indirecto, impliquen una alteración perjudicial de su calidad.
- **Contaminación del aire:** Se entiende por contaminación atmosférica a la presencia en el aire de materias o formas de energía que impliquen riesgo, daño o molestia grave para las personas y bienes de cualquier naturaleza.

- **Contaminación del suelo:** Cuando en el suelo se deposita de forma voluntaria o accidental diversos productos como papel, vidrio, plástico, materia orgánica, solventes, plaguicidas, residuos peligrosos o sustancias radioactivas, etc.

2.2.12. RESIDUOS SÓLIDOS

Según el Ministerio del Ambiente (2015) “Los residuos sólidos son todos aquellos materiales que resultan inservibles posterior a su utilización por parte de un generador o usuario, estos desechos no representan ninguna utilidad o valor económico para quien lo origina. De acuerdo a la Ley N° 27314, los residuos sólidos se clasifican por su origen en: residuos domiciliarios, residuos comerciales, residuos de limpieza de espacios públicos, residuos de establecimientos de atención de salud, residuos industriales, residuos de las actividades de construcción, residuos agropecuarios y residuos de instalaciones o actividades especiales.”

A. LEY GENERAL DE RESIDUOS SÓLIDOS

Ministerio del Ambiente (2015) La Ley establece derechos, obligaciones, atribuciones y responsabilidades de la sociedad en su conjunto, para asegurar una gestión y manejo de los residuos sólidos, sanitaria y ambientalmente adecuada, con sujeción a los principios de minimización, prevención de riesgos ambientales y protección de la salud y el bienestar de la persona humana. La gestión de los residuos sólidos en el país tiene como finalidad su manejo integral y sostenible, mediante la articulación, integración y compatibilización de las políticas, planes, programas estrategias y acciones de quienes intervienen en la gestión y el manejo de los residuos sólidos, aplicándolos lineamientos de política que se establecen en el siguiente artículo.

2.2.13. GESTIÓN AMBIENTAL

De acuerdo al Ministerio del Ambiente (2005) “La gestión ambiental es un proceso permanente y continuo, constituido por el conjunto estructurado de principios, normas técnicas, procesos y actividades, orientado a

administrar los intereses, expectativas y recursos relacionados con los objetivos de la política ambiental y alcanzar así, una mejor calidad de vida y el desarrollo integral de la población, el desarrollo de las actividades económicas y la conservación del patrimonio ambiental y natural del país. La gestión ambiental se rige por los principios establecidos en la presente Ley y en las leyes y otras normas sobre la materia.”

2.2.14. METODOLOGÍA V

A. Perez (2006) La metodología V es un modelo para desarrollo de Software embebido, que logra la combinación del software y Hardware de confianza realizada dirigido a diversos campos de aplicación industrial donde priman las limitaciones de coste, tiempo y confiabilidad.

En el siguiente grafico se muestra el ciclo de vida del modelo V:

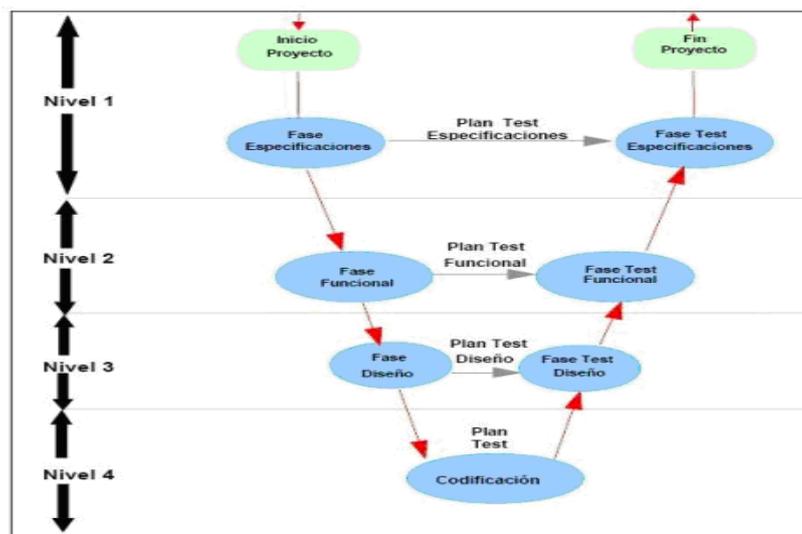


Figura 5: Modelo en V

Fuente: Metodología de Desarrollo de Software

NIVELES:

- **El nivel 1:** Está orientado al “cliente”. El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del análisis de requisitos y especificaciones, se traduce en un documento de requisitos y especificaciones.
- **El nivel 2:** Se dedica a las características funcionales del sistema propuesto. Puede considerarse el sistema como una caja negra, y

caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.

- **El nivel 3:** Define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema.
- **El nivel 4:** Es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.

FASES DEL DESARROLLO

- **Fase 1:** Especificaciones. Se deben definir y documentar los diferentes requisitos del sistema a desarrollar, identificando los valores numéricos más concretos posibles.
- **Fase 2:** Diseño Preliminar. También llamado diseño de alto nivel. Su objetivo es obtener un diseño y visión general de sistema.
- **Fase 3:** Diseño en Detalle. Consiste en detallar de una manera más detenida cada parte de la segunda fase.
- **Fase 4:** Programación. En esta fase procederemos a materializar el diseño a detalle.
- **Fase 5:** Prueba de unidad. En esta fase se verifica cada módulo Hardware y Software de forma unitaria, comprobando su funcionamiento adecuado.
- **Fase 6:** integración. En esta fase se integran los distintos módulos que forman el sistema. Como en el caso anterior, ha de generarse un documento de pruebas. Por una parte, se debe comprobar en todo el sistema el funcionamiento correcto, y por otra, en caso de tratarse con un sistema tolerante a fallos, debe verificarse que ante la presencia de un fallo persiste el funcionamiento correcto.

La metodología V es ideal por su robustez y es la adecuada para el desarrollo de la aplicación ya que para cada fase del desarrollo existe una fase de verificación o validación lo cual permite subsanar errores, requerimientos, modificaciones que se pueda tener.

CAPITULO 3: MATERIALES Y METODOS

La metodología para la gestión del proyecto en la presente investigación denominada DESARROLLO DE UN SISTEMA DE GEOLOCALIZACION DE ALERTA DE RECOJO DE RESIDUOS SÓLIDOS es PMBOK (Guía de los Fundamentos de Gestión de proyectos) sexta edición y para el desarrollo del proyecto se opta la metodología V (Verificación y Validación).

El PMBOK establece 10 áreas de conocimiento, con las cuales se desarrolla el presente proyecto:



Figura 6: Áreas del conocimiento PMBOK
Elaboración propia

3.1. ADMINISTRACIÓN Y GESTIÓN DEL PROYECTO BASADO EN PMBOK

3.1.1. Plan de gestión de la integración

PMBOK (2017) La Gestión de la Integración del Proyecto incluye los procesos y actividades necesarios para identificar, definir, combinar, unificar y coordinar los diversos procesos y actividades de dirección del proyecto dentro de los Grupos de Procesos de la Dirección de Proyectos.

Tabla 2: Plan de Gestión de la Integración

PROYECTO	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.	
PATROCINADO, PREPARADO Y REVISADO POR	Elizabeth Alejandrina Quintana Olarte - Jefe del Proyecto.	
REVISIÓN (Correlativo)	DESCRIPCIÓN (REALIZADA POR) (Motivo de la revisión y entre paréntesis quien la realizó)	
01	Preparación de Acta de Constitución (Elizabeth Alejandrina Quintana Olarte)	
BREVE DESCRIPCIÓN DEL PRODUCTO O SERVICIO DEL PROYECTO (Características, funcionalidades, soporte entre otros)		
La presente Investigación tiene como finalidad desarrollar una aplicación móvil, haciendo uso de las herramientas Ionic y Firebase los cuales permitirán crear una aplicación de interfaz amigable y con interacción en tiempo real. Del mismo modo se hará uso de las herramientas tecnológicas disponibles para lograr la integración entre el dispositivo móvil y el hardware, permitiendo a los ciudadanos determinar la ubicación del vehículo recolector de residuos sólidos urbanos a su vez notifique al usuario que retire los residuos en el momento indicado, así mismo mediante el módulo reporte el usuario podrá visualizar e informar de la acumulación de residuos sólidos para su posterior recojo.		
ALINEAMIENTO DEL PROYECTO		
1. OBJETIVOS ESTRATEGICOS DE LA ORGANIZACIÓN (A qué objetivo estratégico se alinea el proyecto)	2. PROPÓSITO DEL PROYECTO (Beneficios que tendrá la organización una vez que el producto del proyecto esté operativo o sea entregado)	
1.1. Desarrollar una aplicación donde se pueda observar el vehículo en movimiento 1.2. Investigar sobre la arquitectura Raspberry Pi	2.1. Impulsar la creación de aplicaciones con el uso de framework y aplicarlos en la mejora de los las mismas. 2.2. Promover la investigación del desarrollo de aplicaciones haciendo uso de hardware y software libre para automatizar procesos.	
3. OBJETIVOS DEL PROYECTO (Principalmente en términos de costo, tiempo alcance, calidad)		
3.1. Desarrollar una aplicación móvil híbrida. 3.2. Realizar talleres para capacitar al personal de la municipalidad de San Jerónimo en el uso de la aplicación e instalación del hardware. 3.3. Lograr que la aplicación envíe alertas (notificaciones). 3.4. Desarrollar el módulo de reporte residuos sólidos.		

3.5. Finalizar el proyecto en un plazo de 5 meses a partir de la fecha del Acta de Constitución del Proyecto.	
4. FACTORES CRÍTICOS DE ÉXITO DEL PROYECTO (Componentes o características que deben cumplirse en el proyecto para considerar exitoso)	
4.1. Proceso de adquisición de software y hardware dentro de las fechas establecidas y con las características especificadas. 4.2. Diseño del software conforme a los requerimientos establecidos por parte de los interesados del proyecto (stakeholder). 4.3. Notificación a los usuarios de la distancia en la que se encuentra en vehículo recolector con respecto a su punto de origen. 4.4. Reporte de acumulación de residuos sólidos en la calle.	
5. REQUERIMIENTOS DE ALTO NIVEL (Principales condiciones y/o capacidades que debe cumplir el producto o servicio y la gestión del proyecto)	
5.1. Investigación de las herramientas que se usarán en el proyecto. 5.2. Diseñar el módulo de porte de residuos. 5.3. Implementar e implantar los dispositivos (hardware). 5.4. Realizar las pruebas del funcionamiento. 5.5. Mediante el gps ublox neo6m se podrá determinar la latitud, longitud y altitud para posteriormente ser visualizada en la aplicación.	
EXTENSIÓN Y ALCANCE DEL PROYECTO	
6. FASES DEL PROYECTO (Agrupamiento lógico de actividades relacionadas que usualmente culminan elaborando un entregable principal).	7. PRINCIPALES ENTREGABLES (Un único y verificable producto, resultado o capacidad de realizar un servicio que debe ser elaborado para contemplar un proceso, una fase o un proyecto).
Fase I – Gestión del Proyecto	Acta de Constitución del Proyecto.
Fase II – Diseño	Proceso de definición de las interfaces, y de la arquitectura.
Fase III – Requerimiento de Adquisiciones	Adquisición de laptop, raspberry Pi, Gps ublox neo6m, otros componentes. Adquisición del entorno del desarrollador(Firebase, Ionic , entre otros)
Fase IV – Implementación	Implementar el Módulo de la aplicación.
Fase V – Arranque del Sistema	Pruebas a la aplicación.
Fase VI – Término de Obra	Aplicación móvil.
8. INTERESADOS CLAVE (Persona u organización que ésta activamente involucrado en el proyecto o cuyos intereses pueden ser afectados positivamente o negativamente por la ejecución del proyecto o por el producto que elabora).	
8.1. Habitantes del distrito de San Jerónimo – Andahuaylas.	

8.2. Municipalidad del distrito de San Jerónimo - Andahuaylas.
9. RIESGOS (Eventos o condiciones inciertas que, si ocurriese, tiene un efecto positivo o negativo sobre los objetivos del proyecto)
9.1. Demora en la adquisición de las herramientas de hardware para el desarrollo. 9.2. Cambio de las políticas y licencia en el uso de los frameworks. 9.3. Falta de energía eléctrica y de cobertura de internet 9.4. Restricciones de uso limitado en la plataforma firebase
10. HITOS PRINCIPALES DEL PROYECTO (Un evento significativo para el proyecto)
1. Aprobación del acta de constitución del proyecto 2. Alcanzar todos los objetivos trazados
11. PRESUPUESTO DEL PROYECTO (La estimación aprobada para el proyecto o cualquier otro componente de la estructura de desglose de trabajo, u otra actividad del cronograma)
El costo total de la ejecución del proyecto será asumido por el Jefe del Proyecto.

Elaboración Propia

3.1.2. PLAN DE GESTIÓN DE ALCANCE

Incluye los procesos requeridos para garantizar que el proyecto incluye todo el trabajo requerido y únicamente el trabajo requerido para completarlo con éxito.

Tabla 3: Plan de Gestión de Alcance

NOMBRE DEL PROYECTO:	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.
PREPARADO POR:	Elizabeth Alejandrina Quintana Olarte – Jefe del Proyecto
1. Describir cómo será administrado el alcance del Proyecto:	Primero se estableció a quienes estará dirigido el proyecto, a partir de ello se determinó que la aplicación móvil será usada por los habitantes del distrito de San Jerónimo.
2. Evaluar la estabilidad del alcance del proyecto(cómo manejar los cambios, la frecuencia e impacto de los mismos):	Los cambios del proyecto deben ser evaluados y aprobados. El jefe de proyectos debe cuantificar el impacto y proveer alternativas de solución. Los cambios solicitados serán revisados en las reuniones semanales, se indica el estado de los mismos en la reunión siguiente a la que fue solicitado.
3. ¿Cómo los cambios al alcance, serán identificados y clasificados?:	

El jefe del proyecto, revisará la solicitud de cambios del alcance quien hará una evaluación del mismo, de la misma manera si se necesita información adicional el podrá requerir del solicitante.
4. Descripción cómo los cambios del alcance serán integrados al proyecto:
Si el impacto del campo no modifica la línea base del proyecto será aprobado por el jefe del proyecto, y se actualizará las líneas base de todos los planes del proyecto.
5. Comentarios adicionales:
Se necesita integrar adecuadamente los dispositivos móviles con la plataforma de Rasperry, así como también integrar la conexión con firebase.

Elaboración propia

3.1.3. PLAN DE GESTIÓN DEL CRONOGRAMA

Incluye los procesos involucrados en planificar, estimar, presupuestar, financiar, obtener financiamiento, gestionar y controlar los costos de modo que se complete el proyecto dentro del presupuesto aprobado.

Tabla 4: Plan de Gestión de cronograma

NOMBRE DEL PROYECTO:	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.	
PREPARADO POR:	Elizabeth Alejandrina Quintana Olarte – Jefe del Proyecto	
Persona(s) autorizada(s) a solicitar cambio en cronograma:		
Nombre	Cargo	Ubicación
Elizabeth Alejandrina Quintana Olarte	Jefe del Proyecto	San Jerónimo
Persona(s) que aprueba(n) el requerimiento de cambio de cronograma:		
Nombre	Cargo	Ubicación
Elizabeth Alejandrina Quintana Olarte	Jefe del Proyecto	San Jerónimo
Razones aceptables para cambios en cronograma del proyecto: (por ejemplo, retrasos debido a entrega de materiales o disponibilidad de personal, clima, adelantar el cumplimiento debido a término de fase o proceso, etc.)		
<ul style="list-style-type: none"> - Solicitud de cambio de alcance por parte del usuario y por parte de la municipalidad. - Mal establecimiento de la secuencia de actividades. Fallos o averías del servicio de internet. - Fallos de los equipos de trabajo (laptop, gps, rasperry y otros accesorios). - Situaciones imprevistas ajenas o de fuerza mayor. 		

<p>Describir cómo calcular y reportar el impacto en el proyecto por el cambio en cronograma: (Tiempo, costo, calidad. Etc.):</p> <ul style="list-style-type: none"> - En el informe se indicará a la persona responsable del mismo y fecha de ocurrencia del problema. - Descripción del problema indicando el grado de urgencia. - Impacto del mismo sobre el proyecto (Costo, calidad, tiempo y alcance). - Describir las alternativas de solución detallando el impacto en las diferentes áreas (costo, calidad, tiempo y alcance).
<p>Describir cómo los cambios al cronograma serán administrados:</p> <p>La administración del alcance se realizará de la siguiente manera:</p> <p>1) Designación de Responsabilidades:</p> <ul style="list-style-type: none"> • Planificación (Jefe del proyecto). • Ejecución (Jefe del proyecto). • Seguimiento y control (Jefe del proyecto).

Elaboración propia

3.1.4. PLAN DE GESTIÓN DE LOS COSTOS

Incluye los procesos involucrados en planificar, estimar, presupuestar, financiar, obtener financiamiento, gestionar y controlar los costos de modo que se complete el proyecto dentro del presupuesto aprobado.

Tabla 5: Plan de Gestión de los Costos

PROYECTO	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.	
PREPARADO, REVISADO Y APROBADO POR	Elizabeth Alejandrina Quintana Olarte – Jefe del Proyecto.	
Persona(s) autorizada(s) a solicitar cambio en el costo:		
Nombre	Cargo	Ubicación
Elizabeth Alejandrina Quintana Olarte	Jefe del Proyecto	San Jerónimo
Persona(s) que aprueba(n) el requerimiento de cambio en costo contractual:		
Nombre	Cargo	Ubicación
Elizabeth Alejandrina Quintana Olarte	Jefe del Proyecto	San Jerónimo
Persona(s) que aprueba(n) el requerimientos de cambio de costo interno ofrecido:		
- Jefe del Proyecto		
Razones aceptables para cambios en el Costo del Proyecto: (por ejemplo: Aprobación de cambios en el alcance, incremento de costo en los materiales, etc.):		
<ul style="list-style-type: none"> - Ampliación en el alcance del proyecto. - Ampliación de plazos en el alcance del proyecto. - Cambios en las fechas de entrega. 		

- Adecuación en el alcance del proyecto (cambios).
Describir cómo calcular e informar el impacto en el proyecto por el cambio en costo: (Tiempo, costo, calidad. Etc.):
- El responsable del desarrollo de la aplicación solicita cambio de los costos acerca de las razones aceptables en el costo del proyecto
- Descripción de las características de la situación que requiere una solicitud de cambio de los costos
- Describir las alternativas de solución detallando el impacto en las diversas áreas (Calidad, costo, tiempo y alcance)
- Recomendación en la selección de la alternativa de solución(Propuesta)

Elaboración Propia

3.1.5. PLAN DE GESTIÓN DE LA CALIDAD

Incluye los procesos para incorporar la política de calidad de la organización en cuanto a la planificación, gestión y control de los requisitos de calidad del proyecto y el producto, a fin de satisfacer las expectativas de los interesados.

Tabla 6: Plan de Gestión de la Calidad

PROYECTO	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.
PREPARADO, REVISADO Y APROBADO POR	Elizabeth Alejandrina Quintana Olarte – Jefe del Proyecto
GESTIÓN DE CALIDAD DEL PROYECTO (Descripción de cómo se van a aplicar los procesos de gestión de calidad del proyecto. Herramientas a emplear, normativas, reglamentos, responsables, áreas de aplicación, etc.)	
PLANIFICAR LA CALIDAD: (Explicar y sustentar cómo se ha elaborado el presente plan de gestión de calidad)	
<p>El plan de gestión de la calidad es un componente del plan para la dirección del proyecto que describe cómo se implementarán las políticas, procedimientos y pautas aplicables para alcanzar los objetivos de calidad. Describe las actividades y los recursos necesarios para que el equipo de dirección del proyecto alcance los objetivos de calidad establecidos para el proyecto. El plan de gestión de la calidad puede ser formal o informal, detallado o formulado de manera general. (PMBOK, 2017).</p> <p>Para ello se aplicará, la calidad del software (SQA, software Quality Assurance), es un conjunto de actividades previamente planificadas y sistemáticas necesarias para la contribución a la confianza en que el software satisfaga los requisitos de calidad, es aplicado a lo largo de todo el proceso del software.</p>	

<p>REALIZAR ASEGURAMIENTO DE CALIDAD (Explica cómo se va a realizar el proceso de aseguramiento de calidad)</p> <p>Aseguramiento de Calidad: La ejecución del aseguramiento de calidad durante todo el proyecto es responsabilidad del jefe del proyecto, que supervisa el planeamiento de los procesos del proyecto contra lo que sea ejecutado y plantea acciones preventivas o correctivas según sean necesarias.</p>
<p>REALIZAR CONTROL DE CALIDAD (Explicar cómo se va realizar el proceso de control de calidad)</p> <p>Control de Calidad: El jefe del proyecto es responsable de la ejecución del control de la calidad. Los entregables del proyecto son revisados conforme se vayan presentado, se emiten las observaciones o conformidades en la reunión semanal de calidad. Se definen dos procedimientos para el control de calidad:</p> <ul style="list-style-type: none"> - Revisión de contenidos - Revisión de forma <p>Control de calidad del contenido: Se revisan la calidad de los entregables, las observaciones que se puedan formular, se informan en la reunión semanal de calidad.</p> <p>Control de calidad de redacción y formato: Se revisa la redacción de los documentos entregables, las observaciones que se puedan formular, se informan en la reunión semanal de calidad.</p>

Elaboración Propia

3.1.6. PLAN DE GESTIÓN DE LOS RECURSOS

Incluye los procesos para identificar, adquirir y gestionar los recursos necesarios para la conclusión exitosa del proyecto.

Tabla 7: Plan de Gestión de los Recursos

PROYECTO	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.
PREPARADO, REVISADO Y APROBADO POR	Elizabeth Alejandrina Quintana Olarte
ROL EN EL PROYECTO (Denominación del puesto)	Jefe del Proyecto
FUNCION / RESPONSABILIDAD PRINCIPAL (Descripción de los que se espera de este rol)	
<ul style="list-style-type: none"> - El Jefe del Proyecto será responsable de la ejecución y entrega del proyecto - Es responsable de la elección del personal que integra el equipo de proyecto se su organización y gestión - Supervisa el correcto funcionamiento del sistema - Controla la información de gasto del proyecto 	
COMPETENCIAS REQUERIDAS / RESPONSABILIDADES	

(Conocimiento, habilidades, actitudes)				
DISPONIBILIDAD (Posibilidad de contar con el recurso humano en el tiempo requerido)				
Descripción del puesto a requerir	Externo interno	Fecha ingreso	Fecha retiro	Tiempo requerido
Jefe del Proyecto	Interno	30/04/2018	29/08/2018	4 meses
INCORPORACIÓN AL PROYECTO (Cómo se incorpora el personal al proyecto)				
Evento disipador (Hecho que identifica la necesidad de haber gestiones sobre el personal)		Sincronización previa (Acciones requeridas para gestionar al personal)		
Renuncia		Inmediato con la designación de reemplazo por el jefe del proyecto.		
Enfermedad		Inmediato con la designación de reemplazo por el jefe del proyecto.		

Elaboración Propia

3.1.7. PLAN DE GESTIÓN DE LAS COMUNICACIONES

Incluye los procesos requeridos para garantizar que la planificación, recopilación, creación, distribución, almacenamiento, recuperación, gestión, control, monitoreo y disposición final de la información del proyecto sean oportunos y adecuados.

Tabla 8: Plan de Gestión de las Comunicaciones

PROYECTO	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.
PREPARADO, REVISADO Y APROBADO POR	Elizabeth Alejandrina Quintana Olarte
GESTIÓN DE COMUNICACIONES DEL PROYECTO (Descripción de cómo se va a aplicar los procesos de gestión de comunicaciones del proyecto, herramientas a emplear, responsables, áreas de aplicaciones, etc.)	
El responsable es el Jefe del Proyecto, estará a cargo de la integración de la consecución de herramientas hardware para la elaboración del desarrollo del sistema de geolocalización: Raspberry pi 3 modelo B, gps ublox neo6m, cables jumper, modem.	
PLANIFICACIÓN DE LA COMUNICACIÓN (Explicar y sustentar cómo se ha elaborado el presente plan de gestión de comunicación)	
Determinar las necesidades e informaciones y comunicación de los interesados: quién necesita qué información, para cuando la necesita, cómo le será suministrada y por quién. El jefe del proyecto debe	

considerar la cantidad de canales necesarios desde un principio con el fin de que la información fluya y llegue a todos los involucrados.
 Los requisitos de comunicación que influyen:
 - Organigramas
 - Relación de responsabilidades de la organización

Elaboración Propia

3.1.8. PLAN DE GESTIÓN DE RIESGO

Incluye los procesos para llevar a cabo la planificación de la gestión, identificación, análisis, planificación de respuesta, implementación de respuesta y monitoreo de los riesgos de un proyecto.

Tabla 9: Plan de Gestión de Riesgo

PROYECTO	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.
PREPARADO POR	Elizabeth Alejandrina Quintana Olarte
Descripción de la metodología de gestión del riesgo a ser usada:	
Alcances	
- El jefe del proyecto realizará la identificación, priorización y seguimiento de los riesgos más críticos	
Herramientas	
- Lluvia de ideas - Opinión de la alta gerencia - feedback del usuario	
Fuentes de datos	
- La identificación de los riesgos será por parte de todos los integrantes del proyecto según experiencia y juicio de especialistas	
Roles y Responsabilidades:	
- Jefe del proyecto: Es el responsable de la identificación, seguimiento y control en la ejecución del proyecto. - Programador: encargado de desarrollar la aplicación, e integrar las herramientas. - Diseñador: encargado del diseño del sistema.	

Elaboración Propia

3.1.9. PLAN DE GESTIÓN DE LAS ADQUISICIONES

Incluye los procesos necesarios para la compra o adquisición de los productos, servicios o resultados requeridos por fuera del equipo del proyecto.

El presente proyecto tiene cuatro tipos de adquisición:

- Bienes
- Servicios
- Software
- Recursos humanos

Tabla 10: Plan de Gestión de las Adquisiciones

NOMBRE DEL PROYECTO	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.
PLAN DE GESTIÓN DE ADQUISICIONES	
Herramientas	Características
Computadora	Intel Core I5, disco duro de 696 Gb memoria RAM de 6Gb.
Smartphone	Android versión 4.4 o iPhone Con la finalidad de usar la geolocalización.
Raspberry Pi y accesorios	Raspberry Pi 3B Es el hardware del sistema a emplear.
Gps ublox neo6m	Modulo Gps. Para obtener coordenadas exactas de latitud y longitud.
Sistema operativo Windows con licencia	Windows 7 a más, preferentemente de arquitectura de 64Bits. Es importante que cuente con software original para su correcto funcionamiento y así evitar deficiencias.
StarUML	Herramienta de código libre usado para el modelado de software (diagramas) basado en los estándares UML.
Firebase	Base de datos que sincroniza con los usuarios sin tener la necesidad de actualizar constantemente.
Framework Ionic	Versión 3.0.0, usada para el desarrollo de aplicaciones híbridas y multiplataforma.

Elaboración Propia

3.1.10. PLAN DE GESTIÓN DE LOS INTERESADOS

Incluye los procesos requeridos para identificar a las personas, grupos u organizaciones que pueden afectar o ser afectados por el proyecto, para analizar las expectativas de los interesados y su impacto en el proyecto, y para desarrollar estrategias de gestión adecuadas a fin de lograr la participación eficaz de los interesados en las decisiones y en la ejecución del proyecto.

Tabla 11: Plan de Gestión de los Interesados

PROYECTO	Desarrollo de un sistema de geolocalización de alerta de recojo de residuos sólidos en el distrito de San Jerónimo, 2018.
PREPARADO POR	Elizabeth Alejandrina Quintana Olarte – Jefe del Proyecto
REVISADOR POR	Elizabeth Alejandrina Quintana Olarte – Jefe del Proyecto
APROBADO POR	Elizabeth Alejandrina Quintana Olarte – Jefe del Proyecto

Elaboración propia

Identificación de los interesados: Los interesados del proyecto son:

- Los ciudadanos del distrito de San Jerónimo
- Municipalidad

Planificación de los interesados: Se analizó las expectativas de los interesados, que será tomando en cuenta a la hora del desarrollo del sistema.

3.2. METODOLOGÍA DE DESARROLLO

Para la elaboración de este proyecto se utilizará la metodología en V (verificación y validación). El método en v es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto, para este proyecto constituye la metodología más adecuada ya que como característica propone la combinación de software y hardware.

3.3.1. NIVEL 1 DESARROLLO: FASE DE ESPECIFICACIONES

Después de analizar la realidad problemática y ver las necesidades de los usuarios se destacaron los siguientes requerimientos:

A. REQUERIMIENTOS FUNCIONALES

- El sistema será capaz de mostrar a los usuarios la ubicación del vehículo recolector de residuos en tiempo real.
- El sistema será capaz de enviar notificaciones al usuario cuando el vehículo recolector de residuos sólidos se encuentre cerca de su vivienda o barrio (cerca de su casa).

- El sistema permitirá al usuario reportar incidencias de residuos sólidos.
- El sistema tiene incorporado el gps ublox neo6m el cual captura las coordenadas mediante el cual se determina la ubicación del vehículo.
- El móvil recibe la información de latitud, longitud que son transmitidos por la Raspberry.

B. Requerimientos no funcionales

Eficiencia:

- El sistema responderá con una latencia no mayor a los 20 segundos
- El sistema funcionará adecuadamente en todo momento sin dificultad
- El sistema será multiplataforma
- El sistema será híbrida

Seguridad:

- Los datos de la aplicación solo podrán ser modificados por personas autorizadas (administrador).
- La licencia de uso del software donde se aloje y con la que se realice la aplicación debe ser lo menos restrictiva posible, preferentemente software libre.

Usabilidad:

- La App será de interfaz amigable y fácil de usar para el usuario
- Contará con un manual de usuario.

C. RASPBERRY PI Y ARDUINO

Arduino: Es una plataforma electrónica de hardware libre basada en una placa con un microcontrolador. El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de

entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado, el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing / Wiring y el cargador de arranque que es ejecutado en la placa.

Raspberry: Es un ordenador de placa reducida El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos “Turbo” para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) Video Core IV, y 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB). El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa.

Diferencias entre microcontrolador (Arduino) y un microprocesador (raspberry Pi): un microcontrolador es un circuito integrado diseñado con el propósito de tareas específicas. Es principalmente usado en productos que requieren un grado de control impuesto por el usuario. Los microprocesadores en cambio, son usados para ejecutar aplicaciones grandes y genéricas, también las capacidades y salidas, como el rendimiento de la CPU.

Tanto Arduino como Raspberry Pi tienen funcionalidades distintas a la hora de elegir a uno u otro para hacer el proyecto, debemos usar cada uno en la tarea que mejor sabe hacer por ejemplo en el desarrollo de este proyecto de investigación se usa el interfaz gráfico de usuario y entorno CLI, tratamiento de datos para lo que es mejor desarrollar con el microprocesador Raspberry pi.

Tabla 12: Características Arduino y Raspberry Pi

Características	Arduino leonardo	Raspberry Pi 3
Core	ATmega 32U4	4x ARM Cortex-A53
Reloj principal	16Mhz	1.2Ghz
Tamaño de Registros	8 bits	32 bits
RAM	2.5Kb	1Gb
Flash	32kB	16Gb, 32Gb (SD)
GPIOs	20 IO +9(Power+Aux)	40 (IO+Power)
Tamaño(LaxAnxAl mm)	68,6 X 53,3 x 10	85 X 56 X 18
Real Time	Si	No
Consumo	0.125 W	1.75W(Wife +BT on)
Precio	S/. 99.00	S/.250
Sistema Operativo	Posible	Obligatorio

Elaboración propia

3.3.2. NIVEL 2 DESARROLLO: DISEÑO DE ALTO NIVEL

El proyecto se desarrolló bajo la plataforma de desarrollo del Framework Ionic, Raspberry Pi con el sistema operativo Raspbian y el módulo gps ublox neo6m, bajo estas tecnologías se dio solución al problema planteado.

Para un mejor entendimiento se busca identificar a los interesados del proyecto, los cuales son:

A. ESTABLECIMIENTO DE ACTORES

Se identifica a los actores y al equipo encargado de desarrollar el proyecto, quienes participaron de forma activa durante el desarrollo de la aplicación móvil:

- **Director del proyecto:** Es el encargado de brindar viabilidad y disponibilidad de los recursos que se necesita en el proyecto
- **Analista de sistemas:** encargado de diseñar la infraestructura de la aplicación móvil, considerando los requerimientos funcionales, alcance y las limitaciones que la aplicación tiene.

- **Programador:** Encargado de implementar la aplicación considerando la infraestructura diseñada por el analista del sistema.

B. ESTABLECIENDO USUARIOS

Identificamos a los usuarios, quienes serán las personas que harán uso de la aplicación, son:

- **Usuarios directos:** ciudadanos que vivan en el distrito de San Jerónimo
- **Usuarios Indirectos:** ciudadanos de los distintos distritos de la provincia de Andahuaylas.

C. DIAGRAMA DE CASO DE USO GENERAL

El diagrama de casos de usos es la descripción de las actividades que se realizarán para definir un conjunto de funcionalidades que el sistema debe cumplir para satisfacer los requerimientos, los cuales son:

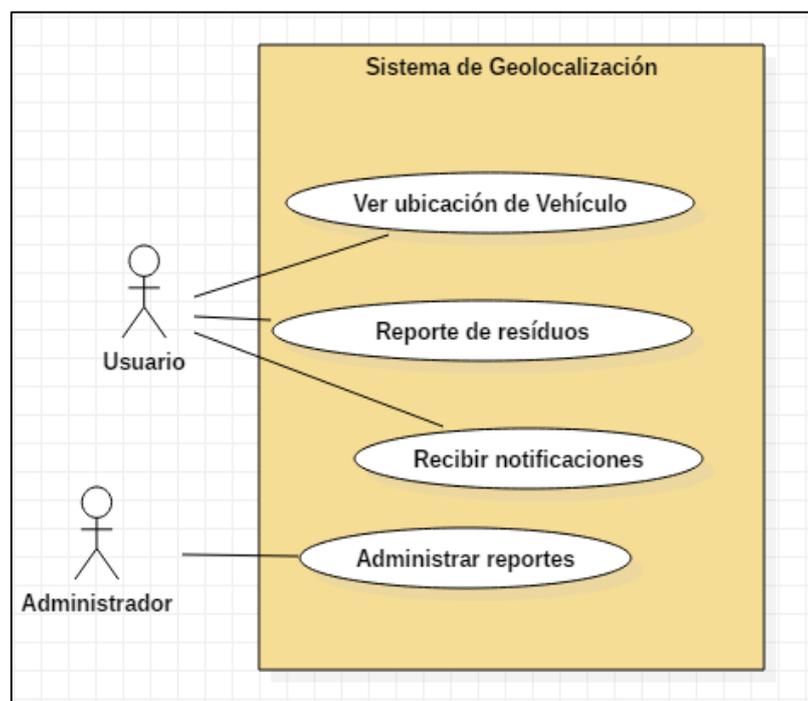


Figura 7: Diagrama caso de uso general
Elaboración propia

D. SUB CASO DE USO 1

En el siguiente caso de uso se realiza la descomposición del caso de uso general para especificar el comportamiento del sistema.

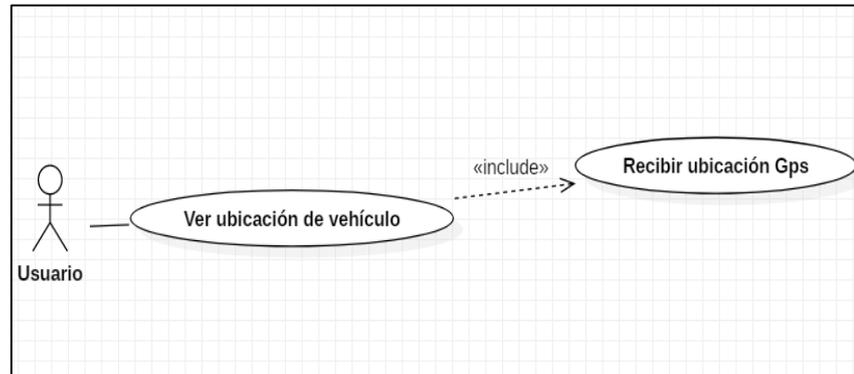


Figura 8: Diagrama de sub caso de uso 1
Elaboración propia

E. SUB CASO DE USO 2:

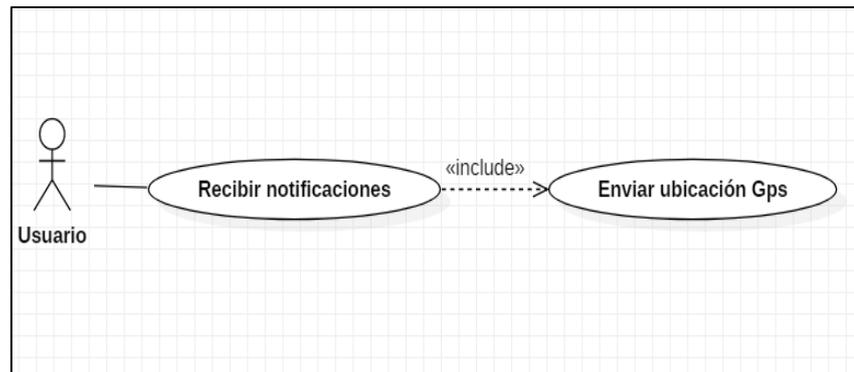


Figura 9: Diagrama de sub caso de uso 2
Fuente: Elaboración propia

F. DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue permite mostrar la arquitectura en tiempo de ejecución del sistema respecto al hardware y software, con el diagrama de despliegue se muestra básicamente la estructura del sistema modelando los aspectos físicos.

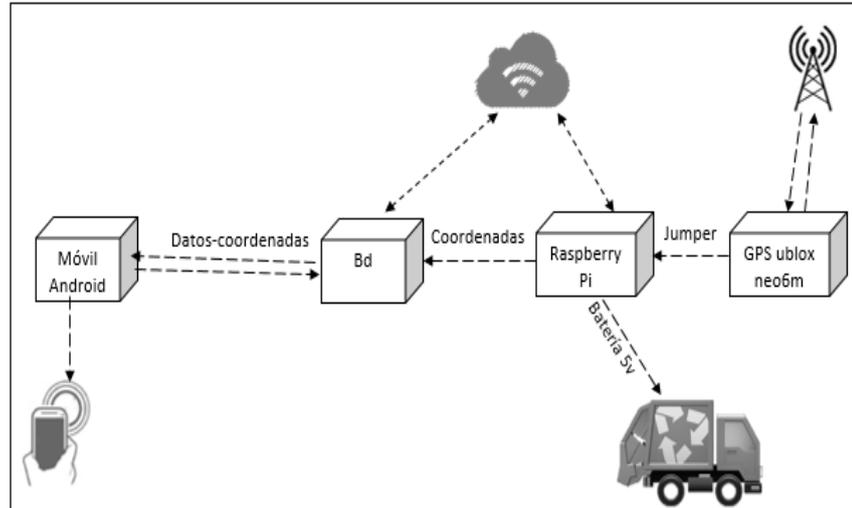


Figura 10: Diagrama de Despliegue
Fuente: Elaboración propia

Descripción:

- **Nodo Raspberry Pi:** El nodo Raspberry es la placa donde se llevara a cabo el cargado de las coordenadas la cual es alimentada con una batería de 5v.
- **Nodo gps ublox neo6m:** Este nodo es el encargado de obtener coordenadas de latitud y longitud.
- **Nodo Bd:** bases de datos Firebase es el nodo encargado de lograr el funcionamiento en tiempo real.
- **Nodo móvil:** Este nodo es necesario para el aprovechamiento de sus funcionalidades.

3.3.3. NIVEL 3 DESARROLLO: DISEÑO EN DETALLE

A. **DIAGRAMA DE ACTIVIDADES:** Un diagrama de actividades muestra un flujo de acciones (nodos ejecutan un proceso), generalmente secuenciales además presenta los resultados de dichas acciones.

- Diagrama de actividad Mostrar vehículo

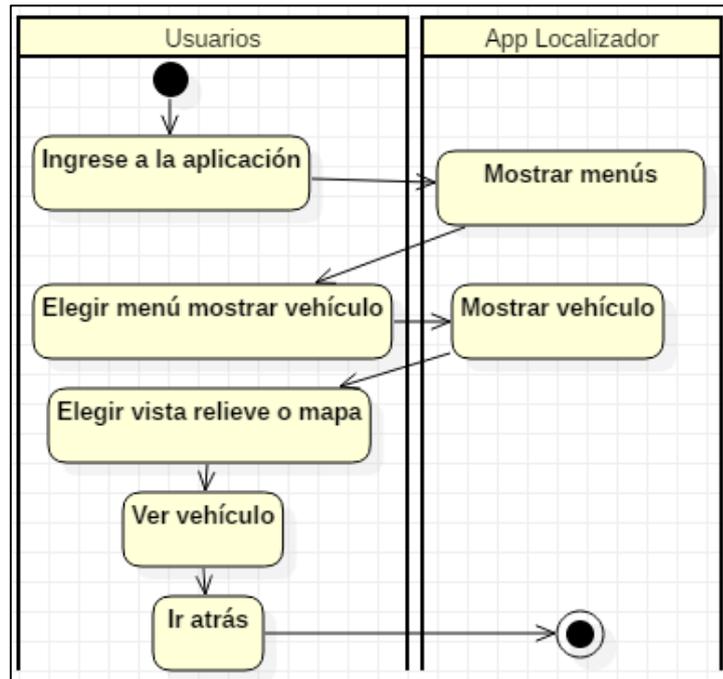


Figura 11: Diagrama de actividad mostrar vehículo
Elaboración propia

- Diagrama de actividad mi ubicación

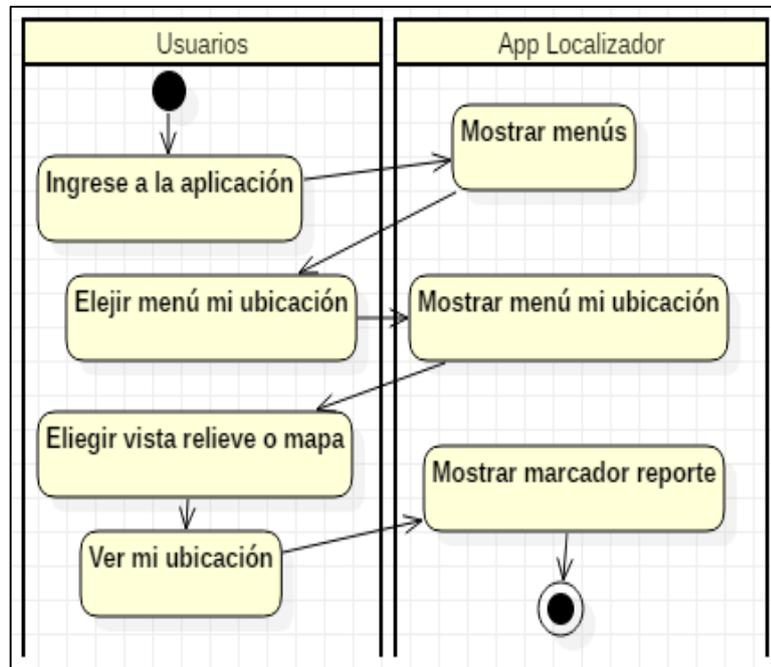


Figura 12: Diagrama de actividad mi ubicación
Fuente elaboración propia

- Diagrama de actividad reporte

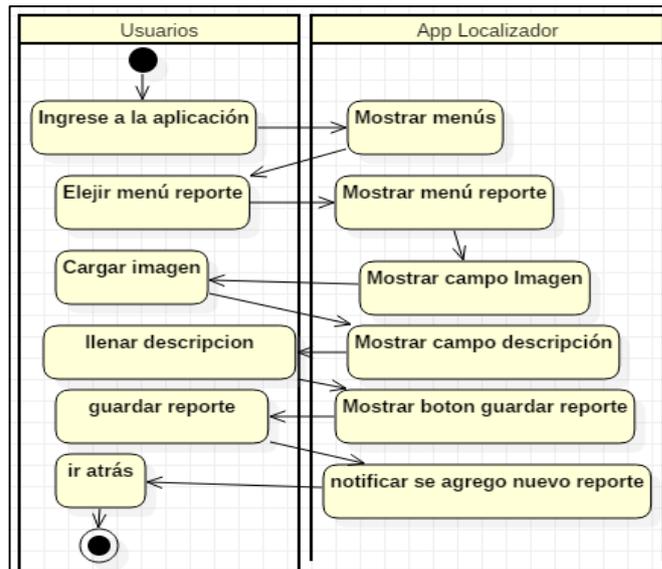


Figura 13: Diagrama de actividad reporte
Elaboración propia

En el diseño del proyecto se hizo uso de los diagramas de casos de uso, diagrama de despliegue y diagrama de actividades, si bien no fueron muchos diagramas, si fueron muy útiles y se convirtieron en la base fundamental del desarrollo. Se utilizó la herramienta starUML.

En la tabla N° 13 se detallan los casos de uso identificados:

Tabla 13: Casos de uso

SISTEMA DE GEOLOCALIZACIÓN	
DESCRIPCIÓN DE CASO DE USO 1	
NOMBRE DE CASO 1	Ver ubicación de vehículo
ACTORES:	Usuario
FUNCIÓN:	Visualizar el trayecto del vehículo
DESCRIPCIÓN:	El usuario al ingresar a la aplicación tiene la opción de visualizar en Google Maps
PRECONDICIONES:	Debe haber ingresado con su usuario y contraseña
DESCRIPCIÓN DE CASO DE USO 2	
NOMBRE DE CASO 2	Reporte de residuos
ACTORES:	Usuario
FUNCIÓN:	Enviar reporte de residuos Solidos
DESCRIPCIÓN:	El sistema pone en funcionamiento el módulo reporte en el cual el usuario mediante los campos

	descripción, imagen, envía reporte de acumulación.
PRECONDICIONES:	Identificarse mediante usuario y contraseña
DESCRIPCIÓN DE CASO DE USO 3	
NOMBRE DE CASO 3	Recibir notificaciones
ACTORES:	usuario
FUNCIÓN:	Recibe notificación
DESCRIPCIÓN:	El usuario recibe la notificación haciéndole saber que el vehículo se encuentra cerca a su domicilio
PRECONDICIONES:	No cerrar sesión de la App
DESCRIPCIÓN DE CASO DE USO 4	
NOMBRE DE CASO 4	Administrar reportes
ACTORES:	Administrador
FUNCIÓN:	gestionar los reporte realizados por los usuarios
DESCRIPCIÓN:	El administrador puede visualizar el trayecto del vehículo como también administrar los reportes, dispone de la opción eliminar los reportes
PRECONDICIONES:	Loguearse validando usuario

Elaboración propia

3.3.4. NIVEL 4 DESARROLLO: IMPLEMENTACIÓN

A. IMPLEMENTACIÓN DEL HARDWARE

Como base del hardware se utilizó el microprocesador Raspberry Pi 3 modelo B la cual presenta varias ventajas a diferencia de otras versiones y con otros dispositivos de fines similares.

Materiales necesarios: Como mínimo para funcionar la Raspberry Pi se necesita

- Micro SD clase 10 de 16GB
- Adaptador de alimentación con micro USB
- Teclado
- Cable HDMI
- Mouse
- Televisor o monitor

CONFIGURACIÓN DE LA RASPBERRY PI

Descargamos la **NOOBS** versión 2.8.2 de la página oficial de raspberry pi <https://www.raspberrypi.org/downloads/>, a diferencia de descargar directamente el sistema operativo Raspbian la NOOBS nos proporciona una selección de SO descomprimidos entre ellos también esta Raspbian.

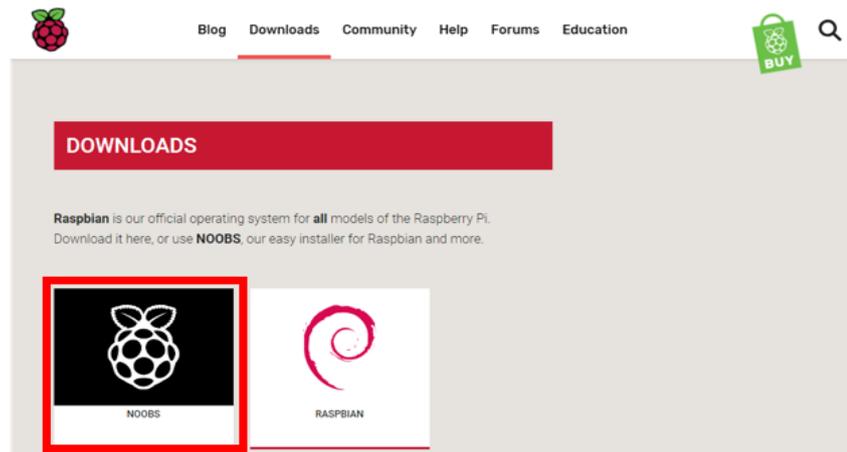


Figura 14: NOOBS
Elaboración propia



Figura 15: Descarga NOOBS
Fuente: Elaboración propia

Existen dos versiones:

- NOOBS es un instalador de sistema operativo fácil que contiene Raspbian y LibreELEC. También proporciona una selección de sistemas operativos alternativos que luego se descargan de Internet y se instalan.
- NOOBS Lite contiene el mismo instalador de sistema operativo sin Raspbian precargado. Proporciona el mismo

menú de selección de sistema operativo que permite descargar e instalar Raspbian y otras imágenes.

Instalar NOOBS: cargar Noobs en el micro SD la cual pondremos en la ranura de la Raspberry Pi como se muestra en la imagen.

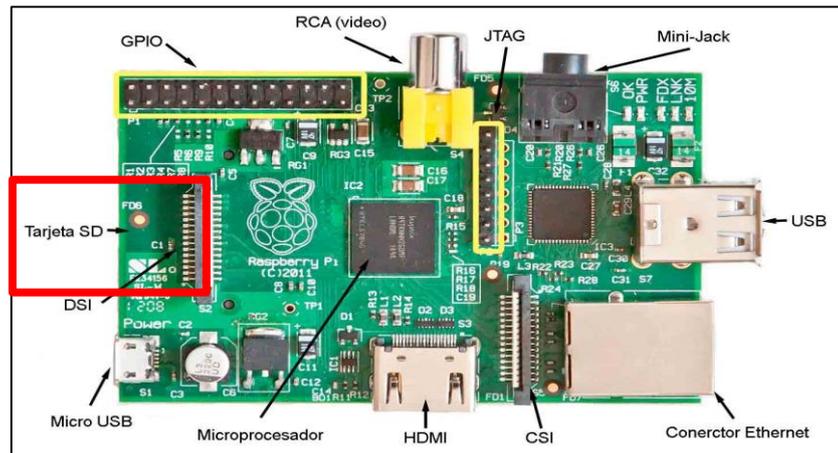


Figura 16: Partes de la Raspberry
Fuente: www.Raspberry.com

Una vez cargado Noobs instalamos el sistema operativo Raspbian

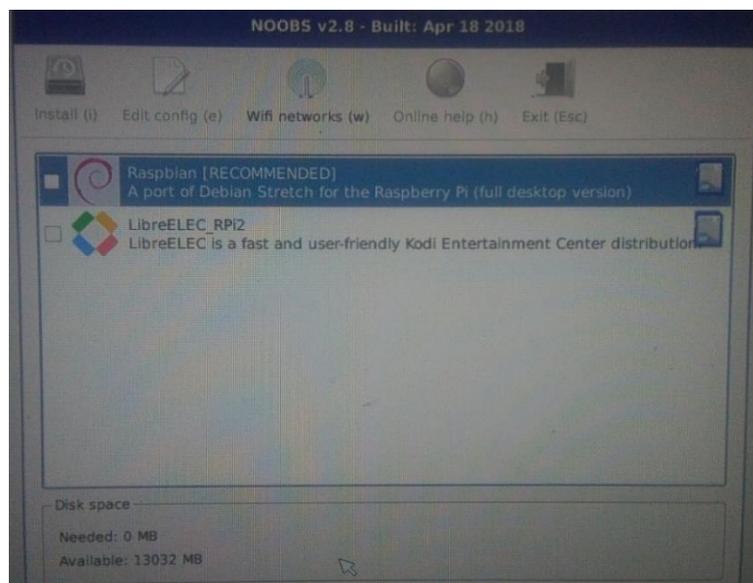


Figura 17: Instalación de Raspbian
Elaboración propia

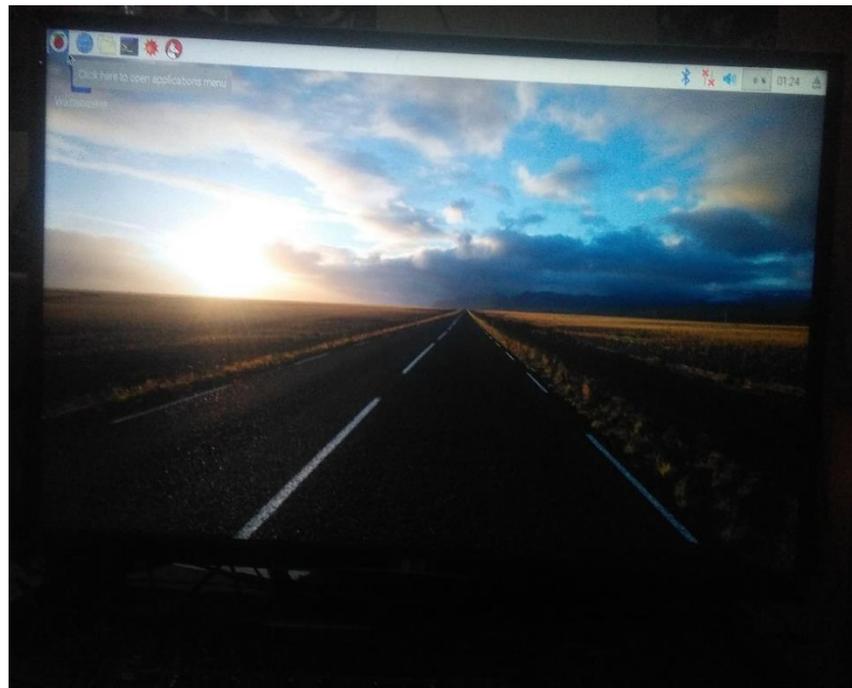


Figura 18: Vista Principal
Elaboración propia

Configuración mediante la terminal usuario: pi, contraseña: raspberry, en el módulo “raspi-config”

```

Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem          Ensures that all of the SD card s
2 Change User Password       Change password for the default u
3 Enable Boot to Desktop/Scratch Choose whether to boot into a des
4 Internationalisation Options Set up language and regional sett
5 Enable Camera              Enable this Pi to work with the R
6 Add to Rastack             Add this Pi to the online Raspber
7 Overclock                  Configure overclocking for your P
8 Advanced Options           Configure advanced settings
9 About raspi-config         Information about this configurat

<Select>                    <Finish>

```

Figura 19: Configuración Raspberry Pi
Elaboración propia

Integración con el módulo GPS:

Para este propósito se usó el modulo Gps ublox neo6m, con la finalidad de la captura de datos de los pgs disponibles.

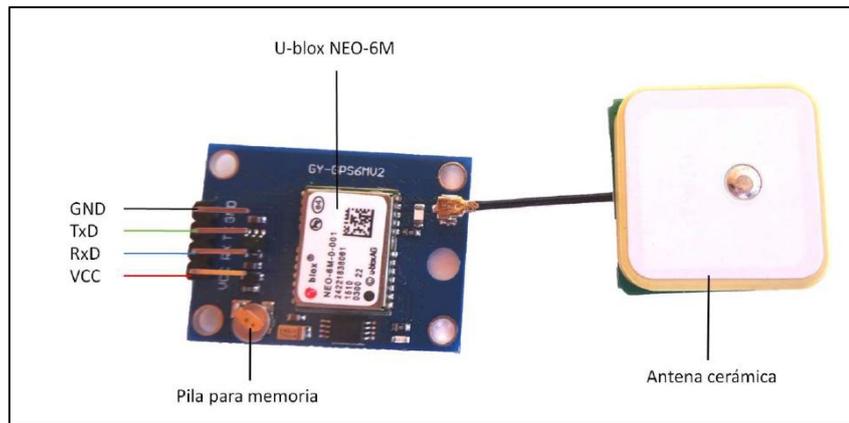


Figura 20: GPS U-Blox NEO
 Fuente: <https://bit.ly/2JfSkpJ>

La cual se integró al microprocesador Raspberry Pi 3 modelo B usada para la transmisión a la base de datos mediante la conexión a internet.

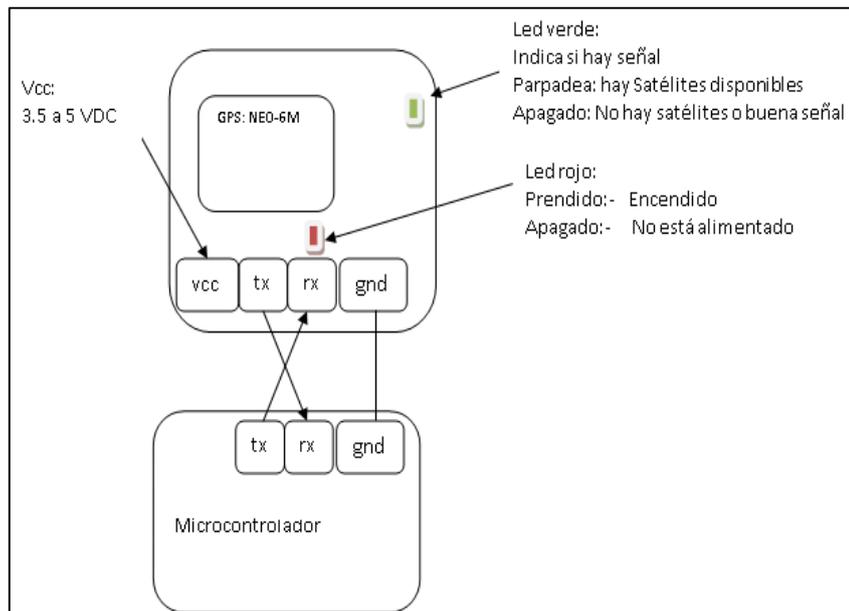


Figura 21: Diagrama de conexión
 Fuente: <https://bit.ly/2z5B4P4>

Captura de datos de la red de satélites después de ensamblar pasamos a la fase de procesamiento de datos, los datos se reciben de una red de satélites utilizando el método de triangulación que consisten en averiguar la distancia de cada una

de las tres señales de los satélites respecto a un punto de medición, en la actualidad existen dos tipos de red la de EE.UU y la red de Rusia. Para la manipulación de datos entregados por el GPS el código NMEA (National Marine Electronics Association) que pertenece a los EE.UU y es uno de los formatos de interfaz más recomendado ya que este entrega todos los datos que maneja el GPS en forma ordenada, las principales sentencias del estándar NMEA es:

\$ GPBOD: recorrido de origen a destino

\$ GPBWC: rumbo y distancia al punto de ruta, gran círculo

\$ GPGGA: datos de localización

\$ GPGLL: datos de latitud y longitud

\$ GPGSA: GPS DOP y satélites activos

\$ GPGSV: satélites GPS en vista

\$ GPHDT: encabezado o titulo

\$ GPR00: lista de puntos en la ruta actualmente activa

\$ GPRMA: datos específicos mínimos recomendados de Loranc

\$ GPRMB: datos mínimos de navegación

\$ GPRMC: datos específicos GPS

\$ GPRTE: datos de una ruta (datos de entrada-salida)

\$ GPTRF: datos de corrección de tránsito

\$ GPSTN: ID de datos múltiples

\$ GPVBW: velocidad terrestre

\$ GPVTG: datos del "track" y velocidad de movimiento

\$ GPWPL: ubicación del punto de referencia

\$ GPXTE: error de seguimiento, medido

\$ GPZDA: fecha y hora

\$ sudo apt-get install gpsd gpsd-clients python-gps

Time: 2013-09-19T22:29:16.000Z	PRN: 10	Elev: 81	Azim: 277	SNR: 30	Used: Y
Latitude: N	13	64	066	00	Y
Longitude: W	2	51	271	00	Y
Altitude: 79.6 m	7	45	152	29	Y
Speed: 1.0 kph	4	45	204	32	Y
Heading: 328.6 deg (true)	23	33	067	35	Y
Climb: 0.0 m/min	5	25	292	24	N
Status: 3D FIX (18 secs)	8	20	174	00	N
Longitude Err: +/- 15 m	9	19	183	00	N
Latitude Err: +/- 17 m	16	13	053	30	N
Altitude Err: +/- 20 m	29	11	332	00	N
Course Err: n/a	20	04	116	00	N
Speed Err: +/- 128 kph	34	00	000	00	N
Time offset: 0.598					
Grid Square: IO82vm					

Figura 23: Transmisión GPS
Elaboración propia

Transmisión de datos para testear el funcionamiento

Ahora que tenemos las coordenadas transmitiremos esos puntos al estado inicial para visualizar en tiempo real la ubicación del dispositivo haciendo uso un script desarrollado en python, en la url detalla <https://bit.ly/2SiY5Hg>.

```
import serial
import pynmea2
import time
from ISSStreamer.Streamer import Streamer

serialStream = serial.Serial("/dev/ttyS0", 9600, timeout=0.5)

# construct a streamer instance with information to append to or create
# a bucket and an ini file location that contains the Initial State
# Account Access Key.
streamer = Streamer(bucket_name="GPS_UNAJMA", bucket_key="BGV0vphrvpr2vCRZdz2oK0CxIEVfhQU9", ini_file_location="/isst

try:
    while True:
        sentence = serialStream.readline()
        if sentence.find('GGA') > 0:
            data = pynmea2.parse(sentence)
            print (data)
            streamer.log("Location", "{lat},{lon}".format(lat=data.latitude,lon=data.longitude))
            streamer.log("Altitude ({unit})".format(unit=data.altitude_units), data.altitude)
            print (lat)
            print (lon)
except KeyboardInterrupt:
    streamer.close()
```

Figura 24: Script envio.py
Elaboración propia

B. IMPLEMENTACIÓN DEL SOFTWARE

Dependencias de paquetes de la aplicación móvil los paquetes son de uso general en el proyecto y se hará uso de acuerdo a la necesidad, como se muestra en la imagen.

```
"dependencies": {
  "@angular/animations": "5.2.11",
  "@angular/common": "5.2.11",
  "@angular/compiler": "5.2.11",
  "@angular/compiler-cli": "5.2.11",
  "@angular/core": "5.2.11",
  "@angular/forms": "5.2.11",
  "@angular/http": "5.2.11",
  "@angular/platform-browser": "5.2.11",
  "@angular/platform-browser-dynamic": "5.2.11",
  "@ionic-native/camera": "^4.13.0",
  "@ionic-native/core": "~4.10.0",
  "@ionic-native/geolocation": "^4.12.2",
  "@ionic-native/google-maps": "^4.13.0",
  "@ionic-native/launch-navigator": "^4.14.0",
  "@ionic-native/splash-screen": "~4.10.0",
  "@ionic-native/status-bar": "~4.10.0",
  "@ionic/lab": "^1.0.0-rc.13",
  "@ionic/storage": "2.1.3",
  "ajv": "^6.5.3",
  "angular-common": "^0.1.0",
  "angularfire2": "^5.0.1",
  "cordova-android": "7.0.0",
  "cordova-plugin-actionsheet": "^2.3.3",
  "cordova-plugin-camera": "^4.0.3",
  "cordova-plugin-device": "^2.0.2",
  "cordova-plugin-dialogs": "^2.0.1",
  "cordova-plugin-geolocation": "^4.0.1",
  "cordova-plugin-googlemaps": "^2.4.3",
  "cordova-plugin-ionic-keyboard": "^2.1.2",
  "cordova-plugin-ionic-webview": "^2.1.3",
  "cordova-plugin-splashscreen": "^5.0.2",
  "cordova-plugin-whitelist": "^1.3.3",
  "firebase": "^5.5.0",
  "ionic-angular": "3.9.2",
  "ionicons": "3.0.0",
  "promise-polyfill": "^8.1.0",
  "rxjs": "^6.3.2",
  "rxjs-compat": "^6.3.2",
  "sw-toolbox": "3.6.0",
  "uk.co.workingedge.phonegap.plugin.launchnavigator": "^4.2.2",
  "zone.js": "0.8.26"
},
```

Figura 27: Package.json
Elaboración propia

Cordova – Plugins necesarias para la aplicación móvil

Una de las claves de Apache cordova son los plugins que permiten acceder a las características nativas del sistema operativo desde el código JavaScript, los plugins se instalan para los dispositivos móviles de acuerdo a la plataforma en la cual se compilará y acorde a las necesidades.

```
"cordova": {
  "plugins": {
    "cordova-plugin-whitelist": {},
    "cordova-plugin-device": {},
    "cordova-plugin-splashscreen": {},
    "cordova-plugin-ionic-webview": {},
    "cordova-plugin-ionic-keyboard": {},
    "cordova-plugin-googlemaps": {
      "API_KEY_FOR_ANDROID": "AIzaSyDZXwL3MLwSgrZsphSR1YhV3bD4aULQKWQ",
      "API_KEY_FOR_IOS": "AIzaSyDZXwL3MLwSgrZsphSR1YhV3bD4aULQKWQ",
      "PLAY_SERVICES_VERSION": "15.0.1",
      "ANDROID_SUPPORT_V4_VERSION": "27.+"
    },
    "cordova-plugin-geolocation": {
      "GEOLOCATION_USAGE_DESCRIPTION": "La app necesita geolocalización"
    },
    "cordova-plugin-camera": {},
    "uk.co.workingedge.phonegap.plugin.launchnavigator": {
      "OKHTTP_VERSION": "3.+"
    }
  },
  "platforms": [
    "android"
  ]
}
```

Figura 28: Cordova - Plugins
Elaboración propia

Configuración Inicial del Proyecto

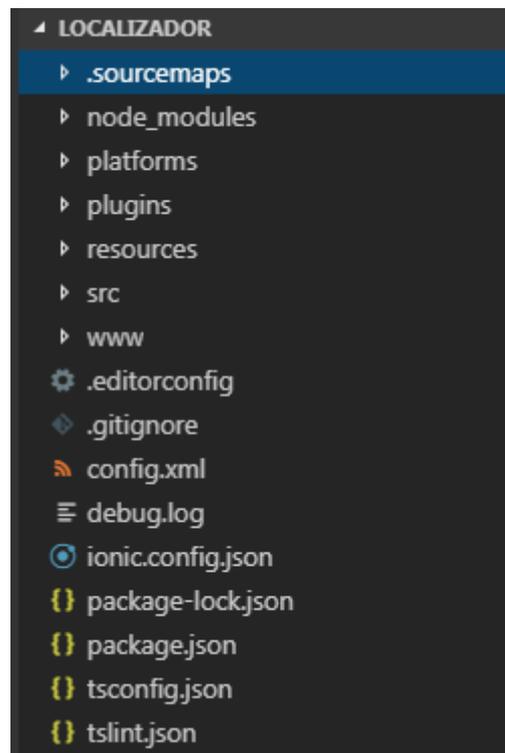
Ionic facilita la creación de proyectos usando su intérprete de línea de comandos (CLI), mediante el cual debe crear el proyecto haciendo uso del comando:

```
$ Ionic start myApp blank
```

Al ejecutar el comando se crea el proyecto llamado myApp con toda la estructura definida de carpetas y directorios que necesita Ionic.

Estructura de directorios y archivos del proyecto

Al crear un nuevo proyecto en Ionic se crea una estructura definida que permite organizar el código y las carpetas son:



*Figura 29: Directorio del Proyecto "Localizador"
Elaboración propia*

- **sourcemaps:** Ionic genera el archivo al instalar los paquetes de mapas.
- **Node_modules:** contiene las dependencias del proyecto por ejemplo @ionic, @angular, @firebase, @ionic-native otros.

- **platforms:** contiene el código específico de las plataformas móviles como por ejemplo Android, iOS y otros.
- **plugins:** contiene los plugins de cordova y los módulos instalados para la aplicación los cuales se utilizan para añadir funcionalidad por ejemplo cordova-plugins-camera, cordova-plugins-geolocation, cordova-plugins-googlemaps.
- **resources:** recursos específicos de las plataformas, en esta carpeta contiene los iconos por defecto de la aplicación.
- **src:** esta carpeta es la más importante ya que contiene todos los archivos de configuración, estilo y de programación.
- **www:** en la carpeta assets contiene imágenes para el icono de la aplicación y en la carpeta build contiene los archivos javascript.
- **package.json:** contiene los paquetes, dependencias y plugins instalados de nodeJS.
- **.editorconfig, .gitignore, config.xml, ionic.config.json, package-lock.json, tsconfig.json, tslint.json:** los archivos son creados y usados por Ionic 3.

Tipos de archivos manejados en el proyecto

- **Archivos .html:** contiene la plantilla html de la pagina
- **Archivos Sass .scss:** es una extensión de CSS escrita en RUBY, es el mejor ayudante para el desarrollador web a la hora de trabajar con CSS gracias a las múltiples funcionalidades como funciones, herencias, mixins.
- **Archivos TypeScript .ts:** Es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft está escrito sobre Javascript que busca otorgar características adicionales como el tipado estático.

Integración con Firebase

Para interactuar con la base de datos de Firebase se procede de la siguiente manera:

- Tener instalado Node.js para tener disponible la terminal de comandos npm, mediante el cual instalamos angularfire2 y firebase, se ejecuta el comando:

```
npm install firebase angularfire2 --save
```

Figura 30: Instalar AngularFire2 Firebase

Fuente: <https://bit.ly/2OORhmo>

Una vez instalado el plugin acceder a la consola de firebase y crear un nuevo proyecto

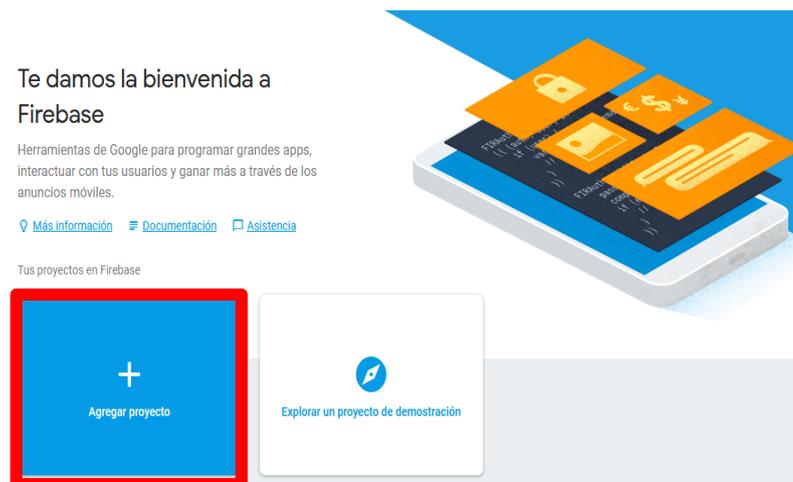


Figura 31: Nuevo Proyecto Firebase

Elaboración propia

Copiar el script que proporciona firebase a nuestro archivo de configuración.

```
<script src="https://www.gstatic.com/firebasejs/${JSCORE_VERSION}/firebase.  
  
<script>  
  var app = firebase.initializeApp({  
    apiKey: '<your-api-key>',  
    authDomain: '<your-auth-domain>',  
    databaseURL: '<your-database-url>',  
    projectId: '<your-cloud-firestore-project>',  
    storageBucket: '<your-storage-bucket>',  
    messagingSenderId: '<your-sender-id>'  
  });  
  // ...  
</script>
```

Figura 32: Script Firebase
Elaboración propia

```
<script src="https://www.gstatic.com/firebasejs/5.5.4/firebase.js"></script>  
<script>  
  // Initialize Firebase  
  var config = {  
    apiKey: "AIzaSyDZXwL3MLwSgrZsphSR1YhV3bD4aULQKWQ",  
    authDomain: "localizador-1a26f.firebaseio.com",  
    databaseURL: "https://localizador-1a26f.firebaseio.com",  
    projectId: "localizador-1a26f",  
    storageBucket: "localizador-1a26f.appspot.com",  
    messagingSenderId: "1006611742920"  
  };  
  firebase.initializeApp(config);  
</script>
```

Figura 33: Script de configuración de Firebase
Elaboración propia

```
const firebaseConfig = {  
  apiKey: "AIzaSyDZXwL3MLwSgrZsphSR1YhV3bD4aULQKWQ",  
  authDomain: "localizador-1a26f.firebaseio.com",  
  databaseURL: "https://localizador-1a26f.firebaseio.com",  
  projectId: "localizador-1a26f",  
  storageBucket: "localizador-1a26f.appspot.com",  
  messagingSenderId: "1006611742920"  
};
```

Figura 34: Archivo de configuración Firebase
"src/app/module.ts"
Elaboración propia

Estructura de la base de datos en firebase

Firebase es una base de datos no relacional (NoSQL). Todos los datos de firebase Realtime Database se almacenan como objetos JSON alojado en la nube, a diferencia de una base de datos relacional SQL no hay tablas ni registros.

Cuando se agrega datos al árbol JSON estos se convierten en una rama de la estructura JSON con una clave (ID) asociada.

<https://firebase.google.com/docs/database/web/structure-data?hl=es>

La base de datos que se utiliza en la aplicación es como se muestra en la imagen:

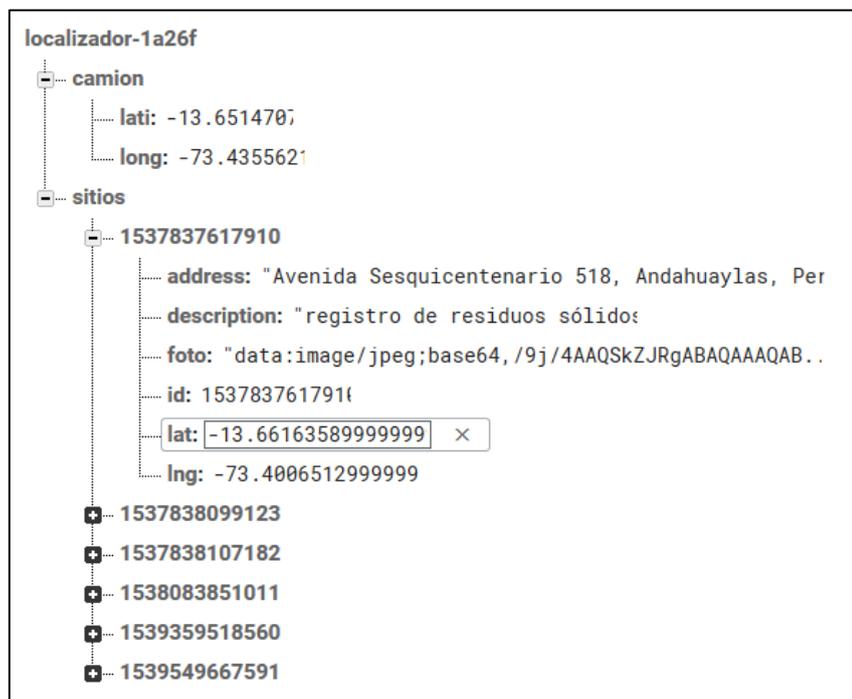


Figura 35: Localizador Database
Elaboración propia

Integración de los módulos de acceso a la aplicación

Login:

Se establece como página principal a LoginPage en app, component.ts significa que al cargar la aplicación en cualquier

plataforma la página de login al momento de iniciar la aplicación se abra automáticamente la página con el siguiente código:

```
@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  rootPage:any = LoginPage;
```

Figura 36: rootPage: Login
Elaboración propia

Autenticamos al usuario mediante email y password Importando la librería “AuthProvider” la cual también es importada en “app/module.ts” ya que es la carpeta que contiene todas las librerías a usar, como se muestra en anexos.

```
ionViewDidLoad() {
  console.log('ionViewDidLoad LoginPage');
}

signin(){
  this.auth.registerUser(this.user.email,this.user.password)
  .then((user) => {
    // El usuario se ha creado correctamente
  })
  .catch(err=>{
    let alert = this.alertCtrl.create({
      title: 'Error',
      subTitle: err.message,
      buttons: ['Aceptar']
    });
    alert.present();
  })
}

login()
{
  this.auth.loginUser(this.user.email,this.user.password ).then((user) => {
  }
)
  .catch(err=>{
    let alert = this.alertCtrl.create({
      title: 'Error',
      subTitle: err.message,
      buttons: ['Aceptar']
    });
    alert.present();
  })
}
```

Figura 37: login.ts
Elaboración propia

Integración de los módulos de la página principal

Una vez autenticado el usuario se inicia la página principal con un event para la visualización del vehículo en el mapa en tiempo

real, esto incluye el acceso a ver su ubicación, la acción reporte de incidencia y visualizar la lista de reportes.

Página Loggedin: el formulario de Logedin está diseñado con etiquetas de Ionic para la adaptación responsive y está vinculada a las otras páginas.

```
<ion-navbar>
  <ion-title>PAGINA PRINCIPAL</ion-title>
</ion-navbar>

</ion-header>

<ion-content padding>

  <P>IDENTIFICADO: {{email}}</P><br><br><br>

  <div id="topButtons">
    <button ion-button color="secondary" (click)="VercamionIn()" outline round>MOSTRAR VEHICULO</button><br><br><br>
    <button ion-button color="secondary" (click)="MiubiIn()" outline round>MI UBICACIÓN</button><br><br><br>
    <button ion-button color="secondary" (click)="ReporteIn()" outline round>REPORTAR</button><br><br><br>
    <button ion-button color="secondary" (click)="ListaIn()" outline round>LISTA DE REPORTES</button><br><br><br>
    <button ion-button color="danger" (click)="cerrarSesion()" outline round>CERRAR SESIÓN</button>
  </div>
</ion-content>
```

Figura 38: *loggedin.html*
Elaboración propia

Importar las librerías de Ionic y angular para poder usar los parámetros y vincular las páginas: ver camión, reporte, mi ubicación, lista de reportes.

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ModalController } from 'ionic-angular';
import { AngularFireAuth } from 'angularfire2/auth';

import{ VercamionPage } from '../vercamion/vercamion';
import{ ReportePage } from '../reporte/reportes';
import{ ListaPage } from '../lista/lista';
import{ MiubiPage } from '../miubi/miubi';
import { Geolocation } from '@ionic-native/geolocation';
import { Platform } from 'ionic-angular';
import { AuthProvider } from '../providers/auth/auth';
```

Figura 39: *loggedin.ts*
Elaboración propia

En la página de loggedin.ts se da acceso a los demás formularios del sistema.

```

VercamionIn() {
  this.navCtrl.push(VercamionPage);
}

ReporteIn() {
  // aquí vamos a abrir el modal para añadir nuestro sitio.
  let mimodal = this.modalCtrl.create( ReportePage,this.coords );
  mimodal.present();
}

ListaIn() {
  this.navCtrl.push(ListaPage);
}

MiubiIn() {
  this.navCtrl.push(MiubiPage);
}

```

Figura 40: acceso a loggedin
Elaboración propia

Mostrar vehículo:

Mediante el módulo ver camión se visualiza el recorrido del vehículo recolector de residuos por las diferentes arterias de la ciudad, representándose en un mapa interactivo de google Maps:

```

import { Component, ViewChild, ElementRef } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { AngularFireDatabase } from 'angularfire2/database';
import { FirebaseDbProvider } from '../providers/firebase-db/firebase-db';

```

Figura 41: librería vercamion.ts
Elaboración propia

Aprovechando el uso del api de google cargamos el mapa con sus especificaciones: marcador centrado, nos ubique en donde se encuentra en el punto donde indicamos, que se basa en el mapa y en la posición.

```

@IonicPage()
@Component({
  selector: 'page-vercamion',
  templateUrl: 'vercamion.html',
})
export class VercamionPage {

  //camion: any;
  camion : any = { lati: 0, long: 0 }
  @ViewChild('map') mapRef: ElementRef;

  constructor(public navCtrl: NavController,
    public navParams: NavParams,
    public afDB: AngularFireDatabase,
    public dbFirebase :FirebaseDbProvider
  ) {

  }
}

```

Figura 42: vercamion.ts
Elaboración propia

Mi ubicación:

En el módulo mi ubicación se recoge los datos que está enviando los datos del gps del celular (coordenadas) los cuales se almacenan en la nube, desde la ubicación actual podemos acceder al modal realizar reportes.

```

import { Component } from '@angular/core';
import { NavController, NavParams, ModalController } from 'ionic-angular';

import { Geolocation } from '@ionic-native/geolocation';
import { Platform } from 'ionic-angular';
import { ReportePage } from '../reporte/reportes';

```

Figura 43: Librerías ubicación
Elaboración propia

```

loadMap(){
  let mapContainer = document.getElementById('map');
  this.map = new google.maps.Map(mapContainer, {
    center: this.coords,
    zoom: 12
  });
  // Colocamos el marcador
  new google.maps.Marker({
    icon : 'assets/imgs/ico_estoy_aqui.png',
    map: this.map,
    position: this.coords
  });
}

obtenerPosicion():any{
  this.geolocation.getCurrentPosition().then(res => {
    this.coords.lat = res.coords.latitude;
    this.coords.lng = res.coords.longitude;

    this.loadMap();
  })
  .catch(
    (error)=>{
      console.log(error);
    }
  );
}
}

```

Figura 44: *ubicacion.ts*
Elaboración propia

Reportar:

Una vez que se toca en reporte, esta acción abre un modal que contiene las coordenadas obtenidas del celular del usuario y los campos descripción de reporte, cargar imagen.


```

ionViewDidLoad() {
  console.log('ionViewDidLoad ModalDetalleSitioPage');
}

cerrarModal(){
  this.viewCtrl.dismiss();
}

comoLlegar(){
  let destino = this.sitio.lat+', '+this.sitio.lng;
  this.launchNavigator.navigate(destino)
  .then(
    success => console.log('Launched navigator'),
    error => console.log('Error launching navigator', error)
  );
}
}

```

Figura 47: modal-detalle-sitio.ts
Elaboración propia.

Notificación:

La notificación es enviada al usuario obteniendo las coordenadas de la posición en la cual se encuentra su vivienda del ciudadano, así el sistema podrá alertar mediante notificaciones cuando el vehículo se encuentre cerca de su vivienda.

```

this.platform.ready().then(() => {
  this.statusBar.styleDefault();
  this.splashScreen.hide();
  this.registerToken();
  this.getNotifications();
});
}

private registerToken(){
  this.push.register().then((t: PushToken) => {
    return this.push.saveToken(t,{
      ignore_user: true
    });
  }).then((t: PushToken) => {
    console.log('Token saved:', t.token);
  });
}

private getNotifications(){
  this.push.rx.notification()
  .subscribe((msg) => {
    alert(msg.title + ': ' + msg.text);
  });
}
}

```

Figura 48: notificación.ts
Elaboración propia

Integración del módulo administrador a la página principal

El módulo administrador es una aplicación distinta al del usuario, las funciones que tiene son ver el recorrido del vehículo recolector de residuos.

```
ionViewDidLoad() {
  console.log('ionViewDidLoad ModalDetalleSitioPage');
}

cerrarModal(){
  this.viewCtrl.dismiss();
}

comoLlegar(){
  let destino = this.sitio.lat+', '+this.sitio.lng;
  this.launchNavigator.navigate(destino)
  .then(
    success => console.log('Launched navigator'),
    error => console.log('Error launching navigator', error)
  );
}
```

Figura 49: src/page/modal-detalle-sitio.ts
Elaboración propia

Ver la lista de reporte de residuos para poder gestionar con las opciones eliminar un reporte.

```
ionViewDidLoad() {
  console.log('ionViewDidLoad ReportePage');
  this.coords.lat = this.navParams.get('lat');
  this.coords.lng = this.navParams.get('lng');

  this.getAddress(this.coords).then(results=> {
    this.address = results[0]['formatted_address'];
  }, errStatus => {

  });
}

cerrarModal(){
  this.viewCtrl.dismiss();
}
```

Figura 50: src/pages/reporte.ts
Elaboración propia

CAPÍTULO 4: RESULTADOS Y DISCUSIÓN

4.1. TEST UNITARIO

4.1.1. PRUEBAS DE HARDWARE

Antes de comenzar con la implementación se realizó la revisión de cada dispositivo:

- **Raspberry Pi 3 modelo B:** se realizaron pruebas de los pines y se habilitó los puertos UART para conectar el gps como se muestra en los anexos, obteniendo el resultado exitoso.
- **Gps ublox neo6m:** para probar el funcionamiento de este dispositivo primero se conectó mediante el jumper a la raspberry Pi a los pines 8 y 10.
- Los demás dispositivos como cable USB fueron comprados con garantías y se comprobó su funcionamiento mientras se iba armando el módulo.

4.1.2. PRUEBAS DE SOFTWARE

Mientras se desarrollaba el código fuente de la aplicación y se iba realizando la compilación, en caso hubiese errores quedaba en manos del programador ir corrigiéndolos.

Se tuvo mayor cuidado en 2 casos (dependencias y librerías). El primero fue en el directorio myApp/src/app donde se encuentra el archivo "app.module.ts" que contiene todas las dependencias y páginas que se harán uso como se muestra en la imagen:

```

import { NgModule, ErrorHandler } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { IonicApp, IonicModule, IonicErrorHandler } from 'ionic-angular';
import { MyApp } from './app.component';

import { LoginPage } from '../pages/login/login';
import { HomePage } from '../pages/home/home';
import { TabsPage } from '../pages/tabs/tabs';

import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

import{ RegisterPage } from '../pages/register/register';
import{ LoggedinPage } from '../pages/loggedin/loggedin';
import { AngularFireModule } from 'angularfire2';
import { AngularFireDatabaseModule } from 'angularfire2/database';
import { AngularFireAuthModule } from 'angularfire2/auth';

import{ VerbusPage } from '../pages/verbus/verbus';
import{ ReportePage } from '../pages/reporte/reporte';
import{ ListaPage } from '../pages/lista/lista';
import{ MiubiPage } from '../pages/miubi/miubi';
import{ AuthProvider } from '../providers/auth/auth';
import { ModalDetalleSitioPage } from '../pages/modal-detalle-sitio/modal-detalle-sitio';

import { Geolocation } from '@ionic-native/geolocation';
import { Camera } from '@ionic-native/camera';
import { FirebaseDbProvider } from '../providers/firebase-db/firebase-db';
import { LaunchNavigator } from '@ionic-native/launch-navigator';

```

Figura 51: *app.module.ts*
Elaboración propia

El segundo caso fue con la instalación de los paquetes y las librerías

1. Para agregar la plataforma de android

```
ionic cordova platform add android
```

2. Appi de google

```
<script
src="https://maps.google.com/maps/api/js"></script>
```

3. Instalación de nuevas páginas en el proyecto

```
ionic g page modalNuevoSitio
```

4. instalando paquetes y dependencias de cámara

```
ionic cordova plugin add cordova-plugin-camera --save  
npm install --save @ionic-native/camera
```

5. iniciando un nuevo modal

```
ionic g page modalDetalleSitio
```

6. instalación de plugins para cordova

```
ionic cordova plugin add  
uk.co.workingedge.phonegap.plugin.launchnavigator  
npm install --save @ionic-native/launch-navigator
```

7. paquete firebase

```
npm install angularfire2 firebase --save
```

8. instalación de promise-polyfill en angular

```
npm install promise-polyfill --save-exact
```

9. instalación de provider para firebase

```
ionic g provider firebaseDb
```

4.2. TEST DE INTEGRACIÓN

Partimos de la conexión entre la raspberry Pi y el módulo gps

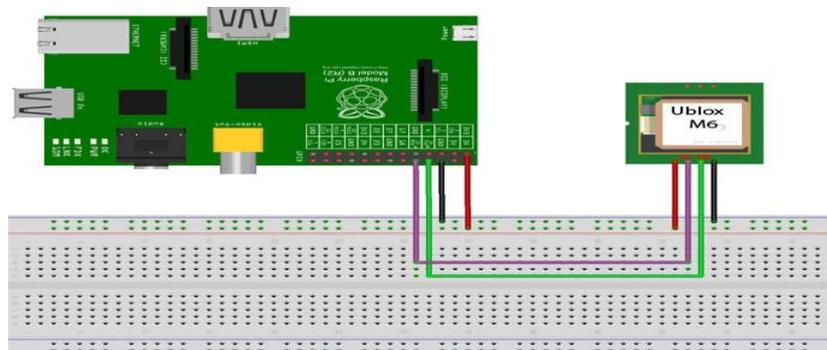


Figura 52: Diagrama de conexión en Fritzing
Elaboración propia

Los cables TX y RX deben estar cruzados, el pin RX del módulo GPS va conectado al pin TX de la raspberry Pi y el pin TX del módulo GPS va conectado al pin RX de la raspberry Pi.

Transmisión a la base de datos: al correr el script para la transmisión a la base de datos.

```
enviocamion.py
1 import serial
2 import pynmea2
3 from firebase import firebase
4
5 firebase=firebase.FirebaseApplication('https://localizador-1a26f.firebaseio.com/')
6
7
8 serialStream = serial.Serial("/dev/ttyS0", 9600, timeout=0.5)
9
10 a=0
11 b=0
12 c=0
13 d=0
14
15 while True:
16     sentence = serialStream.readline()
17     if sentence.find("GGA") > 0:
18         data = pynmea2.parse(sentence)
19         print "{time}: {lat},{lon}".format(time=data.timestamp, lat=data.latitude, lon=data.longitude)
20         a = "{time}".format(time=data.timestamp)
21         b = "{lat}".format(lat=data.latitude)
22         c = "{lon}".format(lon=data.longitude)
23         d = "{unit}".format(unit=data.altitude)
24
25         result = firebase.put('/camion','/',{ 'hora':a, 'long':c, 'lati':b, 'altitud':d})
26
```

Figura 53: scripts transmisión base de datos
Elaboración propia

Con la creación del script realizado en python el objetivo es transmitir los datos obtenidos a un servidor (firebase) y estos puedan ser procesados en tiempo real e interactuar con la aplicación móvil.

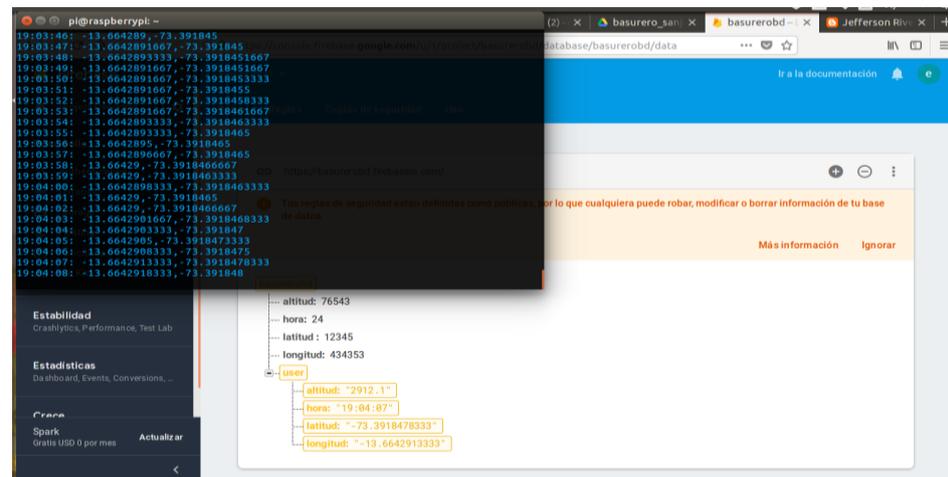


Figura 54: Transmisión a Firebase
Elaboración propia

En la imagen 55 se puede observar que los datos transmitidos se almacenan en los servidores de google los cuales obtendremos para la sincronización con la aplicación móvil.

Se realizó las pruebas del módulo mostrar vehículo

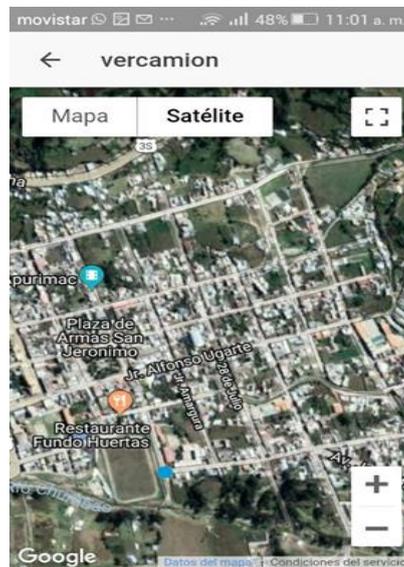


Figura 55: prueba del módulo mostrar vehículo

Elaboración propia

Seguidamente hacemos la verificación del módulo mi ubicación

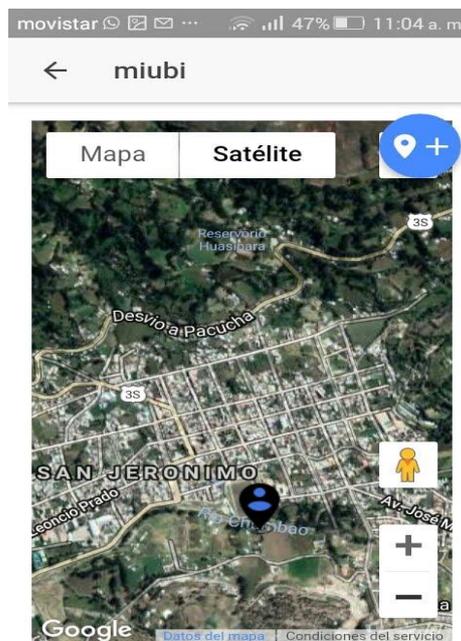


Figura 56: Verificación del módulo mi ubicación

Elaboración propia

Módulo Reporte

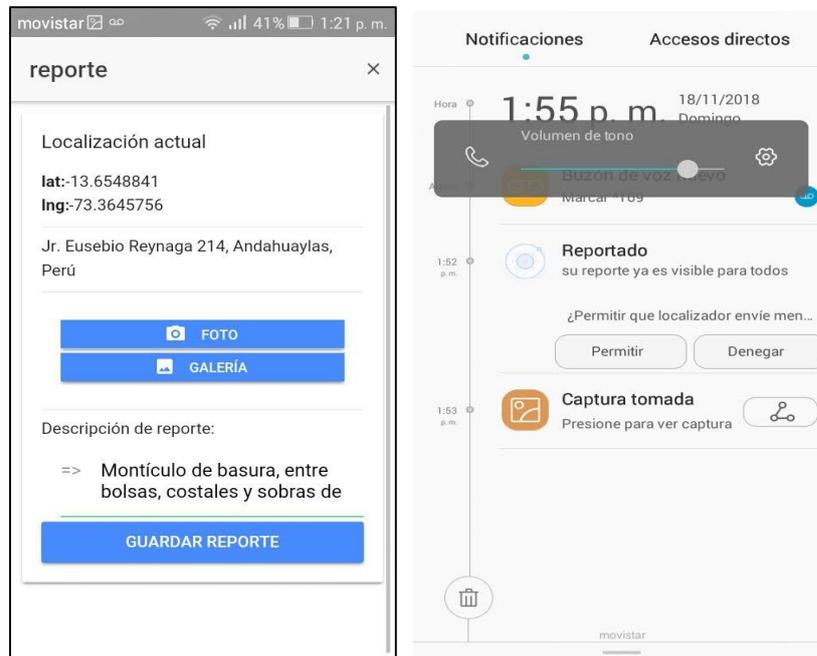


Figura 57: Reporte
Elaboración propia

También se realizó la verificación Lista de reportes



Figura 58: Lista de reportes
Elaboración propia

Notificaciones



Figura 59: Notificaciones
Elaboración propia

Vista del administrador, gestiona los reportes

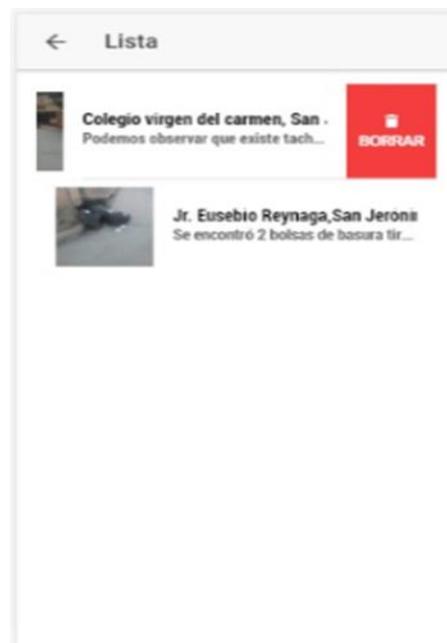


Figura 60: Eliminar Reporte
Elaboración propia

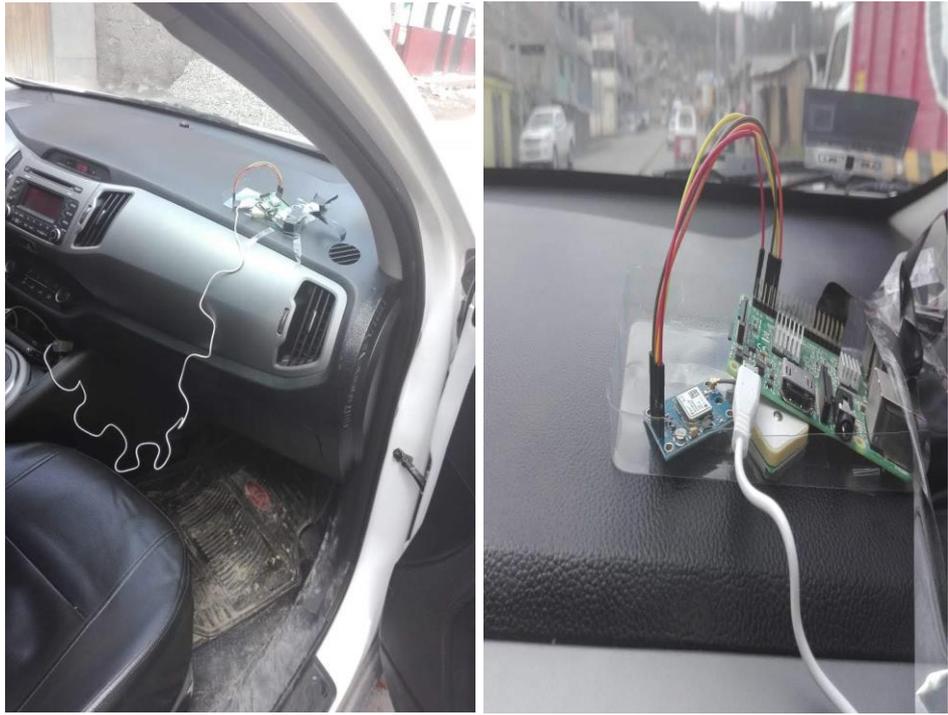


Figura 61: Pruebas
Elaboración propia

CONCLUSIONES

El objetivo de esta tesis era desarrollar un sistema de geolocalización para determinar la ubicación del vehículo recolector de residuos, en tiempo real, de esta manera disminuir la contaminación y tener una participación más activa del ciudadano, logrando cumplirlo gracias a que se ha alcanzado cada sub objetivo planteado inicialmente:

- Se analizó la arquitectura raspberry Pi y se ha logrado desarrollar el sistema con la arquitectura ya que se vio su gran funcionalidad de adaptabilidad con otros dispositivos, contar con un sistema operativo lo cual permitió hacer uso de script desarrollados en python, así como también por su reducido tamaño.
- Se ha diseñado y desarrollado el módulo reporte con una interfaz amigable y sencilla de usar, cuya funcionalidad es permitir al usuario realizar reporte de residuos sólidos que pueda ver en la calle.
- En esta tesis se desarrolló el módulo mostrar vehículo que permite al usuario visualizar en un mapa interactivo el recorrido en tiempo real del vehículo recolector de residuos.
- Se logró realizar las pruebas del funcionamiento de la geolocalización haciendo uso del gps, llegando a obtener las coordenadas exactas de latitud y longitud del punto en el que se encontraba el usuario.

RECOMENDACIONES

Para habilitar los puertos seriales de la raspberry Pi 3 se recomienda tener cuidado con la línea de códigos porque esto podría matar el sistema operativo que está instalado en la raspi, así como también tener en consideración que este procedimiento de activación varía según el tipo de Raspberry Pi.

Cada módulo que se va desarrollando se tiene que ir probando y solucionando los errores que pudiera presentar, como exige la metodología V, la cual es recomendada para el desarrollo de sistemas embebidos.

Los objetivos planteados en el trabajo de tesis fueron enfocados exclusivamente al sistema de geolocalización sin embargo se observa la necesidad de que en un futuro se incorpore un módulo de optimización de las rutas del vehículo recolector de residuos haciendo uso de grafos y árboles de decisiones, este módulo permitiría reducir el tiempo que emplea en el recorrido por las arterias de la ciudad, también optimizaría el uso del combustible conllevando esto a la reducción de la contaminación.

REFERENCIAS BIBLIOGRÁFICAS

A. Perez, O. (2006). Una metodología para el desarrollo de hardware y sistemas.

Acuri, G., & et-al. (1997). DIAGNÓSTICO DE LA SITUACIÓN DEL MANEJO DE RESIDUOS SÓLIDOS.

Alonso, A., & et al. (2011). Dispositivos móviles .

amazon. (2018). amazon. Obtenido de www.aws.amazon.com/es/es/nosql/

Bermúdez, M. (2010). Contaminacion y turismo sostenible.

Chiliquina, V. (2010). “Estudio comparativo de las tecnologías asp, ajax, j2ee, j2me y .net para el desarrollo de aplicaciones móviles, aplicado al desarrollo de una aplicación móvil web para registro de calificaciones y asistencia de los alumnos de la escuela de ingeniería de . Ecuador.

Datadec. (2018). Aplicación en movilidad para la gestión de Residuos. Obtenido de <https://www.datadec.es/expertmovil-residuos/>

Diana, B., & Germán , G. (2015). Prototipo de sistema de vigilancia basado en la internet de las cosas con aplicativo para dispositivos móviles. Bogotá.

Esquivel, L. J. (2016). Sistema de geo localización vía web y móvil para mejorar la búsqueda de personas en desastres naturales en la ciudad de Trujillo 2016. Trujillo: Universidad Cesar Vallejo.

Firebase. (7 de junio de 2016)). Firebase Realtime Database. Obtenido de <https://firebase.google.com/docs/database/>

Gómez, R. (2013). Sistemas Operativos endispositivos móviles. Madrid: Universidad Pontificia de Salamanca.

González , C. (2015). Aplicaciones orientadas a la domótica con Raspberry Pi. Sevilla.

Gonzales, J., & Marin, C. (2015). Prototipo de sistema de seguridad en vehículos de transporte público que permite la captura de imagen fotográfica, posicionamiento global y almacenamiento en base de datos. Colombia.

González, C., & Palomo, B. (2015). Aplicaciones orientadas a la domótica con Raspberry Pi. Sevilla.

Guerra, F. (2013). Diseño de un sistema de control domótico y video. Lima.

Guerra, F. (2013). Diseño de un sistema de control domótico y video vigilancia supervisado por el teléfono móvil. Lima.

Huaraca, R., & et-al. (2017). Sistema de Geolocalización en tiempo real, para el servicio de transporte universitario, Universidad Nacional José María Arce. Provincia de Andahuaylas.

Ionic. (2018). Ionic. Obtenido de <https://ionicframework.com>

Martínez, F. (2011). Aplicaciones para dispositivos Mviles.

Ministerio del Ambiente. (2005). Ley General del Ambiente. Gestión Ambiental, 27.

Ministerio del Ambiente. (2015). Anuario de Estadísticas Ambientales . Perú.

Ministerio del ambiente. (2015). Ley de residuos sólidos. Obtenido de <http://www.minam.gob.pe/>

Pedrozo Petrazzini, G. (2012). Sistemas operativos en dispositivos móviles. Argentina: Universidad Nacional Nordeste.

PMBOK. (2017). PMBOK.

Sancho, H. (2017). Desarrollo de un sistema de localización y aplicación móvil para vehículos en aparcamientos. Universidad Politécnica de Valencia, Valencia.

Upton, E., & Halfacree, G. (2014). Raspberry Pi Guía Del Usuario.

Wolf, G., & et-al. (2014). Fundamentos de Sistemas Operativos.

ANEXOS

ANEXO 1: GLOSARIO DE ACRÓNIMOS

GPS	Sistema de Posicionamiento Global (Global Positioning System)
APP	Aplicación Informática
PMBOOK	Proyecto Organismo de Gestión del Conocimiento (Project Management Body of Knowledge)
PMI	Instituto de Manejo de Proyectos (Project Management Institute)
TIC	Tecnologías de la Información
MIT	Instituto De Tecnología De Massachusetts
USB	Universal Serial Bus (Universal Serial Bus)
PC	Computadora Personal (Personal Computer)
SD	Seguro Digital (Secure Digital)
SDHC	Seguridad Digital de Alta Capacidad (Secure Digital High Capacity)
SDXC	Capacidad Digital Segura Extendida (Secure Digital eXtended Capacity)
ARM	Ordenador con Conjunto Reducido de Instrucciones (Reduced Instruction Set Computer)
CSI	Interfaz Serie para Cámaras (Camera Interface Serie)
HDMI	Interfaz Multimedia de Alta Definición (High Definition Multimedia Interface)
DSI	Display serial Interface (Display Serial Interface)
GPIO	Entrada/Salida de Propósito General (General Purpose Input/Output)
PDAS	Ayudante personal digital (Personal Digital Assistant)
SO	Sistema Operativo

API	Interfaz de Programación de Aplicaciones (Application Programming Interface)
JSON	Notación de objetos de JavaScript (JavaScript Object Notation)
SDK	Kit de Desarrollo de Software (Software Development Kit)
GNSS	Sistema Global de Navegación por Satélite (Sistema Global De Navegacion Por Satelite)
SQA	Aseguramiento de la calidad del Software (Software De Calidad Assurance)
RAM	Memoria de Acceso Aleatorio (Random Access Memory)
UML	Lenguaje de modelado unificado (Unified Modeling Language)
NMEA	Asociación Nacional de Electrónica Marina (National Marine Electronics Association)
CLI	Interfaz de línea de comandos (Command line interface)
AQL	Límite de Calidad Aceptable (Acceptance Quality Limit)
UART	Transmisor-Receptor Asíncrono Universal (Universal Asynchronous Receiver)

ANEXO 2: CÓDIGO FUENTE

CÓDIGO FUENTE 1 MÓDULO 1: mostrar vehículo

“código Html”

```
<ion-header>

  <ion-navbar>
    <ion-title>vercamion</ion-title>
  </ion-navbar>

</ion-header>

<ion-content padding>

  <ion-list>
    <ion-item *ngFor="let sitio of sitios"
(click)="muestraSitio(sitio)">
      <ion-thumbnail item-left>
```

```

        <img [src]="sitio.foto">
    </ion-thumbnail>
    <h2>{{ camion.lati }}</h2>
    <p>{{ camion.long }}</p>

    </ion-item>
</ion-list>

    <div #map id="map"></div>
</ion-content>

```

“código ts”

```

import { Component, ViewChild, ElementRef } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { AngularFireDatabase } from 'angularfire2/database';
import { FirebaseDbProvider } from '../providers/firebase-
db/firebase-db';

declare var google: any;
/**
 * Generated class for the VercamionPage page.
 *
 * See https://ionicframework.com/docs/components/#navigation for
more info on
 * Ionic pages and navigation.
 */

@IonicPage()
@Component({
  selector: 'page-vercamion',
  templateUrl: 'vercamion.html',
})
export class VercamionPage {

  //camion: any;
  camion : any = { lati: 0, long: 0 }

  @ViewChild('map') mapRef: ElementRef;

```

```

constructor(public navCtrl: NavController,
  public navParams: NavParams,
  public afDB: AngularFireDatabase,
  public dbFirebase :FirebaseDbProvider
) {

}
//-----

ionViewDidEnter(){

  this.dbFirebase.getLati().subscribe(camion=>{
    this.camion = camion;//alert(this.camion.lati);

    const location = new google.maps.LatLng(this.camion.lati,
this.camion.long);

    const option={
      center: location,
      zoom: 15,
      streetViewControl: false,
      mapTypeId: 'hybrid'
    };

    const map = new
google.maps.Map(this.mapRef.nativeElement,option);
    this.addMarker(location, map);
  })
}

ionViewDidLoad() {
  this.showMap();
}

showMap() {
  this.camion.lati= '';
  //ubicacion con lat y long
  const location = new google.maps.LatLng();

  const option={
    center: location,
    zoom: 15,

```

```

        streetViewControl: false,
        mapTypeId: 'hybrid'
    };

    const map = new
google.maps.Map(this.mapRef.nativeElement,option);
    this.addMarker(location, map);
    }

    addMarker(position, map){
        return new google.maps.Marker({
            position,
            map
        })
    }
}
}

```

CÓDIGO FUENTE 2 MÓDULO 2: mi ubicación

“código Html”

```

<ion-header>

    <ion-navbar>
        <ion-title>miubi</ion-title>
    </ion-navbar>

</ion-header>

<ion-content padding>
    <div #map id="map"></div>
    <ion-fab right top>
        <button ion-fab (tap)="nuevoSitio()">
            <ion-icon name="pin"></ion-icon>
            <ion-icon name="add"></ion-icon>
        </button>
    </ion-fab>
</ion-content>

```

“código ts”

```
import { Component } from '@angular/core';
import { NavController, NavParams, ModalController } from 'ionic-
angular';

import { Geolocation } from '@ionic-native/geolocation';
import { Platform } from 'ionic-angular';
import { ReportePage } from '../reporte/reportes';

declare var google: any;
/**
 * Generated class for the MiubiPage page.
 *
 * See https://ionicframework.com/docs/components/#navigation for
more info on
 * Ionic pages and navigation.
 */

@Component({
  selector: 'page-miubi',
  templateUrl: 'miubi.html',
})
export class MiubiPage {

  map: any; // Manejador del mapa.
  coords : any = { lat: 0, lng: 0 }

  constructor(public navCtrl: NavController, public NavParams:
NavParams, public platform: Platform, private geolocation:
Geolocation, public modalCtrl : ModalController) {

    platform.ready().then(() => {
      // La plataforma esta lista y ya tenemos acceso a los plugins.
      this.obtenerPosicion();
    });
  }

  loadMap(){
    let mapContainer = document.getElementById('map');
    this.map = new google.maps.Map(mapContainer, {
      center: this.coords,
      zoom: 12
    });
  }
}
```

```

    });
    // Colocamos el marcador
    new google.maps.Marker({
      icon : 'assets/imgs/ico_estoy_aqui.png',
      map: this.map,
      position: this.coords
    });
  }

  obtenerPosicion():any{
    this.geolocation.getCurrentPosition().then(res => {
      this.coords.lat = res.coords.latitude;
      this.coords.lng = res.coords.longitude;

      this.loadMap();
    })
    .catch(
      (error)=>{
        console.log(error);
      }
    );
  }

  nuevoSitio(){
    // aquí vamos a abrir el modal para añadir nuestro sitio.
    let mimodal = this.modalCtrl.create( ReportePage,this.coords
  );
    mimodal.present();
  }
}

```

CÓDIGO FUENTE 3 MÓDULO 3: reporte

“código html”

```

<ion-header>

<ion-navbar>
  <ion-title>reporte</ion-title>
  <ion-buttons start>
    <button ion-button (click)="cerrarModal()">
      <ion-icon name="md-close"></ion-icon>
    </button>
  </ion-buttons>

```


“código ts”

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ViewController }
from 'ionic-angular';
import { Camera, CameraOptions } from '@ionic-native/camera';
import { FirebaseDbProvider } from '../providers/firebase-
db/firebase-db';

declare var google: any;

/**
 * Generated class for the ReportePage page.
 *
 * See https://ionicframework.com/docs/components/#navigation for
more info on
 * Ionic pages and navigation.
 */

@IonicPage()
@Component({
  selector: 'page-reporte',
  templateUrl: 'reporte.html',
})
export class ReportePage {

  coords : any = { lat: 0, lng: 0 }
  address: string;
  description: string = '';
  foto: any = '';
  image: string = null;

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,
    private viewCtrl : ViewController,
    private camera: Camera,
    private dbFirebase :FirebaseDbProvider) {}

  ionViewDidLoad() {
    console.log('ionViewDidLoad ReportePage');
    this.coords.lat = this.navParams.get('lat');
    this.coords.lng = this.navParams.get('lng');

    this.getAddress(this.coords).then(results=> {
      this.address = results[0]['formatted_address'];
    });
  }
}
```

```

    }, errStatus => {
        // Aquí iría el código para manejar el error
    });
}

cerrarModal(){
    this.viewCtrl.dismiss();
}

getAddress(coords):any {
    var geocoder = new google.maps.Geocoder();

    return new Promise(function(resolve, reject) {
        geocoder.geocode({'location': coords} , function (results,
status) { // llamado asincronamente
            if (status == google.maps.GeocoderStatus.OK) {
                resolve(results);
            } else {
                reject(status);
            }
        });
    });
}

sacarFoto(){

    let cameraOptions : CameraOptions = {
        quality: 50,
        encodingType: this.camera.EncodingType.JPEG,
        targetWidth: 800,
        targetHeight: 600,
        destinationType: this.camera.DestinationType.DATA_URL,
        sourceType: this.camera.PictureSourceType.CAMERA,
        correctOrientation: true
    }

    this.camera.getPicture(cameraOptions).then((imageData) => {
        // imageData is a base64 encoded string
        this.foto = "data:image/jpeg;base64," + imageData;
    }, (err) => {
        console.log(err);
    });
}

Galeria(){

```

```

let cameraOptions : CameraOptions = {
  quality: 50,
  encodingType: this.camera.EncodingType.JPEG,
  targetWidth: 800,
  targetHeight: 600,
  destinationType: this.camera.DestinationType.DATA_URL,
  sourceType: this.camera.PictureSourceType.SAVEDPHOTOALBUM,
  correctOrientation: true
}

this.camera.getPicture(cameraOptions).then((imageData) => {
  // imageData is a base64 encoded string
  this.foto = "data:image/jpeg;base64," + imageData;
}, (err) => {
  console.log(err);
});
}

guardarSitio(){
  let sitio = {
    lat: this.coords.lat,
    lng: this.coords.lng ,
    address: this.address,
    description: this.description,
    foto: this.foto
  }

  this.dbFirestore.guardaSitio(sitio).then(res=>{
    console.log('Sitio guardado en firebase:');
    this.cerrarModal();
  })
}
}

```

CÓDIGO FUENTE 4 MÓDULO 4: lista de reporte

“código html”

```

<ion-header>

  <ion-navbar>
    <ion-title>Lista</ion-title>
  </ion-navbar>

</ion-header>

```

```

<ion-content padding>
  <ion-list>
    <ion-item *ngFor="let sitio of sitios"
(click)="muestraSitio(sitio)">
      <ion-thumbnail item-left>
        <img [src]="sitio.foto">
      </ion-thumbnail>
      <h2>{{ sitio.address }}</h2>
      <p>{{ sitio.description }}</p>

    </ion-item>
  </ion-list>
</ion-content>

```

“código ts”

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ModalController }
from 'ionic-angular';
import { FirebaseDbProvider } from '../providers/firebase-
db/firebase-db';
import { ModalDetalleSitioPage } from '../modal-detalle-
sitio/modal-detalle-sitio';

/**
 * Generated class for the ListaPage page.
 *
 * See https://ionicframework.com/docs/components/#navigation for
more info on
 * Ionic pages and navigation.
 */

@IonicPage()
@Component({
  selector: 'page-lista',
  templateUrl: 'lista.html',
})
export class ListaPage {

  sitios: any;

  constructor(public navCtrl: NavController,
    public NavParams: NavParams,

```

```

public modalCtrl : ModalController,
public dbFirebase :FirebaseDbProvider

) {
}

ionViewDidLoad() {
  console.log('ionViewDidLoad ListaPage');
}

ionViewDidEnter(){

  this.dbFirebase.getSitios().subscribe(sitios=>{
    this.sitios = sitios;
  })

}

muestraSitio(sitio){
  let modalSitio = this.modalCtrl.create( ModalDetalleSitioPage,
sitio );
  modalSitio.present();
}
}

```

Habilitar puertos UART raspberry

La comunicación entre la raspberry pi y el gps se realiza a través de un puerto serie (UART), por lo tanto el primer paso ha sido configurar el módulo UART de la raspberry pi, mediante los siguientes comandos:

1. Editar nano config.txt archivo:


```
sudo nano /boot/config.txt
```
2. Agregue la siguiente línea al archivo y guardar los cambios:


```
enable_uart=1
```
3. Una vez modificado el config.txt que necesitamos para deshabilitar el registro en el enlace serie, abra esta cdmline.txt


```
sudo nano /boot/cmdline.txt
```
4. Coloque una # delante de la línea original, y luego añadir la segunda línea del ejemplo a:

```
#dwc_otg.lpm_enable=0      console=serial0,115200      console=tty1  
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes  
rootwait
```

```
dwc_otg.lpm_enable=0      console=tty1      root=/dev/mmcblk0p2  
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait
```

5. Reiniciar el Raspberry Pi:

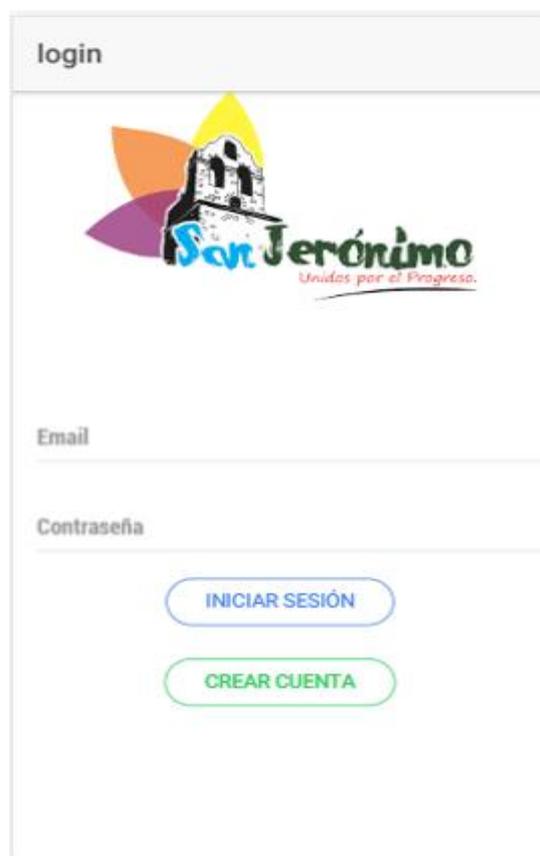
```
sudo reboot -h 0
```

MANUAL DE USUARIO

MANUAL DE USO DEL SISTEMA GEOLOCALIZACIÓN

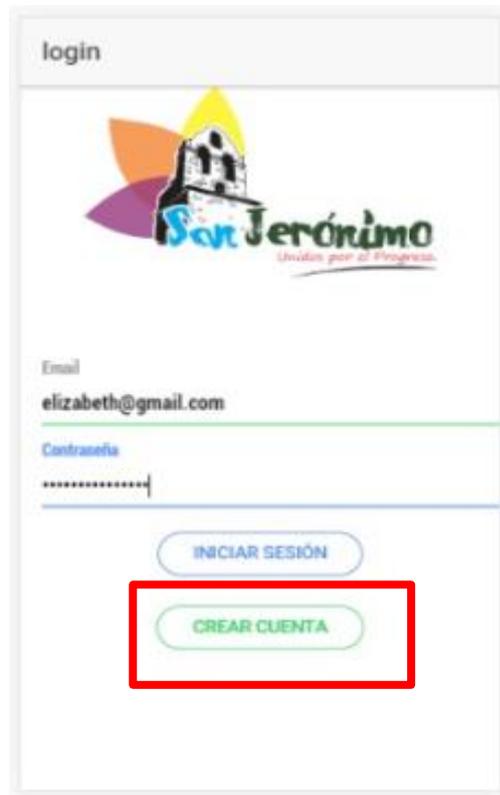
Manual de usuario

1. Ingresar al menú del dispositivo móvil, ejecutar la aplicación e iniciar sesión, el usuario para iniciar sesión deberá registrarse mediante email y password.



The image shows a mobile application login screen. At the top, there is a header with the word "login". Below the header is the logo for "San Jerónimo", which features a stylized building and the tagline "Unidos por el Progreso." Underneath the logo, there are two input fields: one labeled "Email" and another labeled "Contraseña". At the bottom of the screen, there are two buttons: a blue button labeled "INICIAR SESIÓN" and a green button labeled "CREAR CUENTA".

2. Registro de un nuevo usuario: ingresar correo y una contraseña, pulse en el botón registrar.



The image shows a login page titled 'login' for 'San Jerónimo'. The logo features a stylized building with a yellow flame above it, and the text 'San Jerónimo' in blue and red, with the tagline 'Unidos por el Progreso.' below it. The page has two input fields: 'Email' with the text 'elizabeth@gmail.com' and 'Contraseña' with a masked password '*****'. Below the fields are two buttons: 'INICIAR SESIÓN' in blue and 'CREAR CUENTA' in green. The 'CREAR CUENTA' button is highlighted with a red rectangular box.

3. Menú principal: en este apartado tiene las opciones:
 - Mostrar vehículo: al pulsar en este botón le muestra la ubicación del recorrido que hace el vehículo
 - Mi ubicación: desde este botón podrá cargar y visualizar la posición actual en la que se encuentra, permitiéndole generar un reporte.
 - Reportar: mediante este módulo podrá reportar la incidencia de acumulación de residuos en la calle.
 - Lista de reportes: al elegir esta opción puede visualizar la lista de reportes realizados
 - Cerrar sesión: desde esta opción podrá salir del sistema.



4. Módulo reporte: desde aquí podrá enviar el reporte ingresando los campos descripción y una imagen(galeria o foto)

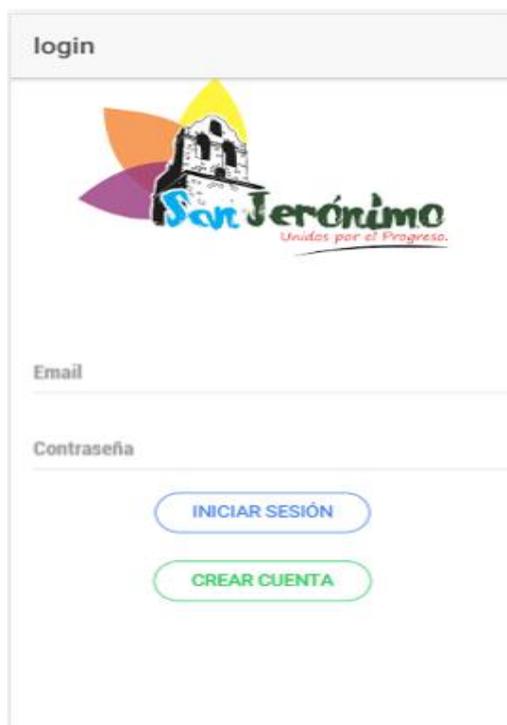


5. Módulo lista de reportes



MANUAL DEL ADMINISTRADOR

1. Iniciar sesión: Ingresar al aplicativo del administrador, ejecutar e iniciar sesión, el administrador para iniciar sesión deberá ingresar mediante email y password.



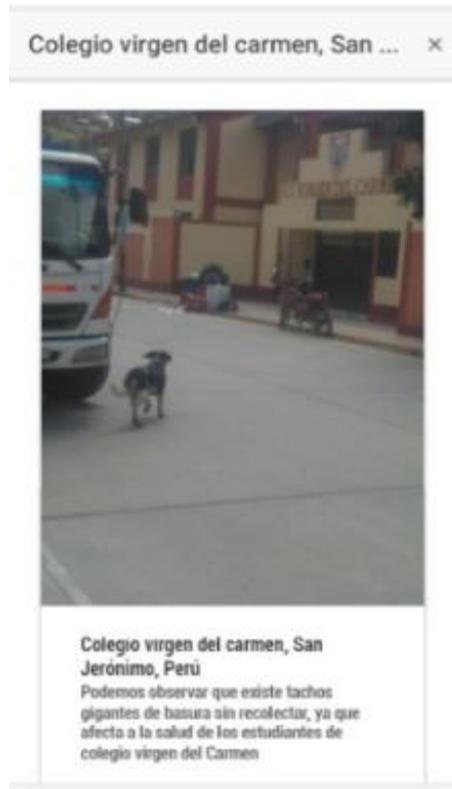
2. Menú principal: en este apartado tiene las opciones de visualizar el recorrido del vehículo y administrar los reportes (trazar la ruta, lista reporte, eliminar reporte).



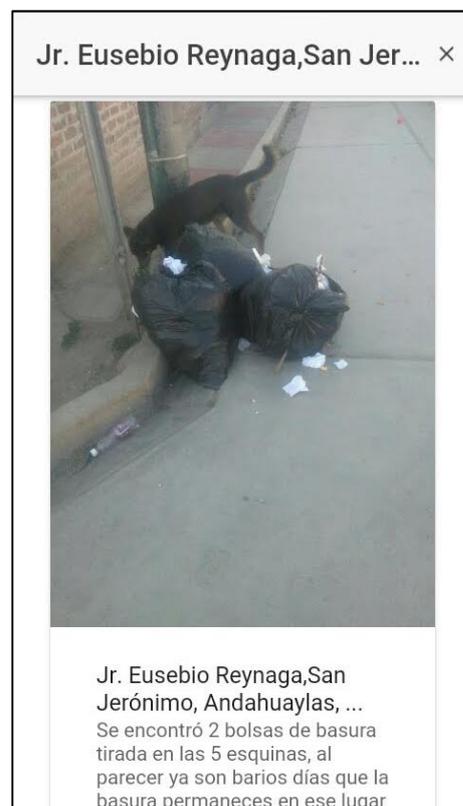
3. Lista de reportes

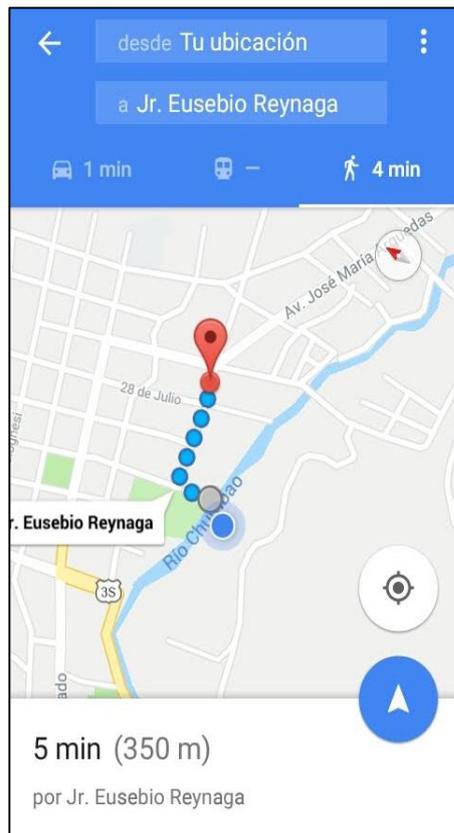


4. Ver detalle de reporte



5. Como llegar





6. Eliminar reporte

