



UNIVERSIDAD
PRIVADA
DEL NORTE

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

“APLICACIÓN MÓVIL DE ALGORITMOS DE
RUTAS ÓPTIMAS Y SU EFECTO EN EL
DESPLAZAMIENTO DE LOS CONDUCTORES DE
VEHÍCULOS EN LA CIUDAD DE TRUJILLO”

Tesis para optar el título profesional de:

Ingeniero de Sistemas Computacionales

Autor(es):

Br. Raúl David Mera Dávila
Br. Wilder Eder Salinas Acosta

Asesor:

Ms. Ing. Lourdes Díaz Amaya

Trujillo – Perú
2018

APROBACIÓN DE LA TESIS

El (La) asesor(a) y los miembros del jurado evaluador asignados, **APRUEBAN** la tesis desarrollada por los Bachilleres **Raúl David Mera Dávila y Wilder Eder Salinas Acosta**, denominada:

**“APLICACIÓN MOVIL DE ALGORITMOS DE RUTAS ÓPTIMAS Y EN SU
EFECTO EN EL DESPLAZAMIENTO DE LOS CONDUCTORES DE
VEHÍCULOS EN LA CIUDAD DE TRUJILLO”**

Mg. Ing. Lourdes Díaz Amaya
ASESORA

Mg. Ing. Víctor Enemesio Dávila Rodríguez
**JURADO
PRESIDENTE**

Mg. Ing. Luis Mauricio Gutiérrez Magán
JURADO

Mg. Ing. Rolando Javier Berrú Beltrán
JURADO

DEDICATORIA

A mi madre, quien me enseñó a valorar desde lo más insignificante a lo más valioso de la vida, aquella que con su educación me demostró la responsabilidad y compromiso hacia a mí, además a nunca rendirme sobre las metas propuestas que fue pieza clave en esta investigación; mi padre por brindarme los buenos consejos que me ayudaron a tomar las mejores decisiones para cuando las cosas sean necesarias, a mis tíos y primos y demás familiares que con sus buenos ánimos y consejos han aportado en mi formación como persona y como profesional.

David.

DEDICATORIA

A Dios maravilloso y todopoderoso del cielo y de la tierra; porque su misericordia es mejor que la vida, porque cuya voluntad fue permitirme lograr alcanzar una meta más en mi vida. Brindándome la salud, inteligencia, entendimiento y el gozo para la culminación del presente proyecto. Muchas gracias.

A mi familia por su incondicional apoyo, a mi madre por su amor y por creer en mí en los momentos más difíciles, a mi tía por su inagotable apoyo y fe, a mis hermanas por su cariño y alegría, a mi padre por su apoyo. Muchas gracias.

Eder.

AGRADECIMIENTO

A nuestra asesora de tesis Mg. Ing. Lurdes Díaz Amaya por su apoyo y paciencia durante el inicio, desarrollo y culminación del presente proyecto. Muchas gracias.

Al Mg. Ing. César Liza Ávila por los conocimientos impartidos durante nuestra vida universitaria que fueron pieza clave y fundamental en el desarrollo del presente proyecto. Muchas gracias.

ÍNDICE DE CONTENIDOS

APROBACIÓN DE LA TESIS.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS.....	viii
ÍNDICE DE FIGURAS	ix
RESUMEN.....	x
ABSTRACT	xi
CAPÍTULO 1. INTRODUCCIÓN.....	13
1.1. Realidad problemática	13
1.2. Formulación del problema.....	16
1.3. Justificación.....	16
1.4. Limitaciones	17
1.5. Objetivos	17
1.5.1. <i>Objetivo general</i>	17
1.5.2. <i>Objetivos específicos</i>	17
CAPÍTULO 2. MARCO TEÓRICO.....	18
2.1. Antecedentes	18
2.2. Bases teóricas.....	20
2.2.1. <i>Algoritmo</i>	20
2.2.2. <i>Algoritmos de Rutas Cortas</i>	21
2.2.3. <i>Algoritmo DFS (Búsqueda en profundidad)</i>	21
2.2.4. <i>Algoritmo BFS (Búsqueda en anchura)</i>	23
2.2.5. <i>Algoritmo Dijkstra</i>	24
2.2.6. <i>Algoritmo Floyd-Warshall</i>	25
2.2.7. <i>Algoritmo Bellman-Ford</i>	26
2.2.8. <i>Algoritmo A*</i>	27
2.2.9. <i>Grafo</i>	29
2.2.10. <i>Android</i>	32
2.2.11. <i>Java</i>	34
2.2.12. <i>PostgreSQL</i>	37
2.2.13. <i>Android Studio</i>	39
2.2.14. <i>Eclipse</i>	40
2.2.15. <i>pgAdmin</i>	43
2.2.16. <i>Sistema de Posicionamiento Global (GPS)</i>	43
CAPÍTULO 3. HIPÓTESIS.....	52

3.1.	Formulación de la Hipótesis.....	52
3.2.	Operacionalización de variables.....	52
CAPÍTULO 4. DESARROLLO.....		53
4.1.	Proceso de desarrollo de software Móvil y web	53
4.1.1.	<i>Disciplinas aplicadas</i>	53
4.2.	Documento Visión	54
4.2.1.	<i>Introducción</i>	54
4.2.2.	<i>Posicionamiento</i>	54
4.2.3.	<i>Definición de Stakeholders</i>	55
4.2.4.	<i>Descripción del producto</i>	55
4.2.5.	<i>Características del producto</i>	56
4.2.6.	<i>Requerimientos del producto</i>	56
4.3.	Prototipos	57
4.3.1.	<i>Móvil</i>	57
4.3.2.	<i>Web</i>	57
4.4.	Modelo del Dominio	58
4.5.	Modelo de Casos de Uso.....	58
4.5.1.	<i>Descripción Global</i>	58
4.5.2.	<i>Requerimientos Específicos</i>	60
4.6.	Fase de Planificación	61
4.6.1.	<i>Planificación Inicial</i>	61
4.6.2.	<i>Estimación de Tiempos</i>	61
4.7.	Fase de Iteraciones.....	62
4.7.1.	<i>Iteración 01</i>	62
4.7.2.	<i>Iteración 02</i>	70
4.7.3.	<i>Iteración 03</i>	80
CAPÍTULO 5. METODOLOGÍA.....		85
5.1.	Diseño de investigación	85
5.2.	Unidad de estudio	85
5.3.	Población	85
5.4.	Muestra (muestreo o selección).....	85
5.5.	Técnicas, instrumentos y procedimientos de recolección de datos	86
5.6.	Métodos, instrumentos y procedimientos de análisis de datos	86
CAPÍTULO 6. RESULTADOS		87
CAPÍTULO 7. DISCUSIÓN.....		98
CONCLUSIONES.....		101
RECOMENDACIONES		102
REFERENCIAS.....		103
ANEXOS		106

ÍNDICE DE TABLAS

Tabla 1. <i>Tabla de Ranking de horas de Congestionamiento del año 2017.</i>	13
Tabla 2. <i>Parque automotor en circulación a nivel nacional, según departamento, 2008 - 2016.</i>	14
Tabla 3. <i>Tabla Parque automotor de Trujillo del año 2015.</i>	15
Tabla 4. <i>Cuadro Comparativo entre los algoritmos de rutas cortas.</i>	29
Tabla 5. <i>Cuadro Comparativo entre Android vs IOS.</i>	34
Tabla 6. <i>Cuadro Comparativo entre C# vs Java.</i>	37
Tabla 7. <i>Cuadro Comparativo entre PostgreSQL vs MySQL.</i>	38
Tabla 8. <i>Cuadro Comparativo entre Eclipse vs Netbeans vs Visual Studio.</i>	42
Tabla 9. <i>Cuadro Comparativo entre Android Studio vs Eclipse.</i>	42
Tabla 10. <i>Cuadro Comparativo de las Metodologías de desarrollo.</i>	46
Tabla 11. <i>Operacionalización de variable independiente.</i>	52
Tabla 12. <i>Operacionalización de variable dependiente.</i>	52
Tabla 13. <i>Fases de Metodología ICONIX.</i>	53
Tabla 14. <i>Enunciado del problema.</i>	54
Tabla 15. <i>Enunciado del posicionamiento del producto.</i>	55
Tabla 16. <i>Resumen Stakeholders.</i>	55
Tabla 17. <i>Resumen de Usuarios.</i>	55
Tabla 18. <i>Beneficios del producto.</i>	56
Tabla 19. <i>Listado de casos de uso.</i>	59
Tabla 20. <i>Planificación Inicial de tareas.</i>	61
Tabla 21. <i>Estimación de tiempos de tarea.</i>	61
Tabla 22. <i>Plan de entrega.</i>	62
Tabla 23. <i>Prueba funcional de caja negra – Gestionar Incidencia.</i>	67
Tabla 24. <i>Cuadro detallado de técnicas, instrumentos y procedimientos de recolección de datos</i>	86
Tabla 25. <i>Cuadro detallado de métodos, instrumentos y procedimientos de análisis de datos</i>	86
Tabla 26. <i>Tipos de Indicadores.</i>	87
Tabla 27. <i>PreTest – PostTest Tiempo de Recorrido.</i>	88
Tabla 28. <i>PreTest – PostTest Distancia en recorrido.</i>	91
Tabla 29. <i>PreTest – PostTest Número de incidencias por conductor.</i>	95

ÍNDICE DE FIGURAS

Figura 1. Algoritmo DFS, Pseudocódigo.....	22
Figura 2. Algoritmo BFS, Pseudocódigo.....	23
Figura 3. Algoritmo Dijkstra, Pseudocódigo.....	24
Figura 4. Algoritmo Floyd-Warshall, Pseudocódigo.....	26
Figura 5. Algoritmo Bellman-Ford, Pseudocódigo.....	27
Figura 6. Algoritmo A*, Pseudocódigo.....	28
Figura 7. Representación de un Grafo.....	30
Figura 8. Grado Dirigido.....	31
Figura 9. Grafo No Dirigido.....	31
Figura 10. Matriz de Adyacencia.....	32
Figura 11. Lista de Adyacencia.....	32
Figura 12. Arquitectura del Sistema Operativo Android.....	33
Figura 13. Metodología y Fases de XP.....	44
Figura 14. Ciclo de vida de la metodología RUP.....	45
Figura 15. Flujo de Proceso ICONIX.....	46
Figura 16. GPS.....	48
Figura 17. GPS, Error ruta múltiple.....	50
Figura 18. GPS, Corrección Diferencial.....	51
Figura 19. Prototipo – CU Generar Ruta Óptima. (Fuente Propia).....	57
Figura 20. Prototipo – CU Gestionar Incidencias (Registrar Incidencia). (Fuente Propia).....	57
Figura 21. Prototipo – CU Gestionar Incidencias (Editar Incidencia – Listar Incidencias). (Fuente Propia).....	58
Figura 22. Modelo de Dominio (Fuente Propia).....	58
Figura 23. Diagrama de casos de uso. (Fuente Propia).....	59
Figura 24. Diagrama de Clases de Datos UC – Gestionar Incidencias. (Fuente Propia).....	64
Figura 25. Diagrama de Robustez UC – Gestionar Incidencias. (Fuente Propia).....	65
Figura 26. Diagrama de secuencia UC – Gestionar Incidencias. (Fuente Propia).....	66
Figura 27. Pruebas Unitarias UC – Gestionar Incidencias. (Fuente Propia).....	68
Figura 28. Prueba Funcional UC – Gestionar Incidencias. (Fuente Propia).....	69
Figura 29. Diagrama de Clases de Datos UC – Generar Ruta Óptima - Android. (Fuente Propia).....	71
Figura 30. Diagrama de Clases de Datos UC – Generar Ruta Óptima - WPS. (Fuente Propia).....	72
Figura 31. Diagrama de Robustez UC – Generar ruta óptima - Android. (Fuente Propia).....	73
Figura 32. Diagrama de Robustez UC – Generar ruta óptima - WPS. (Fuente Propia).....	73
Figura 33. Diagrama de secuencia UC – Generar ruta óptima - Android. (Fuente Propia).....	74
Figura 34. Diagrama de secuencia UC – Generar ruta óptima - WPS. (Fuente Propia).....	74
Figura 35. Pruebas Unitarias UC – Generar Ruta Óptima - Móvil. (Fuente Propia).....	75
Figura 36. Pruebas Unitarias UC – Generar Ruta Óptima - Móvil. (Fuente Propia).....	76
Figura 37. Pruebas Funcional UC – Generar Ruta Óptima - Móvil. (Fuente Propia).....	76
Figura 38. Pruebas Funcional UC – Generar Ruta Óptima - Geoserver. (Fuente Propia).....	77
Figura 39. Pruebas Funcional UC – Generar Ruta Óptima - Geoserver. (Fuente Propia).....	78
Figura 40. Pruebas Funcional UC – Generar Ruta Óptima - Geoserver. (Fuente Propia).....	79
Figura 41. Diagrama de flujo – Algoritmo A*. (Fuente Propia).....	80
Figura 42. Diagrama de Modelo de Datos. (Fuente Propia).....	81
Figura 43. Diagrama de Base de Datos. (Fuente Propia).....	82
Figura 44. Diagrama de Componentes. (Fuente Propia).....	83
Figura 45. Diagrama de Despliegue. (Fuente Propia).....	84
Figura 46. Gráfico de Distribución Z. (Fuente Propia).....	90
Figura 47. Gráfico de Distribución Z. (Fuente Propia).....	93
Figura 48. Gráfico de Distribución Z. (Fuente Propia).....	97
Figura 49. Promedio de medición para el indicador Tiempo en recorrido. (Fuente Propia).....	98
Figura 50. Promedio de medición para el indicador Distancia recorrida. (Fuente Propia).....	99
Figura 51. Promedio de medición para el indicador Número de incidencias evitadas. (Fuente Propia).....	100

RESUMEN

El presente proyecto se enmarca en el ámbito de la ingeniería, en el campo de “Análisis y Diseño de Algoritmos”, donde a través del procesamiento de datos geoespaciales y el uso de algoritmos de rutas óptimas se generarán rutas alternas óptimas que se proporcionarán en tiempo real mediante una aplicación móvil con la finalidad de proponer mejores rutas viales a los conductores evitando atascos por incidentes como tráfico vehicular, obras municipales, accidentes de tránsito y otros, dentro de la malla vial de la ciudad de Trujillo.

El tipo de estudio es experimental con diseño pre-test y post-test. La muestra está constituida por un total de 33 conductores que se escogieron aleatoriamente en la ciudad de Trujillo. Para la obtención de datos necesarios se utilizó un cuestionario gráfico de mapas, donde son plasmadas las rutas dentro de los desplazamientos de los conductores de la empresa mencionada. Para el análisis de datos se utilizó la herramienta FreeOffice 2016.

Los resultados obtenidos de acuerdo a las pruebas de hipótesis que se desarrollaron tuvieron un efecto de mejora dentro de los indicadores como son: El tiempo en recorrido de desplazamiento por conductor hay una reducción del 34%, la distancia recorrida de desplazamiento por conductor se reduce al 8.6%, el número de incidencias evitadas por conductor aumenta a un 60,32% con la aplicación propuesta.

Por todo lo mencionado podemos concluir que la aplicación móvil de algoritmos de rutas óptimas tuvo un efecto de mejora el desplazamiento de los conductores de vehículos de la ciudad de Trujillo, 2018.

El proyecto estará conformado por seis partes fundamentales:

- Obtención y preparación de datos geoespaciales.
- Diseño y presentación de mapas
- Generación de rutas óptimas, a través de algoritmos de rutas óptimas.
- Gestor de datos de incidencias.
- Gestor de datos estadísticos del manejo de rutas óptimas.
- La aplicación móvil.

ABSTRACT

The present project is part of the field of engineering, in the field of "Analysis and Design of Algorithms", where through the processing of geospatial data and the use of optimal route algorithms will generate optimal alternate routes that will be provided in time real by means of a mobile application with the purpose of proposing better road routes to the drivers avoiding traffic jams due to incidents such as vehicular traffic, municipal works, traffic accidents and others, within the road network of the city of Trujillo.

The type of study is experimental with pre-test and post-test design. The sample consists of a total of 33 drivers who were randomly chosen in the city of Trujillo. To obtain the necessary data, a graphic map questionnaire was used, where the routes within the trips of the drivers of the mentioned company are shown. The FreeOffice 2016 tool was used to analyze the data.

The results obtained according to the hypothesis tests that were developed had an effect of improvement within the indicators, such as: The travel time per driver is a reduction of 34%, the distance traveled from travel per driver is reduced to 8.6%, the number of incidents avoided per driver increases to 60.32% with the proposed application.

For all the above we can conclude that the mobile application of optimal route algorithms had an effect of improving the displacement of drivers of vehicles in the city of Trujillo, 2018

The project will be made up of four fundamental parts:

- Obtaining and preparing geospatial data.
- Design and presentation of maps
- Generation of optimal routes, through optimal route algorithms.
- Incident data manager.
- Statistical data management of optimal routes management.
- The mobile application







CAPÍTULO 1. INTRODUCCIÓN

1.1. Realidad problemática

En la actualidad las ciudades se ven influenciados por el crecimiento y desarrollo de las tecnologías emergentes, dichos avances se han venido insertando en la mayoría de organizaciones e instituciones públicas y privadas, dando origen a grandes cambios a nivel estructural. En las grandes ciudades del mundo el crecimiento de la población desarrolló como consecuencia que crezcan demográficamente y por lo tanto también su infraestructura e infraestructura vial. (PUCP, 2015)

En el ámbito internacional las ciudades de EEUU el tránsito vehicular es un problema grave y a los conductores les costó casi 305 billones de dólares en el año 2017, un promedio de 1.445 de dólar por conductor gracias a la información obtenida de INRIX Global Traffic una aplicación que analiza y clasifica el impacto de la congestión del tráfico donde presentó 10 de las 25 ciudades más importantes del mundo con la peor congestión de tráfico. (INRIX Global Traffic Scorecard, 2017)

Tabla 1. *Tabla de Ranking de horas de Congestionamiento del año 2017.*

CIUDAD	2017 RANGOS DE TODAS LAS CIUDADES (2016)	2017 INRIX RANGOS DE PUNTUACION DE TRAFICO (2016)	HORAS PASADAS EN CONGESTIÓN	INDICE DE CONGESTION	PICO	DIURNO	GENERAL
 Los Angeles, CA	1 (1)	1 (1)	102	18.3	21%	11%	12%
 Moscow	2 (2)	2 (2)	91	20.1	32%	30%	26%
 New York, NY	3 (3)	3 (3)	91	17.4	19%	11%	13%
 Sao Paulo	4 (6)	4 (6)	86	16.9	30%	23%	22%
 San Francisco, CA	5 (4)	5 (4)	79	13.7	21%	10%	12%
 Bogotá	6 (5)	6 (5)	75	16.2	36%	30%	30%

Fuente: (INRIX Global Traffic Scorecard, 2017)

En Europa, Abertis líder mundial en la gestión de autopistas ha firmado un acuerdo mundial para adherirse al programa Connected Citizens de Waze, la aplicación de navegación social y tecnología móvil, el acuerdo de colaboración permitirá tanto a los usuarios de la aplicación Waze como a las concesionarias del Grupo Abertis suministrar y recibir información del tráfico a través de la aplicación y otros medios. (EUROPA PRESS, 2017)

En el ámbito nacional la demanda vehicular en el parque automotor entre los años 2008 – 2016 aumentó a un 60% según estudios del Instituto Nacional de Estadística e Informática (INEI), eso da como consecuencia unos de los factores más problemáticos que es el tránsito

vehicular en las ciudades más importantes del país, así mismo eso hace que genere un fastidio para los conductores ya que optan en tomar rutas alternas para llegar a su destino sin saber si es óptima y pueda llegar en menor tiempo posible. (INEI, 2017)

Tabla 2. Parque automotor en circulación a nivel nacional, según departamento, 2008 - 2016.

Departamento	2008	2009	2010	2011	2012	2013	2014	2015	2016
Total	1,640,970	1,732,834	1,849,690	1,979,865	2,137,837	2,295,809	2,453,781	2,611,753	2,769,725
Amazonas	2,218	2,292	2,390	2,407	2,400	2,393	2,386	2,379	2,372
Áncash	21,001	21,309	22,086	23,322	25,418	27,514	29,610	31,706	33,802
Apurímac	3,934	3,973	3,969	3,966	4,039	4,112	4,185	4,258	4,331
Arequipa	91,674	98,270	106,521	118,985	134,533	150,081	165,629	181,177	196,725
Ayacucho	5,404	5,572	5,716	5,784	5,941	6,098	6,255	6,412	6,569
Cajamarca	12,383	13,563	15,107	17,320	19,673	22,026	24,379	26,732	29,085
Cusco	39,688	42,175	45,090	48,491	53,675	58,859	64,043	69,227	74,411
Huancavelica	1,216	1,291	1,319	1,317	1,323	1,329	1,335	1,341	1,347
Huánuco	11,255	11,382	11,864	12,576	13,476	14,376	15,276	16,176	17,076
Ica	25,498	25,691	26,135	26,419	26,551	26,683	26,815	26,947	27,079
Junín	47,769	49,404	51,094	53,118	56,237	59,356	62,475	65,594	68,713
La Libertad	155,411	156,646	158,672	162,026	167,325	172,624	177,923	183,222	188,521
Lambayeque	41,920	43,689	45,881	49,440	53,902	58,364	62,826	67,288	71,750
Lima 1/	1,036,850	1,106,444	1,195,353	1,287,454	1,395,576	1,503,698	1,611,820	1,719,942	1,828,064
Loreto	5,132	5,089	5,089	5,211	5,313	5,415	5,517	5,619	5,721
Madre de Dios	913	941	986	1,027	1,062	1,097	1,132	1,167	1,202
Moquegua	12,202	12,692	13,348	14,003	14,608	15,213	15,818	16,423	17,028
Pasco	6,807	7,187	7,351	7,292	7,238	7,184	7,130	7,076	7,022
Piura	33,497	34,650	36,367	39,099	42,404	45,709	49,014	52,319	55,624
Puno	29,889	31,645	34,169	37,074	40,543	44,012	47,481	50,950	54,419
San Martín	9,917	9,977	10,151	10,418	10,926	11,434	11,942	12,450	12,958
Tacna	35,911	38,457	40,465	42,318	44,430	46,542	48,654	50,766	52,878
Tumbes	3,040	3,054	3,086	3,119	3,257	3,395	3,533	3,671	3,809
Ucayali	7,441	7,441	7,481	7,679	7,987	8,295	8,603	8,911	9,219

Fuente: Instituto Nacional de Estadísticas e Informática (INEI, 2017)

Actualmente la ciudad de Trujillo se encuentra en crecimiento, asimismo observamos vías de mayor congestión del tránsito en donde sus entornos son atractores de viaje por la concentración de la actividad comercial, zonas bancarias, instituciones educativas, entidades públicas, hospitales y otros, generando la molestia de muchos conductores al no encontrar una ruta óptima para llegar a su destino. (Transportes Metropolitanos de Trujillo, 2015)

Este problema se genera también por el alto flujo vehicular en las rutas, que sobrepasa la capacidad de la vía; según reportes en la actualidad se estima que la capacidad óptima de una vía urbana es considerada en 1,800 vehículos/hora/carril. Asimismo, el estacionamiento de vehículos en zonas no permitidas y proliferación de actividades ambulatorias reduce la capacidad de la vía. (Transportes Metropolitanos de Trujillo, 2015)

Como efecto del crecimiento del parque automotor para el año 2015 el flujo vehicular alcanzó la circulación de 139,713 vehículos/día, el 32% son vehículos menores entre moto taxis y motos lineales, seguido de 20% de autos, 8% de taxis y otros, que en total alcanzan una distancia recorrida 2,957 millones km, generando congestión vehicular, contaminación ambiental y accidentes de tránsito en varios puntos de la ciudad.

(Transportes Metropolitanos de Trujillo, 2015)

Ante este problema brevemente mencionado se plantea una solución a través del desarrollo de una aplicación móvil basada en algoritmos de rutas óptimas dentro de la región geográfica de la ciudad de Trujillo.

Tabla 3. *Tabla Parque automotor de Trujillo del año 2015*

Tipo de Vehículo	Número de Vehículos	Pct.	Distancia recorrida (km/año)	Recorrido Promedio (km/día)
Automóvil	27,463	20%	150,357,432.00	15.21
Station Wagon	8,046	6%	176,206,727.00	60.83
Camioneta	25,416	18%	648,140,442.00	70.84
Camión	12,256	9%	268,406,400.00	60.83
Remolcador	4,402	3%	96,403,800.00	60.83
Taxi	11,009	8%	400,839,096.00	101.14
Bus	3,289	2%	95,784,231.00	80.90
Microbús	1,214	1%	35,354,836.00	80.90
Camioneta rural	1,657	1%	52,282,724.00	87.65
Vehículos Menores	44,961	32%	1,033,878,195.00	63.88
Total	139,713	100%	2,957,653,883.00	

Fuente: (Transportes Metropolitanos de Trujillo, 2015)

1.2. Formulación del problema

¿Cómo afectará el desarrollo de una aplicación móvil basado en algoritmos de rutas óptimas en el desplazamiento de los conductores de vehículos privados en la ciudad de Trujillo, 2018?

1.3. Justificación

Con la elaboración de esta aplicación móvil-web pretende brindar una herramienta de ayuda para los conductores de vehículos en mejorar su desplazamiento, proporcionándoles rutas alternas óptimas e indirectamente el tránsito vehicular, esperando lograr una mejor distribución del flujo vehicular dentro de la malla vial.

El proyecto es de gran ayuda para toda la sociedad trujillana, pues la aplicación permitirá tener información actualizada del estado de tráfico en la ciudad y proveerá de rutas óptimas al conductor.

Esta investigación se realiza con el propósito de aportar al conocimiento sobre el uso de algoritmos de rutas óptimas y de datos geoespaciales teniendo en cuenta dos puntos geográficos (punto geográfico de partida y punto geográfico de destino) se podrá trazar rutas alternas óptimas en tiempo real, evadiendo algún incidente en la malla vehicular como accidentes de tráfico, obras municipales, tráfico vehicular (este último proporcionado por Transportes Metropolitano de Trujillo o usuarios de la aplicación) de esta forma se estaría contribuyendo con la mejora del tránsito vehicular.

El desarrollo de este proyecto tiene como resultado una propuesta para ser incorporado como conocimiento a las ciencias de la computación sirviendo para futuros estudiantes y autores permitiendo poner en práctica sobre el uso de algoritmos de búsquedas, grafos y conocimientos adquiridos durante la vida profesional, que quieran mejorar el diseño de este trabajo de investigación o tomarlo de guía para su uso en otras áreas de control.

El uso de este proyecto beneficia toda la ciudad de Trujillo, no solo a las personas que cuentan con un dispositivo móvil con sistema operativo Android al usar la aplicación, sino también a las personas que no lo posean ya que podrán vivir en un medio ambiente más sano ya que se reducirá la emisión de gases contaminantes por parte de los vehículos.

Así mismo este proyecto se justifica económicamente porque mejora los tiempos de desplazamiento por la ciudad teniendo como resultado una mayor productividad en cuanto el

tiempo no se pierde en congestionamientos de tráfico y permitirá un ahorro en el consumo de combustible.

1.4. Limitaciones

- Acceso a los datos en tiempo real del tráfico obtenidos por la institución Transportes Metropolitanos de Trujillo. Para contrarrestar se trabajará con las empresas de taxis que se mantienen informadas con las incidencias ocurridas dentro de la malla vial.
- Costo computacional de algoritmos en dispositivos móviles. Para contrarrestar se implementó un servicio web RESTful (Representational State Transfer Web Services) que básicamente es una arquitectura de comunicación cliente/servidor de mensajes HTTP que será necesario en este proyecto ya que el proceso del algoritmo estará alojado en un servidor para un mejor rendimiento.

1.5. Objetivos

1.5.1. Objetivo general

Determinar el efecto en el desplazamiento de los conductores de vehículos privados en la ciudad de Trujillo, mediante el desarrollo de una aplicación móvil basada en algoritmos de rutas óptimas

1.5.2. Objetivos específicos

- Determinar la distancia promedio recorrido en ruta óptima por el conductor.
- Determinar el tiempo promedio en recorrido en ruta óptima, por el conductor.
- Determinar el número de incidencias evitadas por el conductor.
- Determinar la eficiencia de los algoritmos de rutas óptimas de la aplicación móvil.
- Determinar la funcionalidad esperada de la aplicación móvil.

CAPÍTULO 2. MARCO TEÓRICO

2.1. Antecedentes

(Marchena Pejerrey D. J., 2015) **Desarrollo de un sistema para la optimización de rutas de trabajo utilizando el algoritmo de Dijkstra y Diagramas de Voronoi.** Esta tesis propuso la utilización del algoritmo de Dijkstra y Diagramas de Voronoi para obtener rutas óptimas de trabajo en diferentes caminos recorribles que existen y que hacen posible el acceso de un lugar a otro para una entidad privada prestadora de servicios EPSEL.

Los resultados que se obtuvieron fueron positivos al realizar la comparación de datos del pre análisis (forma tradicional del trabajo de los operarios) con los datos del post análisis (utilizando los datos obtenidos en los reportes del sistema), los cuales para los notificadores tienen una ganancia de trabajo de un 16% para el día viernes y 13% para el día sábado, los verificadores de un 13.78% para el día viernes y 12% para el día sábado de igual manera para los inspectores una ganancia de trabajo de un 18% para el día jueves y 24% para el día sábado.

Esta tesis es considerada como un antecedente ya que profundiza en el uso del algoritmo de Dijkstra en su versión mejorada y resalta el uso diagramas algoritmo de Voronoi para la sectorización de regiones de trabajo.

(Aguado Aranda A. Jiménez de Vega J., 2013) **Optimización de rutas de transporte.** Este proyecto propuso el desarrollo de una aplicación de escritorio que genera una ruta de transporte de pasajeros, determinando la mejor forma de realizar el recorrido. Utilizando algoritmos de Clustering para trazar rutas de recojo de pasajeros y algoritmos Genéticos para trazar rutas de transporte de pasajeros.

Los resultados que se obtuvieron demostraron, que tras el uso de los algoritmos de Clustering y Genéticos, la disminución del tiempo de la ruta ya que se optimiza el número y la posición de paradas que no suelen ser consideradas en la toma de decisiones.

Este artículo es considerado como un antecedente ya que considera el uso de algoritmos de Clustering para realizar trazos de rutas parciales óptimas.

(Guzmán Luna J. A. Arango Sánchez R. E. Jiménez Pinzón L. D., 2012) **Búsqueda de la ruta óptima mediante los algoritmos: genéticos y dijkstra utilizando mapas de visibilidad.** Este artículo propuso el uso de mapas de visibilidad para la generación de trayectorias entre dos puntos y una cantidad cualquiera de obstáculos. Plantea la comparación entre el algoritmo genético y el algoritmo de Dijkstra para encontrar la ruta óptima entre las

trayectorias generadas con el mapa de visibilidad con la finalidad de proveer una libre navegación que permitirá guiar el curso de un robot desde un punto inicial a un punto final.

Los resultados que se obtuvieron sobre los métodos de búsqueda arrojaron la misma ruta óptima, la diferencia entre ellos es que el algoritmo de Dijkstra tuvo un costo computacional mucho menor que el algoritmo genético.

Este artículo es considerado como un antecedente ya que plantea la comparativa entre algoritmos de búsqueda de rutas óptimas. Donde, a partir de la percepción del entorno, se puede optar por el uso de uno de ellos.

(Taquiá Valdivia J. A., 2013) **Optimización de rutas en una empresa de recojo de residuos sólidos en el distrito de los Olivos.** Esta tesis propuso implementar el método más adecuado de optimización de rutas para una empresa de transporte de residuos sólidos con el fin de mejorar su rentabilidad. La modelación de este problema se basa en el problema del agente viajero (TSP) donde un solo vehículo debe visitar a todos los puntos asignados y donde el punto final es el punto de inicio en una sola ruta y a costo mínimo sin tener en cuenta restricción alguna.

Los resultados que se obtuvieron demostraron que la optimización de rutas es fundamental y trae consigo ahorro, ganancias y otros beneficios.

Este artículo es considerado como un antecedente ya que hace uso de la representación del problema del agente viajero (TSP) para brindar una solución de rutas óptimas.

(Amoako, 2011) **Determination of shortest path using Dijkstra's**

Algorithm for Emergency Service in Kumasi Metropolis. El objetivo de esta tesis fue usar el algoritmo de Dijkstra para construir el árbol de alcance mínimo. Propone el desarrollo de un programa en Visual Basic para encontrar la ruta más corta desde cualquier punto hacia un destino teniendo en cuenta las formas de doble carril en la red de las carreteras dentro de la metrópolis de Kumasi – Ghana, disminuyendo el tiempo de llegada del servicio de emergencia debido a casos de accidente de tráfico.

Los resultados que se obtuvieron demostraron una reducción en la distancia real en comparación con la ruta normal. Esto demuestra la importancia de la optimización de los flujos de red de la ciudad.

Esta tesis es considerada como un antecedente ya que se centra en el uso del algoritmo Dijkstra para encontrar rutas cortas que proveen una mejora en el tránsito vehicular.

(Aguayo Moreno, 2008) **Desarrollo de un sistema de localización de rutas óptimas entre dos puntos geográficos.** Esta tesis propuso el desarrollo de una aplicación web que permitirá encontrar la ruta óptima entre dos puntos geográficos de un área del Distrito Metropolitano de Quito. Haciendo uso de varios algoritmos (Dijkstra, Floyd-Warshall, Bellman-Ford, A*) que permiten el cálculo de ruta óptima.

Los resultados que se obtuvieron demostraron una aceptación por parte de los usuarios, tanto por el tiempo de respuesta, información proporcionada y facilidad de uso de esta aplicación.

Esta tesis es considerada como un antecedente ya que se centra en el uso de los algoritmos rutas óptimas y a su vez proporcionan herramientas que proveen información geográfica, renderización de mapas y servicios GIS en entornos de internet.

2.2. Bases teóricas

2.2.1. Algoritmo

“Un algoritmo es una serie de pasos bien organizados que describe el proceso que se debe seguir, para dar solución a un problema específico” (Universidad Tecnológica del Perú, 2016).

Según (Flores Cueto & Tataje Salas, 2011) un algoritmo constituye una lista completa de los pasos y una descripción de los datos que son necesarios para resolver un determinado problema en el ámbito de un método de clase.

Informalmente un algoritmo es cualquier procedimiento computacional bien definido que toma algunos valores o conjunto de valores, como entradas y produce algún valor, o conjunto de valores como salida. Un algoritmo es entonces una secuencia de pasos computacionales para transformar la entrada en una salida. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

Formas de Representación

Un algoritmo puede ser representado mediante:

Pseudocódigo

“Un pseudocódigo permite expresar un algoritmo con palabras en castellano que son semejantes a las sentencias de un lenguaje de programación” (Flores Cueto J.J. y Bertolotti, 2006).

Diagrama de Flujo

Un diagrama de flujo permite ilustrar la secuencia de pasos de un algoritmo por medio de símbolos especializados y líneas de flujo. La combinación de símbolos especializados y líneas de flujo describe la lógica para la solución del problema (algoritmo). Entonces se puede afirmar que el Diagrama de Flujo es la representación gráfica del algoritmo. (Flores Cueto J.J. y Bertolloti, 2006)

Diagrama de Actividad

Los Diagramas de Actividades son parte del Lenguaje Unificado de Modelado (UML) y se utilizan comúnmente en dos formas:

- En el modelado de flujos de trabajo, haciendo hincapié en las actividades tal y como son vistas por los actores que colaboran con el sistema, esto es, modelando procesos de negocio.
- En el modelado de una operación, utilizando los diagramas de actividad como diagramas de flujo para mostrar detalles de un algoritmo, haciendo amplio uso de las condiciones de procesos concurrentes.

Los Diagramas de Actividad están compuestos por símbolos de propósito especial, tales como los símbolos de estado de acción, diamantes, pequeños círculos y barras de sincronización. Estos símbolos están conectados mediante flechas de transición, las cuales representan el flujo de la actividad. (Flores Cueto J.J. y Bertolloti, 2006).

2.2.2. Algoritmos de Rutas Cortas

En muchas áreas como el transporte, diseño topológico de redes de comunicación entre otras, existen problemas relacionados con la búsqueda de algoritmos de caminos o rutas cortas desde un punto de origen hacia otro. El problema del camino más corto tiene que ver con encontrar el menor costo (el coste mínimo) de una ruta o trayectoria desde un nodo de origen de un grafo ponderado a un nodo de destino en ese mismo grafo. (Marchena Pejerrey J. D., 2015).

2.2.2.1. Algoritmo DFS (Búsqueda en profundidad)

“La estrategia del algoritmo de búsqueda en profundidad es como su nombre lo indica buscar más profundamente en el grafo cuando sea posible” (Cormen, Leiserson, Rivest, & Stein, Introducción a Algoritmos, 2001).

“En la búsqueda en profundidad los bordes del vértice más recientemente descubierto v que aún tiene bordes inexplorados son explorados” (Cormen, Leiserson, Rivest, & Stein, Introducción a Algoritmos, 2001).

“Cuando todos los bordes de v han sido explorados, la búsqueda retrocede para explorar los bordes del vértice desde donde se descubrió v ” (Cormen, Leiserson, Rivest, & Stein, Introducción a Algoritmos, 2001).

Este proceso continua hasta que se hayan descubierto todos los vértices que son alcanzables desde el vértice fuente original.

Si quedan vértices no descubiertos entonces uno de ellos es seleccionado como una nueva fuente y la búsqueda se repiten desde esa fuente. El proceso entero es repetido hasta que todos los vértices sean descubiertos. (Cormen, Leiserson, Rivest, & Stein, Introducción a Algoritmos, 2001)

Figura 1. Algoritmo DFS, Pseudocódigo

```

DFS(grafo G)
  PARA CADA vértice  $u \in V[G]$  HACER
    estado[ $u$ ]  $\leftarrow$  NO_VISITADO
    padre[ $u$ ]  $\leftarrow$  NULO
  tiempo  $\leftarrow$  0
  PARA CADA vértice  $u \in V[G]$  HACER
    SI estado[ $u$ ] = NO_VISITADO ENTONCES
      DFS_Visitar( $u$ , tiempo)

DFS_Visitar(nodo  $u$ , int tiempo)
  estado[ $u$ ]  $\leftarrow$  VISITADO
  tiempo  $\leftarrow$  tiempo + 1
  d[ $u$ ]  $\leftarrow$  tiempo
  PARA CADA  $v \in$  Vecinos[ $u$ ] HACER
    SI estado[ $v$ ] = NO_VISITADO ENTONCES
      padre[ $v$ ]  $\leftarrow$   $u$ 
      DFS_Visitar( $v$ , tiempo)
  estado[ $u$ ]  $\leftarrow$  TERMINADO
  tiempo  $\leftarrow$  tiempo + 1
  f[ $u$ ]  $\leftarrow$  tiempo

```

Fuente: (Cormen, Leiserson, Rivest, & Stein, Introducción a Algoritmos, 2001)

2.2.2.2. Algoritmo BFS (Búsqueda en anchura)

La búsqueda en anchura es uno de los más simples algoritmos para buscar en un grafo y el prototipo de muchos otros importantes algoritmos. El algoritmo de árbol de mínima expansión de Prim y el algoritmo de caminos más cortos de Dijkstra usan ideas similares. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

Dado un grafo $G = (V, E)$ y un vértice fuente s , la búsqueda en profundidad sistemáticamente explora las aristas de G para descubrir cada vértice que es alcanzable desde s . Este calcula la distancia (número más pequeño de las aristas) desde s a cada vértice alcanzable. Este produce un primer árbol de amplitud con raíz s que contiene todos los vértices alcanzables. Para cada vértice v alcanzable desde s , el camino en el primer árbol de amplitud desde s a v corresponde a un camino corto desde s a v en G , que es, un camino conteniendo el número más pequeño de aristas. El algoritmo trabaja en grafos dirigidos y no dirigidos. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

Figura 2. Algoritmo BFS, Pseudocódigo

```

BFS( $G, s$ )
1  for each vertex  $u \in G.V - \{s\}$ 
2       $u.color = WHITE$ 
3       $u.d = \infty$ 
4       $u.\pi = NIL$ 
5   $s.color = GRAY$ 
6   $s.d = 0$ 
7   $s.\pi = NIL$ 
8   $Q = \emptyset$ 
9  ENQUEUE( $Q, s$ )
10 while  $Q \neq \emptyset$ 
11      $u = DEQUEUE(Q)$ 
12     for each  $v \in G.Adj[u]$ 
13         if  $v.color == WHITE$ 
14              $v.color = GRAY$ 
15              $v.d = u.d + 1$ 
16              $v.\pi = u$ 
17             ENQUEUE( $Q, v$ )
18      $u.color = BLACK$ 

```

Fuente: (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

2.2.2.3. Algoritmo Dijkstra

El algoritmo de Dijkstra soluciona el problema del camino más corto desde uno nodo fuente en un grafo ponderado y dirigido $G = (V, E)$ donde todos los pesos de las aristas son no negativas $w(u, v) \geq 0$. Con una buena implementación, el tiempo de ejecución del algoritmo de Dijkstra es menor que el tiempo de ejecución del algoritmo de Bellman-Ford. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

El algoritmo de Dijkstra mantiene un conjunto de vértices S cuyos pesos de camino más cortos desde la fuente s ya han sido determinados. El algoritmo repetidamente selecciona el vértice $u \in V - S$ con la estimación mínima del camino más corto, añade u a S , y relaja todo las aristas dejando u . (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

Figura 3. Algoritmo Dijkstra, Pseudocódigo

```
DIJKSTRA( $G, w, s$ )
1 INITIALIZE-SINGLE-SOURCE( $G, s$ )
2  $S = \emptyset$ 
3  $Q = G.V$ 
4 while  $Q \neq \emptyset$ 
5      $u = \text{EXTRACT-MIN}(Q)$ 
6      $S = S \cup \{u\}$ 
7     for each vertex  $v \in G.Adj[u]$ 
8         RELAX( $u, v, w$ )
```

Fuente: (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

El tiempo de ejecución del algoritmo de Dijkstra dependerá de la forma en la que se implemente la cola de prioridad. Consideremos el primer caso en el que mantenemos una cola de prioridad aprovechando que todos los vértices son numerados desde 1 al $|V|$. Simplemente se podría guardar las distancias a los vértices en un arreglo de tamaño $|V|$. Cada operación INSERT y DECREASE-KEY tomara un tiempo de $O(1)$, y cada operación EXTRACT-MIN tomara un tiempo de $O(V)$, para un total de $O(V^2)$ (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

El algoritmo podría mejorarse implementando una cola de prioridad con un MIN-HEAP binario. Cada operación EXTRACT-MIN tomara el tiempo de $O(\lg V)$. El tiempo para construir el MIN-HEAP binario es $O(V)$. Cada operación DECREASE-KEY toma un tiempo de $O(\lg V)$. El tiempo total de ejecución es $O(E \lg V)$ si todos los vértices son alcanzables desde la fuente. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

2.2.2.4. Algoritmo Floyd-Warshall

“El algoritmo de Floyd-Warshall usa programación dinámica para solucionar el problema de búsqueda de las rutas más cortas entre todos los pares de un grafo dirigido $G = (V, E)$ ” (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009).

“El algoritmo Floyd-Warshall, toma un tiempo de ejecución de $\Theta(V^3)$. Aristas ponderadas negativas pueden estar presentes pero asume que no existen ciclos de peso negativo” (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009).

El algoritmo de Floyd-Warshall considera vértices intermediarios de un camino corto, donde un vértice intermediario de un simple camino $p = \{v_1, v_2, \dots, v_i\}$ es cualquier vértice de p diferente a v_1 o v_i , es decir, cualquier vértice del conjunto $\{v_2, v_3, \dots, v_{i-1}\}$. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

El algoritmo de Floyd-Warshall se basa en la siguiente observación. Bajo la suposición que los vértices de G son $V = \{1, 2, \dots, n\}$ se considera un subconjunto $\{1, 2, \dots, k\}$ de vértices para algún k . Para algún par de vértices $i, j \in V$, se considera todos los caminos desde i a j cuyos vértices intermediarios son todos los trazados desde $\{1, 2, \dots, k\}$, y p sea un camino de peso mínimo de entre ellos. (Camino p es simple) El algoritmo explota una relación entre el camino p y el camino más corto desde i a j con todos los vértices intermediarios en el conjunto $\{1, 2, \dots, k-1\}$. La relación depende en sí o no k es un vértice intermediario del camino p . (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

- Si k no es un vértice intermedio del camino p , entonces todos los vértices intermedios del camino p están en el conjunto $\{1, 2, \dots, k-1\}$. Así, un camino corto desde el vértice i al vértice j con todos los vértices intermedios en el conjunto $\{1, 2, \dots, k-1\}$ es también un camino corto desde i a j con todos los vértices intermedios en el conjunto $\{1, 2, \dots, k\}$. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

Si k no es un vértice intermedio del camino p , entonces descomponemos p en $i \rightarrow p_1 \rightarrow k \rightarrow p_2 \rightarrow j$. p_1 es un camino corto desde i a k con todos los vértices intermedios en el conjunto $\{1, 2, \dots, k-1\}$. Similarmente, p_2 es un camino corto desde el vértice k al vértice j con todos los vértices intermedios $\{1, 2, \dots, k-1\}$. (Cormen, Leiserson, Rivest, & Stein, Introducción a Algoritmos, 2001)

Figura 4. Algoritmo Floyd-Warshall, Pseudocódigo.

```

FLOYD-WARSHALL( $W$ )
1   $n = W.rows$ 
2   $D^{(0)} = W$ 
3  for  $k = 1$  to  $n$ 
4      let  $D^{(k)} = (d_{ij}^{(k)})$  be a new  $n \times n$  matrix
5      for  $i = 1$  to  $n$ 
6          for  $j = 1$  to  $n$ 
7               $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$ 
8  return  $D^{(n)}$ 

```

Fuente: (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

“El tiempo de ejecución del algoritmo de Floyd-Warshall es determinado por el triple bucle **for**. Cada ejecución de la línea 7 toma un tiempo de $O(1)$, el algoritmo se ejecuta en un tiempo de $O(n^3)$ ” (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009).

2.2.2.5. Algoritmo Bellman-Ford

El algoritmo de Bellman-Ford soluciona el problema del camino más corto desde uno nodo fuente en el caso genera en el cual los pesos de las aristas pueden ser negativos. Dado un grafo ponderado y dirigido $G = (V, E)$ con fuente s y la función peso $w: E \rightarrow \mathbb{R}$, el algoritmo Bellman-Ford retorna un valor booleano indicando sí o no hay un ciclo con peso negativo que es alcanzable desde la fuente. Si hay un ciclo, el algoritmo indica que no existe solución. Si no hay un ciclo, el algoritmo produce el camino más corto y sus ponderaciones. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

El algoritmo relaja las aristas, progresivamente disminuye un estimado $v.d$ en el peso de un camino más corto desde la fuente s a cada vértice $v \in V$ hasta que logre el peso de la ruta más corta real $g(s, v)$. El algoritmo retorna TRUE si y solo si el grafo contiene ciclos no negativos que son alcanzables desde la fuente. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

Figura 5. Algoritmo Bellman-Ford, Pseudocódigo.

```

BELLMAN-FORD( $G, w, s$ )
1  INITIALIZE-SINGLE-SOURCE( $G, s$ )
2  for  $i = 1$  to  $|G.V| - 1$ 
3      for each edge  $(u, v) \in G.E$ 
4          RELAX( $u, v, w$ )
5  for each edge  $(u, v) \in G.E$ 
6      if  $v.d > u.d + w(u, v)$ 
7          return FALSE
8  return TRUE

```

Fuente: (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

El algoritmo de Bellman-Ford se ejecuta en un tiempo $O(VE)$, desde la inicialización en la línea 1 toma un tiempo de $\Theta(V)$ cada uno de los $|V| - 1$ pasan por las aristas en las líneas 2 – 4 toma un tiempo de $\Theta(E)$ y el bucle **for** de las líneas 5 – 7 toma un tiempo de $O(E)$. (Cormen, Leiserson, Rivest, & Stein, Introduction to Algorithms, 2009)

2.2.2.6. Algoritmo A*

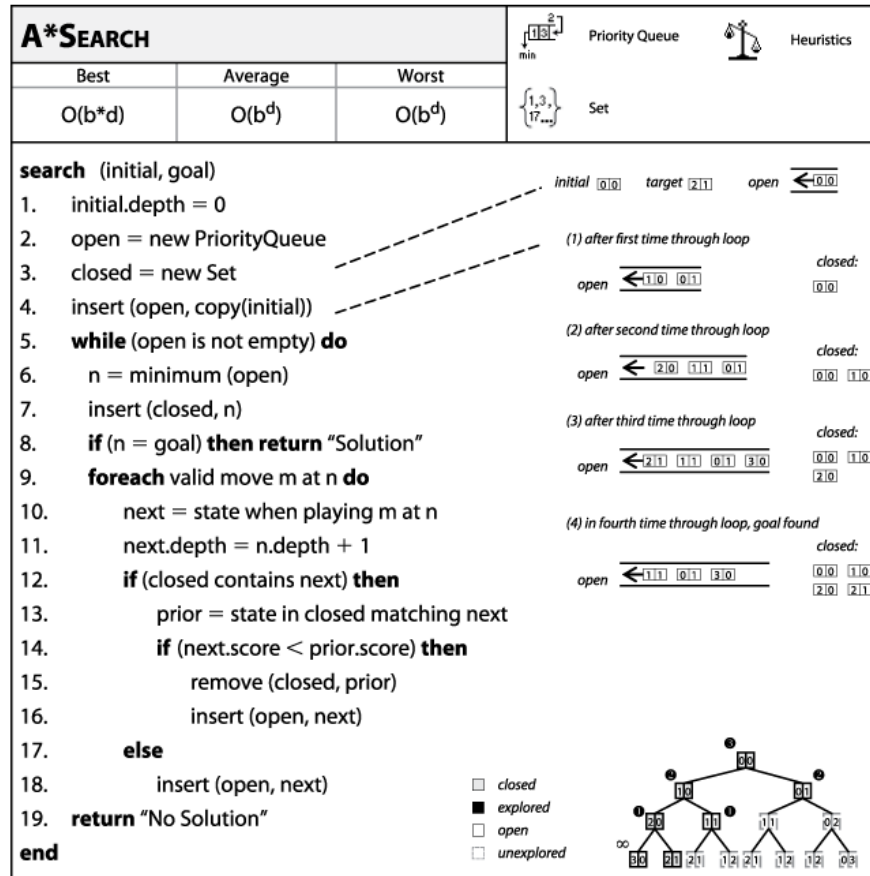
A* es una búsqueda iterativa, ordenada que mantiene un conjunto de estados abiertos para explorar en un intento de alcanzar el estado objetivo. En cada iteración, la búsqueda A* usa una función de evaluación $f^*(n)$ para seleccionar un estado n desde el estado abierto cuyo $f^*(n)$ tenga el valor más corto. $f^*(n)$ tiene la estructura distintiva $f^*(n) = g^*(n) + h^*(n)$ (Pollice, Selkow, & Heineman, 2008), donde:

“ $g^*(n)$ estima la secuencia de movimientos más corta desde el estado inicial a n ” (Pollice, Selkow, & Heineman, 2008).

“ $h^*(n)$ estima la secuencia de movimientos más corta desde n al estado objetivo” (Pollice, Selkow, & Heineman, 2008).

“ $f^*(n)$ estima la secuencia de movimientos más corta desde el estado inicial al estado objetivo a través de n ” (Pollice, Selkow, & Heineman, 2008).

Figura 6. Algoritmo A*, Pseudocódigo



Fuente: (Pollice, Selkow, & Heineman, 2008)

Tabla 4. Cuadro Comparativo entre los algoritmos de rutas cortas

Floyd-Warshall	Bellman – Ford	Dijkstra	Dijkstra Mejorado	Dijkstra A*
COMPLEJIDAD				
$O(n^3)$	$O(n^4)$	$O(n^2)$	$O(V^2) - O(E \lg V)$	$O(V + E)$
CARACTERÍSTICAS				
*Iterativo y de fácil implementación.	*Calcula pesos negativos.	*Cada paso del proceso encuentra ruta óptima.	*Tiene las mismas características del Dijkstra pero varía su complejidad debido a una diferente implementación en la cola de prioridad.	*Acelera el proceso de búsqueda entre 2 nodos (inicio-fin)
*No genera la ruta más corta entre 2 puntos.	*Su complejidad es alta.	*Bajo desempeño al manejar Grafos más grandes		*Bajo desempeño al manejar información geográfica.
*Calcula entre cada par de nodos, en caso de necesitar hallar el algoritmo solo para algún nodo de origen específico debe ejecutarse de nuevo para todos los nodos.				
*Se obtiene rutas al final del proceso del algoritmo.				

Fuente: (Eric Manuel Aguayo Moreno, 2008).

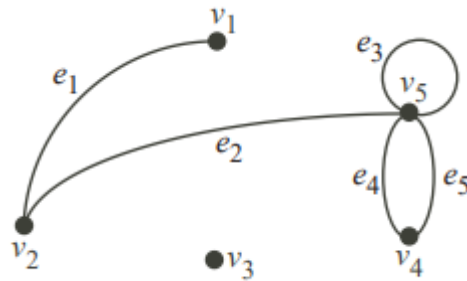
Para el presente proyecto se decidió usar el algoritmo de Dijkstra mejorado ya que tiene mejor rendimiento en costo computacional.

2.2.3. Grafo

Un grafo S consiste de un conjunto $N(S)$ no vacío de elementos llamados vértices (nodos o puntos) y un conjunto $M(S)$ de elementos llamados miembros (bordes, arcos o aristas) unidos con una relación de incidencia que asocia a cada par de nodos (no necesariamente distintos), llamados sus finales. (Godsil & Royle, 2001)

Un grafo G se define como un conjunto M de pares no ordenados de elementos distintos y otro conjunto de elementos N . El conjunto N es el conjunto de vértices (nodos) del grafo y se denota por $N(G)$. El conjunto M es el conjunto de aristas (miembros, arcos o bordes) del grafo y se denota por $M(G)$.

Figura 7. Representación de un Grafo.



Fuente: (Ruohonen, 2013)

Los grafos resultan ser extremadamente útiles para analizar problemas muy diversos (Gallardo, 2003)

Algunos ejemplos son: problemas de asignación de tareas, construcción de redes, problema de horarios, y otros.

Grado de un Grafo

“El grado de un grafo es determinado por el número de vértices o nodos. Grafos pueden ser finitos o infinitos de acuerdo a su orden” (Amoako, 2011).

Grado de un Vértice (Nodo)

Un nodo v es incidente con un enlace e si $v \in e$; entonces e es un enlace a v . Los dos nodos incidentes con un enlace son sus nodos finales. El conjunto de vecinos de un nodo v en G es denotado por $N(v)$. El grado $d(v)$ de un nodo v es el número $|E(v)|$ de enlaces incidentes en v . (Amoako, 2011)

Grafos Dirigidos

Un grafo dirigido o dígrafo D es un conjunto de nodos $N(D)$, un conjunto de miembros (bordes, arcos o vértices) $M(D)$, unidos con una relación que asocia un par de nodos ordenados con cada miembro. El primer nodo de un par ordenado es llamado el nodo inicio, y el segundo es conocido con el nodo final de un miembro dirigido. Un miembro es dirigido de su inicio a su final. (Godsil & Royle, 2001)

Según (Ruohonen, 2013) un grafo dirigido se denomina también un Dígrafo, el cual consta de un conjunto de vértices V y un conjunto de arcos E los cuales son pares ordenados de elementos de V .

En un grafo dirigido las flechas representan a los miembros (bordes, arcos o vértices), las cuales indican la dirección u orientación de los mismos.

$$V = \{x, y, v, w\}$$

$$E = \{(v, w), (w, y), (y, x), (x, v)\}$$

Ecuación 1. Grafo Dirigido - Conjuntos de Vértices "V" y Arcos "E"

Fuente: (Ruohonen, 2013)

Se tiene que tener en cuenta la diferencia entre un grafo y un dígrafo; en un grafo una arista es un par no ordenado de vértices, mientras que en un dígrafo un arco (arista dirigida u orientada) es un par ordenado de vértices.

Figura 8. Grafo Dirigido.



Fuente: (Ruohonen, 2013)

Grafos No Dirigidos

Según (Gusfield, 2014) "Es un objeto combinatorio que consta de un conjunto nodos V , y un conjunto de bordes E . Cada borde en E es especificado por un desordenado par de nodos (u, v) de V , donde u es diferente de v ."

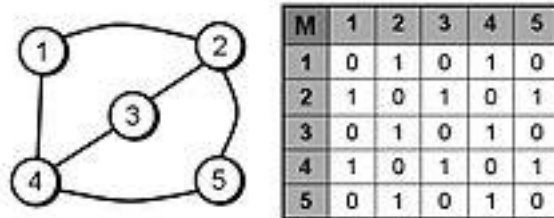
Figura 9. Grafo No Dirigido.



Fuente: (Gusfield, 2014)

Las dos representaciones principales de grafos son a través de Matrices de Adyacencia y

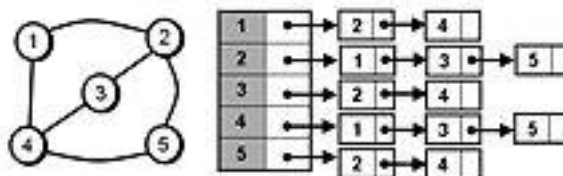
Figura 10. Matriz de Adyacencia.



Fuente: (Gusfield, 2014)

Lista de Adyacencias.

Figura 11. Lista de Adyacencia.



Fuente: (Gusfield, 2014)

Grafos Conexos

“Un grafo se dice que es conexo si cada par de sus vértices están conectados. Es decir G es conexo $\Leftrightarrow \forall u, v : \exists \mu = (u, v)$ En caso contrario, diremos que G es un grafo no conexo” (González Gutiérrez, 2004)

2.2.4. Tecnología Aplicada

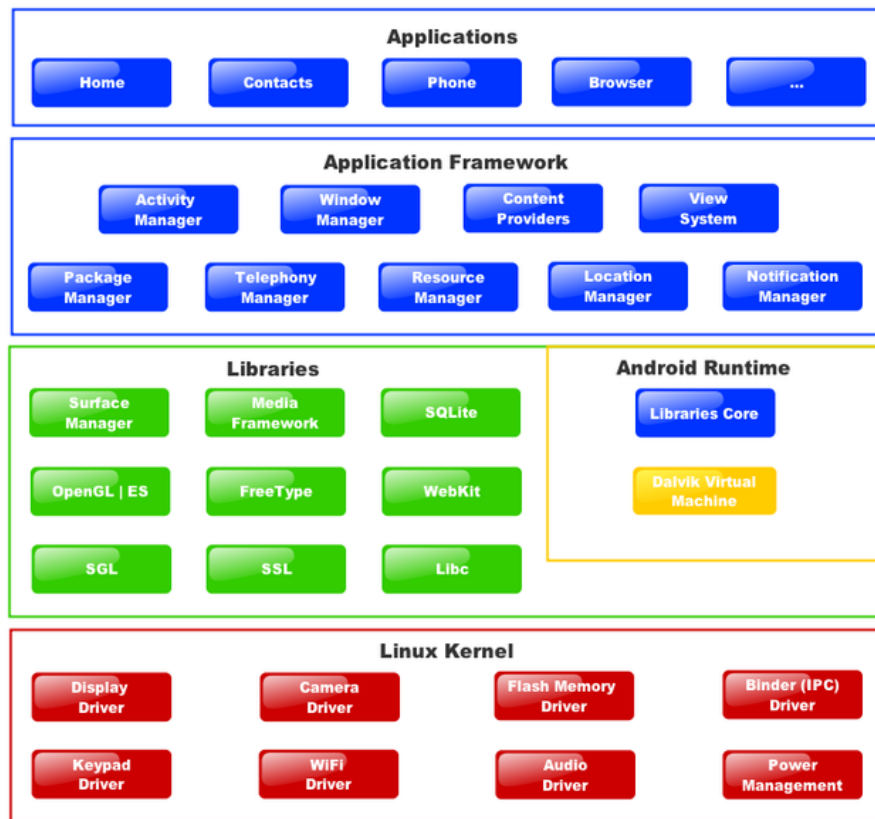
2.2.4.1. Android

Android es una pila de software de código abierto creado para una amplia gama de dispositivos con diferentes factores de forma. Los principales propósitos de Android son crear una plataforma de software abierta para carriers, fabricantes de equipos originales, y desarrolladores para hacer realidad sus ideas innovadoras y para introducir un producto real y exitoso que mejore la experiencia móvil de los usuarios. (Google Inc, The Android Source Code, 2017)

Android fue originado por un grupo de empresas conocida como la Open Handset Alliance, liderada por google. Hoy en día muchas de estas empresas, tanto miembros originales de OHA y otros, han invertido mucho en Android. Estas empresas han asignado importantes

recursos de ingeniería para mejorar Android y llevar dispositivos Android al mercado. (Google Inc, The Android Source Code, 2017)

Figura 12. *Arquitectura del Sistema Operativo Android*



Fuente: (Google Inc, The Android Source Code, 2017)

Algunas de sus características más importantes son listadas a continuación:

- Es de código abierto.
- Cualquiera puede personalizar la plataforma Android.
- Existen muchas aplicaciones disponibles para este sistema operativo.
- Provee interesantes características como detalles del clima, RSS, Servicio de mensajería (SMS y MMS), navegador web, almacenamiento (SQLite), conectividad (GSM, CDMA, Bluetooth, WI-FI), multimedia, y otras más.

“El proyecto de Android de código abierto usa unas cuantas iniciativas de código abierto aprobando licencias de código abierto para nuestro software” (Google Inc, Contenido de Licencia, 2017).

“La licencia preferida para el proyecto Android de código abierto es Apache Software License, Versión 2.0 (Apache 2.0) y la mayoría del software de Android es licenciado con Apache 2.0” (Google Inc, Contenido de Licencia, 2017).

Tabla 5. Cuadro Comparativo entre Android vs IOS.

ANDROID	IOS
FACILIDAD DE USO	
Es más intuitivo.	Es Complejo.
DISPONIBILIDAD DE DESCARGA DE APP'S	
Tiene disponibilidad de descarga en Google Play y en otras plataformas F-Droid, Amazon Underground, etc.	No tiene disponibilidad de descarga, solo si instalas JailBreak (No recomendado).
PERSONALIZACIÓN	
Permite personalizar en todos los aspectos desde iconos, tamaños del dock, animaciones y velocidad, etc.	En este aspecto es muy cerrado, lo único que puede variar es por el color de la carcasa y el fondo de pantalla.

Fuente: Propia.

Para el presente proyecto se decidió usar Android por su facilidad de uso en el sistema operativo y tener mayor demanda en el mercado móvil.

2.2.4.2. Java

Java fue concebido por James Gosling, Patrick Naughton, Chris Warth, Ed Frank, y Mike Seridan en Sun Microsystem en 1991. Este lenguaje fue inicialmente llamado Oak pero fue renombrado a Java en 1995” (Schildt, 2017).

El ímpetu de Java no fue el internet, en lugar, la motivación principal fue la necesidad de un lenguaje independiente de la plataforma que pueda ser usado para crear software para ser embebido en varios dispositivos electrónicos de clientes, tales como tostadoras, hornos microondas, y control remotos. (Schildt, 2017)

En 1993 llegó a ser obvio para los miembros del equipo de diseño de Java que los problemas de portabilidad frecuentemente encontrados cuando se crea código para controladores embebidos serían también encontrados cuando se intenta crear código para el Internet. Esta realización causo que el enfoque de Java cambiase de electrodomésticos de cliente a programación de Internet. (Schildt, 2017)

“Java está directamente relacionado con C y C++. Java hereda su sintaxis de C. Su modelo de objeto es adaptado de C++” (Schildt, 2017).

“Muchos programadores están familiarizados con la sintaxis de C/C++. Esto hace fácil para los programadores de C/C++ aprender Java, y a la inversa, programadores Java aprender C/C++” (Schildt, 2017).

“Los diseñadores de Java no “reinventaron la rueda”. En lugar, ellos refinaron aún más un paradigma de programación altamente exitoso. La edad moderna de programación comenzó con C. Se movió a C++, y ahora a Java” (Schildt, 2017).

“Programación orientada a objetos es el core de Java. De hecho, todos los programas Java son en cierta medida orientados a objetos” (Schildt, 2017).

Java pretende que las aplicaciones de los desarrolladores sean escriba una sola vez y puedan correr sobre cualquier plataforma lo que ofrece portabilidad en una red.

Java fue desarrollado teniendo en cuenta algunos principios claves en mente:

- Fácil de usar

Lo fundamental de Java viene del lenguaje de programación C++. Aunque C++ es un lenguaje potente, es complejo en sintaxis e inadecuado de los requerimientos de Java. Java mejoro la idea de C++ proporcionando un lenguaje de programación que es potente y simple de usar. (Rouse, Java, 2016)

- Confiabilidad

Java necesito reducir la probabilidad de errores fatales de las equivocaciones de los desarrolladores. Con esto en mente se introdujo la programación orientada a objetos. Cuando los datos y su manipulación se empaquetaron juntos en un solo lugar, Java fue robusto. (Rouse, Java, 2016)

- Security

“Java originalmente estaba dirigido a dispositivos móviles que estarían intercambiando datos a través de redes, fue construido para incluir un alto nivel de

seguridad. Java es probablemente el lenguaje de programación más seguro hasta la fecha” (Rouse, Java, 2016).

- Independencia a la plataforma

Los programas deben funcionar independientemente de las máquinas en las que se está ejecutando. Java se escribió para ser un lenguaje portátil que no se preocupa por el sistema operativo o el hardware de la computadora en el que se está ejecutando. (Rouse, Java, 2016)

Existen tres plataformas principales en las cuales los programadores pueden desarrollar aplicaciones Java:

- Java SE (Java Standart Edition)

“Formalmente se le conoce como J2SE, Java SE provee todas las APIs necesarias para el desarrollo tradicional de aplicaciones para escritorio” (Rouse, Java, 2016).

- Java EE (Java Enterprise Edition)

“Formalmente conocido como J2EE, provee la habilidad para crear componentes sever-side que puedan responder a un ciclo de solicitud-respuesta basado en la web” (Rouse, Java, 2016).

- Java ME (Java Micro Edition)

“Formalmente conocido como J2ME. Java ME ofrece un entorno flexible y sólido para aplicaciones que se ejecutan en dispositivos móviles e integrados: teléfonos móviles, TDT, reproductores Blu-ray, dispositivos multimedia digitales, módulos M2M, impresoras y mucho más” (Oracle, ¿Qué es J2ME o Java ME?, 2017).

Java se encuentra bajo licencia GPL lo que permite a la comunidad de programadores/desarrolladores de aplicaciones Java distribuirlas libremente.

Tabla 6. Cuadro Comparativo entre C# vs Java

C#	Java
CREACIÓN	
Creado como parte de Microsoft .NET inicialmente, diseñado para una infraestructura de lenguaje común.	Creado por Sun Microsystems, diseñado para permitir a los desarrolladores “escribir una vez, ejecutar en cualquier lugar”.
ECOSISTEMA	
Usado para desarrollar en plataformas de Microsoft.	Tiene un enorme ecosistema de código abierto.
SOPORTE	
Integrado en la CLI (command-line) y permite que el tipo de la información esté disponible en tiempo de ejecución.	Requiere el uso de un interfaz para lograr un funcionamiento similar.
EXCEPCIONES VERIFICADAS	
Solo tiene un tipo de excepción.	Distingue entre excepciones verificadas y sin verificar.
POLIMORFISMO	
Invoca la palabra clave "virtual" en una clase base y la palabra clave "sobrescribir" en una clase derivada.	Habilita el polimorfismo por defecto.
ENUMERACIONES	
Listas simples de constantes nombradas donde el tipo debe ser integral.	Instancia nombrada de un tipo, lo que facilita agregar comportamiento personalizado.
CUANDO USAR	
En .NET Framework, Mono y Portable.NET	En Unix, Linux u otras plataformas fuera de Microsoft.

Fuente: Propia.

La elección de Java se dio principalmente por ser multiplataforma, flexible en el código, es abierto, expandible, etc.

2.2.4.3. PostgreSQL

PostgreSQL es un potente sistema gestor de base de datos de objetos relacionados que usa y extiende en lenguaje combinado SQL con muchas características que seguramente guarda y escala las cargas de trabajo de datos más complicados.

PostgreSQL se ha ganado una sólida reputación por su arquitectura comprobada, confiabilidad, integridad de datos, sólidos conjunto de características, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para entregar

constantemente soluciones eficaces e innovadoras. PostgreSQL puede ser ejecutado en todos los principales sistemas operativos y tiene complementos potentes, como el popular extensor de base de datos geoespaciales PostGIS. No sorprende que PostgreSQL se haya convertido en la base de datos relacional de código abierto de elección para muchas personas y organizaciones. (The PostgreSQL Global Development Group, 2018)

PostgreSQL se lanza bajo la licencia PostgreSQL, una licencia liberal de código abierto, similar a las licencias BSD o MIT. (The PostgreSQL Global Development Group, 2018)

2.2.4.3.1 PostGIS

PostGIS es un extensor de base de datos espacial para la base de datos de objetivos relacionales PostgreSQL. Agrega soporte para los objetos geográficos que permiten que las consultas se ejecuten en SQL.

Además del conocimiento de la ubicación básica, PostGIS ofrece muchas características que raramente se encuentran en otras bases de datos espaciales competidoras como Oracle Locator, MySQL Spatial y SQL Server. (PostGIS Project Steering Committee, 2018)

Tabla 7. Cuadro Comparativo entre PostgreSQL vs MySQL.

PostgreSQL	MySQL
CODIGO ABIERTO	
Completamente de código abierto.	De código abierto, pero propiedad de Oracle y ofrece versiones comerciales.
CUMPLIMIENTO DE ACID (atomicidad, consistencia, aislamiento, durabilidad)	
Cumple completamente con ACID.	Algunas versiones cumplen.
CUMPLIMIENTO DE SQL	
Cumple casi completamente.	Algunas versiones cumplen.
SOPORTE PARA CONCURRENCIA	
Implementación de MVCC (control de concurrencia multiversión) soporte múltiples peticiones sin bloquear lecturas.	Soporte en algunas versiones.

Tabla 7. Continuación.

SEGURIDAD	
Seguro desde las bases con soporte SSL.	Soporte SSL en algunas versiones.
SOPORTE PARA NoSQL/JSON	
Varias funcionalidades soportadas.	Solo soporte para datos JSON.
METODOS DE ACCESO	
Soporta todos los estándares.	Soporta todos los estándares.
REPLICACIÓN	
Múltiples tecnologías de replicación disponibles:	Replicación estándar maestro-standby:
<ul style="list-style-type: none"> • Un solo maestro a un standby • Un solo maestro a varios standbys • Replicación de flujo/hot standby • Replicación bidireccional • Replicación lógica por flujo de logs 	<ul style="list-style-type: none"> • Un solo maestro a un standby • Un solo maestro a varios standbys • Un solo maestro a un standby a uno o más standbys • Replicación circular (A hacia B hacia C y de vuelta hacia A) • Maestro a maestro
VISTAS MATERIALIZADAS	
Soportado.	No Soportado.
TABLAS TEMPORALES	
Soportado.	Soportado.
DATOS GEOESPACIALES	
Soportado.	Soportado.
LENGUAJES DE PROGRAMACIÓN	
Soportado.	No soportado.
SISTEMA DE TIPOS EXTENSIBLE	
Soportado.	No soportado.

Fuente: Propia.

La elección de la herramienta PostgreSQL se dio principalmente porque tiene mayor soporte en cuestión de rendimiento y datos geoespaciales que son de mayor importancia para la elaboración de nuestro proyecto.

2.2.4.4. Android Studio

“Android Studio es el entorno integrado de desarrollo (IDE) oficial para el desarrollo de aplicaciones Android, basado en IntelliJ IDEA” (Google Inc, Conóce Android Studio, 2017).

Es el reemplazo para ADT (Eclipse Android Development Tools) como principal IDE para el desarrollo de aplicaciones nativas Android.

Además del potente editor de código y herramientas de desarrollo de IntelliJ IDEA, Android Studio ofrece más características que mejoran la productividad cuando se construyen aplicaciones Android, como son:

- Un sistema de compilación basado en Gradle.
- Un emulador rápido y lleno de funciones.
- Un entorno unificado en el que se puede desarrollar aplicaciones para todos los dispositivos Android.
- Ejecución instantánea para introducir cambios en la aplicación en ejecución sin crear un nuevo APK.
- Amplias herramientas y frameworks de pruebas.
- Compatibilidad integrada con Google Cloud Platform.
- Disponible para Windows, Mac, Linux

“Al ser Android Studio basado en IntelliJ IDEA edición community es completamente libre y de código abierto, licenciado bajo licencia Apache 2 y puede ser usado para cualquier tipo de desarrollo” (Jemerov, 2013).

2.2.4.5. Eclipse

La mayoría de la gente conoce Eclipse como un entorno de desarrollo integrado para Java. Eclipse es creado por una comunidad de código abierto y es usada en muchas diferentes áreas, como por ejemplo: IDE para Java o para Android o como una plataforma de aplicaciones Eclipse RCP. (Vogel, 2013)

“Es posible usar Eclipse como una base para crear aplicaciones de propósito general. Estas aplicaciones son conocidas como Eclipse Rich Client Platform (Eclipse RCP)” (Vogel, 2013)

Eclipse es uno de los IDE's Java más usados en el planeta. Con características claves como autocompletar, pruebas unitarias integradas, y más, Eclipse empaca un arsenal potente de herramientas para el desarrollo de Java. Eclipse es una plataforma para herramientas, un punto de integración que provee características y funcionalidades en una manera consistente y poderosa. (Vogel, 2013)

“Eclipse puede ser extendido para ser lo que se necesite, ya sea, agregando plug-ins via el Marketplace de Eclipse, o creándolos” (Vogel, 2013).

“Eclipse es más que una simple tecnología. La fundación Eclipse existe para fomentar comunidades de usuarios, adoptadores, y desarrolladores; y un ecosistema de individuales, organizaciones, y compañías que hacen una vida trabajando con software de código abierto” (Vogel, 2013).

Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones.

Eclipse Public License (EPL) es una licencia de software de código abierto usado por la fundación Eclipse para su software. La licencia EPL es diseñada para ser favorable a los negocios. Programas con licencia EPL pueden ser usados, modificados, copiados y distribuidos libre de cargos. El consumidor de software con licencia EPL puede elegir usar este software en programas de código cerrado. (Vogel, 2013)

Tabla 8. Cuadro Comparativo entre Eclipse vs Netbeans vs Visual Studio.

Eclipse	Netbeans	Visual Studio
Tamaño de 150 MB.	Tamaño de 79.3 MB	Tamaño de 2.5 GB
Licencia libre	Licencia libre	
Plataforma ligera para componentes de software	Menor consumo de memoria	Mayor consumo de memoria
Es fácil de usar, ligero y estable	Es inestable lento y pesado además de que el código no se puede manipular al 100%	Es fácil de usar y lento al momento de compilar.
Ambos son sistemas multiplataforma para 32 y 64 bits en Windows, Linux, Mac OS.		Multiplataforma 32 y 64 bits para Windows.
Más rápido	Más pesado	Más pesado.
Más flexible	Herramienta Swing estándar	Más flexible
Mejor soporte de desarrollo Android	SVC más intuitivo GIT	
Más plugins	GUI más intuitivo GIT	

Fuente: Propia.

Tabla 9. Cuadro Comparativo entre Android Studio vs Eclipse.

Android Studio	Eclipse
Android Nativo	Se tiene que instalar SDK
Actualizaciones constantes	Actualizaciones menos frecuentes
Rendimiento óptimo	Rendimiento bajo
Más rápido	Más pesado
Más intuitivo y fácil de usar	Un poco compleja
Código más ordenado, estructurado y mejores sugerencias.	Código nada ordenado y estructurado.
Recomendado para diseñar interfaces	Diseño de interfaces complejas
Creación de varios emuladores, no consume demasiados recursos	Consume demasiado recursos

Fuente: Propia.

La elección de la herramienta en el entorno web se eligió Eclipse ya que tiene una plataforma ligera para cualquier componente software y es fácil de usar.

La elección de la herramienta en el entorno móvil se eligió Android Studio ya que es un IDE de código nativo para su propio lenguaje de programación Android y está preparado para ser el más óptimo en la elaboración de proyectos móviles.

2.2.4.6. pgAdmin

La herramienta pgAdmin es la plataforma de administración de desarrollo de código abierto más popular y rica en características para PostgreSQL, la base de datos de código abierto más avanzada de todo el mundo.

La herramienta pgAdmin puede ser usado en los sistemas operativos Linux, Unix, Mac OS X y Windows para gestionar PostgreSQL 9.2 y superior. (The pgAdmin Development Team, 2018)

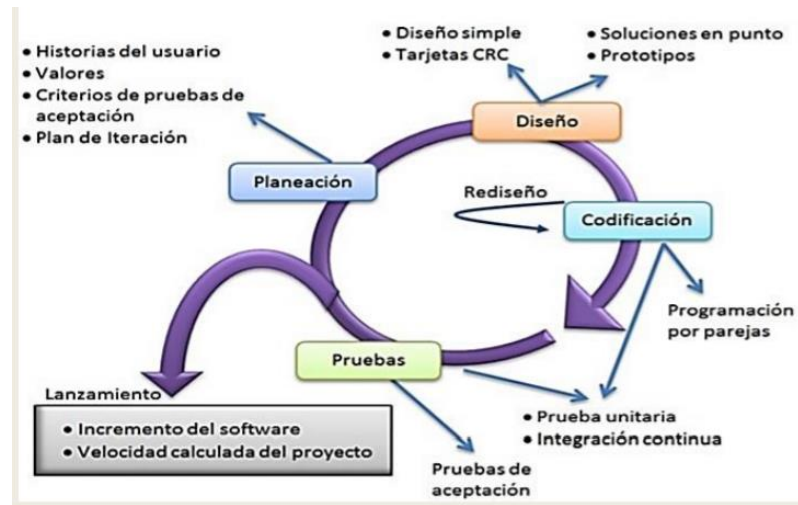
2.2.5. Metodologías de Desarrollo

En el desarrollo de software existen varias metodologías que nos ayudan a construir un software de calidad los más usados son XP, ICONIX y RUP. En este proyecto se hará una breve comparativa de estas metodologías.

Extreme Programing (XP)

Es una disciplina de desarrollo de software basada en los valores de simplicidad, comunicación, retroalimentación y valor. En XP cada participante del proyecto es una parte integral del equipo. El equipo se forma alrededor de un representante llamado el cliente, que se sienta con el equipo y trabaja con ellos diariamente. Los equipos de XP usan una forma simple de planificación y seguimiento para decidir que se debe hacer a continuación y para predecir cuando el proyecto será finalizado. Focalizado en el valor del negocio, el equipo produce software en una serie de pequeños entregables integrados, que aprueban todos los tests que ha definido el cliente. Los programadores de XP trabajan juntos en pares y como un grupo, con un código testeado de forma obsesiva y de diseño simple, mejorando el diseño continuamente para mantenerlo siempre acorde a las necesidades actuales. (Pérez Ramírez, D., Sepúlveda, J. C., & I. Oliveros, Y, 2011)

Figura 13. Metodología y Fases de XP.



Fuente: (Pérez Ramírez, D., Sepúlveda, J. C., & I. Oliveros, Y, 2011)

Rational Unified Process (RUP)

Es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

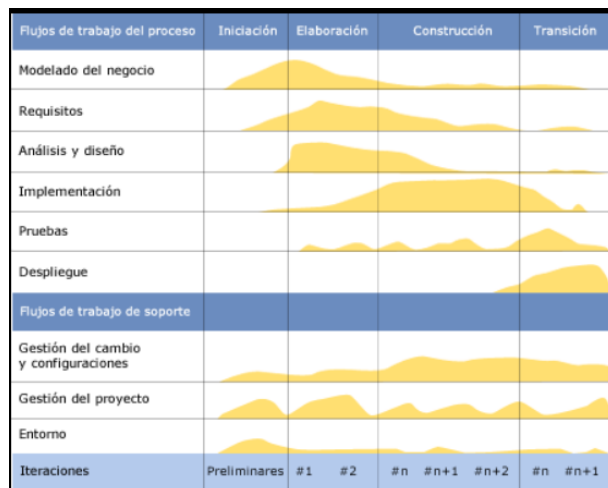
RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. También se conoce por este nombre de software, también desarrollado por Rational, que incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Esta metodología comprende 3 principios claves: Dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

Dirigido por los casos de uso, significa que los requerimientos están enfocados a dar valor al cliente y que el proceso debe garantizar que todo el desarrollo, pruebas, planificación, documentación, etc., está orientado a cubrir estas expectativas del cliente y asegurar que los requerimientos de valor se ponen en producción.

Centrado en arquitectura, significa que hay un énfasis a diseñar una arquitectura de calidad, y es la arquitectura también la que guía la forma como se debe planear y hacer el desarrollo.

Iterativo e incremental, significa que el proyecto se divide en varios ciclos de vida (Iteraciones) que deben dar como resultado un ejecutable. Por cada una de las iteraciones se va agregando requerimientos y sobre todo valor al cliente; por este motivo es incremental. (Darryl K. Taft, 2002)

Figura 14. Ciclo de vida de la metodología RUP.



Fuente: (Darryl K. Taft, 2002)

ICONIX (RUP)

Es una metodología pesada-ligera de Desarrollo de Software que se halla a medio camino entre RUP (Rational Unified Process) y un XP (eXtreme Programing).

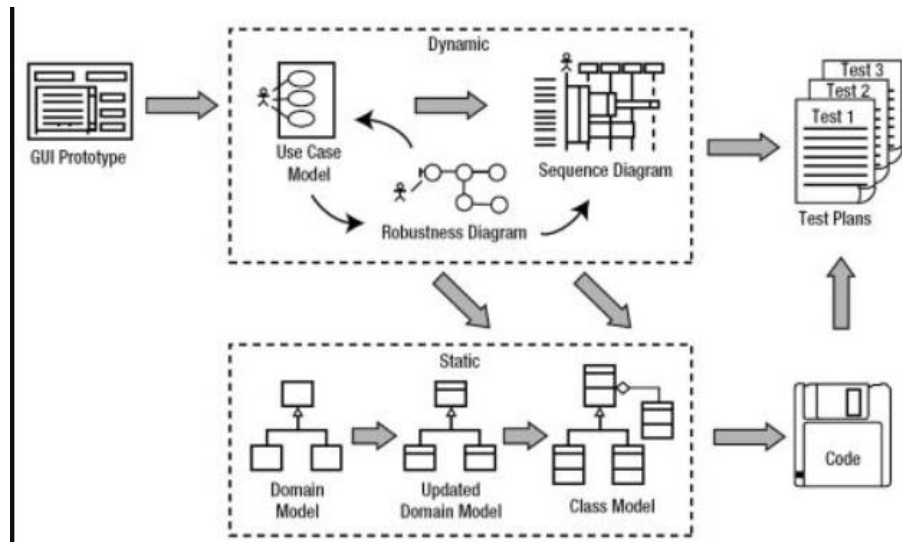
ICONIX deriva directamente del RUP y su fundamento es el hecho de que un 80% de los casos pueden ser resueltos tan solo con un uso del 20% del UML, con lo cual se simplifica muchísimo el proceso sin perder documentación al dejar solo aquello que es necesario. Esto implica un uso dinámico del UML de tal forma que siempre se pueden utilizar otros diagramas además de los ya estipulados si se cree conveniente.

En cuestión de ciclo de vida presenta 2 enfoques "Iterativo" e "Incremental":

El ciclo de vida Incremental, consiste en desarrollar por partes el producto de manera que se pueda integrar funcionalmente.

El ciclo de vida Iterativo, consiste que en cada ciclo de iteración se revisa y mejora el producto.

Figura 15. Flujo de Proceso ICONIX.



Fuente: (Matt Stephens, 2006)

De acuerdo a las metodologías que se mencionaron anteriormente se procedió en hacer una comparativa entre ellas:

Tabla 10. Cuadro Comparativo de las Metodologías de desarrollo.

XP	ICONIX	RUP
CARACTERÍSTICAS		
<ul style="list-style-type: none"> *Desarrollo Iterativo e incremental. *Pruebas Unitarias continuas, frecuentemente repetidas y automatizadas. *Programación por parejas. *Interacción con el usuario final. *Refactorización del código. *Propiedad del código. *Simplicidad del código. 	<ul style="list-style-type: none"> *Desarrollo Iterativo e Incremental. *Trazabilidad entre los artefactos de software. *Dinámica del UML, uso de algunos diagramas. 	<ul style="list-style-type: none"> *Desarrollo interno en etapas interactivas. *Está integrado en todo el ciclo de vida. *Programación por equipos. *Interacción con el usuario estratégico.

Tabla 10. Continuación.

VENTAJAS		
*Comunicación.	*Proceso ágil para obtener un sistema informático.	*Mayor documentación.
*Simplicidad.		*Verificar la calidad de software.
*Realimentación.	*Dedicada a la construcción de sistema de gestión de pequeña y media complejidad con la participación de los usuarios finales.	*Configuración y control de cambios.
*Coraje (Satisfacción de los programadores).		*Es modelado, guiado por casos de uso.
*Disminuye traza de errores.		*Es centrado en arquitectura, guiado por riesgos.
*Alta calidad mínimo de tiempo.		
DESVENTAJAS		
*Dificultad para determinar el costo de proyecto.	*Información rápida y puntual de los requisitos, del diseño y de las estimaciones.	*Los cambios son en una fase
*Se usa principalmente en proyectos pequeños.	*Se usa en proyectos de poca duración.	*Proyectos grandes

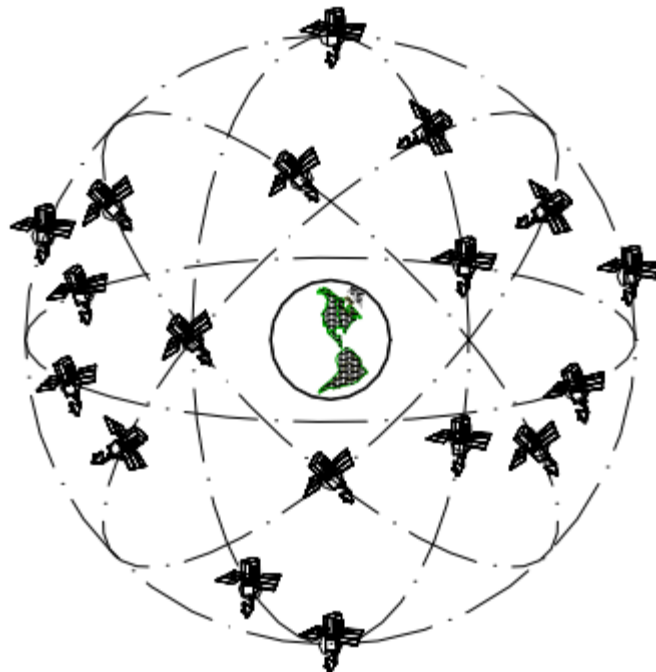
Fuente: Propia.

Para el presente proyecto se decidió usar la metodología ICONIX por su simplicidad y proceso ágil en la obtención de requerimientos del usuario final, cabe resaltar que la participación del usuario será continua durante el ciclo de vida.

2.2.6. Sistema de Posicionamiento Global (GPS)

El sistema de posicionamiento global (GPS), es un sistema mundial de navegación desarrollado por el departamento de defensa de los Estados Unidos. Actualmente este sistema consta de 24 satélites artificiales (21 regulares más 3 de respaldo) y sus respectivas estaciones en tierra, proporcionando información para el posicionamiento las 24 horas del día sin importar las condiciones del tiempo. (Casanova Matera, 2002)

Figura 16. GPS



Fuente: (Casanova Matera, 2002)

“Los satélites artificiales son utilizados por el G.P.S, como punto de referencia para el cálculo de posiciones de puntos sobre la superficie de la tierra con precisiones cada día mejores” (Casanova Matera, 2002).

“Desde sus inicios puramente militares en el año 1978, sus aplicaciones han ido incrementándose constantemente en diversas áreas y los equipos receptores de G.P.S han ido disminuyendo tanto en tamaño como en costo” (Casanova Matera, 2002).

Errores del Sistema de Posicionamiento Global

Según (Casanova Matera, 2002), los errores que afectan las mediciones con G.P.S. se pueden agrupar en tres tipos diferentes:

1. Errores propios del satélite

- **Errores orbitales** o de efemérides, que afectan la determinación de la posición del satélite en un instante determinado con respecto a un sistema de referencia dado. Estos errores se originan debido a que no se conocen con la exactitud necesaria las órbitas de los satélites.
- **Errores del reloj.** Aunque sumamente precisos, los relojes atómicos pueden presentar variaciones debido a la deriva propia del instrumento y a la acción de los efectos relativísticos que originan un diferencial del tiempo entre el sistema del satélite y del sistema del G.P.S. Este diferencial de tiempo no es constante para todos los satélites, sin embargo, estos errores, de muy poca magnitud, son ajustados por el Departamento de Defensa de los Estados Unidos.
- **Errores de configuración geométrica.** El efecto de la geometría en la ubicación de los satélites utilizados en la determinación de un posicionamiento queda expresado por los parámetros de la dilación de precisión geométrica (DPG).

Los parámetros de la DPG resultan en una medida compuesta que refleja la influencia de la geometría de los satélites sobre la precisión combinada de las estimaciones del tiempo y posición de la estación.

Los cuatro parámetros de la dilación de precisión geométrica son:

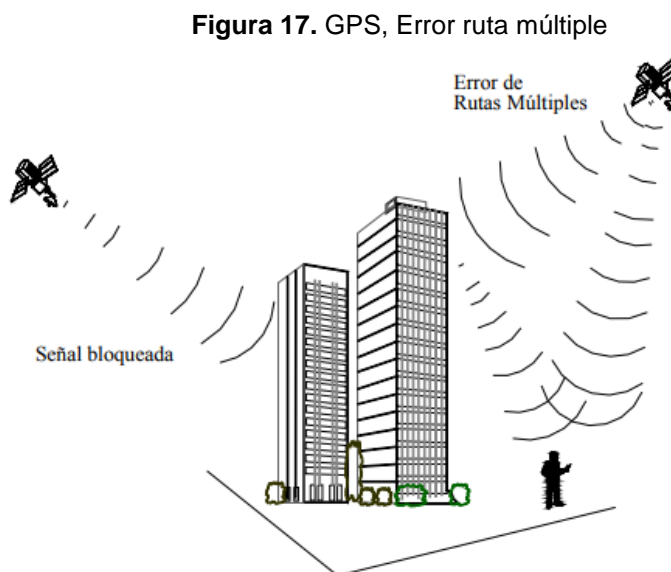
- PDOP: dilación de precisión para la posición
- HDOP: dilación de precisión horizontal
- VDOP: dilación de precisión vertical
- TDOP: dilación de precisión del tiempo

La configuración geométrica ideal se da cuando la posición relativa entre satélites forma ángulos agudos. Por lo contrario, una configuración geométrica pobre se da cuando los satélites están alineados o su posición relativa forma ángulos llanos.

2. Errores originados por el medio de propagación

Como se mencionó anteriormente, los cálculos en el posicionamiento por satélite asumen que la señal viaja a una velocidad constante igual a la de la luz. Sin embargo, la velocidad de la luz se mantiene constante solamente en el vacío. Cuando la señal penetra la ionosfera y la troposfera, debido a los cambios en densidades de las diferentes capas, se producen las refracciones ionosféricas y troposféricas, reduciendo la velocidad de la señal. Actualmente los receptores de G.P.S. toman en cuenta estas demoras haciendo las correcciones pertinentes.

- **El error por ruta múltiple (multipath)** se origina debido a la posibilidad de que una señal reflejada por objetos ubicados en la superficie de la tierra lleguen al receptor por dos o más trayectorias diferentes



Fuente: (Casanova Matera, 2002)

Para minimizar los efectos del error multipath se han desarrollado técnicas avanzadas de procesamiento de señales así como antenas para filtrar las señales que llegan de diferentes direcciones.

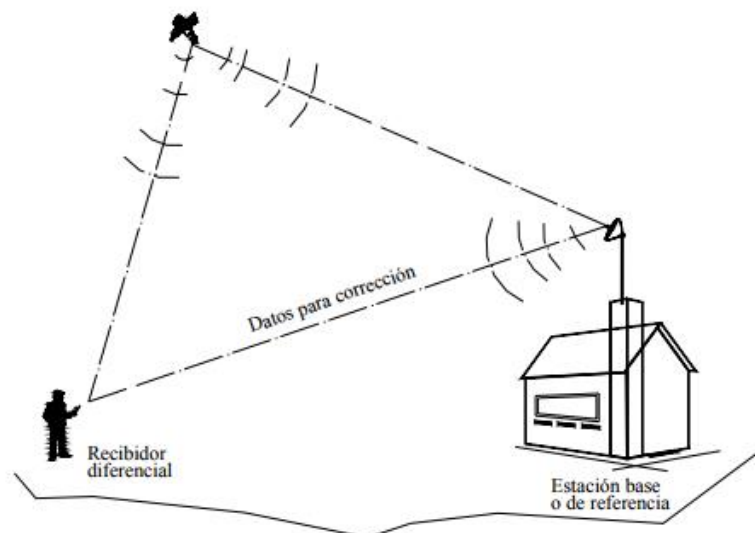
3. Errores en la recepción

Los errores en la recepción son el ruido, centro de fase de la antena, errores de reloj oscilador y el error de disponibilidad selectiva (S/A), el cual es una degradación de la señal del satélite causada en forma intencional por el Departamento de Defensa de los Estados Unidos.

El error de disponibilidad selectiva se corrige mediante la técnica de la corrección diferencial, en la cual se usa un receptor en una estación base cuya posición sea conocida con precisión y un receptor en el punto que se desea ubicar, recolectando datos simultáneamente.

Con la información obtenida en la estación base se calculan los diferenciales o correcciones que deben aplicarse a las mediciones del receptor en la estación del punto a ubicar.

Figura 18. GPS, Corrección Diferencial



Fuente: (Casanova Matera, 2002)

CAPÍTULO 3. HIPÓTESIS

3.1. Formulación de la Hipótesis

Una aplicación móvil basada en algoritmos de rutas óptimas afectara los desplazamientos de los conductores.

3.2. Operacionalización de variables

Tabla 11. Operacionalización de variable independiente.

VARIABLE	DEFINICIÓN CONCEPTUAL	DIMENSIONES	INDICADORES
Aplicación móvil basada en algoritmos de rutas óptimas.	Programa informático para dispositivos móviles como herramienta para permitir al usuario tomar decisiones sobre que rutas tomar para dirigirse de un punto inicial a uno final.	Eficiencia	Tiempo de respuesta de la aplicación.
		Funcionalidad	Precisión esperada de la aplicación.

Fuente: Propia.

Tabla 12. Operacionalización de variable dependiente.

VARIABLE	DEFINICIÓN CONCEPTUAL	DIMENSIONES	INDICADORES
Desplazamiento de los conductores de vehículos en la ciudad de Trujillo.	En mecánica, el desplazamiento es el vector que define la posición de un punto o partícula en relación a un origen A con respecto a una posición B.	Rutas óptimas	Distancia recorrida.
			Tiempo en recorrido.
			Número de incidencias evitadas.

Fuente: Propia.

CAPÍTULO 4. DESARROLLO

4.1. Proceso de desarrollo de software Móvil y web

De acuerdo a lo expuesto en los puntos anteriores, esta investigación propone la implementación de un sistema móvil y web para la gestión de los procesos especificados. Se consideró utilizar la metodología ligera **ICONIX** ya que constituye un nuevo enfoque del desarrollo de software y es una de las más aceptadas por los desarrolladores debido a su simplicidad en sus reglas y prácticas.

4.1.1. Disciplinas aplicadas

A continuación se resume las disciplinas que se aplicaran durante el proceso de desarrollo del software usando la metodología ICONIX, además se muestra los artefactos relacionados por cada disciplina.

Tabla 13. Fases de Metodología *ICONIX*.

FASES	ARTEFACTOS/TAREAS
Análisis de Requisitos	<ul style="list-style-type: none"> • Documento Visión • Modelo del Dominio • Modelo de casos de uso
Análisis y Diseño Preliminar	<ul style="list-style-type: none"> • Especificación de requerimientos del software • Diagrama de robustez
Diseño	<ul style="list-style-type: none"> • Descripción de la arquitectura del software • Diagrama de secuencia
Implementación	<ul style="list-style-type: none"> • Plan de pruebas • Descripción de la plataforma de despliegue

Fuente: Propia.

4.2. Documento Visión

4.2.1. Introducción

4.2.1.1. Propósito

El propósito de este documento es recopilar, analizar y definir las necesidades de alto nivel y las características de la aplicación móvil de algoritmos de rutas óptimas en su efecto en el tránsito vehicular en la ciudad de Trujillo.

Además se define el ámbito de la aplicación y la funcionalidad requerida que permita mejorar el tránsito vehicular de manera diferente y eficiente en la ciudad de Trujillo.

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso y otros documentos adicionales.

4.2.1.2. Alcance

El sistema permitirá a los conductores tomar decisiones sobre que rutas tomar para dirigirse de un punto inicial a uno final evadiendo algún incidente en la malla vehicular como accidentes de tráfico, obras municipales, tráfico vehicular.

4.2.1.3. Referencias

Las referencias aplicables son:

- Especificación de requerimientos de software
- Modelo de casos de uso

4.2.2. Posicionamiento

4.2.2.1. Enunciado del problema

Tabla 14. *Enunciado del problema.*

El Problema	<ul style="list-style-type: none"> • Congestionamiento vehicular en las principales avenidas de la ciudad de Trujillo.
Afecta	<ul style="list-style-type: none"> • Todo conductor de transporte privado.
El Impacto es	<ul style="list-style-type: none"> • Molestia de los conductores al no encontrar una ruta óptima para llegar a su destino. • Incomodidad de los conductores al no conocer los estados de las rutas tomadas.
La solución sería	<ul style="list-style-type: none"> • El desarrollo de una aplicación móvil de algoritmos de rutas óptimas.

Fuente: Propia.

4.2.2.2. Enunciado del posicionamiento del Producto

Tabla 15. *Enunciado del posicionamiento del producto.*

Para	<ul style="list-style-type: none"> • Todo conductor de transporte privado.
Quiénes	<ul style="list-style-type: none"> • Necesiten llegar a su destino tomando rutas óptimas libres de incidencias.
Aplicación móvil basada en algoritmos de rutas óptimas	<ul style="list-style-type: none"> • Programa informático para dispositivos móviles diseñado como herramienta para proveer rutas de un punto inicial a uno final
Qué	<ul style="list-style-type: none"> • Provee al usuario una ruta óptima para dirigirse de un punto inicial a uno final.
Diferente a	<ul style="list-style-type: none"> • Otras aplicaciones que no satisfacen las necesidades reales del conductor.
Nuestro producto	<ul style="list-style-type: none"> • Es una solución basada en la tecnología móvil y web, esto permite que nuestro producto sea flexible, de fácil acceso y fácil de usar, dando a conocer al usuario la ruta óptima entre 2 puntos

Fuente: Propia.

4.2.3. Definición de Stakeholders

4.2.3.1. Resumen de Stakeholders

Tabla 16. *Resumen Stakeholders.*

NOMBRE	DESCRIPCION	RESPONSIBILIDAD
Conductor	Persona encargada de conducir un vehículo para transportar a personas.	Conducir vehículo.

Fuente: Propia.

4.2.3.2. Resumen de Usuarios

Tabla 17. *Resumen de Usuarios.*

NOMBRE	DESCRIPCION	RESPONSIBILIDAD	STAKEHOLDER
Conductor	Persona encargada de conducir un vehículo para transportar a personas	Conducir, buscar rutas, buscar información de tráfico vehicular e incidencias.	Conductor
Administrador	Persona encargada de administrar incidencias en la malla vehicular	Agregar, actualizar eliminar información sobre incidencias en la malla vehicular	Administrador

Fuente: Propia.

4.2.4. Descripción del producto

4.2.4.1. Perspectiva del producto

El producto a desarrollar es un sistema basado en tecnología móvil y web para generar rutas de transportes y determinando la mejor forma de realizar el recorrido en función de ciertos parámetros que ingresa el usuario y mediante algoritmos de rutas óptimas que dará como resultado rutas óptimas.

4.2.5. Características del producto

A continuación se mostrara un listado con los beneficios que obtendrá el usuario a partir del producto.

Tabla 18. Beneficios del producto.

BENEFICIOS	CARACTERÍSTICAS DEL PRODUCTO
Proveer rutas óptimas de acuerdo al tráfico vehicular que se presente.	El usuario contara con un interfaz fácil de usar en la cual podrá visualizar las rutas óptimas que desea buscar de un punto de origen y destino de acuerdo al tráfico vehicular.
Visualizar incidencias dentro de las vías de la ciudad de Trujillo.	El usuario contara con un interfaz donde podrá visualizar a detalle sobre las incidencias que ocurren en tiempo real en las principales vías de la ciudad de Trujillo.

Fuente: Propia.

4.2.6. Requerimientos del producto

4.2.6.1. Requerimientos funcionales

Subsistema Móvil

- Autenticar usuario

Subsistema Web

- Autenticar administrador
- Registrar administrador
- Gestionar Incidencias
- Generar reporte de incidencias

Subsistema de Mapas

- Generar ruta óptima.

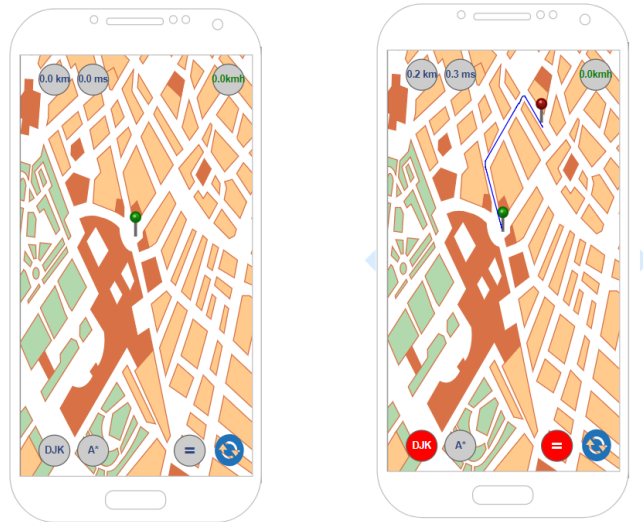
4.2.6.2. Requerimientos no funcionales

- El software debe cumplir con los estándares RUP
- Se usará el lenguaje de programación Java + Spring Framework para la parte web
- Se usará el lenguaje de programación Java + SDK de android para la parte Móvil.
- La base de datos será administrada bajo el motor Postgres SQL + extensiones (PGRouting, postgis, fuzzystmatch).
- Se usara como servidor web APACHE TOMCAT.
- Se usara como Infraestructura web Microsoft Azure.
- El tiempo de respuesta del sistema no deberá exceder los 5 segundos.
- Las interfaces deben ser amigables y sencillas de usar.

4.3. Prototipos

4.3.1. Móvil

Figura 19. Prototipo – CU Generar Ruta Óptima. (Fuente Propia).



4.3.2. Web

Figura 20. Prototipo – CU Gestionar Incidencias (Registrar Incidencia). (Fuente Propia).

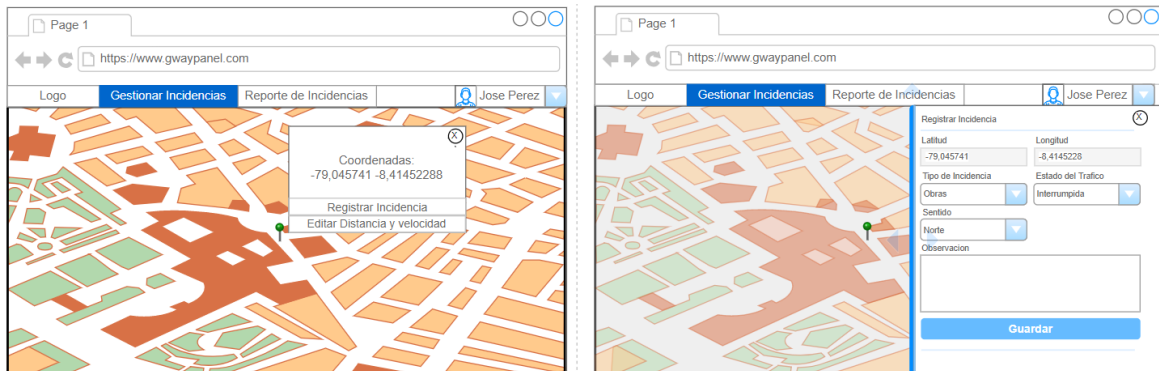
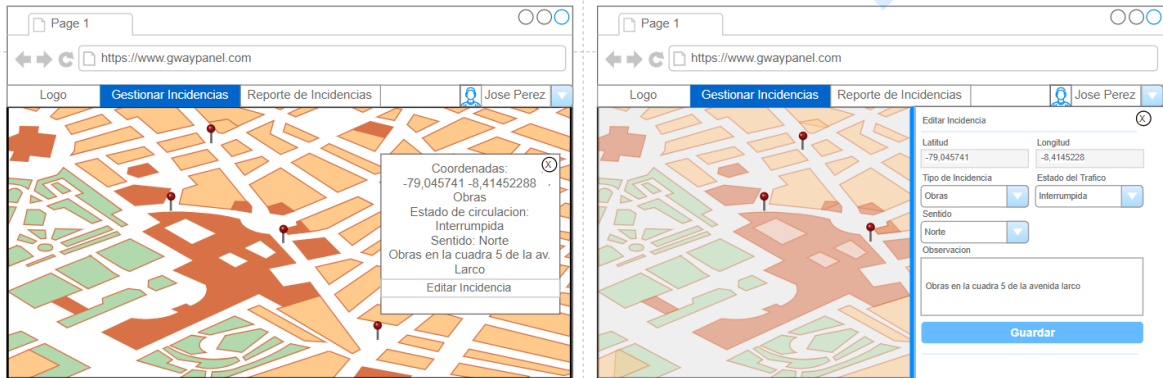


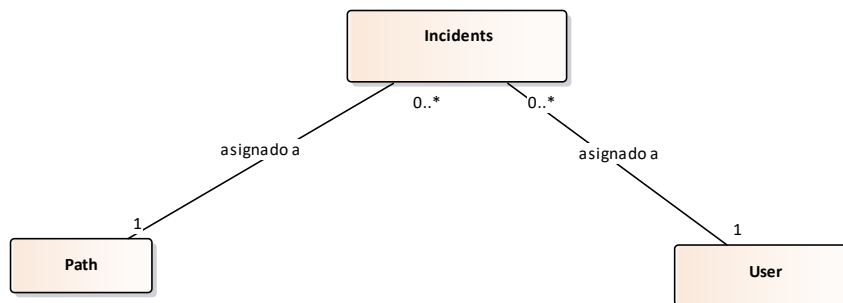
Figura 21. Prototipo – CU Gestionar Incidencias (Editar Incidencia – Listar Incidencias). (Fuente Propia).



4.4. Modelo del Dominio

El informe del Modelo de Dominio ilustra las entidades identificadas y sus relaciones, además de las restricciones de integridad que aplican sobre ellos.

Figura 22. Modelo de Dominio (Fuente Propia).



4.5. Modelo de Casos de Uso

4.5.1. Descripción Global

4.5.1.1. Reporte del modelo de casos de usos

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema de software. Cada caso de uso proporciona uno o más escenarios que indican como debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

Para el presente documento los casos de uso y los actores son:

Figura 23. Diagrama de casos de uso. (Fuente Propia).

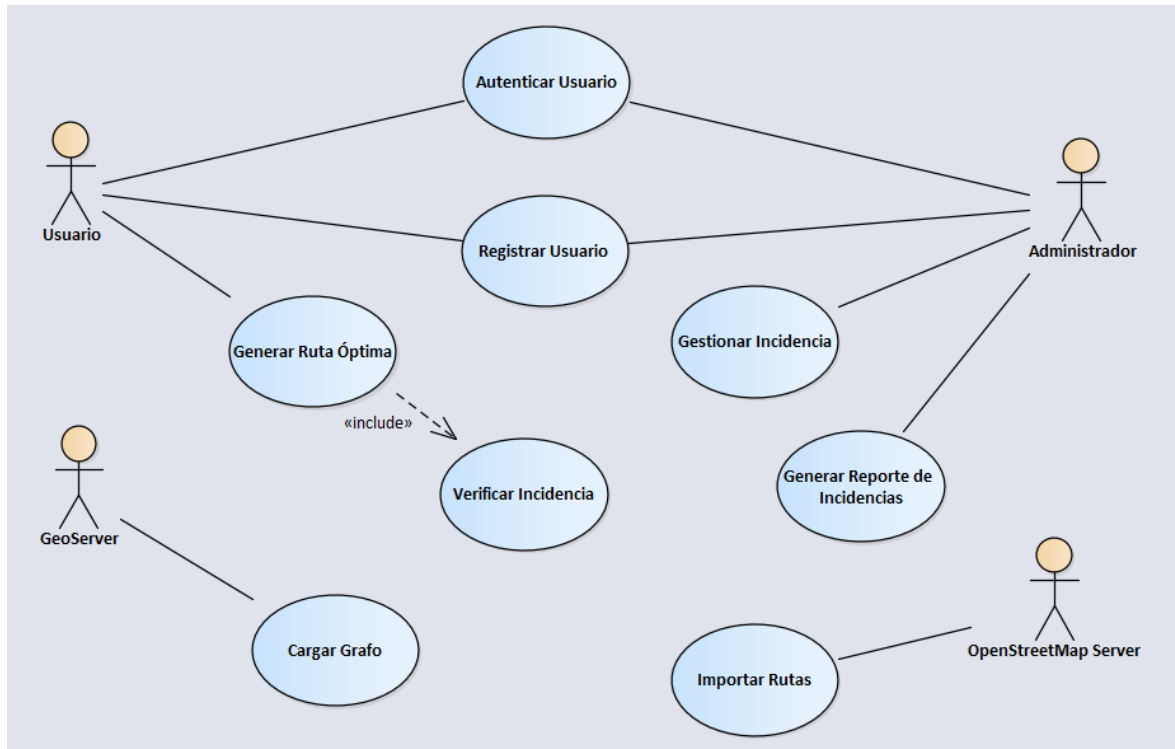


Tabla 19. Listado de casos de uso.

CÓDIGO	CASO DE USO	ACTORES PARTICIPANTES
CU01	Autenticar usuario	Usuario
CU02	Registrar usuario	Usuario
CU03	Gestionar incidencias	Administrador
CU04	Generar reporte de incidencias	Administrador
CU05	Generar ruta óptima	Usuario

Fuente: Propia.

4.5.1.2. Consideraciones y dependencias

Se ha considerado estructurar el modelo de casos de uso en 3 Subsistemas, con el fin de facilitar el desarrollo del análisis y lograr que los lectores de este documento puedan comprender la importancia de cada uno de los casos de uso plasmados.

Se asume que los requisitos descritos en este documento son estables.

Se asume que los usuarios tienen la capacitación o experiencia mínima necesaria para el uso de dispositivos móviles.

4.5.2. Requerimientos Específicos

4.5.2.1. Funcionalidad

Subsistema Móvil

4.5.2.1.1 CU01 - Autenticar usuario

La aplicación móvil deberá permitir al usuario iniciar sesión usando sus credenciales asignadas, para poder acceder a la aplicación.

4.5.2.1.2 CU02 - Registrar usuario

El sistema deberá permitir al usuario registrar sus datos personales.

Subsistema Web

4.5.2.1.3 CU03 - Gestionar Incidencias

El sistema deberá permitir al administrador registrar, actualizar y eliminar las incidencias que se presenta en la ciudad de Trujillo.

4.5.2.1.4 CU04 - Generar reporte de incidencias

El sistema deberá permitir al usuario generar un reporte de incidencias que ocurren en la ciudad de Trujillo de acuerdo a los filtros que el sistema provee.

Subsistema de Mapas

4.5.2.1.5 CU05 - Generar ruta óptima

El sistema deberá permitir a la aplicación móvil generar la ruta óptima entre 2 puntos (ubicación actual y ubicación destino).

4.6. Fase de Planificación

4.6.1. Planificación Inicial

En la planificación inicial listaremos las tareas de acuerdo a su **PRIORIDAD** (Bajo, Media o Alta según la importancia relevancia que tenga), **RIESGO** (Bajo, Medio o Alto según la probabilidad de fallo de cada tarea en el desarrollo), **ESFUERZO** (calificando 1, 2 o 3 según el tiempo y trabajo que demandara en desarrollar la tarea) e **ITERACION** (momento de implementación de cada tarea).

Tabla 20. *Planificación Inicial de tareas.*

Nro.	TAREAS	Prioridad de Negocio	Riesgo	Esfuerzo	Iteración
	Nombre				
1	Implementación del CU05 - Gestionar incidencias	Alta	Medio	2	1
2	Implementación del CU07 - Generar ruta óptima	Alta	Alto	3	2
3	Integración de los módulos	Alta	Alto	3	3

Fuente: Propia.

4.6.2. Estimación de Tiempos

De acuerdo a las ponderaciones de la prioridad, riesgo y esfuerzo se ha estimado el tiempo de desarrollo de cada tarea.

Tabla 21. *Estimación de tiempos de tarea.*

Nro.	TAREAS	Prioridad de Negocio	Tiempo Estimado
	Nombre		
001	Implementación del CU03 - Gestionar incidencias	Alta	35 días
002	Implementación del CU05 - Generar ruta óptima	Alta	45 días
003	Integración de los módulos	Alta	10 días

Fuente: Propia.

Estimación de la velocidad del producto inicial:

- Tiempo calendario: 5 días por semana, de lunes a viernes.
- Tiempo total estimado para la elaboración de las tareas: 90 días.

4.6.3. Plan de Entrega

Tabla 22. *Plan de entrega.*

Entregable	Tareas	Fecha de Inicio	Fecha de Término	Fecha Entrega
Entregable 1	001	07/06/2018	11/07/2018	15/07/2018
Entregable 2	002	07/06/2018	21/07/2018	30/07/2018
Entregable 3	003	02/08/2018	12/08/2018	17/08/2018

Fuente: Propia.

4.7. Fase de Iteraciones

4.7.1. Iteración 01

4.7.1.1. Especificación de requerimientos de software

4.7.1.1.1 Gestionar Incidencias

4.7.1.1.1.1 Descripción

A través de este caso de uso el administrador puede registrar, editar y eliminar incidencias.

4.7.1.1.1.2 Pre-condiciones

Antes de realizar este caso de uso se debe considerar:

- El usuario debe haber sido registrado previamente.

4.7.1.1.1.3 Flujo de eventos principal

Registrar:

1. El sistema muestra el panel con un visor de mapa con una lista de incidencias existentes.
2. El administrador hace Click en algún punto del mapa para registrar una incidencia.
3. El sistema muestra un popup de con un mensaje de "Registrar incidencia aquí", las coordenadas del lugar y una opción de "Agregar".
4. El administrador hace Click en el botón "Agregar".
5. El sistema muestra un slider lateral de registro con los campos: latitud (valor autogenerado), longitud (valor autogenerado), tipo de incidencias, estado del tráfico, sentido, observación.
6. El administrador selecciona el tipo de incidencia, el estado de tráfico, sentido e ingresa un texto en observación y hace Click en el botón "Guardar".
7. El sistema valida la información ingresados por el administrador.
8. El sistema oculta el slider lateral de registro y muestra en el mapa la incidencia ingresada.
9. El administrador termina el caso de uso.

Editar

10. El sistema muestra el panel con un visor de mapa con una lista de incidencias existentes.
11. El administrador pasa el mouse o hace Click en la incidencia existente en el mapa.
12. El sistema muestra un popup de con un mensaje de “Editar incidencia aquí”, las coordenadas del lugar, la información registrada y las opciones de “Editar” y “Eliminar”.
13. El administrador hace Click en el botón “Editar”.
14. El sistema muestra un slider lateral de edición con los campos: latitud (valor autogenerado), longitud (valor autogenerado), tipo de incidencias, estado del tráfico, sentido, observación.
15. El administrador selecciona el campo que desea modificar y hace Click en el botón “Guardar”.
16. El sistema valida la información ingresados por el administrador.
17. El sistema oculta el slider lateral de registro y muestra en el mapa la incidencia editada.
18. El administrador termina el caso de uso.

Eliminar

19. El sistema muestra el panel con un visor de mapa con una lista de incidencias existentes.
20. El administrador pasa el mouse o hace Click en la incidencia existente en el mapa.
21. El sistema muestra un popup de con un mensaje de “Editar incidencia aquí”, las coordenadas del lugar, la información registrada y las opciones de “Editar” y “Eliminar”.
22. El administrador hace Click en el botón “Eliminar”.
23. El sistema muestra una alerta “¿Esta seguro que desea eliminar?” y dos opciones “Aceptar” y “Cancelar”.
24. El administrador hace Click en “Aceptar”.
25. El sistema muestra un mensaje “Se eliminó correctamente”.
26. El administrador termina el caso de uso.

4.7.1.1.1.4 Flujo de eventos alternativos

En el paso (6) y (15) ingreso de datos erróneos:

- El sistema muestra un mensaje “Datos ingresados incorrectos”.

En el paso (23) Click en el botón “Cancelar”:

- El sistema cierra la alerta de eliminación.

Conexión a internet perdida:

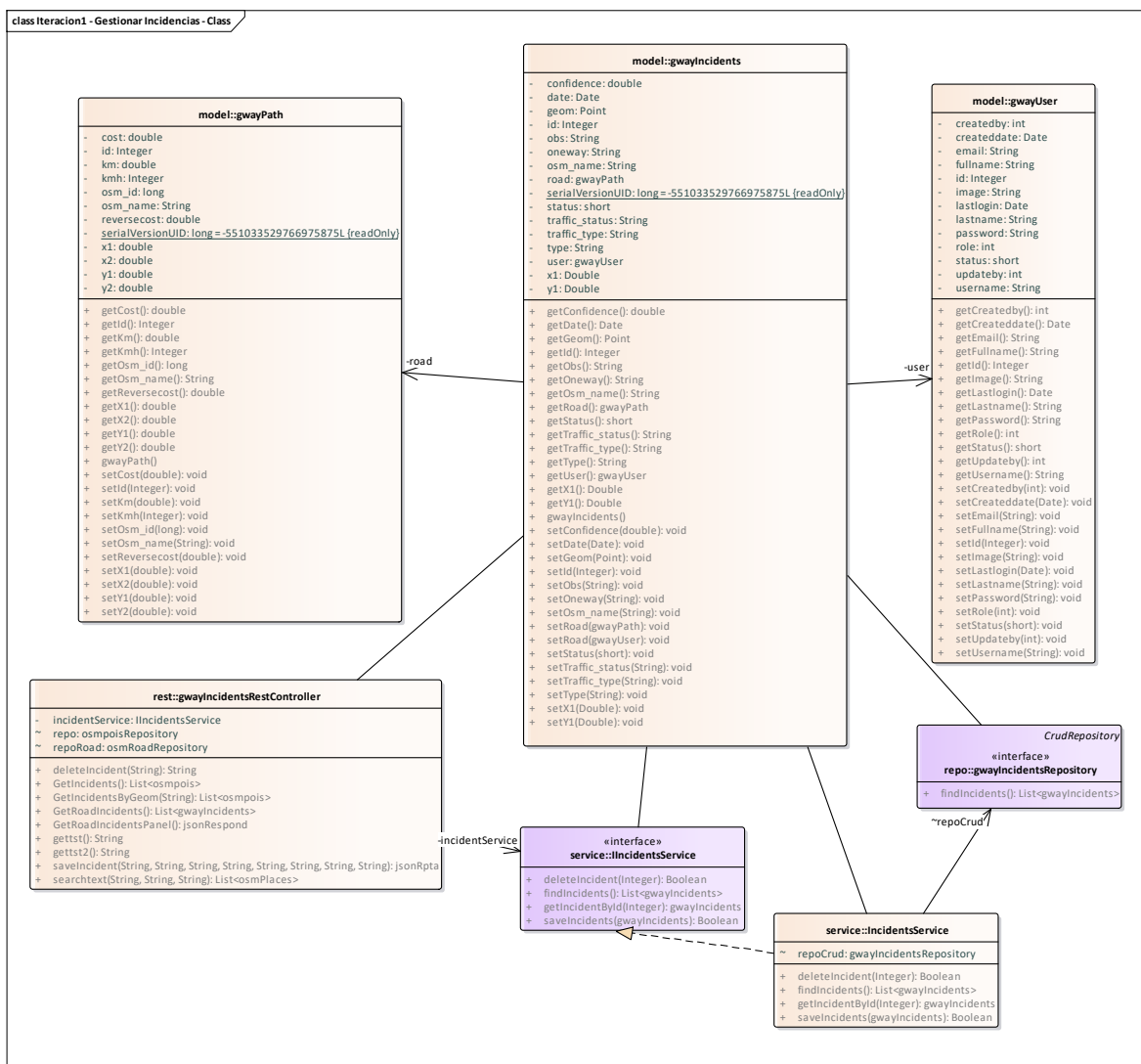
- El sistema mostrara un mensaje indicado “Conexión a internet perdida”.

4.7.1.1.1.5 Post-condiciones

- La incidencia es registrada en la Base de datos.

4.7.1.2. Diagrama de Clases de Datos

Figura 24. Diagrama de Clases de Datos UC – Gestionar Incidencias. (Fuente Propia).

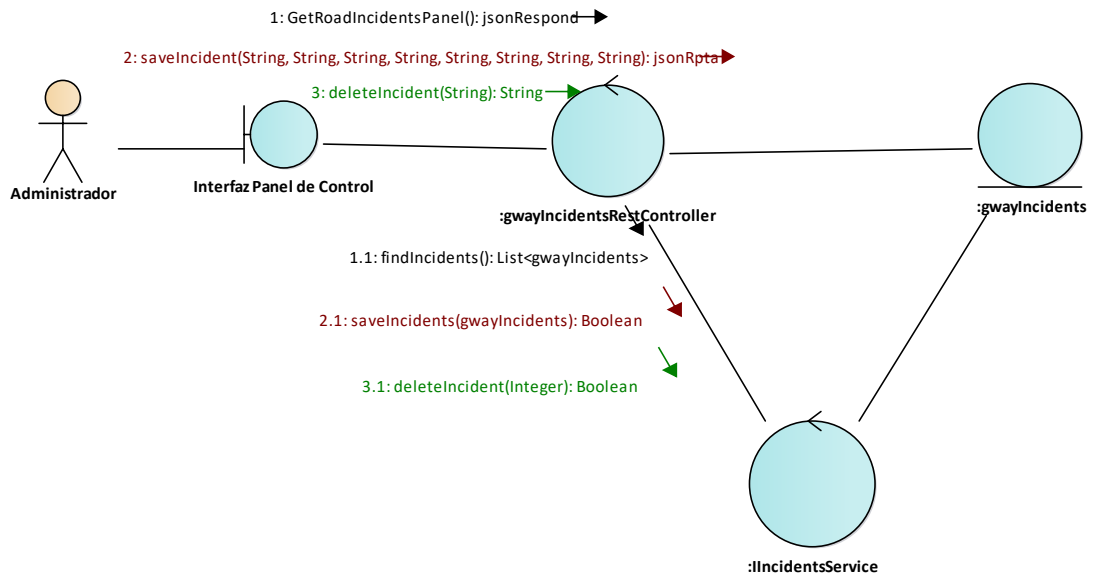


4.7.1.3. Diagrama de Robustez

4.7.1.3.1 Subsistema Web

4.7.1.3.1.1 Gestionar Incidencias

Figura 25. Diagrama de Robustez UC – Gestionar Incidencias. (Fuente Propia).

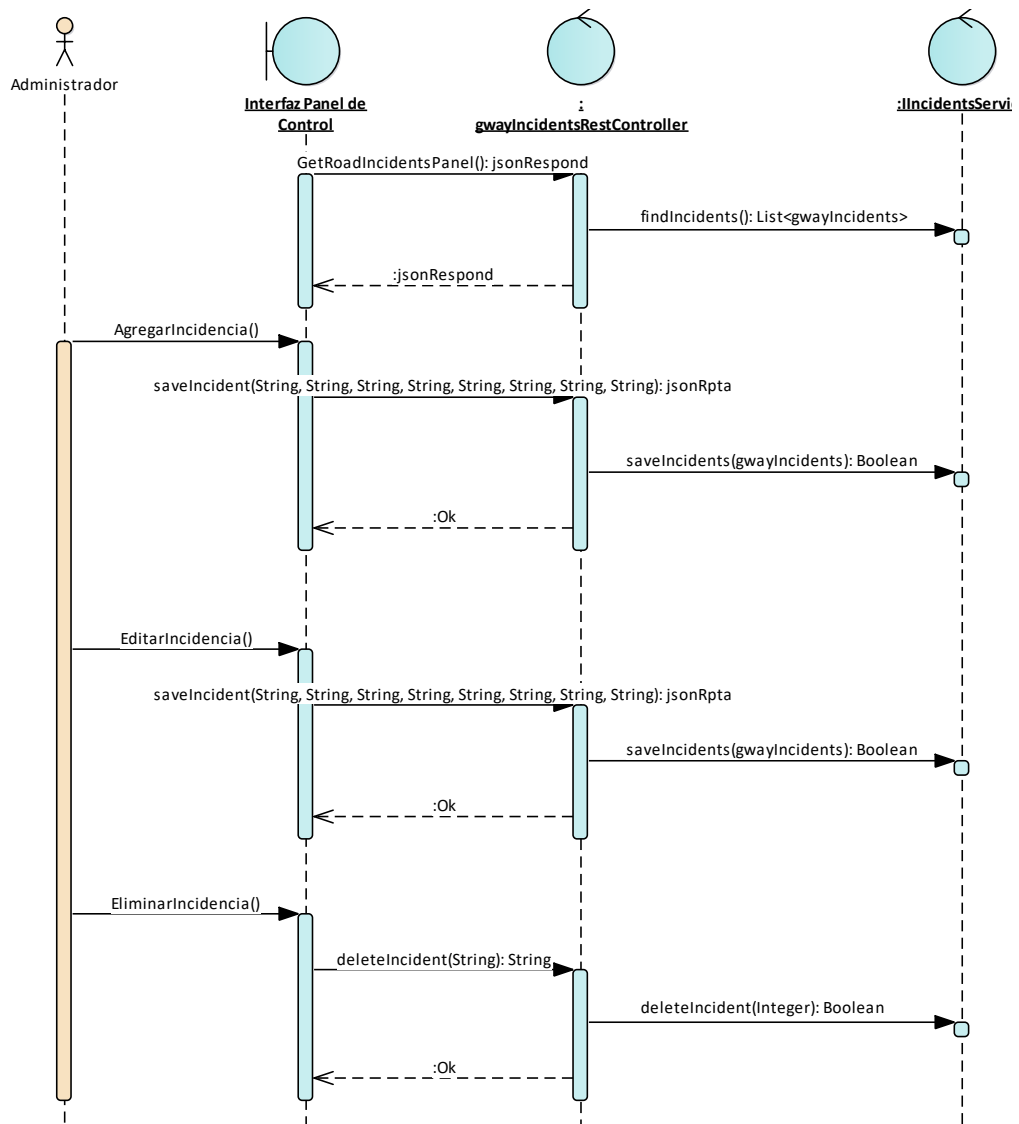


4.7.1.4. Diagrama de Secuencia

4.7.1.4.1 Subsistema Web

4.7.1.4.1.1 Gestionar Incidencias

Figura 26. Diagrama de secuencia UC – Gestionar Incidencias. (Fuente Propia).



4.7.1.5. Plan de pruebas

4.7.1.5.1 Casos de prueba

A continuación, se listan los casos de prueba desarrollados:

- Caso de prueba Gestionar Incidencia

4.7.1.5.1.1 Caso de prueba – Gestionar Incidencia

La interfaz de registro provee los siguientes campos:

- Latitud (Autogenerado)
- Longitud (Autogenerado)
- Tipo de Incidencia
- Tipo de atasco
- Estado del tráfico
- Sentido
- Observación

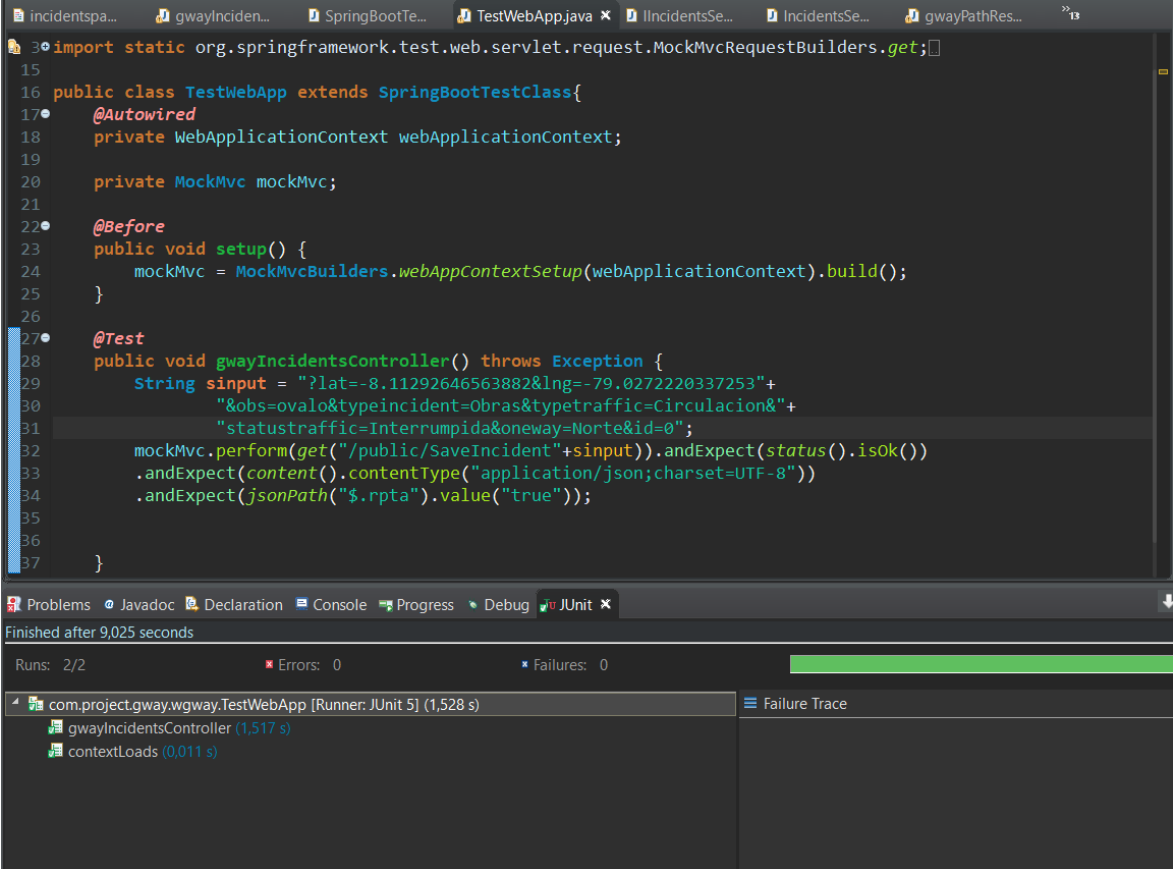
Tabla 23. Prueba funcional de caja negra – Gestionar Incidencia.

NOMBRE CAMPO	VALORES	RESULTADO
Latitud	=Valores predeterminados	Válido
	=Vacío	Inválido
	=Caracteres ASCII	Inválido
Longitud	=Valores predeterminados	Válido
	=Vacío	Inválido
	=Caracteres ASCII	Inválido
Tipo de Incidencia	=Valores predeterminados	Válido
	=Vacío	Inválido
	=Numérico	Válido
Tipo de atasco	=Valores predeterminados	Válido
	=Vacío	Inválido
	=Numérico	Válido
Estado del trafico	=Valores predeterminados	Válido
	=Vacío	Inválido
	=Numérico	Válido
Sentido	=Valores predeterminados	Válido
	=Vacío	Inválido
	=Numérico	Válido
Observación	=Letra mayúscula o minúscula	Válido
	=Vacío	Inválido
	=Numérico	Válido

Fuente: Propia.

Pruebas Unitarias - CU Gestionar Incidencia

Figura 27. Pruebas Unitarias UC – Gestionar Incidencias. (Fuente Propia).



```

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;
15
16 public class TestWebApp extends SpringBootTestClass{
17     @Autowired
18     private WebApplicationContext webApplicationContext;
19
20     private MockMvc mockMvc;
21
22     @Before
23     public void setup() {
24         mockMvc = MockMvcBuilders.webAppContextSetup(webApplicationContext).build();
25     }
26
27     @Test
28     public void gwayIncidentsController() throws Exception {
29         String sinput = "?lat=-8.11292646563882&lng=-79.0272220337253"+
30             "&obs=ovalo&typeincident=Obras&typetraffic=Circulacion"&+
31             "statustraffic=Interrumpida&oneway=Norte&id=0";
32         mockMvc.perform(get("/public/SaveIncident"+sinput)).andExpect(status().isOk())
33             .andExpect(content().contentType("application/json;charset=UTF-8"))
34             .andExpect(jsonPath("$.rpta").value("true"));
35
36     }
37

```

Finished after 9,025 seconds

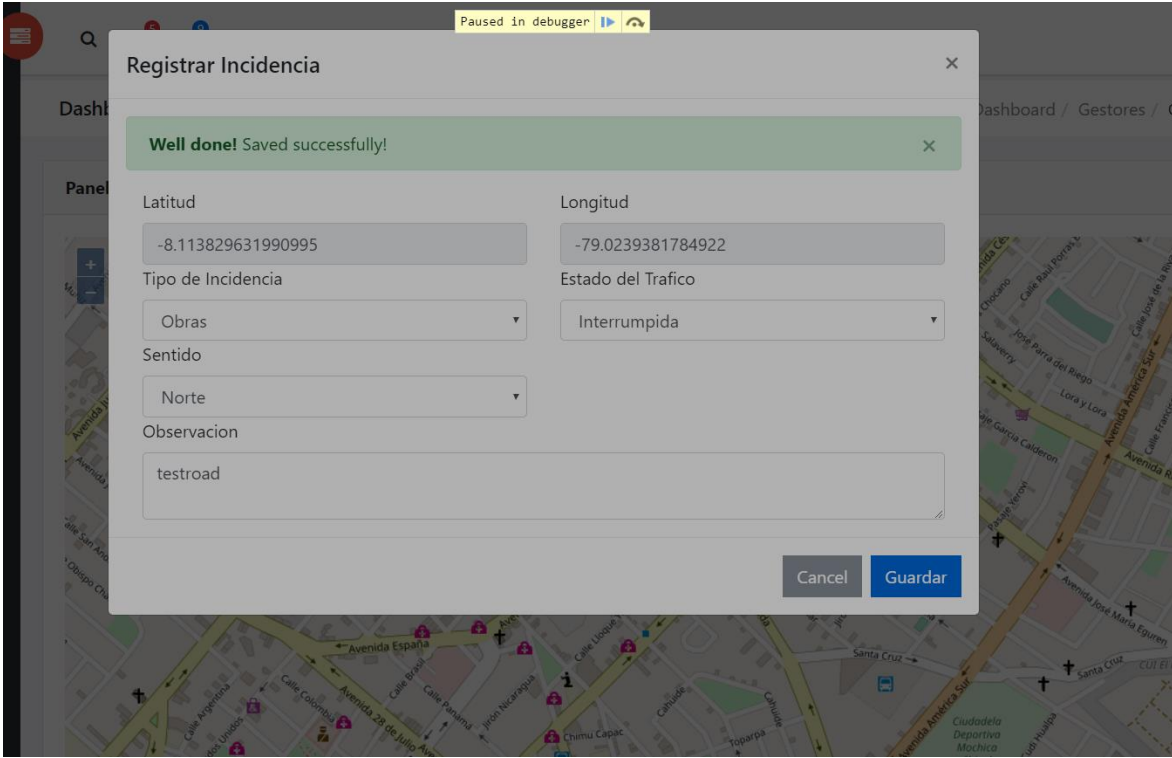
Runs: 2/2 Errors: 0 Failures: 0

com.project.gway.gway.TestWebApp [Runner: JUnit 5] (1,528 s)

- gwayIncidentsController (1,517 s)
- contextLoads (0,011 s)

Prueba Funcional – CU Gestionar Incidencia

Figura 28. Prueba Funcional UC – Gestionar Incidencias. (Fuente Propia).



id	obs	road_id	confidence	date	status	osm_id	osm_name	x1
integer	text	integer	double precision	date	smallint	bigint	character varying (255)	double precision
1	54 testroad	629664	100	2018-08-20	1	[null]	Avenida España	-79.0239381784922

4.7.2. Iteración 02

4.7.2.1. Especificación de requerimientos de software

Subsistema Móvil

4.7.2.1.1 Generar ruta óptima

4.7.2.1.1.1 Descripción

A través de este caso de uso la aplicación móvil podrá generar la ruta óptima de acuerdo a las coordenadas ingresadas del origen y destino.

4.7.2.1.1.2 Pre-condiciones

Antes de realizar este caso de uso se debe considerar:

- GeoServer Habilitado.
- Cargar Datos Geo-espaciales.

4.7.2.1.1.3 Flujo de eventos principal

1. La aplicación móvil hace una solicitud HTTP de búsqueda de rutas.
2. El sistema ejecuta el algoritmo de ruta óptima.
3. El sistema evalúa las rutas sin incidencias.
4. El sistema compara las rutas.
5. El sistema obtiene la ruta óptima.
6. El sistema da como respuesta HTTP la ruta óptima.
7. La aplicación móvil termina el caso de uso.

4.7.2.1.1.4 Flujo de eventos alternativos

Conexión a internet perdida:

- El sistema mostrara un mensaje indicado "Conexión a internet perdida".

4.7.2.1.1.5 Post-condiciones

4.7.2.2. Diagrama de Clases de Datos

Figura 29. Diagrama de Clases de Datos UC – Generar Ruta Óptima - Android. (Fuente Propia).

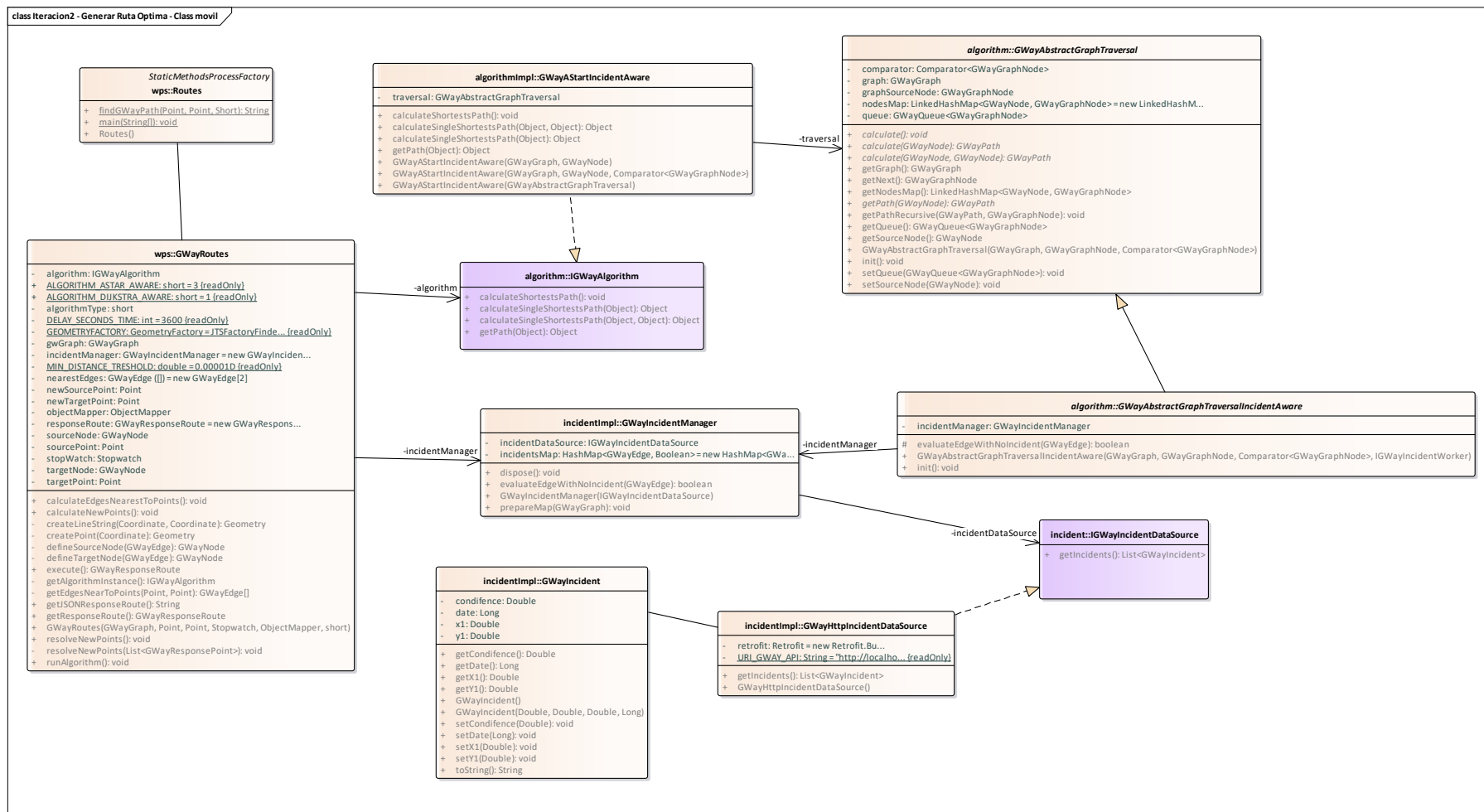
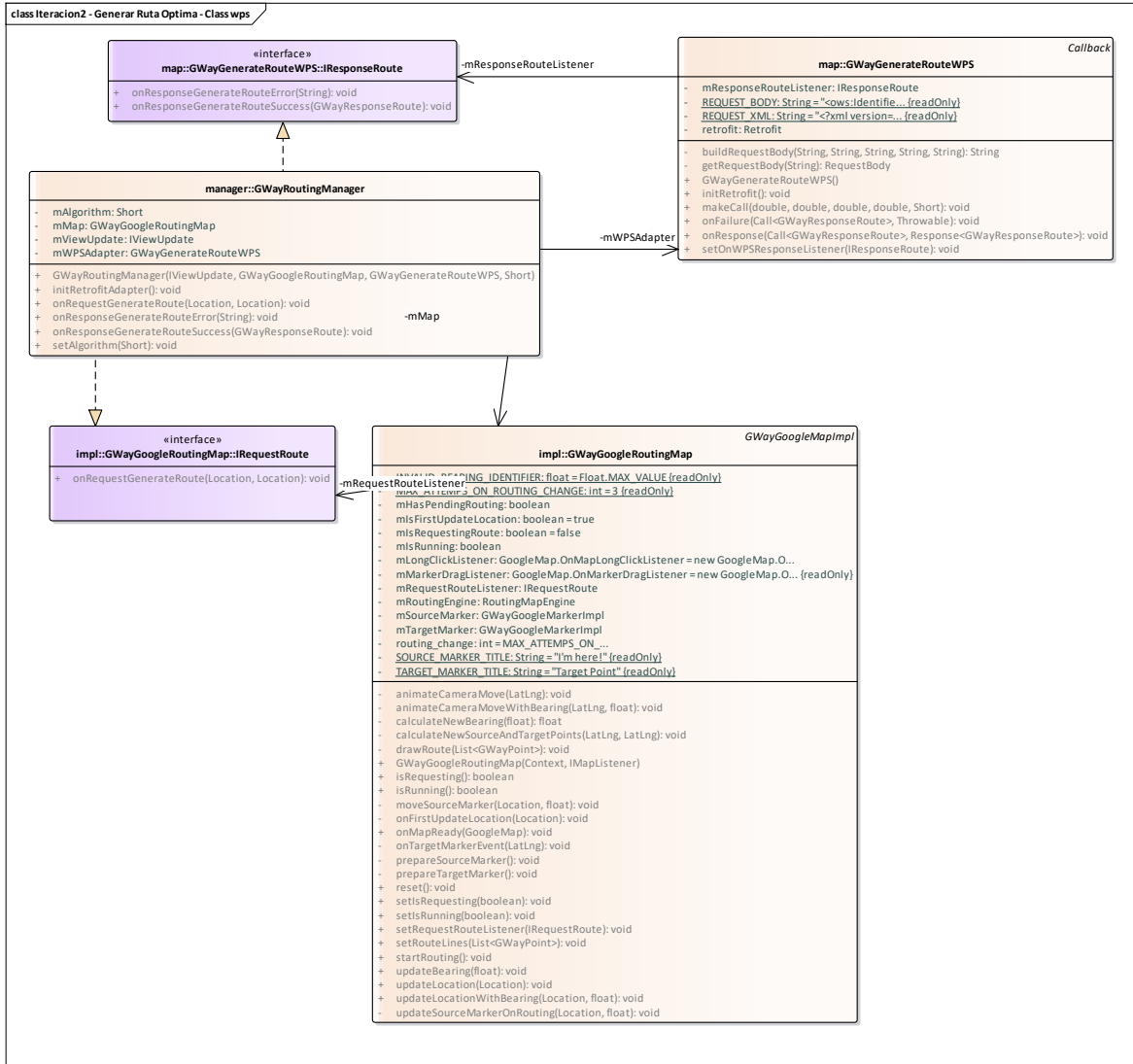


Figura 30. Diagrama de Clases de Datos UC – Generar Ruta Óptima - WPS. (Fuente Propia).



4.7.2.3. Diagrama de Robustez

4.7.2.3.1 Subsistema Móvil

4.7.2.3.1.1 Generar ruta óptima

Figura 31. Diagrama de Robustez UC – Generar ruta óptima - Android. (Fuente Propia).

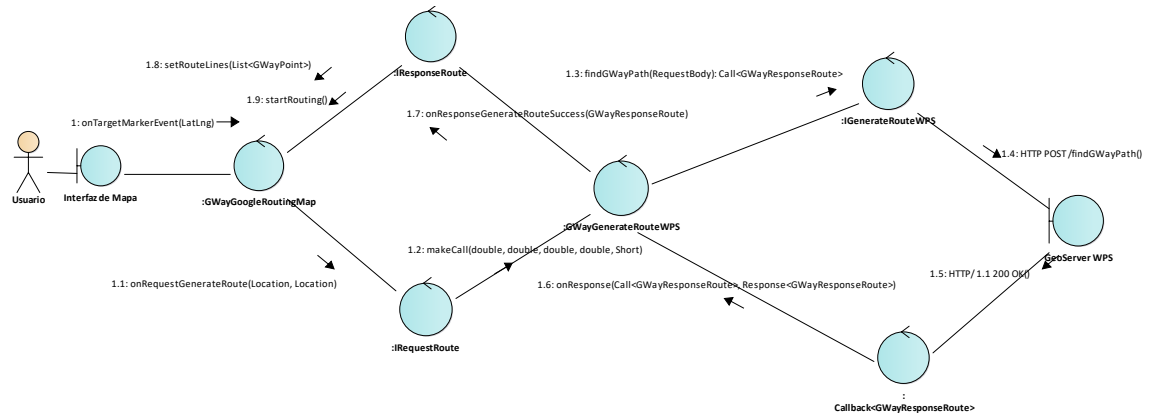
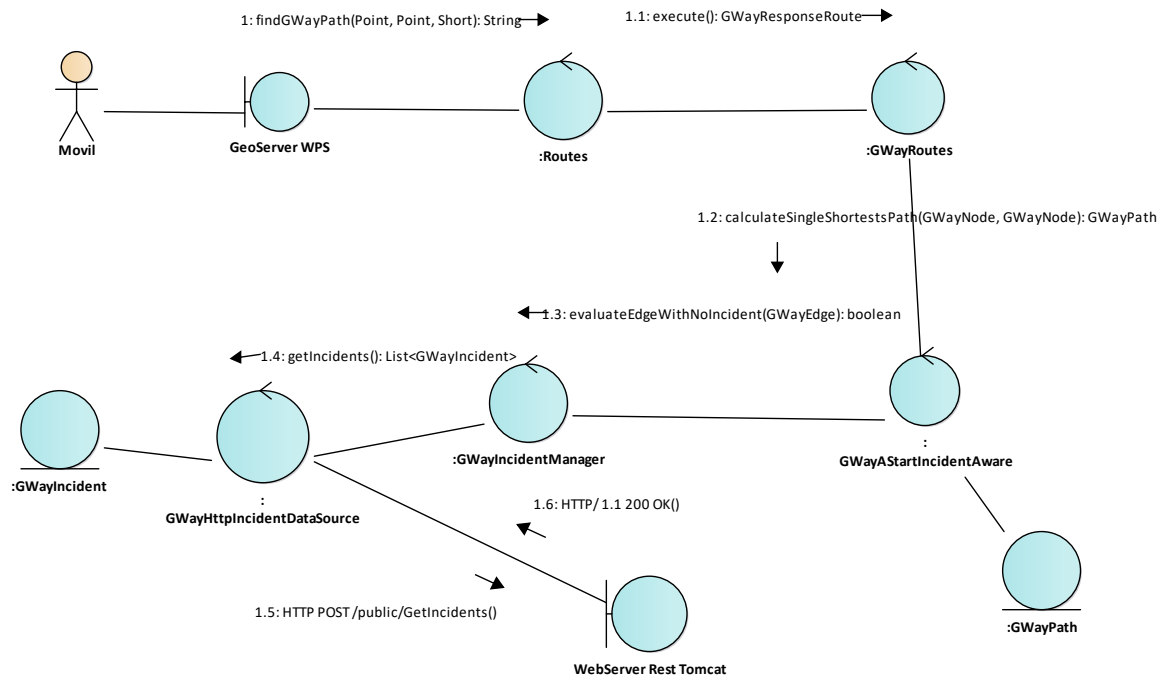


Figura 32. Diagrama de Robustez UC – Generar ruta óptima - WPS. (Fuente Propia).



4.7.2.4. Diagrama de Secuencia

4.7.2.4.1 Subsistema Móvil

4.7.2.4.1.1 Generar ruta óptima

Figura 33. Diagrama de secuencia UC – Generar ruta óptima - Android. (Fuente Propia).

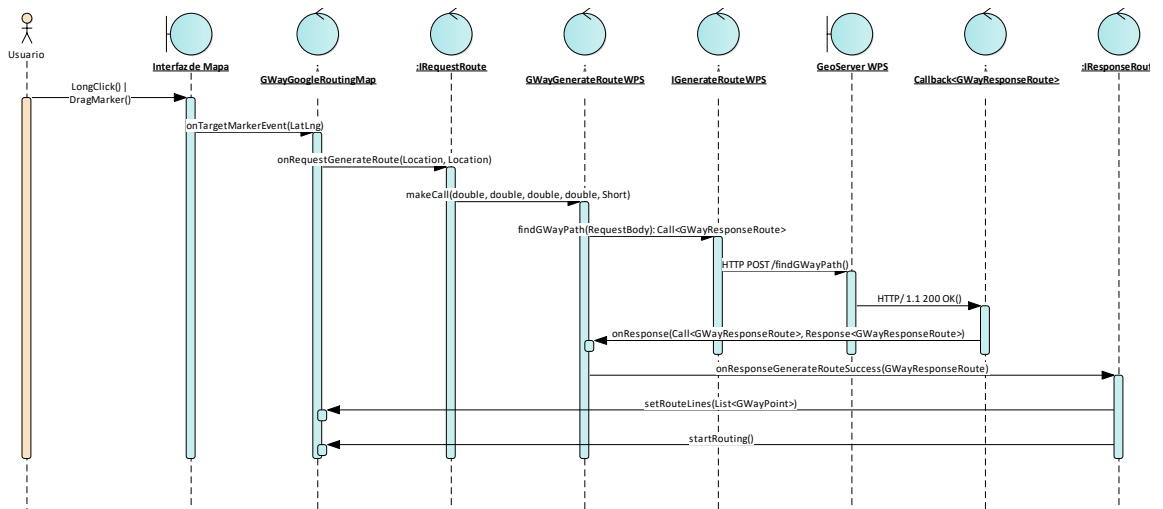
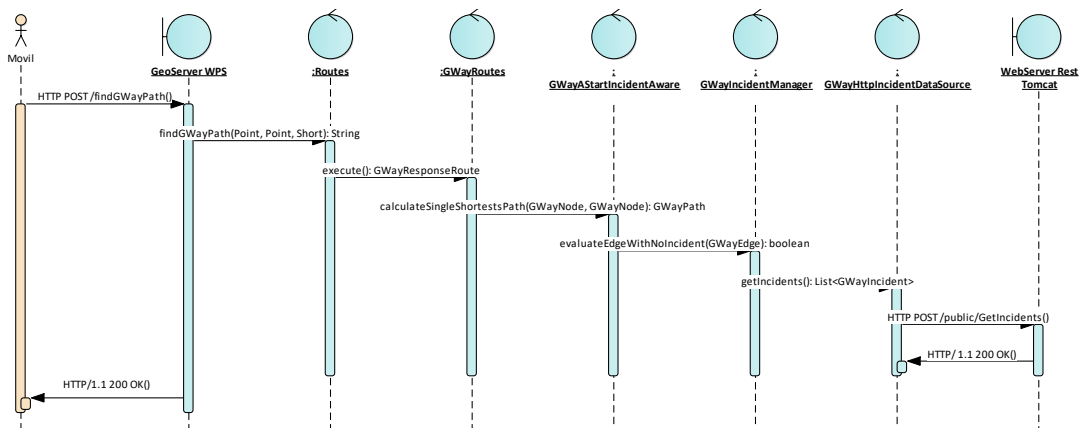


Figura 34. Diagrama de secuencia UC – Generar ruta óptima - WPS. (Fuente Propia).



4.7.2.5. Plan de pruebas

4.7.2.5.1 Casos de prueba

A continuación, se listan los casos de prueba desarrollados para el subsistema móvil y web:

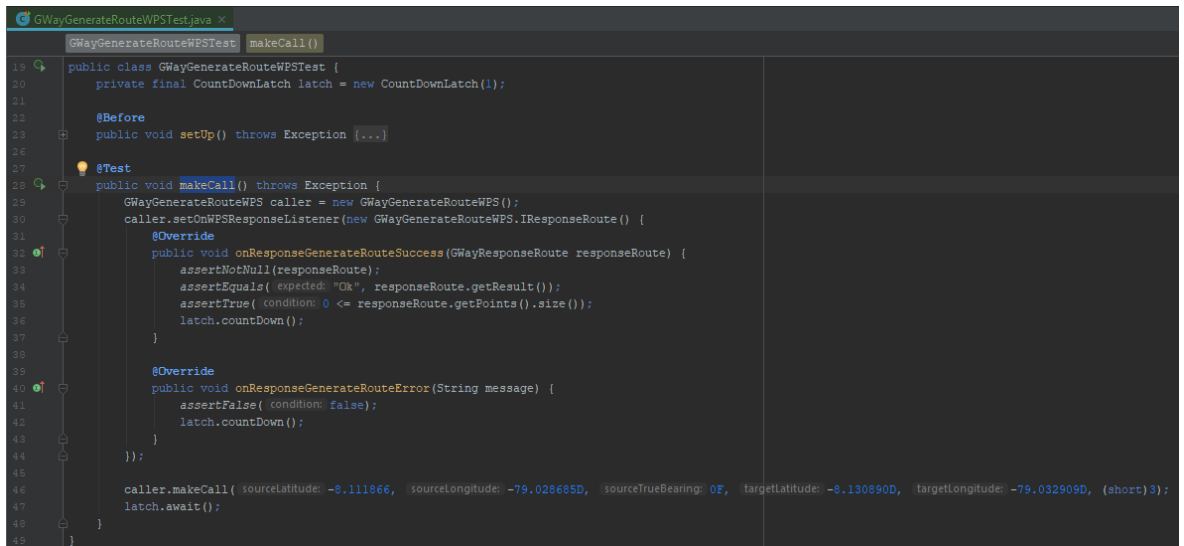
- Caso de prueba Generar ruta óptima

4.7.2.5.1.1 Caso de prueba – Generar ruta óptima

Para este caso de uso se implementó la prueba unitaria y funcional.

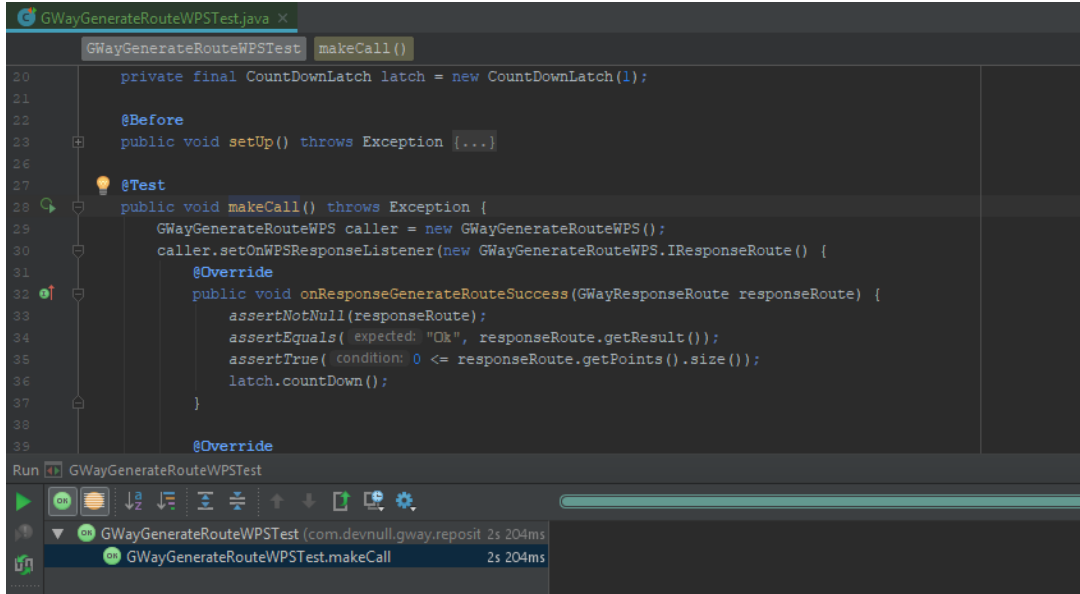
Pruebas Unitarias - CU Generar Ruta Óptima – Móvil

Figura 35. Pruebas Unitarias UC – Generar Ruta Óptima - Móvil. (Fuente Propia).



```
18 public class GWayGenerateRouteWPSTest {
19     private final CountDownLatch latch = new CountDownLatch(1);
20
21
22     @Before
23     public void setUp() throws Exception {
24
25
26
27
28
29     @Test
30     public void makeCall() throws Exception {
31         GWayGenerateRouteWPS caller = new GWayGenerateRouteWPS();
32         caller.setOnWPSResponseListener(new GWayGenerateRouteWPS.IResponseRoute() {
33             @Override
34             public void onResponseGenerateRouteSuccess(GWayResponseRoute responseRoute) {
35                 assertNotNull(responseRoute);
36                 assertEquals("Ok", responseRoute.getResult());
37                 assertTrue("condition: 0 <= responseRoute.getPoints().size()",
38                     0 <= responseRoute.getPoints().size());
39                 latch.countDown();
40             }
41
42             @Override
43             public void onResponseGenerateRouteError(String message) {
44                 assertFalse("condition: false");
45                 latch.countDown();
46             }
47         });
48         caller.makeCall( sourceLatitude: -8.111866, sourceLongitude: -79.028685D, sourceTrueBearing: 0F, targetLatitude: -8.130890D, targetLongitude: -79.032909D, (short)3);
49         latch.await();
50     }
51 }
```

Figura 36. Pruebas Unitarias UC – Generar Ruta Óptima - Móvil. (Fuente Propia).



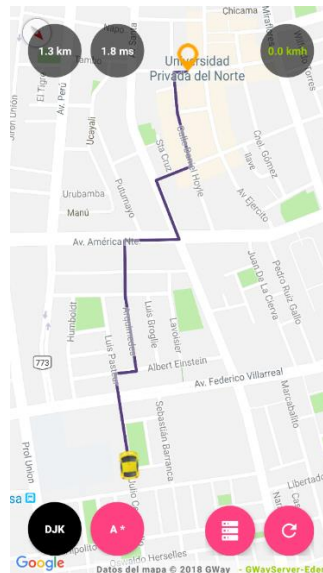
```

GWayGenerateRouteWPSTest.java
GWayGenerateRouteWPSTest makeCall()
20 private final CountDownLatch latch = new CountDownLatch(1);
21
22 @Before
23 public void setUp() throws Exception {...}
24
25 @Test
26 public void makeCall() throws Exception {
27     GWayGenerateRouteWPS caller = new GWayGenerateRouteWPS();
28     caller.setOnWPSResponseListener(new GWayGenerateRouteWPS.IResponseRoute() {
29         @Override
30         public void onResponseGenerateRouteSuccess(GWayResponseRoute responseRoute) {
31             assertNotNull(responseRoute);
32             assertEquals("Ok", responseRoute.getResult());
33             assertTrue("condition: 0 <= responseRoute.getPoints().size()",
34                 0 <= responseRoute.getPoints().size());
35             latch.countDown();
36         }
37     });
38 }
39 @Override
Run GWayGenerateRouteWPSTest
GWayGenerateRouteWPSTest (com.devnull.gway.reposit 2s 204ms)
GWayGenerateRouteWPSTest.makeCall 2s 204ms

```

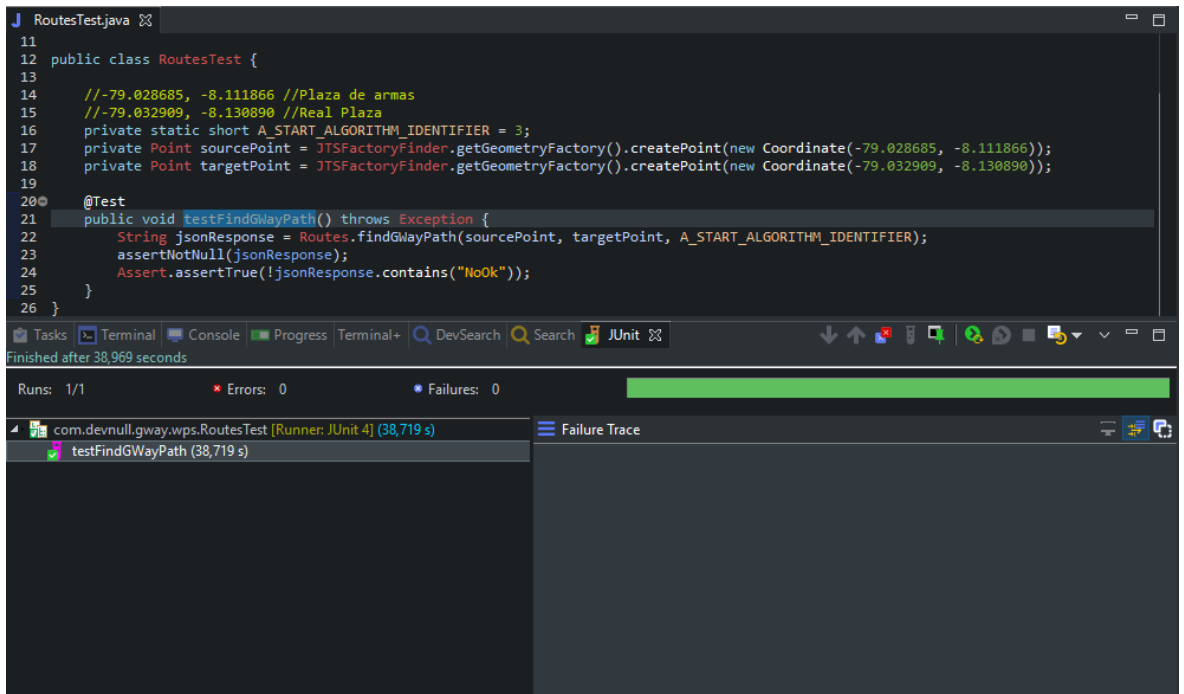
Prueba Funcional – CU Generar Ruta Óptima – Móvil

Figura 37. Pruebas Funcional UC – Generar Ruta Óptima - Móvil. (Fuente Propia).



Pruebas Unitarias - CU Generar Ruta Óptima – Geoserver

Figura 38. Pruebas Funcional UC – Generar Ruta Óptima - Geoserver. (Fuente Propia).



```
11
12 public class RoutesTest {
13
14     //-79.028685, -8.111866 //Plaza de armas
15     //-79.032909, -8.130890 //Real Plaza
16     private static short A_START_ALGORITHM_IDENTIFIER = 3;
17     private Point sourcePoint = JTSFactoryFinder.getGeometryFactory().createPoint(new Coordinate(-79.028685, -8.111866));
18     private Point targetPoint = JTSFactoryFinder.getGeometryFactory().createPoint(new Coordinate(-79.032909, -8.130890));
19
20     @Test
21     public void testFindGWayPath() throws Exception {
22         String jsonResponse = Routes.findGWayPath(sourcePoint, targetPoint, A_START_ALGORITHM_IDENTIFIER);
23         assertNotNull(jsonResponse);
24         Assert.assertTrue(!jsonResponse.contains("NoOk"));
25     }
26 }
```

Finished after 38,969 seconds

Runs: 1/1 * Errors: 0 • Failures: 0

com.devnull.gway.wps.RoutesTest [Runner: JUnit 4] (38,719 s)

testFindGWayPath (38,719 s)

Prueba Funcional – CU Generar Ruta Óptima – Geoserver

Figura 39. Pruebas Funcional UC – Generar Ruta Óptima - Geoserver. *(Fuente Propia).*

Generador de consultas WPS paso a paso.

Elija proceso

Find the shorest path between two places ([Proces de descripción WPS](#))

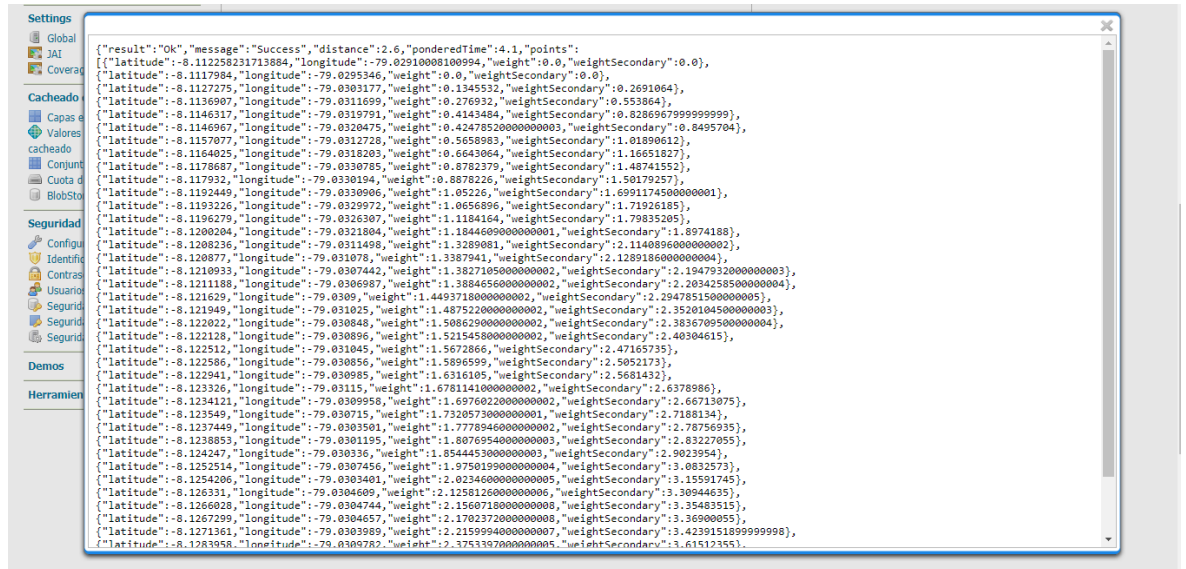
Entradas para el proceso

SourcePoint* - Point
Source Geometry

TargetPoint* - Point
Target Geometry

Algorithm* - Short
Algorithm

Figura 40. Pruebas Funcional UC – Generar Ruta Óptima - Geoserver. (Fuente Propia).



```

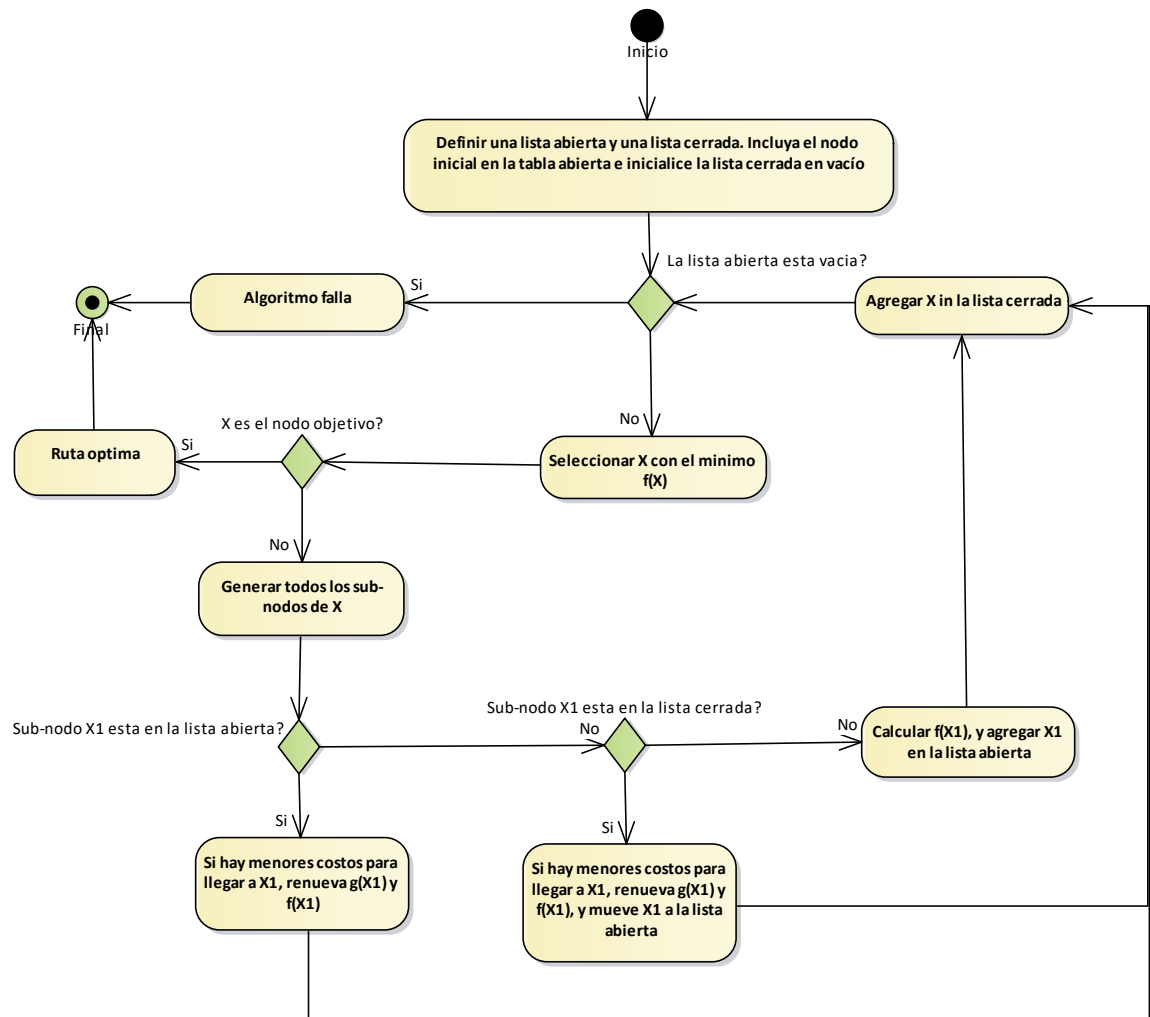
{
  "result": "Ok",
  "message": "Success",
  "distance": 2.6,
  "ponderedTime": 4.1,
  "points": [
    {
      "latitude": -8.11258231713884,
      "longitude": -79.02910088100994,
      "weight": 0.0,
      "weightSecondary": 0.0,
    },
    {
      "latitude": -8.1117984,
      "longitude": -79.0295346,
      "weight": 0.0,
      "weightSecondary": 0.0,
    },
    {
      "latitude": -8.1127275,
      "longitude": -79.0303177,
      "weight": 0.1345532,
      "weightSecondary": 0.2691064,
    },
    {
      "latitude": -8.1136907,
      "longitude": -79.0311699,
      "weight": 0.276932,
      "weightSecondary": 0.553864,
    },
    {
      "latitude": -8.1146317,
      "longitude": -79.0319791,
      "weight": 0.4143484,
      "weightSecondary": 0.8286967999999999,
    },
    {
      "latitude": -8.1146967,
      "longitude": -79.0320475,
      "weight": 0.42478520000000003,
      "weightSecondary": 0.8495704,
    },
    {
      "latitude": -8.1157077,
      "longitude": -79.0312728,
      "weight": 0.5658983,
      "weightSecondary": 1.01890612,
    },
    {
      "latitude": -8.1164025,
      "longitude": -79.0318203,
      "weight": 0.6643064,
      "weightSecondary": 1.16651827,
    },
    {
      "latitude": -8.1178687,
      "longitude": -79.0330785,
      "weight": 0.8782379,
      "weightSecondary": 1.48741552,
    },
    {
      "latitude": -8.117932,
      "longitude": -79.0330194,
      "weight": 0.8878226,
      "weightSecondary": 1.58179257,
    },
    {
      "latitude": -8.1192449,
      "longitude": -79.0330906,
      "weight": 1.05226,
      "weightSecondary": 1.6991174500000001,
    },
    {
      "latitude": -8.1193226,
      "longitude": -79.0329972,
      "weight": 1.0656896,
      "weightSecondary": 1.71926185,
    },
    {
      "latitude": -8.1196279,
      "longitude": -79.0326307,
      "weight": 1.1184164,
      "weightSecondary": 1.79835205,
    },
    {
      "latitude": -8.1200204,
      "longitude": -79.0321804,
      "weight": 1.1844609000000001,
      "weightSecondary": 1.8974188,
    },
    {
      "latitude": -8.1208236,
      "longitude": -79.0311498,
      "weight": 1.3289081,
      "weightSecondary": 2.1140896000000002,
    },
    {
      "latitude": -8.120877,
      "longitude": -79.031078,
      "weight": 1.3387943,
      "weightSecondary": 2.1289186000000004,
    },
    {
      "latitude": -8.1210933,
      "longitude": -79.0307442,
      "weight": 1.3827185000000002,
      "weightSecondary": 2.1947932000000003,
    },
    {
      "latitude": -8.1211188,
      "longitude": -79.0306987,
      "weight": 1.3884656000000002,
      "weightSecondary": 2.2034258500000004,
    },
    {
      "latitude": -8.121629,
      "longitude": -79.0309,
      "weight": 1.4493718000000002,
      "weightSecondary": 2.2947851500000005,
    },
    {
      "latitude": -8.121949,
      "longitude": -79.031025,
      "weight": 1.4875220000000002,
      "weightSecondary": 2.3520104500000003,
    },
    {
      "latitude": -8.122022,
      "longitude": -79.030848,
      "weight": 1.5086290000000002,
      "weightSecondary": 2.3836799500000004,
    },
    {
      "latitude": -8.122128,
      "longitude": -79.030896,
      "weight": 1.5215458000000002,
      "weightSecondary": 2.48904615,
    },
    {
      "latitude": -8.122512,
      "longitude": -79.031045,
      "weight": 1.5672866,
      "weightSecondary": 2.47165735,
    },
    {
      "latitude": -8.122586,
      "longitude": -79.030856,
      "weight": 1.5896599,
      "weightSecondary": 2.5052173,
    },
    {
      "latitude": -8.122941,
      "longitude": -79.030985,
      "weight": 1.6316109,
      "weightSecondary": 2.5681432,
    },
    {
      "latitude": -8.123326,
      "longitude": -79.03115,
      "weight": 1.6781141000000002,
      "weightSecondary": 2.6378906,
    },
    {
      "latitude": -8.123412,
      "longitude": -79.030958,
      "weight": 1.6976022000000002,
      "weightSecondary": 2.66713075,
    },
    {
      "latitude": -8.123549,
      "longitude": -79.030715,
      "weight": 1.7320573000000001,
      "weightSecondary": 2.718134,
    },
    {
      "latitude": -8.1237449,
      "longitude": -79.0303501,
      "weight": 1.7778946000000002,
      "weightSecondary": 2.78756935,
    },
    {
      "latitude": -8.1238853,
      "longitude": -79.0301195,
      "weight": 1.8076954000000003,
      "weightSecondary": 2.83227055,
    },
    {
      "latitude": -8.124247,
      "longitude": -79.030336,
      "weight": 1.8544453000000003,
      "weightSecondary": 2.9023954,
    },
    {
      "latitude": -8.1252514,
      "longitude": -79.0307456,
      "weight": 1.9750199000000004,
      "weightSecondary": 3.0832573,
    },
    {
      "latitude": -8.1254206,
      "longitude": -79.0303401,
      "weight": 2.0234600000000005,
      "weightSecondary": 3.15591745,
    },
    {
      "latitude": -8.126331,
      "longitude": -79.0304609,
      "weight": 2.1258126000000006,
      "weightSecondary": 3.30944635,
    },
    {
      "latitude": -8.1266028,
      "longitude": -79.0304744,
      "weight": 2.1560718000000008,
      "weightSecondary": 3.35483515,
    },
    {
      "latitude": -8.1267299,
      "longitude": -79.0304657,
      "weight": 2.1702372000000008,
      "weightSecondary": 3.36900055,
    },
    {
      "latitude": -8.1271361,
      "longitude": -79.0303989,
      "weight": 2.2159940000000007,
      "weightSecondary": 3.4239151899999998,
    },
    {
      "latitude": -8.1283858,
      "longitude": -79.0300782,
      "weight": 2.3763307000000005,
      "weightSecondary": 3.61613363,
    }
  ]
}

```

4.7.3. Iteración 03

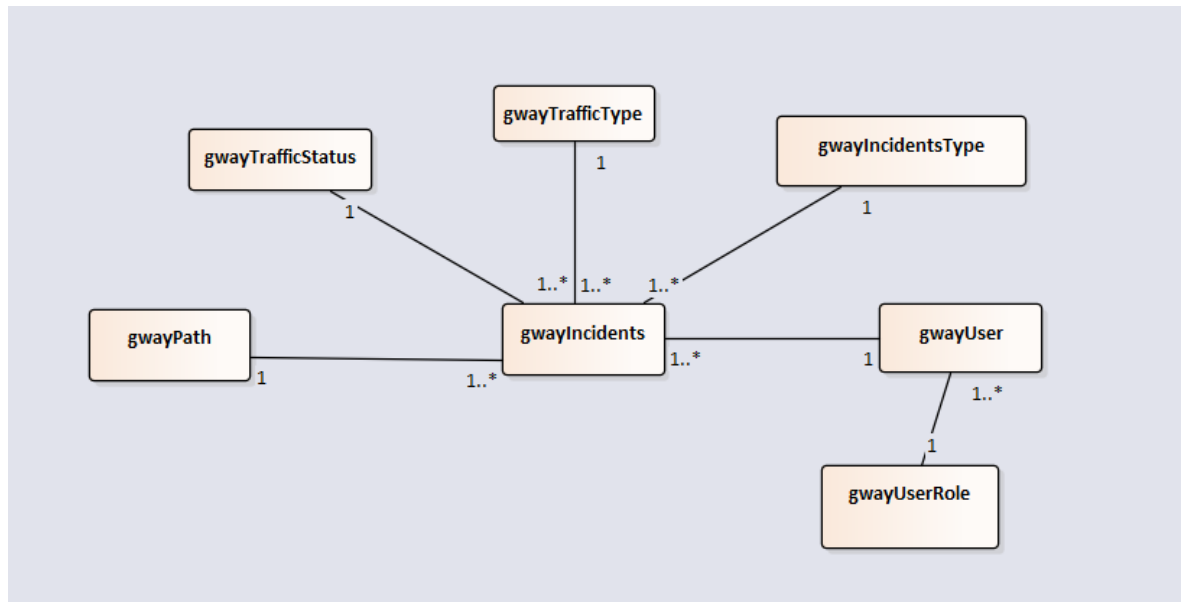
4.7.3.1. Diagrama de Flujo del Algoritmo A*

Figura 41. Diagrama de flujo – Algoritmo A*. (Fuente Propia).



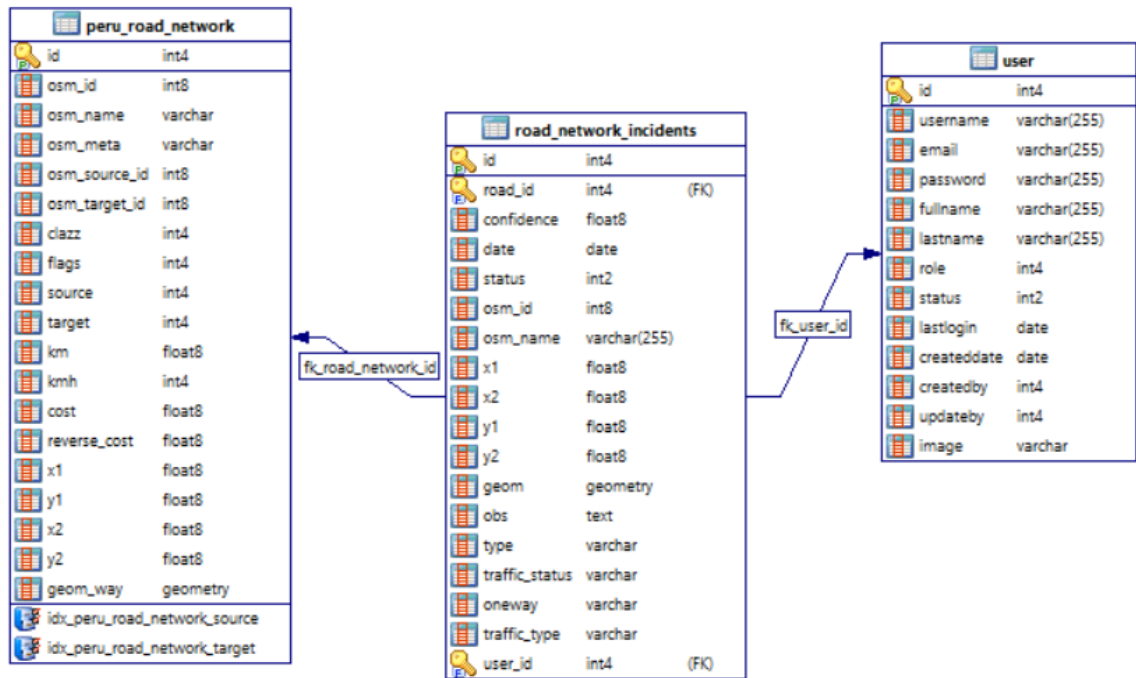
4.7.3.2. Modelo de Datos

Figura 42. Diagrama de Modelo de Datos. *(Fuente Propia).*



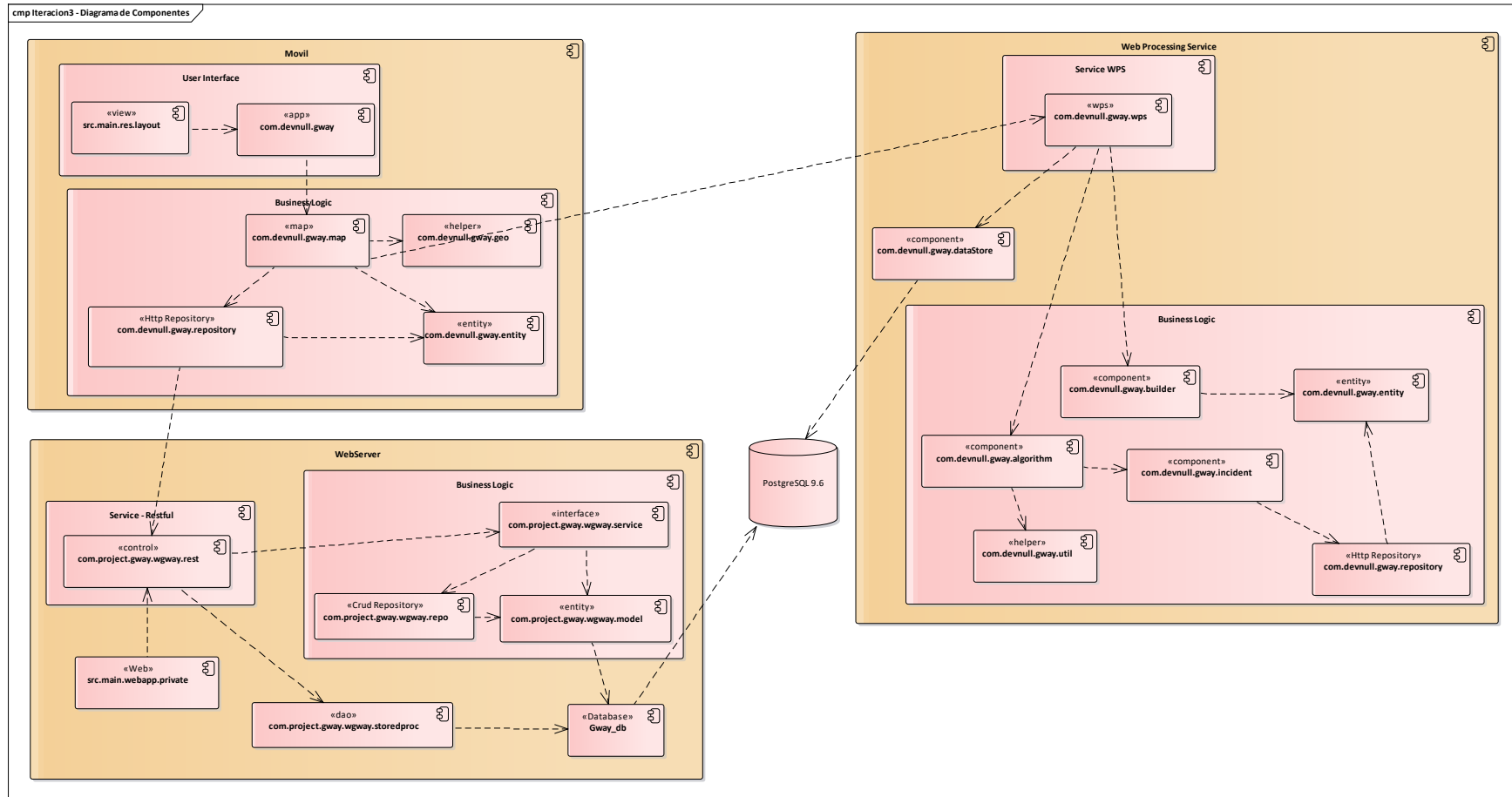
4.7.3.3. Base de Datos

Figura 43. Diagrama de Base de Datos. (Fuente Propia).



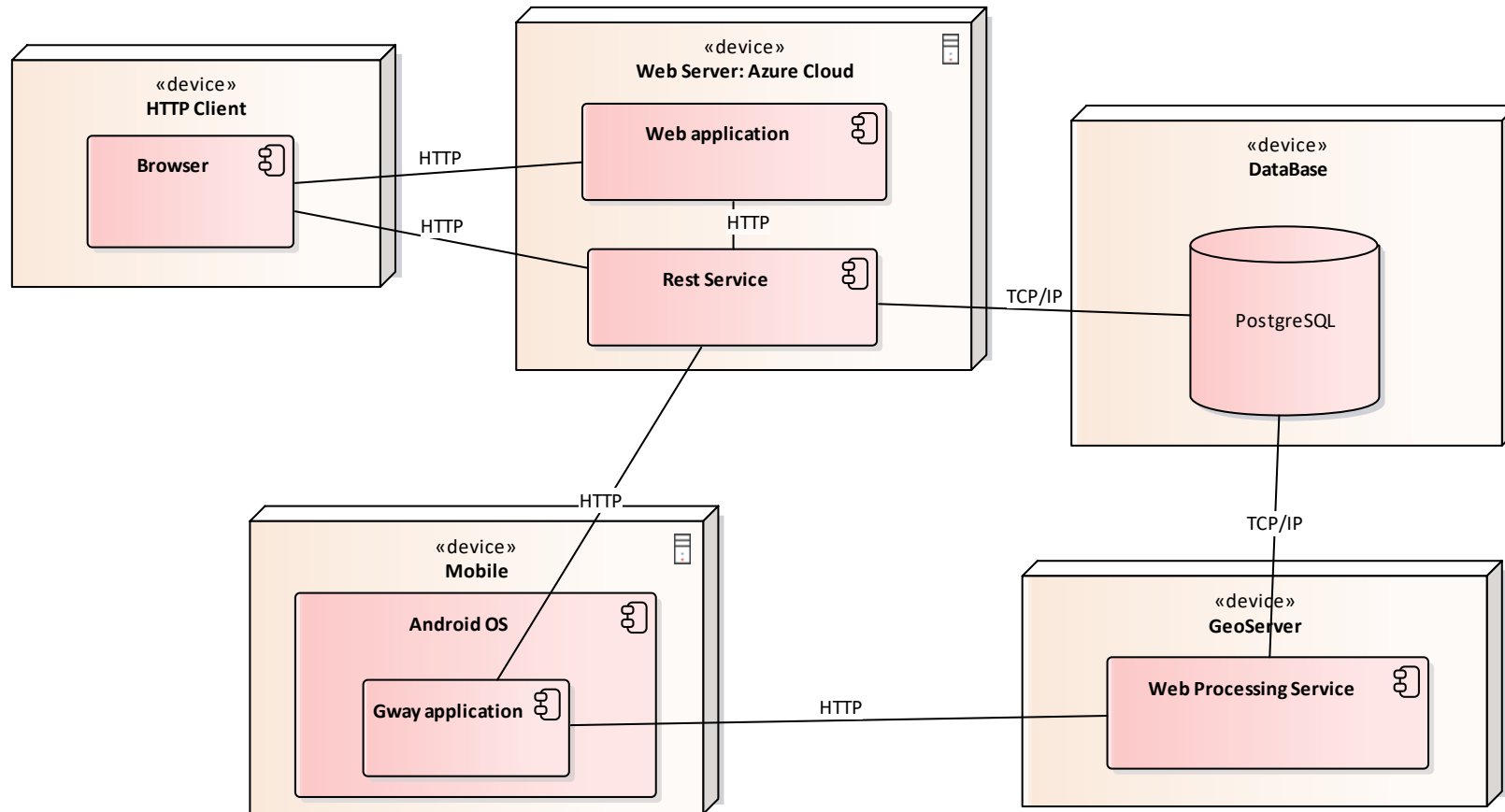
4.7.3.5. Modelo de Componentes

Figura 44. Diagrama de Componentes. (Fuente Propia).



4.7.3.6. Diagrama de Despliegue

Figura 45. Diagrama de Despliegue. *(Fuente Propia).*



CAPÍTULO 5. METODOLOGÍA

5.1. Diseño de investigación

Según el propósito: Investigación aplicada.

Según el diseño: Investigación Experimental.

SIMBOLO	DESCRIPCION
M1	Medición en el desplazamiento de los conductores de vehículos antes de la aplicación móvil.
X	Aplicación móvil basada en algoritmos de rutas óptimas.
M2	Medición en el desplazamiento de los conductores de vehículos después de la aplicación móvil.



5.2. Unidad de estudio

Conductor de transporte privado de la ciudad de Trujillo.

5.3. Población

Está conformado por 33 conductores de taxis que se escogió aleatoriamente en la ciudad de Trujillo.

5.4. Muestra (muestreo o selección)

Por el tamaño de la población la muestra será igual a la población.

5.5. Técnicas, instrumentos y procedimientos de recolección de datos

Tabla 24. Cuadro detallado de técnicas, instrumentos y procedimientos de recolección de datos

DIMENSIONES	INDICADORES	TECNICAS	INSTRUMENTOS
Rutas óptimas	1. Distancia recorrida. 2. Tiempo en recorrido. 3. Número de incidencias evitadas.	Encuesta	Cuestionario

5.6. Métodos, instrumentos y procedimientos de análisis de datos

Tabla 25. Cuadro detallado de métodos, instrumentos y procedimientos de análisis de datos

DIMENSIONES	INDICADORES	TECNICAS	INSTRUMENTOS	PROCEDIMIENTO
Eficiencia	1. Tiempo de respuesta.	TE = Tiempo de espera TR = Tiempo de respuesta $TR = \frac{(TE + Tn)}{n}$	Métrica de Calidad	Se ejecutará una tarea específica. Se mide el tiempo que toma la muestra en completar la operación.
Funcionalidad	2. Precisión esperada.	TP = Total de Pruebas TE = Total de Errores $\frac{TP}{100\%} = \frac{TE}{x}$	Métrica de Calidad	Se hará casos de Test de entrada vs Test de salida y luego se compara las salidas con los resultados esperados.
Rutas óptimas	3. Distancia recorrida. 4. Tiempo en recorrido. 5. Número de incidencias evitadas.	MA: Medición Antes MD: Medición Después D: Diferencia entre MA y MD $D = MA - MD$	Cuestionario	La encuesta se hará a los conductores de la ciudad de Trujillo con una serie de preguntas acerca de los indicadores ya propuestos.

CAPÍTULO 6. RESULTADOS

6.1. Contrastación

6.1.1. Prueba de hipótesis

La Contrastación de Hipótesis se ha realizado de acuerdo al método propuesto Pre Test – Post Test, para poder aceptar o rechazar la hipótesis. Así mismo para la realización de este diseño se identificaron indicadores cuantitativos, los cuales se describen a continuación:

Tabla 26. *Tipos de Indicadores.*

N°	VARIABLE	INDICADORES	TIPO
1	Dependiente	Tiempo en recorrido.	Cuantitativo.
2		Distancia recorrida.	Cuantitativo.
3		Número de incidencias evitadas.	Cuantitativo.

6.1.2. Prueba de hipótesis Indicador Dependiente

6.1.2.1. Prueba de hipótesis para indicador 1: Cuantitativo

Tiempo en recorrido.

a) Definición de Variables

TDR_a = Tiempo de desplazamiento en ruta antes.

TDR_p = Tiempo de desplazamiento en ruta con la aplicación propuesta.

b) Hipótesis estadística

Hipótesis Ho = Tiempo de desplazamiento en ruta antes es menor o igual que el Tiempo de desplazamiento en ruta con la aplicación propuesta. (Minutos)

$$H_0 = TDR_a - TDR_p \leq 0$$

Hipótesis Ha = Tiempo de desplazamiento en ruta antes es mayor que el Tiempo de desplazamiento en ruta con la aplicación propuesta. (Minutos)

$$H_0 = TDR_a - TDR_p > 0$$

c) Nivel de Significancia

Se define el margen de error, confiabilidad **95%**.

Usando un nivel de significancia ($\alpha = 0.05$) del 5%. Por lo tanto el nivel de confianza ($1 - \alpha = 0.95$) será del 95%

d) Estadígrafo de contraste

Puesto que $n = 33$ es mayor que 30 usaremos la distribución estándar normal (Z)

$$Zc = \frac{(\bar{X}_a - \bar{X}_d)}{\sqrt{\left(\frac{\sigma_a^2}{n_a} + \frac{\sigma_s^2}{n_s}\right)}}$$

Resultados: Para calcular el Tiempo en recorrido de desplazamiento se ha estimado un universo de 33 conductores.

Tabla 27. PreTest – PostTest Tiempo de Recorrido.

N°	Antes	Después	Antes	Después	Antes	Después
	TDRai	TDRpi	TDRai - TDRa	TDRpi - TDRp	(TDRai - TDRa) ²	(TDRpi - TDRp) ²
1	88	62	-9,94	-2,28	98,85	5,22
2	88	62	-9,94	-2,28	98,85	5,22
3	88	62	-9,94	-2,28	98,85	5,22
4	88	62	-9,94	-2,28	98,85	5,22
5	88	62	-9,94	-2,28	98,85	5,22
6	88	62	-9,94	-2,28	98,85	5,22
7	88	62	-9,94	-2,28	98,85	5,22
8	88	62	-9,94	-2,28	98,85	5,22
9	89	63	-8,94	-1,38	79,97	1,92
10	94	63	-3,94	-1,78	15,54	3,19
11	94	63	-3,94	-1,78	15,54	3,19
12	94	63	-3,94	-1,78	15,54	3,19
13	94	63	-3,94	-1,78	15,54	3,19
14	94	63	-3,94	-1,78	15,54	3,19
15	108	63	9,96	-1,28	99,15	1,65
16	109	64	10,96	-1,08	120,07	1,18
17	111	65	12,96	0,32	167,90	0,10
18	111	65	12,96	0,32	167,90	0,10
19	111	65	12,96	0,32	167,90	0,10
20	111	65	12,96	0,32	167,90	0,10
21	111	65	12,96	0,32	167,90	0,10

22	112	66	13,96	1,62	194,81	2,61
23	112	66	13,96	1,62	194,81	2,61
24	105	67	6,96	2,62	48,41	6,84
25	105	67	6,96	2,62	48,41	6,84
26	105	67	6,96	2,62	48,41	6,84
27	96	67	-2,04	2,62	4,17	6,84
28	98	67	-0,04	2,62	0,00	6,84
29	92	67	-6,04	2,62	36,51	6,84
30	95	67	-3,04	2,62	9,26	6,84
31	93	67	-5,04	2,62	25,43	6,84
32	93	67	-5,04	2,62	25,43	6,84
33	93	67	-5,04	2,62	25,43	6,84
Sumatoria	3235	2135			2668,28	136,54
Promedio	98,04	64,68				
Varianza					83,38	4,27

Promedio:

$$TP = \frac{\sum_{i=1}^n t_i}{n}$$

$$\overline{TDRa} = \frac{\sum_{i=1}^n TDR_{ai}}{n_a} = \frac{3235}{33} = 98,04$$

$$\overline{TDRp} = \frac{\sum_{i=1}^n TDR_{si}}{n_s} = \frac{2135}{33} = 64,68$$

Varianza:

$$\sigma_a^2 = \frac{\sum_{i=1}^n (TDR_{ai} - \overline{TDRa})^2}{n_a} = \frac{2668,28}{33} = 83,38$$

$$\sigma_s^2 = \frac{\sum_{i=1}^n (TDR_{pi} - \overline{TDRp})^2}{n_s} = \frac{136,54}{33} = 4,27$$

Cálculo de Z:

$$Z_c = \frac{(\overline{TDRa} - \overline{TDRp})}{\sqrt{\left(\frac{\sigma_a^2}{n_a} + \frac{\sigma_s^2}{n_s}\right)}}$$

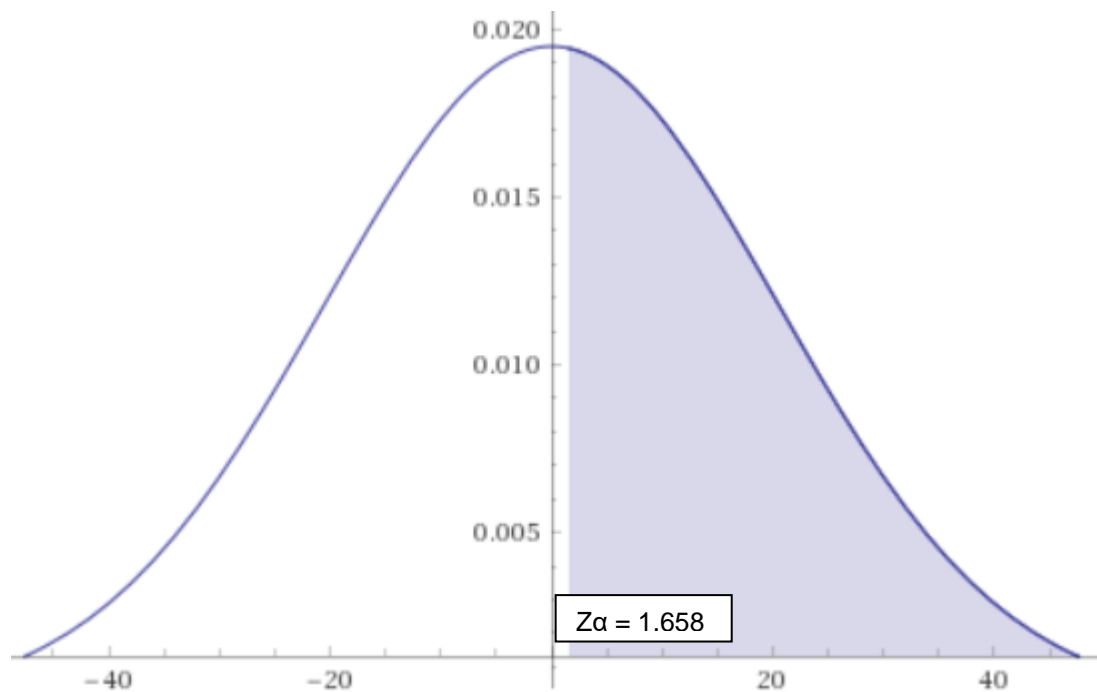
$$Z_c = \frac{(98,04 - 64,68)}{1,629} = 20,467$$

e) Región crítica

Para $\alpha = 0.05$ (Anexo – 05) encontramos $Z\alpha = 1.658$ entonces la región crítica de la prueba es $Z_{tab} = < 1.658 >$.

f) Conclusión

Figura 46. Gráfico de Distribución Z. (*Fuente Propia*).



Puesto que $Z_c = 20,467$ calculado, es mayor que $Z_\alpha = 1.658$ y estando este valor dentro de la región de rechazo $< 1.658 >$, entonces se rechaza H_0 y por consiguiente se acepta H_a . Se concluye entonces que el Tiempo promedio de desplazamiento en ruta óptima es menor con la aplicación propuesta teniendo un nivel de error del **5%** y un nivel de confianza del **95%**.

6.1.2.2. Prueba de hipótesis para indicador 2: Cuantitativo

Distancia recorrida.

a) Definición de Variables

DDRa = Distancia de desplazamiento en ruta antes.

DDRp = Distancia de desplazamiento en ruta con la aplicación propuesta.

b) Hipótesis estadística

Hipótesis Ho = Distancia de desplazamiento en ruta antes es menor o igual que la Distancia de desplazamiento en ruta con la aplicación propuesta. (Km)

$$H_0 = DDR_a - DDR_p \leq 0$$

Hipótesis Ha = Distancia de desplazamiento en ruta antes es mayor que el Distancia de desplazamiento en ruta con la aplicación propuesta. (Km)

$$H_0 = DDR_a - DDR_p > 0$$

c) Nivel de Significancia

Se define el margen de error, confiabilidad **95%**.

Usando un nivel de significancia ($\alpha = 0.05$) del 5%. Por lo tanto el nivel de confianza ($1 - \alpha = 0.95$) será del 95%

d) Estadígrafo de contraste

Puesto que $n = 33$ es mayor que 30 usaremos la distribución estándar normal (Z)

$$Z_c = \frac{(\bar{X}_a - \bar{X}_d)}{\sqrt{\left(\frac{\sigma_a^2}{n_a} + \frac{\sigma_s^2}{n_s}\right)}}$$

Resultados: Para calcular la Distancia de desplazamiento se ha estimado un universo de 33 conductores.

Tabla 28. PreTest – PostTest Distancia en recorrido.

N°	Antes	Después	Antes	Después	Antes	Después
	DDRa _i	DDR _p _i	DDRa _i - DDRa	DDR _p _i - DDR _p	(DDRa _i - DDRa) ^ 2	(DDR _p _i - DDR _p) ^ 2
1	24,50	22,80	-0,96	-0,27	0,93	0,07
2	24,50	22,80	-0,96	-0,27	0,93	0,07
3	24,50	22,80	-0,96	-0,27	0,93	0,07

4	24,50	22,80	-0,96	-0,27	0,93	0,07
5	24,50	22,80	-0,96	-0,27	0,93	0,07
6	24,50	22,80	-0,96	-0,27	0,93	0,07
7	24,50	22,80	-0,96	-0,27	0,93	0,07
8	24,50	22,80	-0,96	-0,27	0,93	0,07
9	24,50	22,90	-0,96	-0,17	0,93	0,03
10	24,40	23,00	-1,06	-0,07	1,13	0,00
11	24,40	23,00	-1,06	-0,07	1,13	0,00
12	24,40	23,00	-1,06	-0,07	1,13	0,00
13	24,40	23,00	-1,06	-0,07	1,13	0,00
14	24,40	23,00	-1,06	-0,07	1,13	0,00
15	25,80	23,30	0,34	0,23	0,11	0,05
16	25,50	23,10	0,04	0,03	0,00	0,00
17	26,00	23,10	0,54	0,03	0,29	0,00
18	26,00	23,10	0,54	0,03	0,29	0,00
19	26,00	23,10	0,54	0,03	0,29	0,00
20	26,00	23,10	0,54	0,03	0,29	0,00
21	26,00	23,10	0,54	0,03	0,29	0,00
22	26,10	23,00	0,64	-0,07	0,40	0,00
23	26,10	23,00	0,64	-0,07	0,40	0,00
24	26,00	23,30	0,54	0,23	0,29	0,05
25	26,10	23,30	0,64	0,23	0,40	0,05
26	26,10	23,30	0,64	0,23	0,40	0,05
27	26,20	23,30	0,74	0,23	0,54	0,05
28	26,80	23,30	1,34	0,23	1,79	0,05
29	26,50	23,30	1,04	0,23	1,07	0,05
30	26,80	23,30	1,34	0,23	1,79	0,05
31	26,60	23,30	1,14	0,23	1,29	0,05
32	26,60	23,30	1,14	0,23	1,29	0,05
33	26,60	23,30	1,14	0,23	1,29	0,05
Sumatoria	840,30	761,20			26,54	1,23
Promedio	25,46	23,07				
Varianza					0,80	0,04

Promedio:

$$TP = \frac{\sum_{i=1}^n t_i}{n}$$

$$\overline{TDRa} = \frac{\sum_{i=1}^n DDR_{ai}}{n_a} = \frac{840,30}{33} = 25,46$$

$$\overline{TDRp} = \frac{\sum_{i=1}^n DDR_{si}}{n_s} = \frac{761,20}{33} = 23,07$$

Varianza:

$$\sigma_a^2 = \frac{\sum_{i=1}^n (DDR_{ai} - \overline{DDR_a})^2}{n_a} = \frac{26,54}{33} = 0,80$$

$$\sigma_s^2 = \frac{\sum_{i=1}^n (DDR_{pi} - \overline{DDR_p})^2}{n_s} = \frac{1,23}{33} = 0,04$$

Cálculo de Z:

$$Z_c = \frac{(\overline{DDR_a} - \overline{DDR_p})}{\sqrt{\left(\frac{\sigma_a^2}{n_a} + \frac{\sigma_s^2}{n_s}\right)}}$$

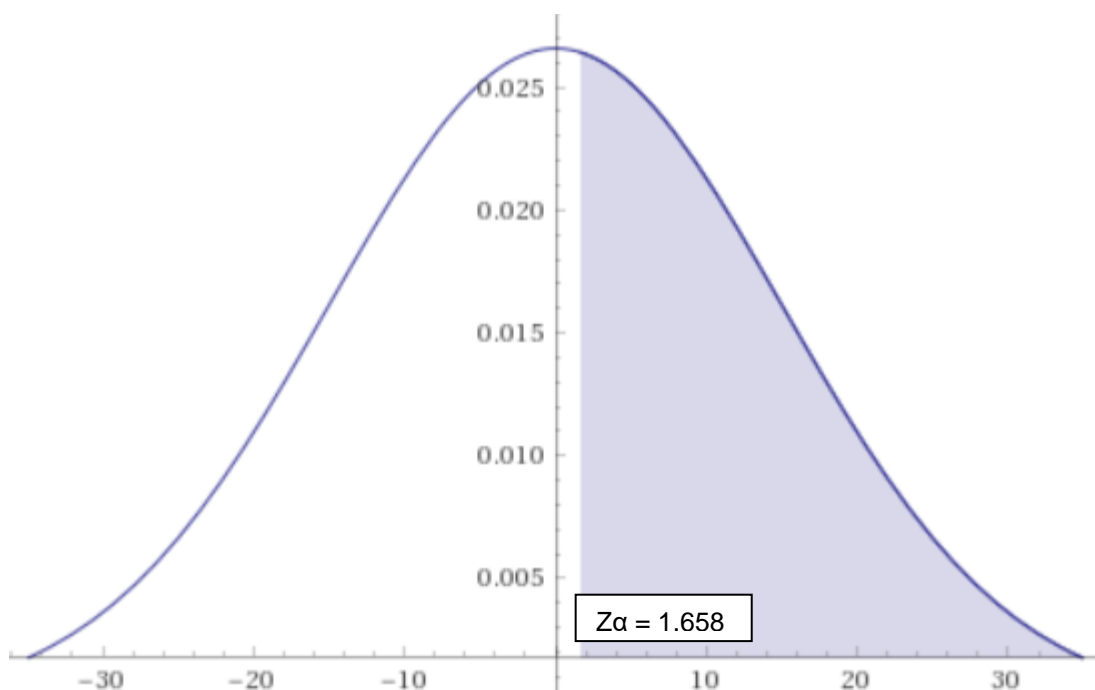
$$Z_c = \frac{(25,46 - 23,07)}{0,15968} = 15,01$$

e) Región crítica

Para $\alpha = 0.05$ (Anexo – 05) encontramos $Z\alpha = 1.658$ entonces la región crítica de la prueba es $Z_{tab} = < 1.658 >$.

f) Conclusión

Figura 47. Gráfico de Distribución Z. (Fuente Propia).



Puesto que $Zc = 15,01$ calculado, es mayor que $Z\alpha = 1.658$ y estando este valor dentro de la región de rechazo $< 1.658 >$, entonces se rechaza H_0 y por consiguiente se acepta H_a . Se concluye entonces que la Distancia promedio de desplazamiento en ruta óptima es menor con la aplicación propuesta teniendo un nivel de error del **5%** y un nivel de confianza del **95%**.

6.1.2.3. Prueba de hipótesis para indicador 3: Cuantitativo

Número de incidencias evitadas.

g) Definición de Variables

NIEPCa = Número de incidencias evitadas por conductor antes.

NIEPCp = Número de incidencias evitadas por conductor usando la aplicación propuesta.

h) Hipótesis estadística

Hipótesis H_0 = Número de incidencias evitadas por conductor antes es mayor o igual que el Número de incidencias evitadas por conductor usando la aplicación propuesta.

$$H_0 = NIEPC_a - NIEPC_p \geq 0$$

Hipótesis H_a = Número de incidencias evitadas por conductor antes es menor que el Número de incidencias evitadas por conductor usando la aplicación propuesta.

$$H_0 = NIEPC_a - NIEPC_p < 0$$

i) Nivel de Significancia

Se define el margen de error, confiabilidad **95%**.

Usando un nivel de significancia ($\alpha = 0.05$) del 5%. Por lo tanto el nivel de confianza ($1 - \alpha = 0.95$) será del 95%

j) Estadígrafo de contraste

Puesto que $n = 33$ es mayor que 30 usaremos la distribución estándar normal (Z)

$$Zc = \frac{(\bar{X}_a - \bar{X}_d)}{\sqrt{\left(\frac{\sigma_a^2}{n_a} + \frac{\sigma_s^2}{n_s}\right)}}$$

Resultados: Para calcular el Número de incidencias por el conductor se ha estimado un universo de 33 conductores.

Tabla 29. PreTest – PostTest Número de incidencias por conductor.

N°	Antes	Después	Antes	Después	Antes	Después
	NIEPCai	NIEPCpi	NIEPCai - NIEPCa	NIEPCpi - NIEPCp	(NIEPCai - NIEPCa) ^ 2	(NIEPCpi - NIEPCp) ^ 2
1	14	40	-0,03	4,64	0,00	21,50
2	14	40	-0,03	4,64	0,00	21,50
3	14	40	-0,03	4,64	0,00	21,50
4	14	40	-0,03	4,64	0,00	21,50
5	14	40	-0,03	4,64	0,00	21,50
6	14	40	-0,03	4,64	0,00	21,50
7	14	40	-0,03	4,64	0,00	21,50
8	14	40	-0,03	4,64	0,00	21,50
9	13	36	-1,03	0,64	1,06	0,40
10	16	38	1,97	2,64	3,88	6,95
11	16	38	1,97	2,64	3,88	6,95
12	16	38	1,97	2,64	3,88	6,95
13	16	38	1,97	2,64	3,88	6,95
14	16	38	1,97	2,64	3,88	6,95
15	13	34	-1,03	-1,36	1,06	1,86
16	11	34	-3,03	-1,36	9,18	1,86
17	13	34	-1,03	-1,36	1,06	1,86
18	13	34	-1,03	-1,36	1,06	1,86
19	13	34	-1,03	-1,36	1,06	1,86
20	13	34	-1,03	-1,36	1,06	1,86
21	13	34	-1,03	-1,36	1,06	1,86
22	14	35	-0,03	-0,36	0,00	0,13
23	14	35	-0,03	-0,36	0,00	0,13
24	14	34	-0,03	-1,36	0,00	1,86
25	14	34	-0,03	-1,36	0,00	1,86
26	14	34	-0,03	-1,36	0,00	1,86
27	15	31	0,97	-4,36	0,94	19,04
28	14	31	-0,03	-4,36	0,00	19,04
29	13	31	-1,03	-4,36	1,06	19,04
30	15	31	0,97	-4,36	0,94	19,04
31	14	29	-0,03	-6,36	0,00	40,50
32	14	29	-0,03	-6,36	0,00	40,50
33	14	29	-0,03	-6,36	0,00	40,50
Sumatoria	463,00	1167,00			38,97	423,64
Promedio	14,03	35,36				
Varianza					1,18	12,84

Promedio:

$$TP = \frac{\sum_{i=1}^n t_i}{n}$$

$$\overline{TDRa} = \frac{\sum_{i=1}^n NIEPC_{ai}}{n_a} = \frac{463}{33} = 14,03$$

$$\overline{TDRp} = \frac{\sum_{i=1}^n NIEPC_{si}}{n_s} = \frac{1167}{33} = 35,36$$

Varianza:

$$\sigma_a^2 = \frac{\sum_{i=1}^n (NIEPC_{ai} - \overline{NIEPC_a})^2}{n_a} = \frac{38,97}{33} = 1,18$$

$$\sigma_s^2 = \frac{\sum_{i=1}^n (NIEPC_{pi} - \overline{NIEPC_p})^2}{n_s} = \frac{423,64}{33} = 12,84$$

Cálculo de Z:

$$Zc = \frac{(\overline{NIEPC} - \overline{NIEPC})}{\sqrt{\left(\frac{\sigma_a^2}{n_a} + \frac{\sigma_s^2}{n_s}\right)}}$$

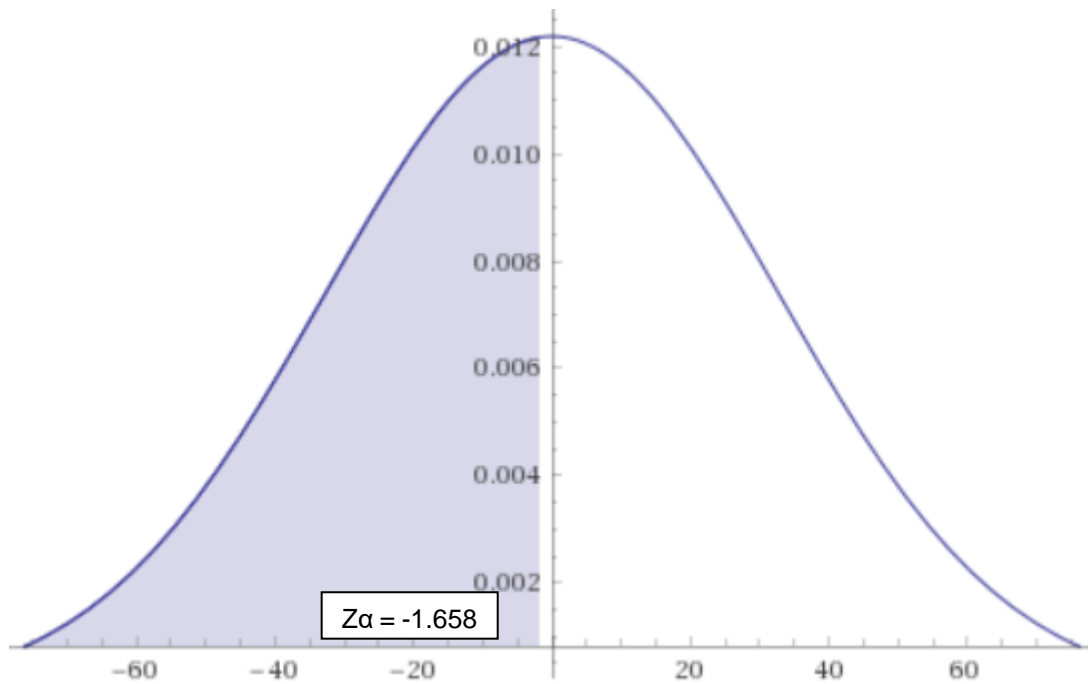
$$Zc = \frac{(1,18 - 12,84)}{0,6517} = -32,73$$

k) Región crítica

Para $\alpha = 0.05$ (Anexo – 05) encontramos $Z\alpha = -1.658$ entonces la región crítica de la prueba es $Z_{tab} = < -1.658 >$.

I) Conclusión

Figura 48. Gráfico de Distribución Z. (*Fuente Propia*).



Puesto que $Z_c = -32,73$ calculado, es menor que $Z_{\alpha} = -1.658$ y estando este valor dentro de la región de rechazo $< -1.658 >$, entonces se rechaza H_0 y por consiguiente se acepta H_a . Se concluye entonces que el Número de incidencias evitadas por conductor antes es menor que el Número de incidencias evitadas por conductor usando la aplicación propuesta teniendo un nivel de error del **5%** y un nivel de confianza del **95%**.

CAPÍTULO 7. DISCUSIÓN

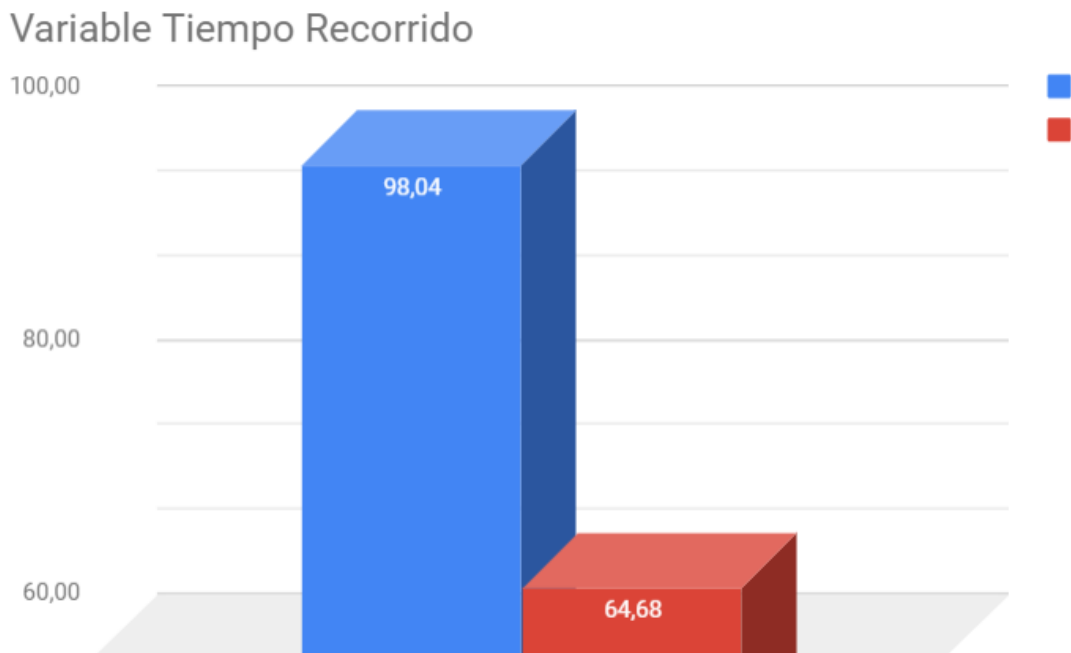
Esta investigación tiene como propósito de determinar el efecto en el desplazamiento de los conductores de vehículos privados en la ciudad de Trujillo, mediante el desarrollo de una aplicación móvil basada en algoritmos de rutas óptimas, luego de haber analizado el uso de la aplicación por cada conductor, se obtuvo lo siguiente.

7.1. Indicador 1 – Tiempo en recorrido

El resultado de la prueba de hipótesis para este indicador, concluye que el tiempo de desplazamiento en una ruta óptima del conductor usando la aplicación propuesta es menor que haciendo uso del criterio del conductor.

Según la tabla N° 27 vemos el contraste entre el pre test y post test, en la figura que se muestra a continuación podemos ver que el tiempo en recorrido promedio de desplazamiento según el criterio de los conductores que se evaluaron es 98,04 minutos y 64,68 minutos como tiempo en recorrido promedio de desplazamiento en la aplicación propuesta, lo que significa que reduce el tiempo de desplazamiento en 33.36 minutos lo que es equivalente al 34%.

Figura 49. Promedio de medición para el indicador Tiempo en recorrido. *(Fuente Propia).*

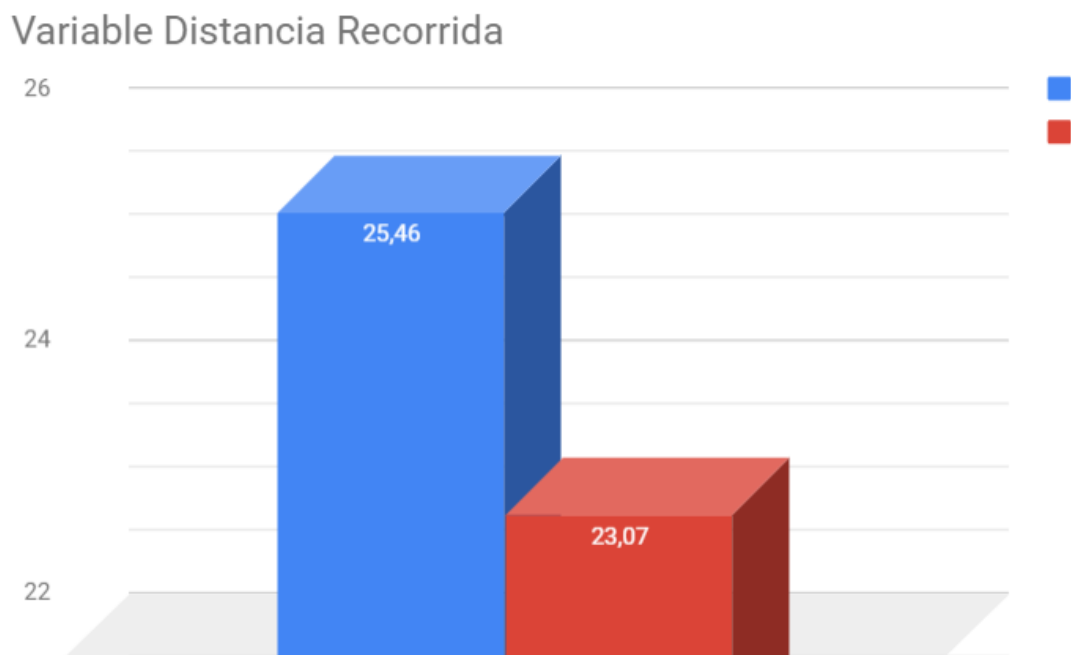


7.2. Indicador 2 – Distancia recorrida

El resultado de la prueba de hipótesis para este indicador, concluye que la distancia de desplazamiento en una ruta óptima del conductor usando la aplicación propuesta es menor que haciendo uso del criterio del conductor.

Según la tabla N° 28 vemos el contraste entre el pre test y post test, en la figura que se muestra a continuación podemos ver que el promedio de la distancia recorrida de desplazamiento según el criterio de los conductores que se evaluaron es de 25,46 km y 23,07 como distancia recorrida promedio de desplazamiento en la aplicación propuesta, lo que significa que reduce la distancia de desplazamiento en 2.39 km lo que es equivalente al 8.6%.

Figura 50. Promedio de medición para el indicador Distancia recorrida. *(Fuente Propia).*

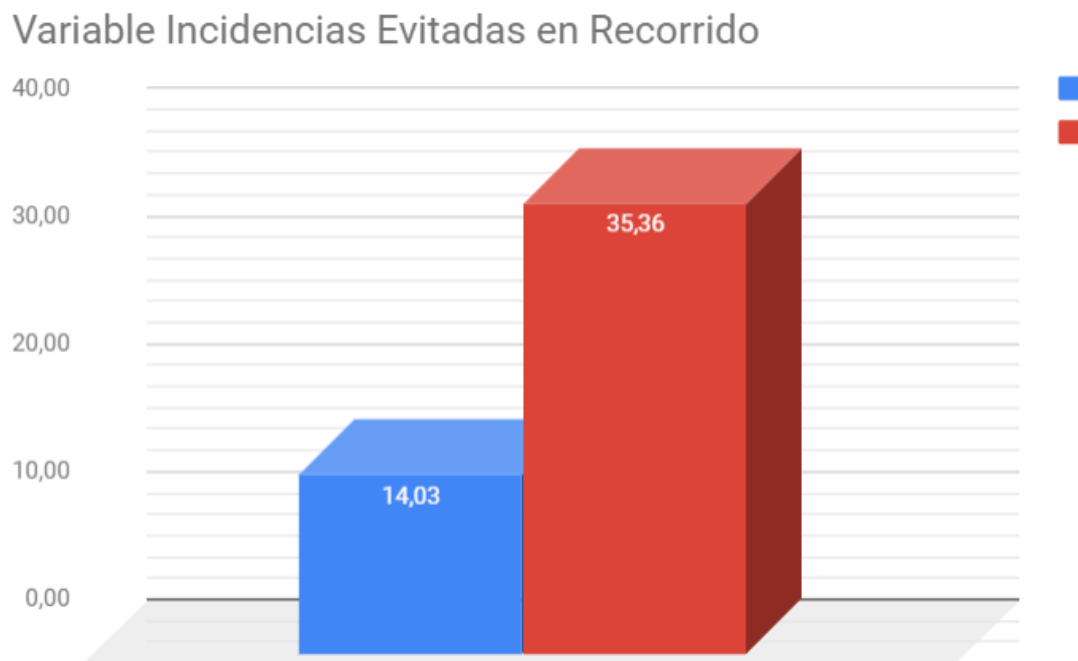


7.3. Indicador 3 – Número de incidencias evitadas

El resultado de la prueba de hipótesis para este indicador, concluye que el número de incidencias evitadas por el conductor usando la aplicación propuesta es mayor que haciendo uso del criterio del conductor.

Según la tabla N° 29 vemos el contraste entre el pre test y post test en la figura que se muestra a continuación podemos ver que el promedio de incidencias evitadas por los conductores que se evaluaron es de 14,03 y 35,36 para el método propuesto, lo que significa que aumenta el número de incidencias evitadas en 21,33 lo que es equivalente al 60,32%.

Figura 51. Promedio de medición para el indicador Número de incidencias evitadas. *(Fuente Propia).*



CONCLUSIONES

Al finalizar este trabajo de investigación, se llegan a las siguientes conclusiones:

- La aplicación propuesta a través de algoritmos de rutas óptimas mejoró los desplazamientos de los conductores en contraste a los desplazamientos basados en sus criterios.
- Se demostró que el efecto de un aplicativo móvil basado en rutas óptimas sobre el tiempo de recorrido promedio de desplazamiento es positivo logrando una notable mejora en la reducción de minutos.
- Se demostró que el efecto de un aplicativo móvil basado en rutas óptimas sobre la distancia de recorrido promedio de desplazamiento es positivo logrando una mejora en la reducción de kilómetros.
- Se demostró que a través de un aplicativo móvil basado en rutas óptimas sobre el número de incidencias evitadas es positivo ya que evito un amplio rango de incidencias en las vías de la ciudad de Trujillo.

RECOMENDACIONES

Se recomienda que el actor administre constantemente las incidencias ocurridas (tráfico, accidentes, obras) en tiempo real para tener una fuente de datos más confiable en la ciudad de Trujillo.

Se recomienda implementar los algoritmos en lenguaje de programación de bajo nivel como c++ para obtener un mejor tiempo de respuesta en el procesamiento.

Contar con herramientas para la gestión de datos geoespaciales (QGIS, ArcGIS, etc.) para su actualización constante y proporcionar un mejor servicio.

Crear un sistema estadístico para la toma de decisiones en el planeamiento de nuevos proyectos en la infraestructura vial.

REFERENCIAS

- Aguado Aranda A. Jiménez de Vega J. (2013). Optimización de rutas de transporte. *Optimización de rutas de transporte*. Madrid, Madrid. Recuperado el 12 de Setiembre de 2017
- Aguayo Moreno, E. M. (Mayo de 2008). Desarrollo de un sistema de localización de rutas óptimas entre dos puntos geográficos. *Desarrollo de un sistema de localización de rutas óptimas entre dos puntos geográficos*. Quito, Quito, Ecuador. Recuperado el 21 de Agosto de 2017
- Amoako, E. O. (2011). *Determination of shortest path using Dijkstra's Algorithm for Emergency Service in Kumasi Metropolis*. KUMASI. Recuperado el 23 de Agosto de 2017
- Casanova Matera, L. (2002). *Topografía Plana*. Mérida: Taller de Publicaciones de Ingeniería, ULA. Recuperado el 24 de Agosto de 2017
- Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2001). *Introducción a Algoritmos*. Massachusetts London, Inglaterra: MIT. Recuperado el 15 de Agosto de 2017
- Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2009). *Introduction to Algorithms* (Third Edition ed.). Massachusetts London, Inglaterra: MIT. Recuperado el 22 de Noviembre de 2017
- Darryl K. Taft. (6 de Diciembre de 2002). *Eweek*. Obtenido de Eweek: <http://www.eweeek.com/pc-hardware/ibm-acquires-rational>
- Delgado, M. (Mayo de 2008). *Grafos*. Recuperado el 20 de Noviembre de 2017, de Grafos: <http://www.dma.fi.upm.es/personal/gregorio/grafos/web/dijkstra/>
- EUROPA PRESS. (29 de Junio de 2017). *Notimérica*. Recuperado el 12 de Setiembre de 2017, de Notimérica: <http://www.notimerica.com/economia/noticia-abertis-colaborara-aplicacion-traffic-waze-brasil-chile-argentina-20170629130425.html>
- Flores Cueto J.J. y Bertolotti, Z. C. (2006). *Google Books*. Recuperado el 21 de Noviembre de 2017, de Google Books: <https://books.google.com.pe/books?id=bntdb3fV31EC&pg=PA74&lpg=PA74&dq=Algoritmo&source=bl&ots=hpQXGqG5DT&sig=V3PC8pvVDKp18ihOVqXtufli5C8&hl=es-419&sa=X&ved=0ahUKEwj9yuilnOfVAhXDTSYKHcBZAh44ChDoAQhMMAc#v=onepage&q&f=false>
- Flores Cueto, J., & Tataje Salas, G. (2011). *Método de la 6'D*. Lima: USMP. Recuperado el 21 de Noviembre de 2017, de Google Books: <https://books.google.com.pe/books?id=bntdb3fV31EC&pg=PA74&lpg=PA74&dq=Algoritmo&source=bl&ots=hpQXGqG5DT&sig=V3PC8pvVDKp18ihOVqXtufli5C8&hl=es-419&sa=X&ved=0ahUKEwj9yuilnOfVAhXDTSYKHcBZAh44ChDoAQhMMAc#v=onepage&q&f=false>
- Gallardo. (24 de Noviembre de 2003). *Grafos*. Madrid, España. Recuperado el 12 de Agosto de 2017
- Gallardo, D. (26 de 11 de 2012). *Iniciándose en la plataforma Eclipse*. Recuperado el 15 de Setiembre de 2017, de Iniciándose en la plataforma Eclipse: <https://www.ibm.com/developerworks/ssa/library/os-ecov/index.html>
- Garrido A. Onaindía E. (2010). Un algoritmo para la optimización de rutas de transporte. *Un algoritmo para la optimización de rutas de transporte*. Valencia, Valencia, España. Recuperado el 12 de Setiembre de 2017
- Gilfillan, I. (s.f.). *La Biblia MySQL*. Obtenido de <http://didepa.uaemex.mx/clases/Manuales/MySql/MySql-La%20biblia%20de%20mysql.pdf>
- Godsil, C., & Royle, G. (2001). *Teoría Algebraica de Grafos*. New York: Springer. Recuperado el 12 de Setiembre de 2017
- González Gutiérrez, F. J. (Octubre de 2004). Apuntes de Matemática Discreta. Cádiz, Andalucía, España. Recuperado el 12 de Setiembre de 2017, de <http://www2.uca.es/matematicas/Docencia/ESI/1711003/Apuntes/Leccion14.pdf>
- Google Inc, A. C. (2017). *Conoce Android Studio*. Recuperado el 12 de Setiembre de 2017, de Conoce Android Studio: <https://developer.android.com/studio/intro/index.html>

- Google Inc, A. C. (27 de Marzo de 2017). *Contenido de Licencia*. Recuperado el 12 de Setiembre de 2017, de Contenido de Licencia: <https://source.android.com/source/licenses>
- Google Inc, A. C. (27 de Marzo de 2017). *The Android Source Code*. Recuperado el 13 de Setiembre de 2017, de The Android Source Code: <https://source.android.com/source/>
- GRTC. (26 de Febrero de 2018). *Gerencia Regional de Transportes y Comunicaciones*. Obtenido de Gerencia Regional de Transportes y Comunicaciones: <http://www.grtclalibertad.gob.pe/servicios/actualidad/transparencia-institucional/informacion-adicional/sub-gerencia-de-transportes>
- Gusfield. (2014). Recuperado el 12 de Setiembre de 2017
- Guzmán Luna J. A. Arango Sánchez R. E. Jiménez Pinzón L. D. (2012). Búsqueda de la ruta óptima mediante los algoritmos: genéticos y dijkstra utilizando mapas de visibilidad. *Búsqueda de la ruta óptima mediante los algoritmos: genéticos y dijkstra utilizando mapas de visibilidad*. Recuperado el 12 de Setiembre de 2017
- Huerta, E., Mangiaterra, A., & Noguera, G. (2005). *GPS Posicionamiento Satelital*. Rosario: UNR Editora. Recuperado el 22 de Setiembre de 2017
- Inam, R. (2010). A* Algorithm for Multicore Graphics Processors. Gotemburgo, Suecia. Recuperado el 19 de Setiembre de 2017
- INEI. (12 de Diciembre de 2017). *Instituto Nacional de Estadística e Informática*. Recuperado el 20 de Noviembre de 2017, de Instituto Nacional de Estadística e Informática: <https://www.inei.gob.pe/estadisticas/indice-tematico/transport-and-communications/>
- INRIX Global Traffic Scorecard. (2017). *INRIX*. Recuperado el 12 de Setiembre de 2017, de INRIX: <http://inrix.com/scorecard/>
- Jemerov, D. (22 de Mayo de 2013). *IntelliJ IDEA and Android Studio FAQ*. Recuperado el 16 de Setiembre de 2017, de IntelliJ IDEA and Android Studio FAQ: <https://blog.jetbrains.com/idea/2013/05/intellij-idea-and-android-studio-faq/#comment-4939>
- Johnsonbaugh, R. (2005). *Matemáticas Discretas* (Sexta ed.). Mexico: Pearson Education. Recuperado el 19 de Setiembre de 2017
- Marchena Pejerrey, D. J. (Septiembre de 2015). Desarrollo de un Sistema para la optimización de rutas de trabajo utilizando el algoritmo de Dijkstra y Diagramas de Voronoi. Chiclayo, La Libertad, Peru. Recuperado el 13 de Setiembre de 2017
- Marchena Pejerrey, J. D. (Setiembre de 2015). Desarrollo de un sistema para la optimización de rutas de trabajo utilizando el algoritmo de Dijkstra y Diagramas de Voronoi. Chiclayo, Peru. Recuperado el 19 de Setiembre de 2017
- Matt Stephens. (6 de Abril de 2006). *RedGate*. Obtenido de RedGate: <https://www.red-gate.com/simple-talk/opinion/opinion-pieces/agile-development-and-iconix/>
- Meza, O., & Ortega, M. (2006). *Google Books*. Recuperado el 21 de Setiembre de 2017, de Google Books: https://books.google.com.pe/books?id=bETqIvP2UuwC&pg=PA130&lpg=PA130&dq=Algoritmo+BFS&source=bl&ots=PQxfrdKZJ-&sig=jjyp3Uhr5d11-qkA4Q2R84UpHVI&hl=es-419&sa=X&ved=0ahUKEwjXuZTQo-fVAhX17CYKHW1_Aqk4ChDoAQhPMAc#v=onepage&q&f=false
- Oracle, C. (2017). *¿Qué es J2ME o Java ME?* Recuperado el 26 de Setiembre de 2017, de ¿Qué es J2ME o Java ME?: https://www.java.com/es/download/faq/whatis_j2me.xml
- Oracle, C. (2017). *About MySQL Workbench*. Recuperado el 23 de Setiembre de 2017, de About MySQL Workbench: <http://mysqlworkbench.org/about/>
- Oracle, C. (26 de Noviembre de 2017). *MySQL 5.7 Reference Manual*. Obtenido de MySQL 5.7 Reference Manual: <https://downloads.mysql.com/docs/refman-5.7-en.pdf>
- Oracle, C. (2017). *MySQL Workbench*. Recuperado el 29 de Setiembre de 2017, de MySQL Workbench: <https://www.mysql.com/products/workbench/>
- Pérez Ramírez, D., Sepúlveda, J. C., & I. Oliveros, Y. (2011). *Extreme Programming (XP)*. Valencia: Editorial Académica Española.

- Pollice, G., Selkow, S., & Heineman, G. (2008 de Octubre de 2008). *Algorithms in a Nutshell*. O'Reilly Media Inc. Recuperado el 29 de Setiembre de 2017
- PostGIS Project Steering Committee. (2018). *PostGIS*. Obtenido de PostGIS: <https://postgis.net/>
- PUCP. (11 de Diciembre de 2015). *Pontificia Universidad Católica del Perú*. Recuperado el 20 de Noviembre de 2017, de Pontificia Universidad Católica del Perú: <http://puntoedu.pucp.edu.pe/noticias/un-nuevo-algoritmo-para-mejorar-el-trafico-en-lima/>
- Rouse, M. (Enero de 2015). *MySQL*. Recuperado el 28 de Setiembre de 2017, de MySQL: <http://searchdatacenter.techtarget.com/es/definicion/MySQL>
- Rouse, M. (December de 2016). *Java*. Recuperado el 27 de Setiembre de 2017, de Java: <http://www.theserverside.com/definition/Java>
- Ruohonen, K. (2013). *Graph Theory*. Recuperado el 28 de Setiembre de 2017
- Schildt, H. (2017). *Java: A Beginner's Guide* (Séptima ed.). New York: McGraw-Hill Education.
- Taquía Valdivia J. A. (3 de Junio de 2013). Optimización de rutas en una empresa de recojo de residuos sólidos en el distrito de los Olivos. *Optimización de rutas en una empresa de recojo de residuos sólidos en el distrito de los Olivos*. Lima, Lima, Peru. Recuperado el 12 de Setiembre de 2017
- The pgAdmin Development Team. (2018). *pgAdmin*. Obtenido de pgAdmin: <https://www.pgadmin.org/>
- The PostgreSQL Global Development Group. (2018). *PostgreSQL*. Recuperado el 28 de Setiembre de 2017, de PostgreSQL: <https://www.postgresql.org/about/>
- Transportes Metropolitanos de Trujillo. (5 de Octubre de 2015). Evaluacion de Area Saturada de la Red Vial Metropolitana de Trujillo. En P. Quiroz, & G. Huerta, *Transportes Metropolitanos de Trujillo* (pág. 5). Trujillo. Recuperado el 12 de Setiembre de 2017, de <http://siar.regionlalibertad.gob.pe/index.php?accion=verElemento&idElementoInformacion=784&verPor=&idTipoElemento=13&idTipoFuente=>
- Universidad Tecnológica del Perú, V. d. (2016). *Algoritmo y Estructura de Datos I. Algoritmo y Estructura de Datos I*. Lima, Lima, Perú: Imprenta Grupo IDAT. Recuperado el 3 de Setiembre de 2017
- Vogel, L. (2013). *Eclipse IDE*.

ANEXOS

ANEXO N° 1: CUESTIONARIO DEL TRANSITO VEHICULAR

CUESTIONARIO DE TRÁFICO VEHICULAR

El presente cuestionario tiene por finalidad recoger información para establecer el nivel de la movilidad vehicular en la ciudad Trujillo. Así mismo se le pide ser extremadamente objetivo (a), honesto (a) y sincero (a) en sus respuestas. Se le agradece por anticipado su valiosa participación y colaboración

Instrucciones: Marque una de las alternativas que se dan a continuación a cada pregunta.

1. ¿Llega usted en el tiempo propuesto a su destino?

- Nunca
- A veces
- Siempre

2. ¿Cuál es el tiempo que tarda al llegar a su punto destino?

- 5 a 10 minutos
- 10 a 30 minutos
- 30 a 60 minutos

3. ¿Determina usted alguna ruta alterna ante congestión vehicular, inaccesibilidad o algún accidente dentro de su ruta?

- Nunca
- A veces
- Siempre

4. Si han tomado alguna ruta alterna, ¿Le fue de buena ayuda?

- Nunca
- A veces
- Siempre

5. ¿Usa algún medio informativo para buscar rutas alternas?

- Nunca
- A veces
- Siempre

6. Si usan algún medio informativo, ¿Le fue de buena ayuda?

- Nunca
- A veces
- Siempre

7. ¿Cómo califica el tráfico vehicular en la ciudad de Trujillo?

- Malo
- Regular
- Bueno

ANEXO N° 2: CUESTIONARIO DEL TRÁNSITO VEHICULAR

CUESTIONARIO DE TRÁFICO VEHICULAR

El presente cuestionario tiene por finalidad recoger información para establecer el nivel de la movilidad vehicular en la ciudad Trujillo. Así mismo se le pide ser extremadamente objetivo (a), honesto (a) y sincero (a) en sus respuestas. Se le agradece por anticipado su valiosa participación y colaboración

1. ¿Cuál es la ruta que toma desde el punto de inicio A hacia el punto destino B?

2. ¿Cuál es la ruta que tomaría en caso de tráfico en la ubicación x desde el punto de inicio A hacia el punto destino B?

3. ¿Qué factores le hacen tomar una ruta diferente a la ruta común?

ANEXO N° 3: ESTRUCTURA DE LA ENTREVISTA

ESTRUCTURA DE LA ENTREVISTA

¿Cuántas rutas existen entre la Av. España y el Centro histórico de la ciudad de Trujillo?

¿Dan aviso a los conductores cuando una(s) de las rutas se encuentra inaccesible por motivos de obras o accidente?

¿Cuentan con algún lugar o centro de comandos para realizar las observaciones del tráfico vehicular?

¿Cuentan con algún sistema que les permita realizar diagnósticos de la situación vehicular?

¿Estaría de acuerdo y comprometido a dar todo el apoyo necesario para el desarrollo de la aplicación móvil?

¿Quién sería el encargado de autorizar el acceso a la información y de supervisar el desarrollo del proyecto?

ANEXO N° 4: RECOLECCIÓN DE DATOS

ESCALA 1 (STO. DOMINGUITO -> PLAZA DE ARMAS)		ANTES					DESPUES				
GRUPO	CONDUCTORES	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen
G1	8	19	3,5	2	AV PERÚ - MERCADO UNIÓN CALLE AMAZONAS		10	3,5	7	AV PERÚ - MERCADO UNIÓN CALLE AMAZONAS MAYORISTA ANTES - DESPUES HUAYNA CAPAC - AV ESPAÑA AV ESPAÑA - 29 DE DICIEMBRE JR FRANCISCO PIZARRO - JR BOLOGNESI	
G2	8	20	3,5	1	CALLE AMAZONAS		10	3,5	6	CALLE AMAZONAS MAYORISTA ANTES - DESPUES HUAYNA CAPAC - AV ESPAÑA AV ESPAÑA - 29 DE DICIEMBRE JR FRANCISCO PIZARRO - JR BOLOGNESI	
G3	12	22	4	3	AV PERU - MERCADO UNIÓN CALLE AMAZONAS VALLEJO DESDE PUNO -> MAYORISTA		11,4	3,5	5	AV PERU - MERCADO UNIÓN CALLE AMAZONAS VALLEJO DESDE PUNO -> MAYORISTA AV VALLEJO - GUZMAN BARRON JR INDEPENDENCIA - JR ORBEGOZO	
G4	5	16	3,7	2	AV PERÚ - MERCADO UNIÓN VALLEJO DESDE PUNO -> MAYORISTA		11,4	3,5	5	AV PERU - MERCADO UNIÓN CALLE AMAZONAS VALLEJO DESDE PUNO -> MAYORISTA AV VALLEJO - GUZMAN BARRON JR INDEPENDENCIA - JR ORBEGOZO	

ESCALA 2 (PLAZA DE ARMAS -> UPAO)		ANTES					DESPUES				
GRUPO	CONDUCTORES	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen
G1	14	7,1	2,5	2	AV. ESPAÑA - BOLIVAR COSTA RICA - AV AMERICA		7,1	2,5	7	"AV. ESPAÑA - BOLIVAR COSTA RICA - AV AMERICA" Jr. Independencia - Jr. Bolognesi Av. España - Independencia (Janos) Av. España - Jr. Pizarro (OR) Av. Larco - Av. Húsares de Junín (Pileta de Agua) Av. Husares de Junin (Clínica San Pablo)	
G2	10	15	2,9	2	AV. ESPAÑA - BOLIVAR AV. HUSARES DE JUNIN		8	2,6	4	AV. ESPAÑA - BOLIVAR AV. HUSARES DE JUNIN AV. ESPAÑA - 29 DE DICIEMBRE AV. AMERICA SUR - AV. COSTA RICA	
G3	2	15	3	2	AV. ESPAÑA - BOLIVAR AV. HUSARES DE JUNIN		8	2,6	4	AV. ESPAÑA - BOLIVAR AV. HUSARES DE JUNIN AV. ESPAÑA - 29 DE DICIEMBRE AV. AMERICA SUR - AV. COSTA RICA	
G4	7	11	2,9	2	INDEPENDENCIA CUADRA 1 HUSARES DE JUNIN		8	2,6	4	AV. ESPAÑA - BOLIVAR AV. HUSARES DE JUNIN AV. ESPAÑA - 29 DE DICIEMBRE AV. AMERICA SUR - AV. COSTA RICA	

ESCALA 3 (UPAO -> UPN)		ANTES					DESPUES				
GRUPO	CONDUCTORES	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen
G1	14	17	5,4	3	AV. JUAN PABLO - AV. JESÚS DE NAZARETH HACIA UNT AV. JUAN PABLO - AV. JESÚS DE NAZARETH HACIA BOMBEROS AV. LARCO - AV. ESPAÑA HACIA EL CENTRO		12,8	5,3	7	AV. JUAN PABLO - AV. JESÚS DE NAZARETH HACIA UNT AV. JUAN PABLO - AV. JESÚS DE NAZARETH HACIA BOMBEROS AV. LARCO - AV. ESPAÑA HACIA EL CENTRO AV. AMERICA SUR - OVALO LARCO AV. AMERICA OESTE - AV JUAN PABLO II AV. AMERICA OESTE - AV ANTENOR ORREGO AV. AMERICA OESTE - FRENTE LA CORTE	
G2	9	21	6	1	AV. JUAN PABLO - AV. PEDRO MUÑIZ - AV. ESPAÑA		12,8	5,3	5	AV. JUAN PABLO - AV. PEDRO MUÑIZ - AV. ESPAÑA AV. AMERICA SUR - OVALO LARCO AV. AMERICA OESTE - AV JUAN PABLO II AV. AMERICA OESTE - AV ANTENOR ORREGO AV. AMERICA OESTE - FRENTE LA CORTE	
G3	10	14	5,9	1	AV. AMERICA OESTE - AV. ANTENOR ORREGO		13,8	5,6	4	AV. JUAN PABLO - AV. PEDRO MUÑIZ - AV. ESPAÑA AV. AMERICA SUR - OVALO LARCO AV. AMERICA OESTE - AV JUAN PABLO II AV. AMERICA OESTE - FRENTE LA CORTE	

ESCALA 4 (UPN - EMTRAFESA)		ANTES					DESPUES				
GRUPO	CONDUCTORES	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen
G1	14	12	3.7	3	Av. Mansiche - Mall Aventura Plaza (Entrada Ripley) Ovalo Mochica Av. Mansiche - Ca. Los Zafiros (Hospital Regional)		10,5	3,5	5	Av. Mansiche - Mall Aventura Plaza (Entrada Ripley) Ovalo Mochica Av. Mansiche - Ca. Los Zafiros (Hospital Regional) AV. NICOLAS DE PIEROLA - AV INDOAMERICA AV. TUPAC AMAERU - AV. INDOAMERICA	
G2	7	14	4.1	2	Av. Túpac Amaru - Av. Indoamérica Av. Nicolás de Pierola - Av. Valcárcel (Interseccion)		11	3,8	3	Av. Túpac Amaru - Av. Indoamérica Av. Nicolás de Pierola - Av. Valcárcel (Interseccion) AV. NICOLAS DE PIEROLA - AV INDOAMERICA	
G3	8	13	4.2	3	Av. Nicolás de Pierola - Av. Valcárcel (Interseccion) Av. América Norte - Av. M. Vera Enriquez Av. España - Av. M. Vera Enriquez		10,5	3,5	4	Av. Nicolás de Pierola - Av. Valcárcel (Interseccion) Av. América Norte - Av. M. Vera Enriquez Av. España - Av. M. Vera Enriquez AV. NICOLAS DE PIEROLA - AV INDOAMERICA	
G4	4	16	4.5	5	Av. Mansiche - Mall Aventura Plaza Av. América Oeste - Av Mansiche Av. Nicolás de Pierola - Av. Indoamérica Av. Pablo Casals - Av. Tupac Amaru Av. Teodoro Varcárcel - Av. Tupac Amaru		10,5	3,5	5	Av. Mansiche - Mall Aventura Plaza Av. América Oeste - Av Mansiche Av. Nicolás de Pierola - Av. Indoamérica Av. Pablo Casals - Av. Tupac Amaru Av. Teodoro Varcárcel - Av. Tupac Amaru	

ESCALA 5 (EMTRAFESA -> REAL PLAZA)		ANTES					DESPUES				
GRUPO	CONDUCTORES	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen
G1	9	19	4,9	3	JIRON INDEPENDENCIA - PLAZA DE ARMAS AV ESPAÑA - VIRREY AV AMERICA SUR (UPAO) HACIA LINEA		12,1	4,1	8	JIRON INDEPENDENCIA - PLAZA DE ARMAS AV ESPAÑA - VIRREY AV AMERICA SUR (UPAO) HACIA LINEA AV. ESPAÑA - AV. PERÚ JR GAMARRA - JR GRAÚ AV. ESPAÑA - AV. 29 DE DICIEMBRE AV. AMERICA SUR - AV. COSTA RICA PROL. CESAR VALLEJO - C. PAISAJISTICA	
G2	6	20	4,4	5	JIRON INDEPENDENCIA - PLAZA DE ARMAS AV ESPAÑA (VIRREY) JIRON GRAU (ESPALDA VIRREY) AV ESPAÑA FRENTE OR AV AMERICA SUR (UPAO) HACIA LINEA		11,7	4,2	9	JIRON INDEPENDENCIA - PLAZA DE ARMAS AV ESPAÑA (VIRREY) JIRON GRAU (ESPALDA VIRREY) AV ESPAÑA FRENTE OR AV AMERICA SUR (UPAO) HACIA LINEA SINCHI ROCA - MERCADO MAYORISTA SINCHI ROCA CON ZELA AV. AMERICA SUR - COSTA RICA PROL. VALLEJO - C. PAISAJISTICA	
G3	12	21	4,1	3	AV ESPAÑA - PEDRO MUÑIZ AV ESPAÑA FRENTE OR AV AMERICA SUR (UPAO) HACIA LINEA		11,9	4	10	AV ESPAÑA - PEDRO MUÑIZ AV ESPAÑA FRENTE OR AV AMERICA SUR (UPAO) HACIA LINEA AV. ESPAÑA - AV. PERÚ AV. ESPAÑA - C. FRANCISCO ZELA AV. ESPAÑA - C. ATAHUALPA AV. ESPAÑA - C. HUAYNA CAPAC AV. ESPAÑA - AV. 29 DE DICIEMBRE AV. AMERICA SUR - AV. COSTA RICA PROL. CESAR VALLEJO - PAISAJISTICA	
G4	6	23	4,7	2	AV ESPAÑA - PEDRO MUÑIZ AV AMERICA SUR (UPAO) HACIA LINEA		11,9	4	9	AV ESPAÑA - PEDRO MUÑIZ AV AMERICA SUR (UPAO) HACIA LINEA AV. ESPAÑA - AV. PERÚ AV. ESPAÑA - C. FRANCISCO ZELA AV. ESPAÑA - C. ATAHUALPA AV. ESPAÑA - C. HUAYNA CAPAC AV. ESPAÑA - AV. 29 DE DICIEMBRE AV. AMERICA SUR - AV. COSTA RICA PROL. CESAR VALLEJO - PAISAJISTICA	

ESCALA 5 (REAL PLAZA -> MALL)		ANTES					DESPUES				
GRUPO	CONDUCTORES	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen	Tiempo (min)	Distancia (km)	N° de Incidencias	Detalle de Incidencias	Imagen
G1	9	14	4.5	1	AV. JUAN PABLO II (PUERTA PRINCIPAL UNT)		9,9	3,9	6	AV. JUAN PABLO II (PUERTA PRINCIPAL UNT) PROL. VALLEJO CON PAISAJISTICA AV. AMERICA SUR - OVALO LARCO AV. AMERICA OESTE - AV. JUAN PABLO II AV. AMERICA OESTE - AV ANTENOR ORREGO	
G2	12	18	4.9	2	AV. AMERICA SUR (UPAO) JUAN PABLO II - PEDRO MUÑIZ Y AV. ESPAÑA		9,9	3,9	7	AV. AMERICA SUR (UPAO) JUAN PABLO II - PEDRO MUÑIZ Y AV. ESPAÑA PROL. VALLEJO CON PAISAJISTICA AV. AMERICA SUR - OVALO LARCO AV. AMERICA OESTE - AV. JUAN PABLO II AV. AMERICA OESTE - AV ANTENOR ORREGO	
G3	5	20	4.9	2	AV. AMERICA SUR (UPAO) AV. JESUS DE NAZARETH (ALBRETCH)		9,9	3,9	7	AV. AMERICA SUR (UPAO) AV. JESUS DE NAZARETH (ALBRETCH) PROL. VALLEJO CON PAISAJISTICA AV. AMERICA SUR - OVALO LARCO AV. AMERICA OESTE - AV. JUAN PABLO II AV. AMERICA OESTE - AV ANTENOR ORREGO	
G4	4	15	5,1	3	AV. AMERICA SUR (UPAO) AV. JUAN PABLO II (PUERTA PRINCIPAL UNT) POR FOTOCOPIAS AV. AMERICA OESTE - AV. ANTENOR ORREGO		11,7	4,1	4	AV. AMERICA SUR (UPAO) AV. JUAN PABLO II (PUERTA PRINCIPAL UNT) POR FOTOCOPIAS AV. AMERICA OESTE - AV. ANTENOR ORREGO	
G5	3	13	4,9	2	AV. AMERICA OESTE (FRENTE A LA CORTE) NATASHA AV. AMERICA OESTE - AV. ANTENOR ORREGO		11,7	4,1	2	AV. AMERICA OESTE (FRENTE A LA CORTE) NATASHA AV. AMERICA OESTE - AV. ANTENOR ORREGO	

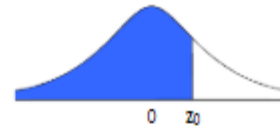
ANEXO N° 5: TABLA DE LA DISTRIBUCION NORMAL

μ = Media

σ = Desviación típica

Tipificación: $z_0 = \frac{x - \mu}{\sigma}$

$$P(z \leq z_0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_0} e^{-\frac{z^2}{2}} dz$$



z_0	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09	z_0
0,0	0,5000	0,5040	0,5080	0,5120	0,5160	0,5199	0,5239	0,5279	0,5319	0,5359	0,0
0,1	0,5398	0,5438	0,5478	0,5517	0,5557	0,5596	0,5636	0,5675	0,5714	0,5753	0,1
0,2	0,5793	0,5832	0,5871	0,5910	0,5948	0,5987	0,6026	0,6064	0,6103	0,6141	0,2
0,3	0,6179	0,6217	0,6255	0,6293	0,6331	0,6368	0,6406	0,6443	0,6480	0,6517	0,3
0,4	0,6554	0,6591	0,6628	0,6664	0,6700	0,6736	0,6772	0,6808	0,6844	0,6879	0,4
0,5	0,6915	0,6950	0,6985	0,7019	0,7054	0,7088	0,7123	0,7157	0,7190	0,7224	0,5
0,6	0,7257	0,7291	0,7324	0,7357	0,7389	0,7422	0,7454	0,7486	0,7517	0,7549	0,6
0,7	0,7580	0,7611	0,7642	0,7673	0,7704	0,7734	0,7764	0,7794	0,7823	0,7852	0,7
0,8	0,7881	0,7910	0,7939	0,7967	0,7995	0,8023	0,8051	0,8078	0,8106	0,8133	0,8
0,9	0,8159	0,8186	0,8212	0,8238	0,8264	0,8289	0,8315	0,8340	0,8365	0,8389	0,9
1,0	0,8413	0,8438	0,8461	0,8485	0,8508	0,8531	0,8554	0,8577	0,8599	0,8621	1,0
1,1	0,8643	0,8665	0,8686	0,8708	0,8729	0,8749	0,8770	0,8790	0,8810	0,8830	1,1
1,2	0,8849	0,8869	0,8888	0,8907	0,8925	0,8944	0,8962	0,8980	0,8997	0,9015	1,2
1,3	0,9032	0,9049	0,9066	0,9082	0,9099	0,9115	0,9131	0,9147	0,9162	0,9177	1,3
1,4	0,9192	0,9207	0,9222	0,9236	0,9251	0,9265	0,9279	0,9292	0,9306	0,9319	1,4
1,5	0,9332	0,9345	0,9357	0,9370	0,9382	0,9394	0,9406	0,9418	0,9429	0,9441	1,5
1,6	0,9452	0,9463	0,9474	0,9484	0,9495	0,9505	0,9515	0,9525	0,9535	0,9545	1,6
1,7	0,9554	0,9564	0,9573	0,9582	0,9591	0,9599	0,9608	0,9616	0,9625	0,9633	1,7
1,8	0,9641	0,9649	0,9656	0,9664	0,9671	0,9678	0,9686	0,9693	0,9699	0,9706	1,8
1,9	0,9713	0,9719	0,9726	0,9732	0,9738	0,9744	0,9750	0,9756	0,9761	0,9767	1,9
2,0	0,9772	0,9778	0,9783	0,9788	0,9793	0,9798	0,9803	0,9808	0,9812	0,9817	2,0
2,1	0,9821	0,9826	0,9830	0,9834	0,9838	0,9842	0,9846	0,9850	0,9854	0,9857	2,1
2,2	0,9861	0,9864	0,9868	0,9871	0,9875	0,9878	0,9881	0,9884	0,9887	0,9890	2,2
2,3	0,9893	0,9896	0,9898	0,9901	0,9904	0,9906	0,9909	0,9911	0,9913	0,9916	2,3
2,4	0,9918	0,9920	0,9922	0,9925	0,9927	0,9929	0,9931	0,9932	0,9934	0,9936	2,4
2,5	0,9938	0,9940	0,9941	0,9943	0,9945	0,9946	0,9948	0,9949	0,9951	0,9952	2,5
2,6	0,9953	0,9955	0,9956	0,9957	0,9959	0,9960	0,9961	0,9962	0,9963	0,9964	2,6
2,7	0,9965	0,9966	0,9967	0,9968	0,9969	0,9970	0,9971	0,9972	0,9973	0,9974	2,7
2,8	0,9974	0,9975	0,9976	0,9977	0,9977	0,9978	0,9979	0,9979	0,9980	0,9981	2,8
2,9	0,9981	0,9982	0,9982	0,9983	0,9984	0,9984	0,9985	0,9985	0,9986	0,9986	2,9
3,0	0,99865	0,99869	0,99874	0,99878	0,99882	0,99886	0,99889	0,99893	0,99896	0,99900	3,0
3,1	0,99903	0,99906	0,99910	0,99913	0,99916	0,99918	0,99921	0,99924	0,99926	0,99929	3,1
3,2	0,99931	0,99934	0,99936	0,99938	0,99940	0,99942	0,99944	0,99946	0,99948	0,99950	3,2
3,3	0,99952	0,99953	0,99955	0,99957	0,99958	0,99960	0,99961	0,99962	0,99964	0,99965	3,3
3,4	0,99966	0,99968	0,99969	0,99970	0,99971	0,99972	0,99973	0,99974	0,99975	0,99976	3,4
3,5	0,99977	0,99978	0,99978	0,99979	0,99980	0,99981	0,99981	0,99982	0,99983	0,99983	3,5
3,6	0,99984	0,99985	0,99985	0,99986	0,99986	0,99987	0,99987	0,99988	0,99988	0,99989	3,6
3,7	0,99989	0,99990	0,99990	0,99990	0,99991	0,99991	0,99992	0,99992	0,99992	0,99992	3,7
3,8	0,99993	0,99993	0,99993	0,99994	0,99994	0,99994	0,99994	0,99995	0,99995	0,99995	3,8
3,9	0,99995	0,99995	0,99996	0,99996	0,99996	0,99996	0,99996	0,99996	0,99997	0,99997	3,9

$1-\alpha$	90%	92%	94%	95%	96%	97%	98%	99%
α	10%	8%	6%	5%	4%	3%	2%	1%
$z_{1-\alpha/2}$	1,645	1,751	1,881	1,960	2,054	2,170	2,326	2,576
z_{α}	1,282	1,405	1,555	1,645	1,751	1,881	2,054	2,326

Siendo:

$1-\alpha$ = Nivel de confianza
 α = Nivel de significación