



FACULTAD DE INGENIERÍA GEOGRÁFICA, AMBIENTAL Y ECOTURISMO

DESARROLLO DE UN PLANIFICADOR DE RUTAS EN BASE A LOS
ALGORITMOS DE DIJKSTRA USANDO EL SISTEMA DE TRANSPORTE
PÚBLICO DE CARÁCTER REGULAR A NIVEL DE LIMA Y CALLAO

Línea de investigación:

**Desarrollo urbano-rural, catastro, prevención de riesgos, hidráulica y
geotecnia**

Trabajo de suficiencia profesional para optar el título profesional de
Ingeniero Geógrafo

Autor:

Roque Huamán, André Valentín

Asesora:

Aylas Humareda, María del Carmen

ORCID: 0000-0002-2063-0005

Jurado:

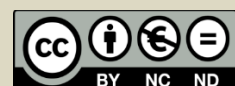
Osorio Rojas, Eberardo Antonio

Valer Silva, José Manuel

Díaz Villalobos, Carlos Alberto

Lima - Perú

2024



DESARROLLO DE UN PLANIFICADOR DE RUTAS EN BASE A LOS ALGORITMOS DE DIJKSTRA USANDO EL SISTEMA DE TRANSPORTE PÚBLICO DE CARÁCTER REGULAR A NIVEL DE LIMA Y CALLAO

INFORME DE ORIGINALIDAD

14%

INDICE DE SIMILITUD

12%

FUENTES DE INTERNET

3%

PUBLICACIONES

5%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	cdn.www.gob.pe Fuente de Internet	1%
2	repositorio.unfv.edu.pe Fuente de Internet	1%
3	Submitted to Universidad de León Trabajo del estudiante	<1%
4	busquedas.elperuano.pe Fuente de Internet	<1%
5	Submitted to Corporación Universitaria Minuto de Dios, UNIMINUTO Trabajo del estudiante	<1%
6	www.coursehero.com Fuente de Internet	<1%
7	Submitted to Universidad Nacional Abierta y a Distancia, UNAD, UNAD Trabajo del estudiante	<1%



Universidad Nacional
Federico Villarreal

VRIN | VICERRECTORADO
DE INVESTIGACIÓN

FACULTAD DE INGENIERÍA GEOGRÁFICA, AMBIENTAL Y ECOTURISMO

**DESARROLLO DE UN PLANIFICADOR DE RUTAS EN BASE A LOS
ALGORITMOS DE DIJKSTRA USANDO EL SISTEMA DE TRANSPORTE
PÚBLICO DE CARÁCTER REGULAR A NIVEL DE LIMA Y CALLAO**

Línea de investigación:

Desarrollo urbano-rural, catastro, prevención de riesgos, hidráulica y geotecnia

Informe de suficiencia profesional para optar el título profesional de Ingeniero Geógrafo

Autor:

Roque Huamán, André Valentín

Asesor:

Aylas Humareda, María del Carmen

(ORCID: 0000-0002-2063-0005)

Jurado:

Osorio Rojas, Eberardo Antonio

Valer Silva, José Manuel

Díaz Villalobos, Carlos Alberto

Lima – Perú

2024

DEDICATORIA

Este trabajo se lo dedico a mi padre, Santos Clotario, y a mi madre, Marfiza, quienes siempre anhelaron ver concluida esta etapa de mi vida. Con este documento, cierro un capítulo importante en la vida de todo profesional, uno que ellos han apoyado con profunda dedicación. Sus acciones y palabras son dignas de ser honradas, ya que a través de ellas me han brindado las enseñanzas que me han permitido avanzar.

En recuerdo de mi padre, quien descansa en paz, agradezco su ejemplo de vida. Sin importar las circunstancias, cumplía cabalmente con su trabajo, y sus sabias palabras, "El hombre es esclavo de sus palabras y dueño de su silencio", me han recordado siempre la importancia de la integridad y de ser prudente cuando no es posible cumplir con lo prometido.

A mi madre, ruego a Dios me permita disfrutar de su compañía por muchos años más. Con su constancia, esfuerzo meticuloso y dedicación, ha ganado el respeto de muchos, y para mí representa un símbolo de superación constante.

AGRADECIMIENTO

Agradezco, en primer lugar, a Dios, quien siempre ha puesto los medios necesarios para ayudarme a superar los obstáculos en mi vida.

A mis padres, Santos Clotario y Marfiza, y a mis hermanos, en especial a Víctor, les extiendo mi más profundo agradecimiento por su apoyo incondicional a lo largo de mi vida académica y personal. Sin su constante respaldo, este logro no habría sido posible.

Deseo también expresar mi gratitud al amor en sus diversas formas, pues ha sido una fuerza que ha influido significativamente en mi vida. Agradezco al Primer Amor, C, por recordarme la importancia de la fe en Dios en los momentos más difíciles. A la Media Naranja, N, por complementar aquellos aspectos de mí que no había podido ver, y que han sido esenciales para mi crecimiento personal. A mi Alma Gemela, F, por ayudarme en esta etapa tan importante, recordándome que con el enfoque adecuado de nuestra esencia podemos cumplir nuestros objetivos. Finalmente, reconozco que el amor en la vida se manifiesta de distintas maneras, brindándonos aprendizajes y apoyo en los momentos clave.

Aprecio profundamente la sincera amistad de quienes han estado a mi lado, en especial la de mi amigo Luis Miguel, quien, como un hermano, me ha brindado su constante apoyo y compañía, escuchando y compartiendo la vida.

INDICE

Resumen	17
Abstract	18
I. INTRODUCCIÓN	19
1.1. Trayectoria del Autor	20
1.2. Descripción de la Empresa	24
1.2.1. Misión	24
1.2.2. Visión	24
1.2.3. Funciones	25
1.2.4. Organigrama	26
1.2.5. Áreas y Funciones Desempeñadas	28
II. DESCRIPCIÓN DE UNA ACTIVIDAD ESPECÍFICA	30
2.1. Listado de Términos	31
2.2. Antecedentes del Proyecto	34
2.3. Planteamiento del Problema	42
2.4. Objetivos	44
2.4.1. Objetivo General	44
2.4.2. Objetivos Específicos	44
2.5. Alcance del Proyecto	45
2.5.1. Sistema de Transporte Público Regular	45
2.5.2. Ámbito	45
2.5.3. Población	45
2.6. Marco Teórico	47
2.6.1. Base Legal	47

2.6.1.1. Base Legal referente al SIT	47
2.6.1.2. Base Legal referente a la ATU	48
2.6.2. Base Teórica	50
2.6.2.1. Historia de la Teoría de Grafos	50
2.6.2.2. El Juego Icosiano	55
2.6.2.3. Fundamentos de la Teoría de Grafos	59
2.6.2.4. Representación y Análisis de los Grafos por Adyacencia	65
2.6.2.5. Ciclos y Caminos Eulerianos	66
2.6.2.6. Ciclos y Caminos Hamiltonianos	68
2.6.2.7. Relación de Ciclos y Caminos Eulerianos y Hamiltonianos	70
2.6.2.8. Aplicaciones Generales de la Teoría de Grafos	72
2.6.2.9. Teoría de Grafos en la Planificación y Desarrollo Territorial	77
2.6.3. Teoría de Grafos en el Transporte	80
2.6.3.1. Teoría de Grafos en el Transporte: TSP	81
2.6.3.2. Teoría de Grafos en el Transporte: Dijkstra	86
2.6.3.3. Relaciones entre el Problema del Viajero y Dijkstra	88
2.6.3.4. Algoritmo de Dijkstra	89
2.6.3.5. Algoritmo de Dijkstra Mejorado	91
2.6.3.6. Aplicación del Algoritmo de Dijkstra	92
2.6.4. Base Técnica	100
2.6.4.1. Teoría de Grafos en la Programación	100
2.6.4.2. Algoritmos en la Programación	101
2.6.4.3. GIS y TIC	106
2.6.4.4. Base de Datos (BD)	109
2.6.4.5. Lenguaje de Programación para Análisis	119

2.6.4.6. Arquitectura Tecnológica del Proyecto	122
2.7. Requerimientos para Implementar Proyecto.....	126
2.7.1. <i>Hardware</i>	126
2.7.2. <i>Software</i>	127
2.7.3. <i>Datos</i>	128
2.7.4. <i>Desarrollo del Proyecto</i>	129
2.7.5. <i>Análisis del Requerimiento</i>	130
2.7.5.1. Análisis de Necesidades y Limitaciones	130
2.7.5.2. Propuesta de Solución	131
2.7.6. <i>Planteamiento de Esquemas en la Base de Datos para la Gestión de Datos</i> 134	
2.7.7. <i>Modelado de Datos</i>	136
2.7.7.1. Modelo de Base de Datos.....	139
2.7.8. <i>Acopio y Procesamiento Automatizado de Datos</i>	143
2.7.8.1. Flujos para datos propios de la entidad (ATU)	143
2.7.8.2. Flujos para datos complementarios de Open Street Map (OSM).....	144
2.7.8.3. Detalle de Procesos Automatizados.....	145
2.7.9. <i>Integración y Estructuración de Datos como Grafos para proceso de</i> <i>Análisis de Ruta</i>	155
2.7.9.1. Integración de datos de la entidad (ATU)	156
2.7.9.2. Integración de datos de complementarios de Open Street Map (OSM) 160	
2.7.9.3. Estructuración de Datos como Objetos del Grafo.....	162
2.7.10. <i>Generación de Procesos de Análisis para la Planificación de Rutas</i>	170
2.7.10.1. Análisis de Datos de Entrada.....	171

2.7.10.2.	Cálculo de Rutas	174
2.7.10.3.	Estructuración de Resultados	179
2.7.11.	<i>Implementación de Herramienta de Planificación de Rutas</i>	182
2.7.11.1.	Implementación de API Python	182
2.7.11.2.	Integración de API Python a la Aplicación Móvil	183
2.7.11.3.	Herramienta de Rutas en la Aplicación Móvil	184
2.8.	Resultados del Proyecto	187
2.8.1.	<i>Estructura de Base de Datos para datos del Sistema de Transporte Público</i> 187	
2.8.2.	<i>Código Base para los Procesos de Análisis de Datos</i>	189
2.8.3.	<i>Herramienta para Planificación de Rutas usando el Servicio de Transporte Público</i>	190
2.8.4.	<i>Integración de búsqueda de rutas con Aplicación Móvil desarrollada por la Entidad</i> 190	
III.	APORTES MÁS DESTACABLES A LA INSTITUCIÓN	193
IV.	CONCLUSIONES	194
V.	RECOMENDACIONES	195
VI.	REFERENCIAS	196
VII.	ANEXOS	197

ÍNDICE DE TABLAS

Tabla 1	71
Comparación de Ciclos y Caminos Eulerianos y Hamiltonianos	71
Tabla 2	88
Comparación de Ciclos y Caminos Eulerianos y Hamiltonianos	88
Tabla 3	111
Algunas Tecnologías de Bases de Datos Relacionales y Aplicaciones GIS.....	111
Tabla 4	117
Algunas Tecnologías de Bases de Datos No SQL y Aplicaciones GIS.....	117
Tabla 5	118
Cuadro comparativo BD Relacional y BD No SQL	118
Tabla 6	120
Ventajas y Desventajas entre Python, R, Julia.....	120
Tabla 7	121
Análisis de Datos con Python, R, Julia	121
Tabla 8	126
Detalle de Hardware	126
Tabla 9	127
Detalle de Software.....	127
Tabla 10	128
Fuente de Datos.....	128
Tabla 11	130
Cuadro de Análisis de Necesidades y Limitaciones del Proyecto	130
Tabla 12	132
Cuadro de Solución a las Necesidades con Tecnología Planteada	132

Tabla 13	133
Cuadro de Solución a las Limitaciones con Tecnología Planteada	133
Tabla 14	135
Distribución de Esquemas de Base de Datos	135
Tabla 15	138
Composición del Sistema Integrado de Transporte	138
Tabla 16	141
Descripción de Objetos de la Base de Datos	141
Tabla 17	146
Secciones OSM Descarga de Datos	146
Tabla 18	147
Secciones OSM Carga de Datos a BD	147
Tabla 19	149
Secciones KML/KMZ exportación a GeoPackage	149
Tabla 20	150
Secciones Integridad de Puntos	150
Tabla 21	152
Secciones Integridad de Puntos	152
Tabla 22	153
Secciones Capas a PostgreSQL	153
Tabla 23	156
Identificación de objetos de Base de Datos referentes al SIT para Centralizar	156
Tabla 24	160
Identificación de objetos de Base de Datos de OSM (Open Street Map) a Centralizar	160
Tabla 25	165

Estructuración de los Nodos para los Grafos	165
Tabla 26	167
Estructuración de los Arcos para los Grafos.....	167
Tabla 27	173
Secciones de la Validación de Datos	173
Tabla 28	175
Secciones del Cálculo de Caminata Inicial y Final.....	175
Tabla 29	178
Secciones del Cálculo de Rutas	178
Tabla 30	179
Secciones del Cálculo de Rutas	179

ÍNDICE DE FIGURAS

Figura 1	Organigrama de la ATU	27
Figura 2	Detalle de la Oficina de Administración (Organigrama de la ATU).....	28
Figura 3	Proyección del 2021 de la población de Lima Metropolitana	34
Figura 4	Proyección del 2021 de la población de Callao.....	35
Figura 5	Extensión y densidad poblacional durante el 2021 para Lima Metropolitana	35
Figura 6	Extensión y densidad poblacional durante el 2021 para Callao	36
Figura 7	Mapa de densidad poblacional a nivel de Lima y Callao	36
Figura 8	Mapa de vías de Lima y Callao.	37
Figura 9	Ranking de las ciudades con mayor tráfico a nivel Mundial en el 2022	38
Figura 10	Ranking de las ciudades con mayor tráfico a nivel de América del Sur en el 2022 38	38
Figura 11	Horas Punta durante el 2009 al 2019	39
Figura 12	Parque Automotor en Lima en el 2020	39
Figura 13	Principal modo de viaje en Lima para ir a estudiar o trabajar durante el 2015 .	40
Figura 14	Principal forma de movilizarse pre y post pandemia durante el 2021	41
Figura 15	Preferencia de medio de transporte para ir a laborar en el 2021.....	41
Figura 16	Mapa de la red del Sistema de Transporte Regular y ámbito según densidad poblacional.....	46
Figura 17	Los 7 Puentes de Königsberg.....	50
Figura 18	Mapa de Königsberg.....	51
Figura 19	Leonhar Euler (1707 – 1783).....	51
Figura 20	Bosquejo dado por Euler.....	52
Figura 21	Estructuración del Grafo de Königsberg.....	53
Figura 22	Representación de nodos - aristas - grados	54

Figura 23	William Rowan Hamilton (1805 – 1865)	56
Figura 24	Los 5 Sólidos Platónicos	57
Figura 25	Dodecaedro de Madera	58
Figura 26	Grafo del Dodecaedro de Madera	58
Figura 27	Solución al Juego Icosiano.....	59
Figura 28	Estructura de un Grafo	60
Figura 29	Grafo No Dirigido.....	61
Figura 30	Grafo Dirigido.....	61
Figura 31	Grafo Ponderado Dirigido.....	62
Figura 32	Grafos Ponderados Bipartitos	62
Figura 33	Grafos Acíclicos No Dirigidos.....	63
Figura 34	Grafos Acíclicos Dirigidos	63
Figura 35	Grafos de Árboles No Dirigidos	64
Figura 36	Grafos de Árboles con Raíz	64
Figura 37	Matriz y Lista de Adyacencia	66
Figura 38	Circuito Euleriano Simple.....	67
Figura 39	Camino Euleriano Simple	67
Figura 40	Circuitos Hamiltonianos	68
Figura 41	Camino Hamiltoniano Simple.....	69
Figura 42	Camino y Circuito Hamiltoniano Simple	69
Figura 43	Relación entre Grafos Eulerianos y Hamiltonianos.....	70
Figura 44	Análisis de Conexiones de un Placa Computacional.....	72
Figura 45	Topología de Redes.....	73
Figura 46	Diagrama de PERT	74
Figura 47	Modelo de Estado de Reposo del Cerebro Humano	74

Figura 48	Ejemplo de Red de Contactos de una Red Social	75
Figura 49	Estructura Esquemática de Cationes de Amonio Cuaternario	76
Figura 50	Ejemplo de Aplicación de Grafos para la asignación de lugares de reciclaje ...	78
Figura 51	Ejemplo de Aplicación de Grafos para realizar la planificación de asfaltado de calles	79
Figura 52	Representación de Ciclo Hamiltoniano en un Grafo	82
Figura 53	Edsger Wybe Dijkstra (1930 – 2002)	87
Figura 54	Algoritmo para ejemplo	93
Figura 55	Ejercicio 01	93
Figura 56	Ejercicio 01 - Paso 01	94
Figura 57	Ejercicio 01 - Paso 02	95
Figura 58	Ejercicio 01 - Paso 03	96
Figura 59	Ejercicio 01 - Paso 04	97
Figura 60	Ejercicio 01 - Paso 05	98
Figura 61	Ejercicio 01 - Paso 06	99
Figura 62	Ejercicio 01 - Paso 07	99
Figura 63	Representación de Mapa en un Grafo	100
Figura 64	Sección del Grafo formado entre Paraderos y Ruta de Servicios	101
Figura 65	Ejemplo de Algoritmo con Cubo Rubik	102
Figura 66	Seudocódigo para Crear el Grafo.....	104
Figura 67	Seudocódigo del Algoritmo de Dijkstra	106
Figura 68	Ejemplo de Estructura de BD Relacional orientado al Análisis de Redes de Transporte	110
Figura 69	Base de Datos Relacional.....	110
Figura 70	BD NoSQL Documental.....	112

Figura 71	BD NoSQL Clave - Valor.....	113
Figura 72	BD NoSQL de Grafos.....	114
Figura 73	BD NoSQL Orientada a Columnas.....	115
Figura 74	Estructuras de Datos a NoSQL	116
Figura 75	Demanda de Lenguajes de Programación 2023.....	119
Figura 76	GIS Desktop y BD PostgreSQL.....	123
Figura 77	GIS Desktop y Python	124
Figura 78	Diagrama de Arquitectura Tecnológica para Implementación de Herramienta de Búsqueda de Rutas.....	125
Figura 79	Flujo para el Desarrollo del Proyecto	129
Figura 80	Distribución de Base de Datos.....	134
Figura 81	Flujo para la generación del Modelo y Base de Datos.....	136
Figura 82	Jerarquía del Sistema de Transporte Regular.....	137
Figura 83	Diagrama de Esquemas y Modelo de Base de Datos.....	139
Figura 84	Modelo de Datos del Sistema de Transporte Regular.....	140
Figura 85	Flujo de Tratamiento de Datos Vectoriales	143
Figura 86	Flujo de Tratamiento de Datos Tabulares.....	144
Figura 87	Flujo de Preprocesamiento y Carga de Datos OSM	145
Figura 88	Componentes OSM Descarga de Datos.....	146
Figura 89	Flujo OSM Descarga de Datos	147
Figura 90	Componentes OSM Carga de Datos a BD.....	148
Figura 91	Flujo OSM Carga de Datos a BD	148
Figura 92	Componentes KML/KMZ exportación a GeoPackage	149
Figura 93	Flujo OSM Carga de Datos a BD	150
Figura 94	Componentes Integridad de Puntos	151

Figura 95	Flujo Integridad de Puntos	151
Figura 96	Componentes Integridad de Polilíneas.....	152
Figura 97	Flujo Integridad de Polilíneas	153
Figura 98	Componentes Capas a PostgreSQL	154
Figura 99	Flujo Capas a PostgreSQL	154
Figura 100	Flujo para la Integración y Estructuración de Datos	155
Figura 101	Proceso de centralización del Objeto: SIT_I_REGULAR.....	157
Figura 102	Proceso de centralización del Objeto: SIT_I_RUTA.....	158
Figura 103	Proceso de centralización del Objeto: SIT_I_PARADA	158
Figura 104	Proceso de centralización del Objeto: SIT_I_RUTA_PARADA	159
Figura 105	Integración de entidades para el Objeto: OSM_I_RUTA	161
Figura 106	Integración de entidades para el Objeto: SIT_I_VERTICE.....	162
Figura 107	Estructura de Ruta y Paradas.....	164
Figura 108	Procesos de Análisis para la Planificación de Rutas	170
Figura 109	Procesos de Ingreso de Datos para Validación	172
Figura 110	Procesos de Validación de Datos	172
Figura 111	Componentes de la Validación de Datos.....	173
Figura 112	Proceso de Análisis de Rutas	174
Figura 113	Proceso de Cálculo de Caminata para el Punto de Llegada y Salida	175
Figura 114	Componentes del Cálculo de Caminata Inicial y Final.....	176
Figura 115	Ejemplo de Estructuración de Grafo	177
Figura 116	Proceso de Cálculo de Rutas	177
Figura 117	Componentes del Cálculo de Rutas.....	178
Figura 118	Ejemplo de Integración Caminata (Inicial y Final) con Posibles rutas Óptimas	

Figura 119	Proceso de Integración de Resultados	180
Figura 120	Componentes de Integración de Resultados.....	180
Figura 121	Ejemplo de Integración Caminata (Inicial y Final) con Posibles rutas Óptimas 181	
Figura 122	Representación de API.....	182
Figura 123	Diagrama Integración de FastAPI y Planificador	183
Figura 124	Integración API FastAPI y .NET	184
Figura 125	Diagrama Tecnología de Alto Nivel de Aplicación.....	184
Figura 126	Aplicación: Planificador de Rutas (Ingreso de Datos).....	185
Figura 127	Aplicación: Planificador de Rutas (Visualización de Ruta).....	186
Figura 128	Modelo de Base de Datos para el SIT	188
Figura 129	Flujo de Data y Archivo de Procesamiento de Datos.....	189
Figura 130	Flujo FastAPI y Análisis de Rutas	190
Figura 131	Aplicación ATU Destino: Plataforma IOS Apple Store	191
Figura 132	Aplicación: Búsqueda de Rutas.....	192

Resumen

El presente informe tiene como objetivo sustentar la capacidad profesional del autor a través de un proyecto implementado por el mismo, así como brindar información de carácter público respecto a la entidad en la cual se desarrolló el proyecto. En el presente, también se incluye la trayectoria profesional y educativa del autor. El autor del presente documento, a la fecha, viene desempeñando el rol de Especialista en Sistemas de Información Geográfica, Especialista GIS, para la Unidad de Tecnologías de la Información (UTI) perteneciente a la Autoridad del Transporte Urbano (ATU). La oficina de UTI se encarga de brindar soluciones tecnológicas para la entidad ATU. Es así como se hace responsable de implementar una aplicación para determinar rutas de viaje usando el Sistema de Transporte Público Regular, actualmente concesionado (COSAC I: Metropolitano, Corredores Complementarios, Línea 01 de Metro de Lima y Callao), a nivel de Lima y Callao. Con la finalidad de ofrecer herramientas orientadas al uso del Sistema de Transporte Público por la ciudadanía. Para poder cumplir con lo objetivo principal de la aplicación es necesario integrar herramientas de análisis de planificación de rutas, siendo en este punto donde los Sistemas de Información Geográfica, como parte de las Tecnologías de Información y Comunicación (TIC), juegan un rol importante en el desarrollo de plataformas digitales que integran funcionalidades basadas en el análisis del territorio, por su versatilidad para explotar y analizar la información georreferenciada. Por ende, el documento pretende dar explicación sobre los algoritmos implementados para el cálculo y determinación de rutas, así como precisar información referente a la estructura de datos usada como insumo clave para el desarrollo de los mismo. Considerar que la planificación de rutas de viaje tiene como alcance el Sistema de Transporte Público de carácter Regular.

Palabras clave: planificación de rutas, sistema de información geográfica, tecnologías de la información y comunicación.

Abstract

This report aims to substantiate the professional capacity of the author through a project implemented by them, as well as to provide public information about the entity in which the project was developed. The document also includes the professional and educational trajectory of the author. To date, the author has been serving as a Specialist in Geographic Information Systems, GIS Specialist, for the Information Technology Unit (UTI) of the Urban Transport Authority (ATU). The IT Unit office is responsible for providing technological solutions for the ATU entity. In this role, it is tasked with implementing an application for determining travel routes using the Regular Public Transport System, currently concessioned (COSAC I: Metropolitano, Complementary Corridors, Line 01 of Lima and Callao Metro), in Lima and Callao. The purpose is to offer tools oriented towards the use of the Public Transport System by the citizenry. To achieve the main objective of the application, it is necessary to integrate route planning analysis, at which point Geographic Information Systems, as part of Information and Communication Technologies (ICT), play an important role in the development of digital platforms that incorporate functionality based on territorial analysis, due to their versatility in exploiting and analyzing georeferenced information. Therefore, the document aims to explain the algorithms implemented for the calculation and determination of routes, as well as to provide information regarding the data structure used as a key input for the development of the same. Consider that the planning of travel routes encompasses the Regular Public Transport System.

Keywords: route planning, geographic information system, information and communication technologies.

I. INTRODUCCIÓN

En concordancia al proceso que permite concretar la denominación profesional de ingeniero geógrafo, bajo la modalidad de suficiencia profesional, el presente documento se estructura bajo los lineamientos del Anexo IV del Reglamento General de Grados y Títulos pertenecientes a la Universidad Nacional Federico Villarreal, el cual ha sido aprobado mediante Resolución Rectoral N°2900-2018-CU-UNFV y cuya fecha es del 25 de junio del 2018.

El presente sustenta la experiencia del autor como profesional Especialista GIS en la Unidad de Tecnologías de la Información (UTI) perteneciente a la Autoridad del Transporte Urbano (ATU), sustento que se da mediante el desarrollo de un proyecto de algoritmos en GIS para la determinación de rutas de viaje usando el Sistema de Transporte Público de carácter regular, actualmente concesionado, en el cual se evalúan posibilidades de viaje utilizando la Línea 01 del Metro de Lima y Callao, los Corredores Complementarios y la red de buses del Metropolitano. Así mismo, en el documento se incluyen datos referentes a la trayectoria académica y profesional del compositor del presente, y breves resúmenes de carácter general y público concernientes a la entidad ATU, así como la mención de algunas funciones y actividades desarrolladas por el profesional en la entidad (ATU - UTI).

La necesidad de implementar el algoritmo en GIS, surge con el requerimiento de construir una aplicación de carácter público desde la cual los usuarios del sistema de transporte público a nivel de Lima y Callao, puedan seleccionar un medio de transporte para realizar un recorrido de inicio a fin. Para la generación del algoritmo de búsqueda de rutas, se ha tomado como referencia documentación técnica de cálculo de rutas óptimas en una red de grafos mediante los algoritmos de Dijkstra cuyo fin es encontrar las rutas óptimas en una red interconectada. La tecnología usada es código abierto para evitar incurrir en costos de licenciamiento.

1.1. Trayectoria del Autor

El autor cuenta con diploma de Bachiller en la carrera profesional de Ingeniería Geográfica impartida por la Universidad Nacional Federico Villarreal, documento emitido el 08 de mayo del 2017. Actualmente, el mismo, desempeña el rol de Especialista GIS para la Unidad de Tecnologías de la Información, oficina perteneciente a la Autoridad del Transporte Urbano.

Durante las siguientes líneas se precisa la trayectoria profesional que ha permitido acercar al compositor del presente hacia la especialidad en GIS. Para lo cual, es necesario mencionar que el primer acercamiento del autor con los Sistemas de Información Geográficos Empresariales se da en el año 2015, cuando cursaba el último ciclo de la carrera profesional, en la Compañía Operadora de Gas del Amazonas & Transportadora de Gas del Perú (COGA & TGP), compañía del Sector Hidrocarburos, en dicha empresa apoyó en el mantenimiento de los datos vectoriales y ráster para la Base de Datos GIS Empresarial, así como en otras actividades tales como el análisis y tratamiento LIDAR, generación de modelados ambientales, elaboración de perfiles longitudinales y transversales, elaboración de Cartografía Temática siendo las más destacadas los mapas de modelados ambientales, mapas de riesgos geológicos, mapas de perfil longitudinal y transversal de terreno, y en la elaboración de recorridos virtuales en 3D referente al Derecho de Vía del Proyecto.

Seguido durante el periodo del 2016, ya como egresado de la carrera profesional, se incorpora al proyecto GEOCATMIN, a cargo del Instituto Geológico, Minero y Metalúrgico (INGEMMET), entidad estatal encargada de la promoción Minera, en la cual pudo conocer la arquitectura tecnológica respecto a la distribución de los componentes de ESRI (ArcGIS), identificando el uso de cada componente y como los mismos permiten tener una orquestación armoniosa de un GIS Empresarial, siendo estos la tecnología base para el Proyecto. Así mismo el autor realizó el modelado de tablas georreferenciadas para la Geodatabase Empresarial,

colaboró en la automatización de los procesos de compilación de datos vectoriales, apoyó en el procesamiento de datos ráster y en la preparación de cartografía geológica. Adicionalmente participó en el proceso de análisis y pruebas sobre los productos de Tecnología ESRI, documentando los flujos y procesos de integración de los mismos.

Luego, para el año 2017 y en proceso de obtener el grado de Bachiller, el autor forma parte de la Compañía de Minas Buenaventura CIA, apoyando al área de Permisos Mineros mediante la elaboración de mapas temáticos para la gestión de los permisos mineros, así como en la composición de cartografía referente a los procesos de la operación minera. En paralelo impulsó la compilación e integración de los datos vectoriales en una Geodatabase Corporativa, para luego poder mostrarlos en aplicaciones WEB construidas mediante interfaces denominadas Asistentes de Configuración de Aplicaciones Web, usando principalmente APP-Builder para la construcción de Visores Web.

A mediados del 2017 y con el grado de Bachiller, recibe la propuesta de integrar el equipo de tecnología orientado a la implementación de aplicaciones empresariales GIS para el Sector Transporte, mediante el organismo público Provias Nacional. En dicha entidad, empieza a incorporar conceptos propios base de datos relacionales, entidad – relación, y metodologías propias de estas tecnologías para la depuración, procesamiento y análisis de datos vectoriales para el tratamiento de los mismos e integrándolos con información oriunda de otros sistemas para mejorar los análisis y generar reportes de mapas. Durante ese periodo el equipo logra implementar la aplicación de uso público denominada Visor de Emergencias Viales, donde se visualizan las emergencias en las vías nacionales y departamentales; producto que le permite al equipo recibir diplomas de reconocimiento por parte del entonces director de la entidad.

Ya para mediados del 2018, quien suscribe el presente, recibe una nueva propuesta para apoyar la implementación de tecnología GIS en el Sector de Agua y Saneamiento, esta vez desde el Organismo Técnico de la Administración de los Servicios de Saneamiento (OTASS).

En dicho organismo, forma parte del equipo dedicado a la generación de ideas tecnológicas disruptivas, aportando con la construcción de productos digitales ágiles enfocados a la recopilación de datos en campo y su visualización desde aplicaciones Web GIS, mediante los Asistentes de Configuración de Aplicaciones Web, usando tecnología propia de ESRI (ArcGIS). Las aplicaciones logran ser mejoradas en apoyo del equipo de tecnología, lo cual permite concebir el primer visor Web GIS denominado GOTASS (Geovisor de OTASS), el cual se desarrolla en tecnología nativa de la Web y usando Servicios de Mapas de ESRI.

Próximo a concluir el año 2018, se genera una nueva oportunidad para el Sector de Transporte, en el rubro del Transporte Eléctrico esta vez desde la entidad en su momento denominada Autoridad Autónoma del Sistema Eléctrico de Transporte Masivo de Lima y Callao (AATE), la cual al presente forma parte de la ATU. En dicha entidad el autor se integra al equipo técnico interdisciplinario para la implementación del proyecto SISPRO-L2 (Sistema de Seguimiento del Proyecto de la Línea 2), apoyando en el proceso de construcción de la aplicación Web GIS SISPRO-L2, la cual contiene módulos web para el mantenimiento de la información. En dicha entidad, el autor asume las funciones de crear, implementar y gestionar los datos vectoriales, así como la orquestación de la Geodatabase Empresarial, los Servicios Web GIS y otros componentes asociados a la arquitectura GIS de ESRI que permiten integrar la información georreferenciada en la aplicación Web GIS SISPRO-L2. Sin embargo, el proyecto SISPRO-L2 queda suspendido al culminar el primer trimestre del año 2020 por la integración AATE hacia la Autoridad del Transporte Urbano.

Para ese momento, el autor recibe la propuesta de conformar parte del equipo de tecnología para del Ministerio de Salud (MINSA). Donde apoya en los procesos de implementación de aplicaciones Web GIS ejerciendo actividades de procesamiento, análisis e integración de las bases de datos georreferenciadas y bases de datos relacionales, así como en el desarrollo de componentes tecnológicos, servicios Rest-API, para el consumo de datos en

las aplicaciones Web GIS. Resultado del esfuerzo del equipo se genera el Visor GIS del MINSA orientado al proceso de Vacunación contra la Covid-19, entre otras herramientas Web GIS de uso interno de la entidad, aplicaciones que no llegan a tomar la relevancia como el Carné de Vacunación por las propias circunstancias de la pandemia. En paralelo, apoya al Sector Hidrocarburos con la implementación del Visor Web GIS denominado GEOPETRO, para el Vice-Ministerio de Hidrocarburos perteneciente al Ministerio de Energía y Minas (MINEM), a través de la entidad CAREC, visor que luego sería repotenciado al incluir personal de desarrollo para el mantenimiento perfectivo de la aplicación.

Es así como concluido el periodo de Covid-19, y para terminar el año 2022 recibe la propuesta de retornar a la Autoridad del Transporte Urbano (ATU) para repotenciar el proyecto SISPRO-L2. Sin embargo, durante el proceso es asignado a formar parte de la oficina denominada Unidad de Tecnologías de la Información (UTI) para apoyar al equipo de desarrollo de software en el proceso de construcción de una aplicación móvil, denominada ATU Destino, para el cálculo de rutas usando el sistema de transporte urbano. Cabe precisar que en colaboración del equipo técnico de tecnología se logra repotenciar el SISPRO-L2, construyendo nuevas aplicaciones Web GIS para la gestión de proyectos del Sistema Integrado de Transporte, bajo una nueva plataforma denominada SISPRO.

1.2. Descripción de la Empresa

Como entidad en referencia para el sustento de Suficiencia Profesional, se precisa que será la Autoridad del Transporte Urbano (ATU) cuyo domicilio fiscal se encuentra en la Calle José Gálvez N°550, distrito de Miraflores, provincia de Lima y departamento de Lima, con RUC 20604932964 y con Número de Identificación 001717.

La Autoridad del Transporte Urbano, en adelante ATU, se crea bajo la Ley N°30900 constituyéndose como un organismo de carácter técnico perteneciente al Pliego del Ministerio de Transporte y Comunicaciones (MTC), y es producto de la transferencia de las Gerencias de Transporte Urbano (GTU) de la Municipalidad Metropolitana de Lima y de la Gerencia General de Transporte Urbano (GGTU) de Callao, y de la fusión por absorción del Instituto Metropolitano Protransporte de Lima (PROTRANSPORTE) y de la Autoridad Autónoma del Sistema Eléctrico de Transporte Masivo de Lima y Callao (AATE).

La ATU tiene como fin gestionar, supervisar, fiscalizar y promover de manera eficiente el sistema de transporte integrado buscando brindar calidad en los servicios de transporte público ofertados a nivel de Lima y Callao.

1.2.1. Misión

La entidad tiene como misión establecer y gestionar el sistema de transporte público a nivel de Lima y Callao bajo un enfoque integral, caracterizado por contar con servicios de calidad asegurando la eficiencia, sostenibilidad, accesibilidad y seguridad de los mismo. Esto implica realizar acciones de planificación y coordinación de los diversos modos de transporte, así como la implementación de un sistema de recaudo unificado para el transporte masivo.

1.2.2. Visión

La visión de la entidad es ser reconocida como una autoridad líder respecto a la gestión del transporte público a nivel de Lima y Callao, siendo un modelo de gestión a nivel nacional, así como para Latinoamérica. Para lo cual, se encuentra en proceso de implementar diferentes

metodologías y marcos legales que le permitan gestionar el sistema de transporte público desde un enfoque integral, para asegurar que el sistema de transporte sea:

- Moderno y Eficiente.
- Centrado en sus usuarios.
- Sostenible tanto económica como ambientalmente.
- Accesible e inclusivo para todos.
- Seguro.

1.2.3. Funciones

La entidad desempeña un rol clave en la organización y regulación del sistema de transporte público en Lima y Callao, para lo cual cumple con la ejecución de las siguientes funciones:

- **Planificar y regular:** Diseña planes de movilidad, establece normas y regulaciones para los servicios de transporte.
- **Gestionar y supervisar:** Opera el Sistema Integrado de Transporte y el Sistema de Recaudo Único, y supervisa el cumplimiento de las normas por parte de los operadores.
- **Fiscalizar y sancionar:** Vela por el cumplimiento de la normativa y aplica sanciones cuando sea necesario
- **Invertir y financiar:** Promueve y ejecuta inversiones en infraestructura y servicios de transporte
- **Comunicar y educar:** Informa a la ciudadanía sobre el sistema de transporte y promueve una cultura de movilidad sostenible.

1.2.4. Organigrama

Bajo la Ley N°30900, el organismo técnico (ATU) tiene como objetivo garantizar el funcionamiento de manera eficiente, sostenible, accesible, seguro e integral del Sistema de Transporte a nivel de Lima y Callao, para satisfacer la necesidad de traslado de la población propia de la conurbación entre las provincias de Lima y Callao. Por ende, para lograr cumplir lo señalado, la ATU se conforma principalmente por:

La Presidencia Ejecutiva, máxima autoridad de carácter ejecutivo, cuya responsabilidad es conducir y supervisar el funcionamiento de la entidad.

La Gerencia General, máxima autoridad de carácter administrativo, responsable de la conducción, coordinación y supervisión de la gestión de los sistemas administrativos. Así como asesorar al presidente de la entidad en la gestión administrativa de la misma.

Los Órganos de Línea, encargados de dar soporte técnico orientado al cumplimiento de los objetivos de la entidad. Los cuales son:

Dirección de Integración de Transporte Urbano y Recaudo.

Dirección de Infraestructura.

Dirección de Supervisión de Proyectos.

Dirección de Operaciones.

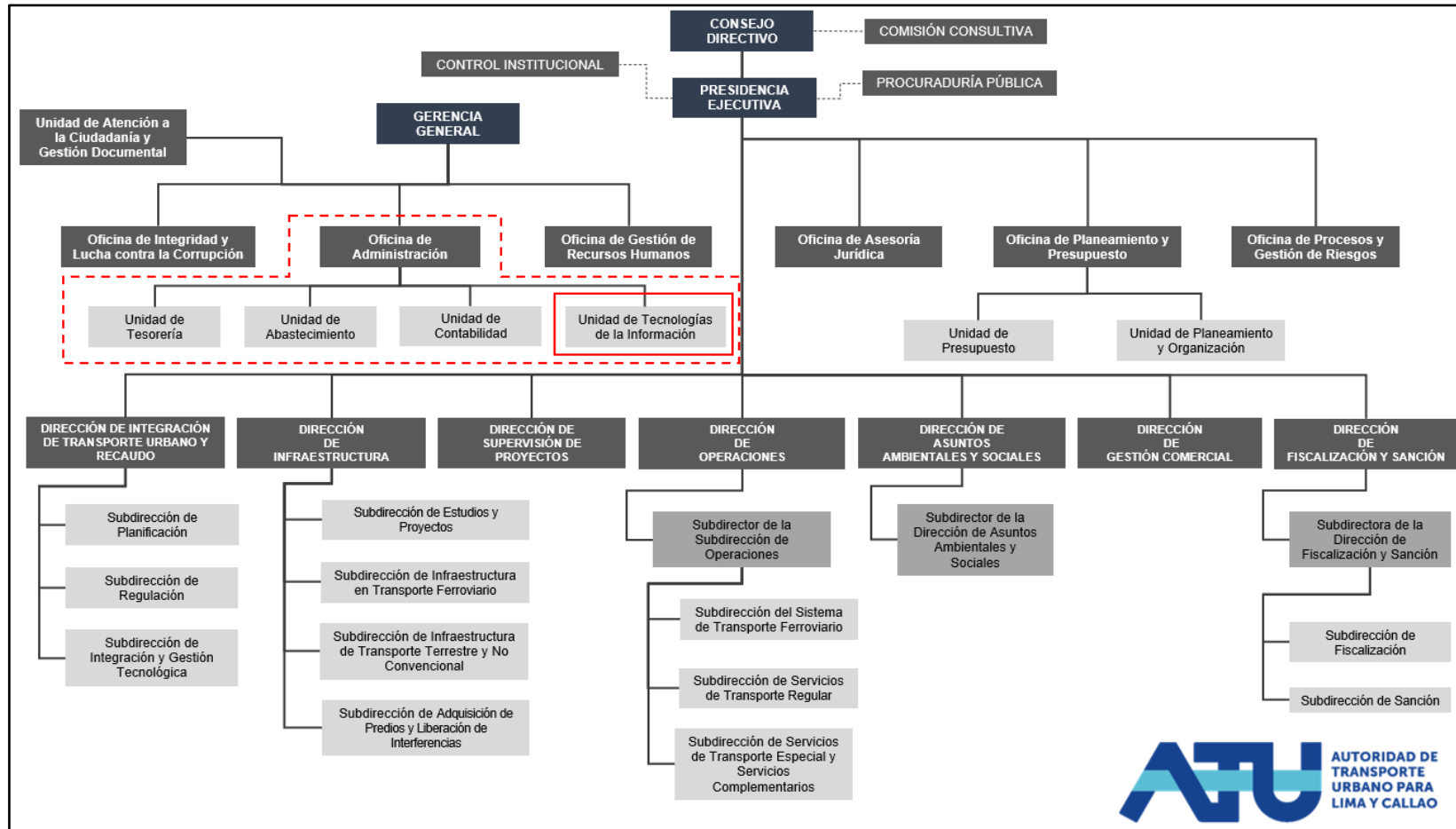
Dirección de Asuntos Ambientales y Sociales.

Dirección de Gestión Comercial.

Dirección de Fiscalización y Sanción.

Figura 1

Organigrama de la ATU



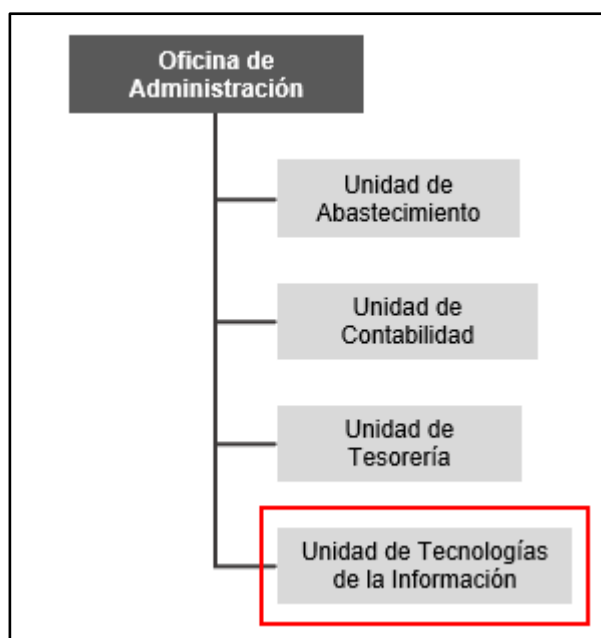
Nota. Resolución de Presidencia Ejecutiva 193-2023-ATU/PE (ROF - ATU)

1.2.5. Áreas y Funciones Desempeñadas

La Unidad de Tecnologías de la Información, en adelante UTI, perteneciente a la ATU, forma parte de la Oficina de Administración como unidad orgánica encargada de velar por la disponibilidad de los recursos informáticos, brindando soporte a los sistemas de información e infraestructura tecnológica y de comunicaciones, siendo un soporte transversal para el funcionamiento de la ATU.

Figura 2

Detalle de la Oficina de Administración (Organigrama de la ATU)



Nota. ATU – ROF (Organigrama)

La UTI tiene como funciones:

- Proponer, formular y evaluar el Plan de Gobierno Digital, considerando los objetivos de la entidad y en función de las necesidades de la Alta Dirección y de las otras áreas de la organización.
- Proponer, diseñar y gestionar la normativa referente a la Seguridad de los Sistemas de Información, y otros componentes informáticos.
- Diseñar, actualizar y dar mantenimiento del Portal Web Institucionales y otras

plataformas digitales.

- Brindar soporte técnico a los usuarios en referencia a los recursos informáticos.
- Asumir funciones según designación del jefe de la Oficina de Administración, y por aquellas que se indique mediante normativa expresa.

Quien suscribe, presta servicios profesionales a la ATU a través de la UTI, a la fecha del presente documento, desempeñando el cargo de especialista en GIS. Teniendo como responsabilidades:

- Gestionar las Bases de Datos Georreferenciada.
- Integración y Análisis de Datos Georreferenciados y Relacionales.
- Automatización de procesos de Análisis de Datos.
- Generación de Servicios Interoperables.
- Apoyo en los Procesos de Desarrollo e Implementación de Aplicaciones Georreferenciadas.
- Apoyo en la generación y propuesta de mejoras respecto a las tecnologías referentes a los Sistemas de Información Geográficos.

II. DESCRIPCIÓN DE UNA ACTIVIDAD ESPECÍFICA

En el contexto actual de la creciente urbanización y movilidad, la optimización de los sistemas de transporte público se ha convertido en un desafío crucial. La planificación de rutas eficientes, que minimicen los tiempos de viaje y maximicen la accesibilidad, es fundamental para garantizar una movilidad sostenible y mejorar la calidad de vida de los ciudadanos.

Sin embargo, la complejidad de los sistemas de transporte público, caracterizados por múltiples rutas, horarios y transferencias, dificulta la tarea de encontrar la mejor ruta para un determinado viaje. Las herramientas existentes, a menudo limitadas en su alcance o en su capacidad de procesamiento, no siempre satisfacen las necesidades de los usuarios y de los planificadores urbanos.

El presente proyecto describe el proceso de desarrollo de una herramienta de planificación de rutas basada en algoritmos eficientes y datos abiertos, capaz de generar soluciones personalizadas y optimizadas para los usuarios del transporte público. Este proyecto tiene como finalidad sustentar la suficiencia profesional del autor.

Como caso de estudio, se ha tomado el Sistema de Transporte Público Regular, donde se utilizará el algoritmo de Dijkstra. Este algoritmo, fundamentado en la Teoría de Grafos, ha sido ampliamente integrado en herramientas GIS debido a su eficacia en cálculos de rutas más cortas y análisis de redes. Las herramientas GIS utilizan la teoría de grafos y algoritmos como Dijkstra para realizar análisis de proximidad, análisis de redes y otros referentes al manejo de redes. La herramienta modela la red de transporte como un grafo y calcula la ruta más corta entre dos puntos, considerando los servicios de transporte público brindados por los sistemas de transporte regular.

Es necesario precisar que el proyecto forma parte de una solución digital integrada para la planificación de rutas de viaje, la cual a la fecha del presente documento es una aplicación móvil para plataformas móviles (IOS & Android) de acceso es público en versión Beta.

2.1. Listado de Términos

Grafo: Estructura que se utiliza para representar relaciones entre objetos, formada por puntos (nodos) conectados por líneas (arcos).

Nodos o Vértices: Los puntos o entidades en un grafo que están conectados por arcos.

Arcos o Aristas: Las conexiones o enlaces entre los nodos en un grafo, que pueden ser dirigidos o no dirigidos.

Grados: Número de arcos que inciden en un nodo en un grafo.

Ciclos en los Grafos (Circuito Cerrado): Una secuencia de arcos en un grafo que comienza y termina en el mismo nodo sin repetir arcos.

Teorema: Declaración matemática que ha sido probada basándose en axiomas y otras declaraciones ya demostradas.

NP: Clase de problemas de decisión para los cuales una solución propuesta puede ser verificada en tiempo polinomial.

NP-completo: Conjunto de problemas que son al menos tan difíciles como los más difíciles en NP, y para los cuales cualquier problema en NP puede transformarse en ellos en tiempo polinomial.

Modelo de Negocio: Plan que describe cómo una organización crea, entrega y captura valor, principalmente en términos económicos.

Aplicación: Un programa informático diseñado para realizar una función específica o un conjunto de funciones. Las aplicaciones pueden ser tan simples como una calculadora o tan complejas como un procesador de texto o un videojuego.

TIC (Tecnologías de la Información y Comunicación): Conjunto de herramientas, tecnologías y recursos utilizados para procesar, almacenar y transmitir información, especialmente en formato digital.

GIS (Sistemas de Información Geográfica): Un sistema diseñado para capturar,

almacenar, analizar, gestionar y presentar todo tipo de datos georreferenciados, es decir, datos vinculados a una ubicación geográfica específica.

Algoritmo: Conjunto de instrucciones ordenadas y finitas que se siguen para resolver un problema o realizar una tarea específica.

Geoespacial: Relacionado con la posición y características de los objetos en la Tierra.

Estructura de Datos: Forma en la que se organizan, gestionan y almacenan los datos para facilitar su acceso y modificación eficiente.

Datos Estructurados: Datos organizados en un formato fijo, como tablas con filas y columnas, lo que facilita su análisis.

Datos Semi Estructurados: Datos que no se ajustan a un formato tabular rígido pero que tienen etiquetas u otras marcas que ayudan a darles cierta estructura.

Datos No Estructurados: Datos que no tienen un formato predefinido, como texto, imágenes, audio o video. Son más difíciles de analizar pero a menudo contienen información valiosa.

Listas: Una colección ordenada de elementos.

Tuplas: Una colección ordenada de elementos, similar a una lista, pero inmutable (no se puede cambiar).

Análisis de Datos: Proceso de examinar, limpiar y modelar datos con el objetivo de descubrir información útil, llegar a conclusiones y apoyar la toma de decisiones.

Lenguaje de Programación: Sistema de comunicación utilizado para escribir instrucciones que una computadora puede ejecutar.

Librería de Programación: Conjunto de funciones y recursos reutilizables que ayudan a los desarrolladores a escribir código de manera más eficiente.

API: Conjunto de reglas y definiciones que permite que diferentes aplicaciones se comuniquen entre sí.

Script: un pequeño programa diseñado para automatizar tareas simples.

Repositorio de Datos: Lugar donde se almacenan y gestionan datos de manera centralizada, permitiendo su acceso y análisis.

Base de Datos: Una colección organizada de datos, estructurada para facilitar su acceso, gestión y actualización.

Esquema de Base de Datos: Estructura que define la organización de datos en una base de datos, incluyendo las tablas, campos y relaciones entre ellos.

Aplicación Back-End: Parte del software que maneja la lógica, base de datos, y la gestión del servidor, trabajando detrás de escena.

Aplicación Front-End: Parte del software que interactúa directamente con los usuarios, mostrando información y aceptando entradas.

Código Fuente: Conjunto de instrucciones y declaraciones escritas por un programador en un lenguaje de programación, que define el comportamiento y las funcionalidades de una aplicación.

SIT: Sistema Integrado de Transporte.

UTI: Unidad de Tecnologías de la Información.

DSP: Dirección de Supervisión de Proyectos.

ATU: Autoridad de Transporte Urbano.

2.2. Antecedentes del Proyecto

La conurbación existente entre los distritos de Lima y Callao concentra aproximadamente la tercera parte de la población del Perú. Así mismo, la capital del país representa un gran centro negocios acogiendo a grandes empresas, instituciones internacionales, entidades públicas referentes a la administración estatal, y en adicional al puerto y aeropuerto ambos de carácter internacional. A su vez, el centro histórico de la capital ha sido declarado Patrimonio Mundial de la Humanidad por la UNESCO donde convergen diferentes centros turísticos, lo cual ha propiciado que se vuelve el segundo destino turístico del Perú, luego de Cusco.

Figura 3

Proyección del 2021 de la población de Lima Metropolitana

DISTRITO	POBLACIÓN ESTIMADA	DISTRITO	POBLACIÓN ESTIMADA
San Juan de Lurigancho	1,203,125	San Borja	129,095
San Martín de Porres	758,216	Lurín	111,789
Ate	687,615	Miraflores	111,202
Comas	581,036	Surquillo	101,448
Villa María del Triunfo	443,687	Pueblo Libre	95,809
Villa El Salvador	428,276	Breña	94,209
San Juan de Miraflores	417,800	Ancón	87,091
Santiago de Surco	414,142	Jesús María	82,842
Carabayllo	412,234	San Isidro	68,604
Puente Piedra	404,081	Magdalena del Mar	66,464
Chorrillos	361,907	Lince	60,647
Los Olivos	355,528	San Luis	56,068
Lurigancho-Chosica	293,377	Chaclacayo	44,634
Lima	267,199	Santa Rosa	40,030
El Agustino	225,749	Cieneguilla	39,633
Independencia	225,536	Barranco	36,200
Santa Anita	224,893	Punta Hermosa	22,855
La Victoria	189,152	Pucusana	17,142
Rímac	181,613	San Bartolo	9,003
San Miguel	177,164	Punta Negra	8,531
La Molina	162,761	Santa María del Mar	1,180

Fuente: INEI

Nota. Entidad Pública INEI

Figura 4*Proyección del 2021 de la población de Callao*

DISTRITO	POBLACIÓN ESTIMADA	DISTRITO	POBLACIÓN ESTIMADA
Callao	520,709	Mi Perú	53,557
Ventanilla	380,894	Carmen de la Legua Reynoso	46,772
Bellavista	81,122	La Punta	3,902
La Perla	64,524		

Fuente: INEI

Nota. Entidad Pública INEI

Es así como los puntos mencionados fortalecen el estatuto de Lima como nodo regional, el cual se expresa en una gran densidad poblacional respecto a la extensión territorial de los distritos de Lima y Callao.

Figura 5*Extensión y densidad poblacional durante el 2021 para Lima Metropolitana*

DISTRITO	KM ²	HAB/KM ²	DISTRITO	KM ²	HAB/KM ²	DISTRITO	KM ²	HAB/KM ²
Breña	3.2	29,440	San Borja	10	12,910	Santa Rosa	21.5	1,862
Surquillo	3.5	28,985	Lima	22	12,145	Lurigancho-Chosica	236.5	1,240
Pueblo Libre	4.4	21,775	Villa El Salvador	35.5	12,064	Carabaylo	346.8	1,188
La Victoria	8.7	21,742	Comas	48.8	11,906	Chaclacayo	39.5	1,130
Santa Anita	10.7	21,018	Santiago de Surco	34.8	11,901	Pachacamac	160.2	919
San Martín de Porres	36.9	20,548	Miraflores	9.6	11,584	Lurín	181.1	617
Lince	3	20,216	Barranco	3.3	10,970	Pucusana	37.8	453
Los Olivos	18.3	19,428	Chorrillos	38.9	9,304	Ancón	299.2	291
Magdalena del Mar	3.6	18,462	San Juan de Lurigancho	131.3	9,163	San Bartolo	45	200
El Agustino	12.5	18,060	Ate	77.7	8,850	Punta Hermosa	119.5	191
Jesús María	4.6	18,009	Villa María del Triunfo	70.6	6,285	Cieneguilla	240.3	165
San Juan de Miraflores	24	17,408	San Isidro	11.1	6,181	Santa María del Mar	9.8	120
San Miguel	10.7	16,557	Puente Piedra	71.2	5,675	Punta Negra	130.5	65
San Luis	3.5	16,019	La Molina	65.8	2,474	Lima Metropolitana		3,685
Independencia	14.6	15,448						
Rímac	11.9	15,262						

Fuente: INEI

Nota. Entidad Pública INEI

Figura 6

Extensión y densidad poblacional durante el 2021 para Callao

DISTRITO	KM ²	HAB/KM ²	DISTRITO	KM ²	HAB/KM ²
La Perla	2.8	23,044	Callao	45.7	11,394
Mi Perú	2.4	22,315	Ventanilla	73.5	5,182
Carmen de la Legua Reynoso	2.1	22,272	La Punta	0.8	4,878
Bellavista	4.6	17,635	Provincia del Callao		8,730

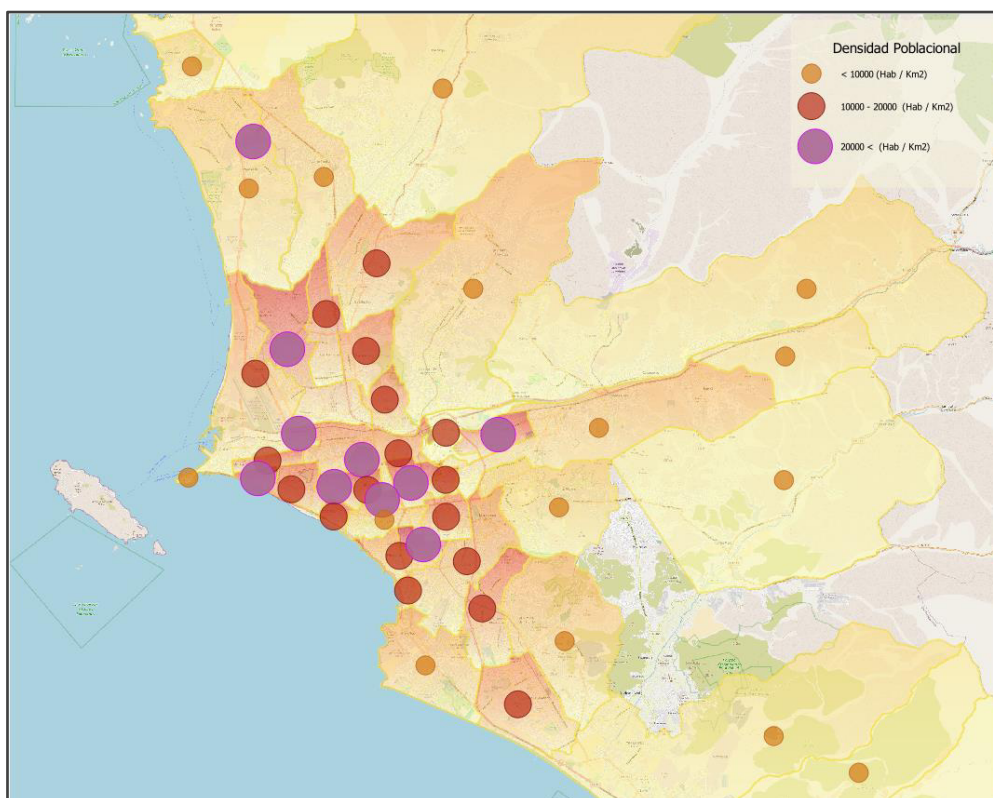
Fuente: INEI

Nota. Entidad Pública INEI

Sin embargo, a pesar de la gran densidad poblacional se sabe bien que Lima y Callao han crecido con falta de planificación territorial impactando el desarrollo urbano sostenible.

Figura 7

Mapa de densidad poblacional a nivel de Lima y Callao

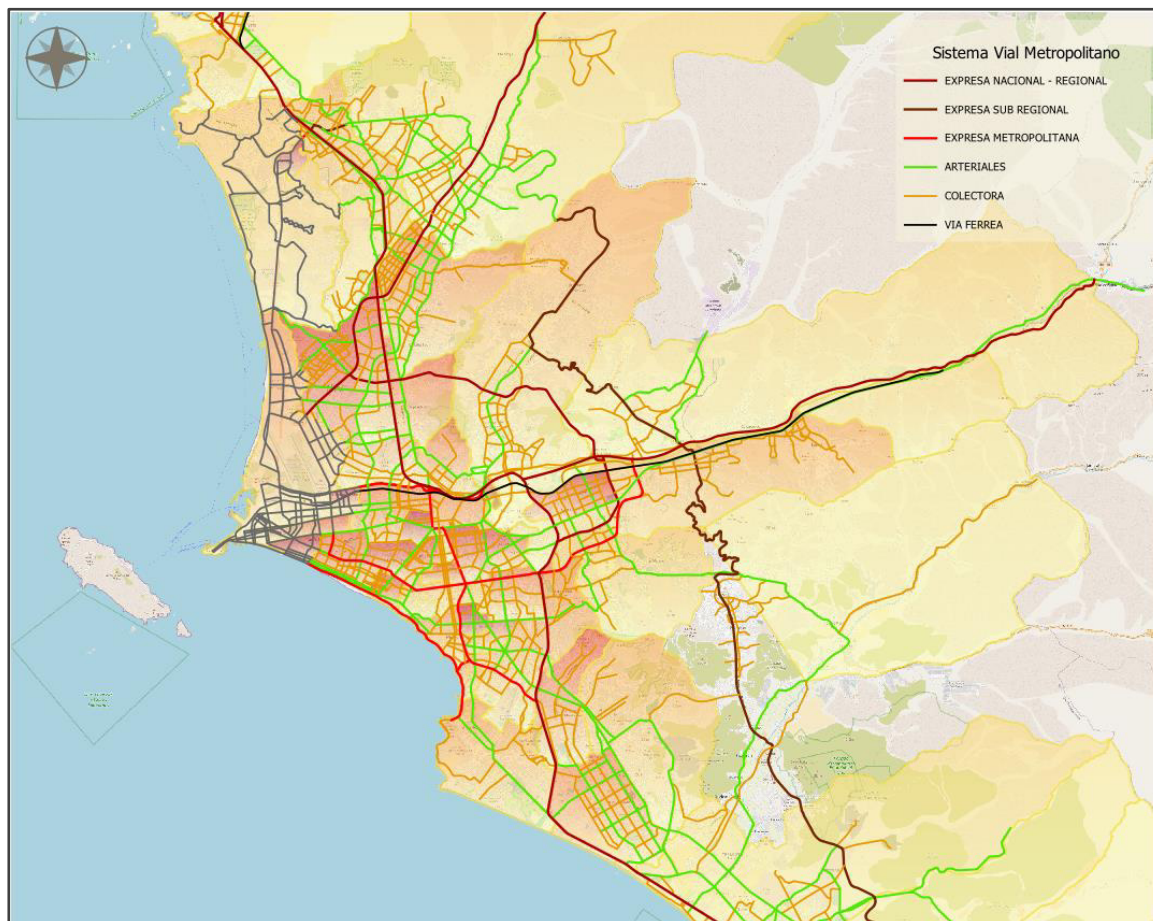


Nota. Elaboración propia con datos de Entidad Pública INEI

Prueba de ello es la dificultosa labor para mejorar y optimizar la infraestructura vial de las vías por las cuales transita a diario los vehículos motorizados.

Figura 8

Mapa de vías de Lima y Callao.





Nota. Elaboración propia con los Servicios de Mapas de la Entidad Pública IMP

En suma, de lo mencionado, la creciente flota del parque automotor ha orillado a la Capital al ranking de las peores ciudades con mayores tiempos de viaje. Según la multinacional Tomtom, Lima figura como una de las 10 peores ciudades a nivel mundial con mayor tiempo de viaje y la primera en América del Sur.

Figura 9



Ranking de las ciudades con mayor tráfico a nivel Mundial en el 2022

Clasificar por filtro	Categoría mundial ▼	Ciudad	Tiempo medio de viaje cada 10 km. ▼	Cambio a partir de 2021 ▼	Tiempo en hora punta al año	Velocidad media en hora punta
1	1	Londres  Reino Unido	36 min 20 s	+ 1 min 50 s	325 horas	14 kilómetros por hora
2	2	Bangalore  India	29 min 10 s	+ 40 segundos	260 horas	18 kilómetros por hora
3	3	Dublín  Irlanda	28 min 30 s	+ 1 min 40 s	277 horas	17 kilómetros por hora
4	4	Sapporo  Japón	27 min 40 s	+ 50 segundos	240 horas	19 kilómetros por hora
5	5	Milán  Italia	27 min 30 s	- 20 segundos	259 horas	18 kilómetros por hora
6	6	Pune  India	27 min 20 s	+ 1 min 10 s	249 horas	19 kilómetros por hora
7	7	Bucarest  Rumania	27 min 20 s	- 10 segundos	277 horas	17 kilómetros por hora
8	8	Lima  Perú	27 min 10 s	+ 2 minutos	254 horas	18 kilómetros por hora
9	9	Manila  Filipinas	27 minutos	+ 40 segundos	241 horas	20 kilómetros por hora
10	10	Bogota  Colombia	26 min 20 s	+ 50 segundos	249 horas	19 kilómetros por hora

Nota. Empresa Privada Internacional Tomtom

Figura 10

Ranking de las ciudades con mayor tráfico a nivel de América del Sur en el 2022

Rank by filter	World rank ▼	City	Average travel time per 10 km ▼	Change from 2021 ▼	Time in rush hour per year	Average speed in rush hour
1	8	Lima  Peru	27 min 10 s	+ 2 min	254 hours	18 km/h
2	10	Bogota  Colombia	26 min 20 s	+ 50 s	249 hours	19 km/h
3	16	Buenos Aires  Argentina	24 min 40 s	+ 20 s	217 hours	22 km/h
4	26	Montevideo  Uruguay	23 min	+ 20 s	197 hours	24 km/h
5	27	Recife  Brazil	22 min 50 s	+ 40 s	220 hours	21 km/h
6	35	Sao Paulo  Brazil	22 min 10 s	+ 1 min 20 s	209 hours	22 km/h
7	37	Curitiba  Brazil	22 min	+ 2 min	207 hours	22 km/h
8	39	Belo Horizonte  Brazil	22 min	+ 1 min	214 hours	21 km/h
9	44	Fortaleza  Brazil	21 min 30 s	+ 40 s	205 hours	23 km/h
10	62	Porto Alegre  Brazil	20 min 20 s	+ 40 s	196 hours	24 km/h

Nota. Empresa Privada Internacional Tomtom

Figura 11*Horas Punta durante el 2009 al 2019*

Nota. Empresa Privada Internacional TEC Corporation

Respecto al parque automotor, el MTC estimó en el 2020 que los vehículos en el departamento de Lima representaban el 66% del nivel nacional. Y a nivel de Lima el 15.6% del total de vehículos es conformado por vehículos registrados para transporte público.

Figura 12*Parque Automotor en Lima en el 2020*

Parque automotor del departamento de Lima, 2020		
TIPO DE VEHÍCULO	UNIDADES	%
Automóvil	884,965	43.7%
Station wagon	382,925	18.9%
Camioneta rural (combi)	255,763	12.6%
Camioneta pick up	196,133	9.7%
Camión	124,340	6.1%
Ómnibus	62,549	3.1%
Remolque/ Semiremolque	42,396	2.1%
Remolcador	33,395	1.6%
Camioneta panel	32,761	1.6%
Total	2'025,227	100%

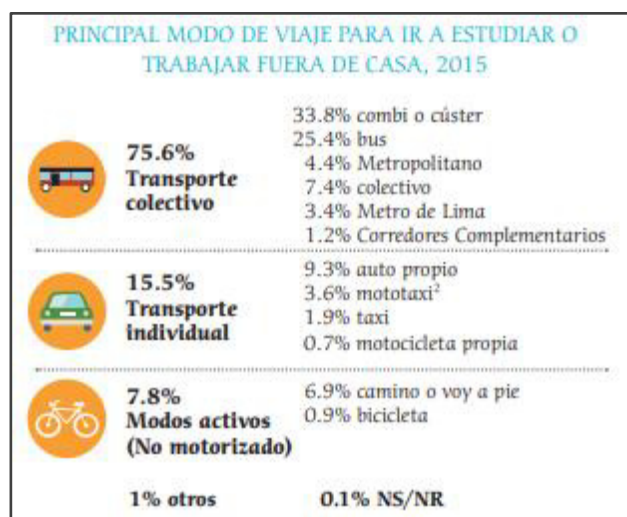
Fuente: MTC

Nota. Entidad Pública MTC

Así mismo, estudios realizados por el Observatorio Lima Cómo Vamos, pre y post pandemia Covid-19, revelaron que el uso del transporte público tradicional es el principal modo de viaje. Pues según estudios previos, durante el periodo 2015, realizados por el mismo Observatorio, se determinó que el 75.6% de personas que estudian o trabajan fuera de casa usan el medio de transporte masivo. En complemento, se realizó otro estudio durante el 2021 por el mismo Observatorio donde se reafirma que los medios de transporte públicos siguen siendo el medio de viaje con mayor preferencia por los usuarios.

Figura 13

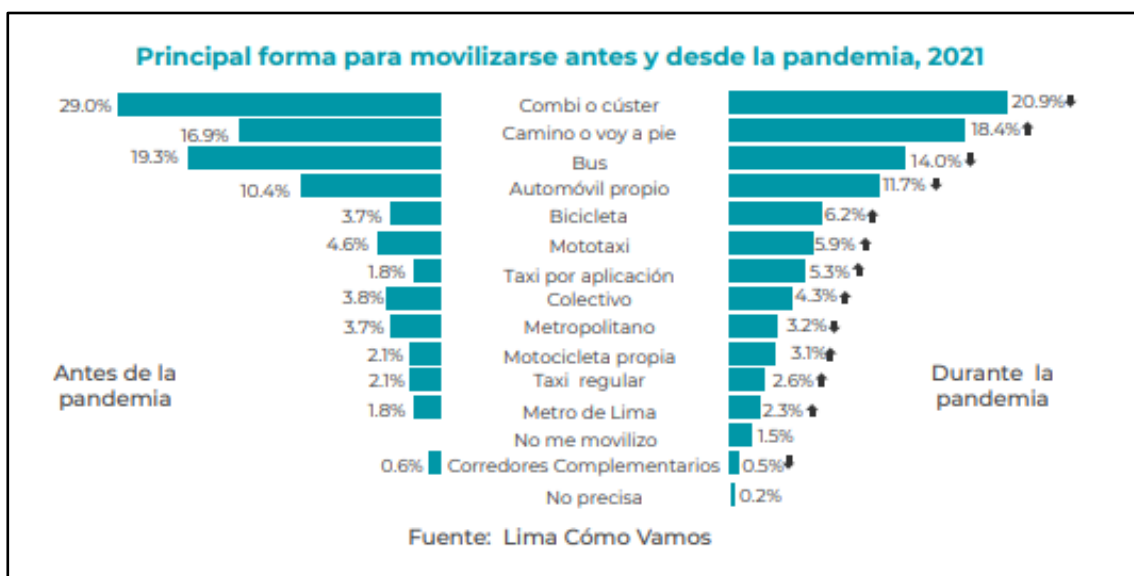
Principal modo de viaje en Lima para ir a estudiar o trabajar durante el 2015



Nota. Entidad Privada Lima Como Vamos

Figura 14

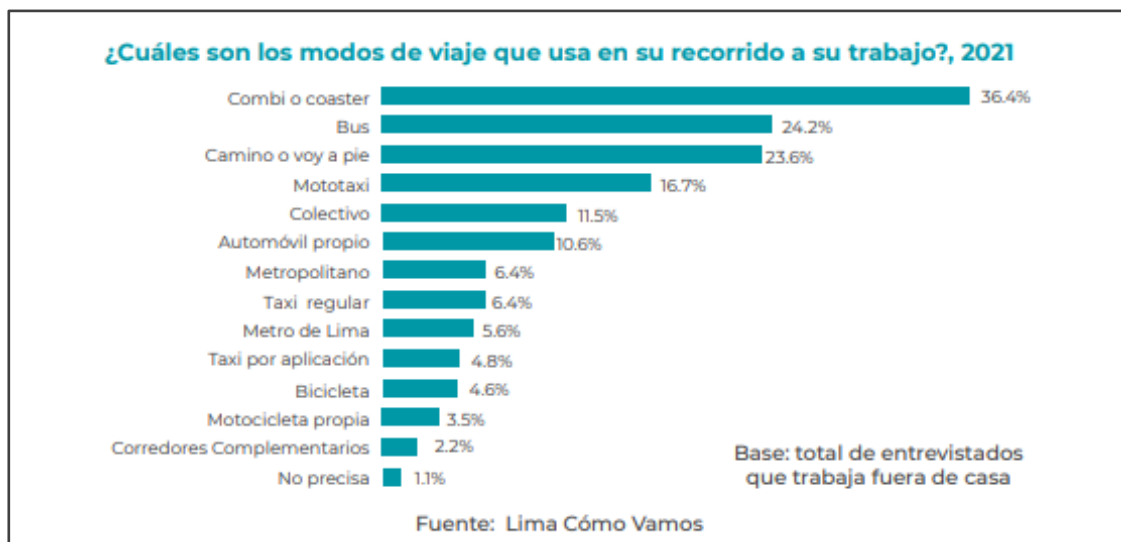
Principal forma de movilizarse pre y post pandemia durante el 2021



Nota. Entidad Privada Lima Como Vamos

Figura 15

Preferencia de medio de transporte para ir a laborar en el 2021



Nota. Entidad Privada Lima Como Vamos

2.3. Planteamiento del Problema

La ATU como Organismo Técnico adscrito al MTC, tiene como función organizar, implementar y gestionar el Sistema de Transporte Urbano de Lima y Callao. Para lo cual, a la fecha, se encuentra en proceso de integrar los Medios de Transporte Regular y Convencional con el fin de contar con un Sistema Integrado de Transporte a nivel de Lima y Callao.

Se debe entender como medio de Transporte Regular al Sistema conformado por los medios de viaje, actualmente concesionados, los cuales son:

- Sistema de Metro de Lima y Callao, conformado por las líneas de metro que articularán la metrópoli y del cual se encuentra en operación la Línea 01 y parte de la Línea 02.
- Sistema del Metropolitano. Cuyas redes operativas son la red Troncal, y las redes de Alimentadores al Norte y Sur de la Capital.
- Corredores Complementarios. El cual se encuentra operando mediante la agrupación de las redes denominadas Corredores Azul, Corredor Rojo y Corredor Morado.

Y como medio de Transporte Convencional, a la agrupación de las distintas líneas de servicio público conformadas por Ómnibus, Minibuses y Microbuses, las cuales son pertenecientes a diferentes empresas de transporte tales como el Grupo Empresarial El Rápido SA, conocido como "El Rápido" y otras más.

Por lo mencionado, en la actualidad, existen diversas líneas de transporte público que ofrecen servicios de movilidad hacia los distintos puntos de la ciudad, lo cual se traduce en diferentes posibilidades y alternativas de viaje. Sin embargo, al considerar los prolongados tiempos de viaje por la congestión vehicular y el bajo nivel de planificación para una adecuada implementación de la infraestructura vial es importante brindar a los usuarios del transporte público herramientas que les permitan optimizar los tiempos de viaje mediante una

planificación eficiente del mismo.

En consecuencia, de lo precisado en párrafos anteriores y en ejercicio de las funciones asignadas, la ATU ha identificado como parte del problema que afecta la calidad del servicio de transporte público, el bajo conocimiento y planificación en los medios de viaje de los usuarios. Por ende, en cumplimiento de sus obligaciones como Organismo Técnico del MTC, se ha visto en la necesidad de elaborar una herramienta tecnológica de carácter público que permita realizar la planificación adecuada de viajes a los usuarios del Sistema de Transporte Público, desde la cual el usuario pueda consultar la ruta más rápida entre dos puntos de origen y destino; para la primera etapa del desarrollo de la aplicación, la planificación de viajes se realizará tomando como base el transporte público regular por tener mayor formalidad en el desarrollo del mismo.

2.4. Objetivos

2.4.1. *Objetivo General*

- Implementar un planificador de rutas para el Sistema de Transporte Regular mediante la aplicación de la teoría de grafos.

2.4.2. *Objetivos Específicos*

- Construir el modelo de datos que permita contemplar la dinámica de la información georreferenciada del Sistema de Transporte Público Regular.
- Desarrollar los scripts que permitan manipular la información georreferenciada para el análisis, procesamiento e integración de la misma.
- Construir el buscador de rutas con los datos del Sistema de Transporte Regular mediante algoritmos basados en la teoría de grafos.
- Validar la factibilidad de realizar la integración del buscador de rutas con aplicaciones digitales.

2.5. Alcance del Proyecto

El algoritmo para la consulta de rutas óptimas se implementará tomando como referencia el ámbito por el cual circula el transporte regular de Lima y Callao.

2.5.1. Sistema de Transporte Público Regular

La información base tomada para la implementación del algoritmo es referente al sistema de transporte público regular, el cual está compuesto por el Sistema de Metro de Lima y Callao, Sistema de Corredores Complementarios, Sistema del Metropolitano y Alimentadores.

2.5.2. Ámbito

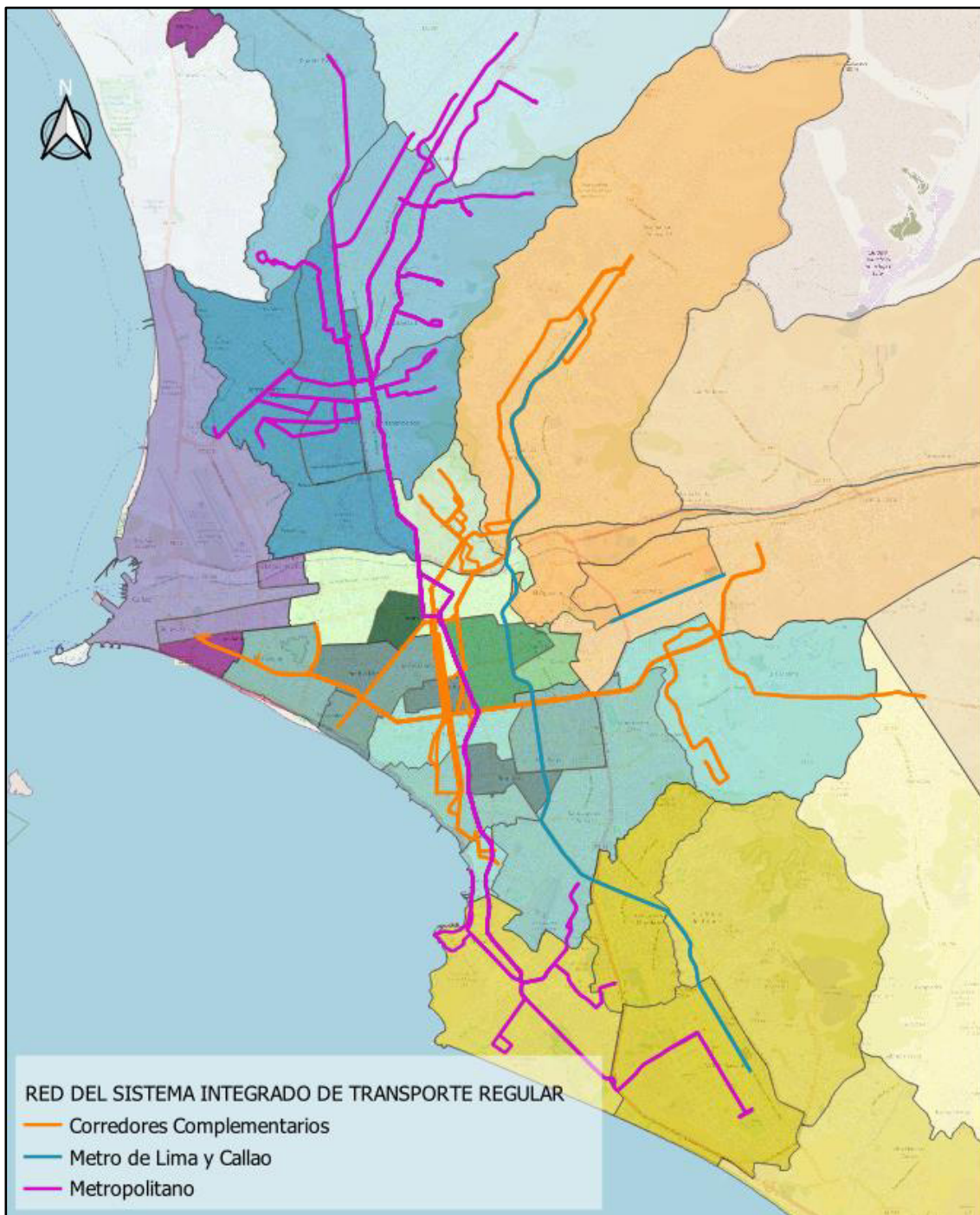
El ámbito es determinado por los distritos que se interconectan mediante cada Sistema del Transporte Regular, siendo estos los que conforman el área de Lima Metropolitana.

2.5.3. Población

La población está determinada por los usuarios con facilidades tecnológicas que usan los Sistemas de Transporte Regular.

Figura 16

Mapa de la red del Sistema de Transporte Regular y ámbito según densidad poblacional



Nota. Elaboración propia

2.6. Marco Teórico

2.6.1. Base Legal

La ATU Autoridad del Transporte Urbano (ATU), es una entidad pública concebida por la necesidad de optimizar la gestión sobre el Transporte Urbano a nivel de Lima y Callao. La cual está conformada por la integración de funciones, y por ende de recursos, de diferentes gerencias y entidades públicas asociadas a la administración del transporte en Lima y Callao, siendo estas las siguientes:

- Gerencia de Transporte Urbano (GTU) de la Municipalidad Metropolitana de Lima.
- Gerencia General de Transporte Urbano (GGTU) de la Municipalidad Provincial del Callao.
- Autoridad Autónoma del Sistema Eléctrico de Transporte Masivo de Lima y Callao (AATE).
- Protransporte

2.6.1.1. Base Legal referente al SIT

Ley 30900.

- Fecha de publicación: 28 de diciembre del 2018.
- Descripción: Esta ley es la norma fundamental que establece la creación de la ATU como una entidad pública descentralizada, adscrita al Ministerio de Transportes y Comunicaciones (MTC). Su objetivo principal es integrar, planificar, regular, gestionar y supervisar el sistema de transporte público en Lima y Callao.

Decreto Supremo N°003-2019-MTC.

- Fecha de publicación: 21 de enero de 2019.
- Descripción: El Decreto Supremo en cuestión, especifica las funciones, estructura organizativa y procedimientos que debe seguir la ATU para cumplir con su mandato. Define también la coordinación que debe mantener la ATU con otras

entidades públicas y privadas.

Resolución Ministerial N°009-2019-MTC/01.

- Fecha de publicación: 13 de febrero de 2019.
- Descripción: La Resolución Ministerial en mención, detalla las funciones específicas de las unidades de organización de la ATU a partir del tercer nivel organizacional

Decreto Supremo N°005-2019-MTC.

- Fecha de publicación: 08 de marzo de 2019.
- Descripción: Este decreto reglamenta la Ley N°30900, especificando las competencias y funciones asignadas a la ATU como entidad encargada del Sistema Integrado de Transporte de Lima y Callao (SIT), y a los servicios complementarios para contar con un sistema de transporte intermodal.

Decreto Legislativo N°1498.

- Fecha de publicación: 10 de mayo de 2020.
- Descripción: Este decreto legislativo, emitido durante la pandemia de COVID-19, refuerza las capacidades de la ATU para gestionar el transporte en situaciones de emergencia, asegurando la continuidad y seguridad del servicio. También se enfoca en la reestructuración y mejora continua del SIT, adaptándolo a nuevas realidades y desafíos.

2.6.1.2. Base Legal referente a la ATU

Ordenanza N°1613 Municipalidad Metropolitana de Lima.

- Fecha de publicación: 27 de junio de 2012.
- Descripción: La Ordenanza 1613, sienta las bases para la integración y modernización del transporte público en Lima, mediante la creación del Sistema Integrado de Transporte Público de Lima Metropolitana dando aprobación al Plan

Regulador de Rutas. Siendo un paso importante hacia la creación de un sistema de transporte más organizado y eficiente para los ciudadanos.

Resolución de Consejo Directivo N°002-2019-SMV/02.

- Fecha de publicación: 24 de enero de 2020.
- Descripción: Esta resolución, planteada por la Superintendencia del Mercado de Valores (SMV), aprueba las disposiciones para la regulación del mercado del transporte público en Lima y Callao, dentro del marco del SIT. Incluye medidas para garantizar la sostenibilidad financiera y operativa del sistema.

Ley 31027.

- Fecha de publicación: 01 de julio de 2020.
- Descripción: Esta ley fortalece el marco legal para la implementación del Sistema Integrado de Transporte (SIT) en Lima y Callao, otorgando a la ATU mayores competencias y facultades para llevar a cabo sus funciones. También establece disposiciones específicas para la integración de servicios y modos de transporte.

2.6.2. Base Teórica

Para iniciar esta sección es importante aclarar que no existe una definición exacta de grafos, pues esta varía según el contexto en que se utilice, considerando que la aplicación de la misma recae en un amplio campo por su propia versatilidad dentro de las matemáticas y la informática. Por ende, para el presente trabajo, la definición de grafos se centrará en el uso y aplicación en los sistemas de transporte.

2.6.2.1. Historia de la Teoría de Grafos

La teoría de grafos se originó en el siglo XVIII, cuando el matemático Leonhard Euler resolvió un famoso problema conocido como los "Siete Puentes de Königsberg".

Figura 17

Los 7 Puentes de Königsberg

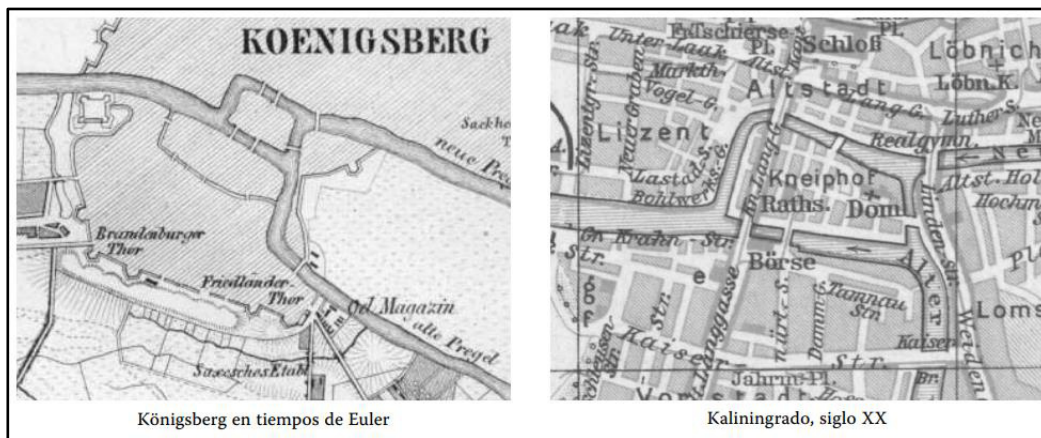


Nota. Revista Española Suma 45

La ciudad de Königsberg en Prusia, actualmente exclave ruso, estaba dividida por un río llamado Pregel, hoy conocido como Pregolya, y conectada por siete puentes. El desafío consistía en encontrar un recorrido que cruzara cada puente una sola vez y regresara al punto de partida.

Figura 18

Mapa de Königsberg



Nota. Revista Española Suma 45

Se sabe cómo dato histórico que un grupo de jóvenes académicos por el año 1735 le solicitaron al matemático Leonhard Euler, nacido en la ciudad de Basilea en 1707, que les ayudara resolviendo el problema de los 7 Puentes. Enterado de la encrucijada, Leonhard, aceptó buscar dar solución a la incógnita en cuestión, dedicándose a estudiar la misma.

Figura 19

Leonhar Euler (1707 – 1783)



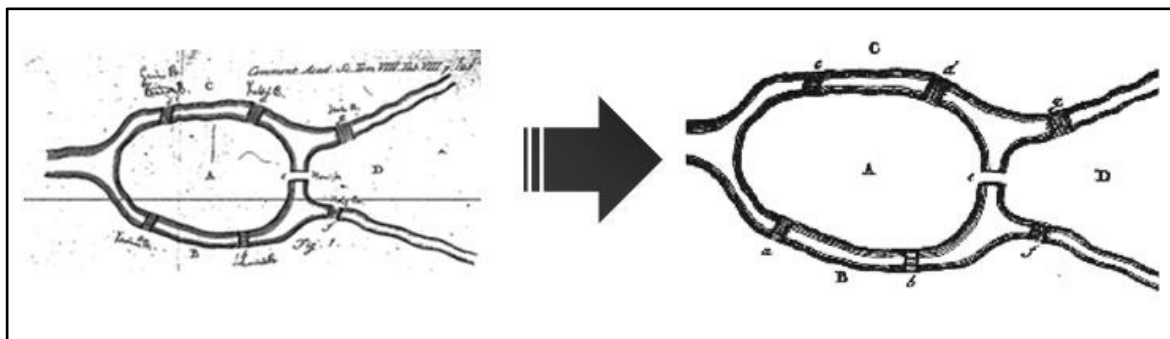
Nota. Fuente externa

Para la resolución del problema, el matemático, planteó una solución ingeniosa para la época, sin saber que la misma sentaría las bases para establecer la teoría de grafos.

Para poder analizar mejor el problema, Euler, plasmó el mapa de la ciudad como un diagrama conectado por vértices y aristas, donde cada vértice eran los puntos a conectar (islas y ciudad) denominándolos nodos, y las aristas representaban a los puentes que permitían conectar dichos nodos. El planteamiento del problema, consistía en el siguiente enunciado: “En la ciudad de Königsberg, perteneciente a Prusia de aquel entonces, hay una isla A llamada Kneiphof, rodeada por los dos brazos del río Pregel. A su vez existen siete puentes a, b, c, d, e, f y g, que cruzan por los dos brazos del río. Por lo tanto, la incógnita consistía en determinar si una persona podía realizar un paseo de tal forma que cruce cada uno de estos puentes sólo una vez”.

Figura 20

Bosquejo dado por Euler



Nota. Revista Española Suma 45

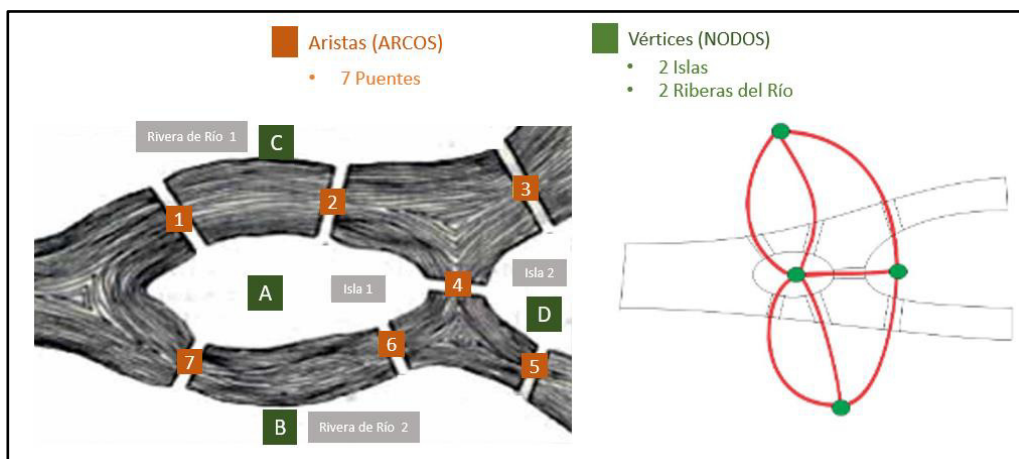
De este planteamiento, el matemático, representó los caminos que conectaban la isla y la ciudad como AB y BD, según correspondiera. Sin embargo, esta notación tenía limitaciones al no considerar la posibilidad de múltiples puentes entre las áreas. Por lo que observó que, dependiendo de cuántos puentes conectaban cada área, las letras A, B, C y D debían aparecer un número específico de veces en una secuencia de ocho letras para representar todas las rutas posibles.

Con esto pudo concluir que no era factible realizar el recorrido deseado considerando las restricciones establecidas. Sin embargo, Euler no se contentó con la solución dada por él mismo, por lo que se dedicó a plantear una solución que sirviera para cualquier número de puentes.

Primero anotó el número total de puentes más uno y listó las letras correspondientes a las áreas (A, B, C, D). Luego, creó una columna con el número de puentes que conectaban cada área (5, 3, 3, 3), marcando con un asterisco las letras con un número par de puentes. Aplicó operaciones a estos números: si eran impares, los aumentó en uno y los dividió por dos; si eran pares, simplemente los dividió por dos. Euler concluyó que el problema solo tendría solución si la suma final de estos números era igual o menor al número inicial (8 en este caso). En Königsberg, esta suma fue nueve, lo que indicó que no había solución posible para cruzar todos los puentes según las reglas establecidas.

Figura 21

Estructuración del Grafo de Königsberg



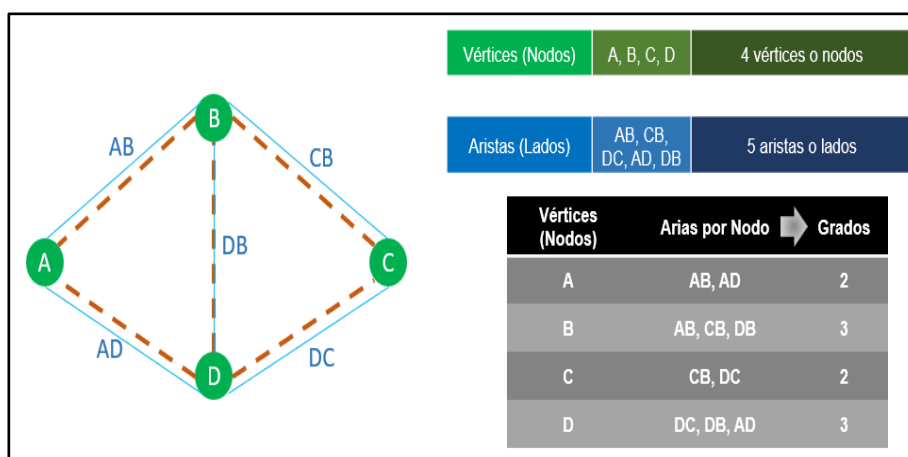
Nota. Elaboración propia

Durante el estudio del matemático suizo Leonhard Euler, este desarrolló el concepto de “grado” de un vértice; donde el “grado” representa el número de aristas que inciden en un determinado vértice. Y a su vez introdujo condiciones para determinar si un camino podía o no pasar por todas las aristas según el número de grados de cada vértice, a lo que actualmente

se conoce como Camino Euleriano, y al mismo tiempo determinó las condiciones para saber si era posible iniciar en cualquier vértice y pasar por todas las aristas una sola vez hasta regresar al mismo punto desde cual se inició, lo que hoy conocemos como Circuito Euleriano.

Figura 22

Representación de nodos - aristas - grados



Nota. Elaboración propia

Aclarar que, durante el estudio realizado por el matemático suizo, este no usó los términos de circuito y camino euleriano como tal. Pues estos términos serían planteados por otros estudiosos, para referirse a dichos teoremas años más tarde.

Es así como se establecen los fundamentos para dar lugar a lo que hoy se conoce como la teoría de grafos, pues mediante un trabajo innovador para aquel entonces, se permitió abstraer problemas específicos y generalizarlos, proporcionando una nueva manera de representarlos, analizarlos y resolverlos mediante el uso de estructuras conectadas.

Durante el siglo XIX, otros matemáticos como Gustav Kirchhoff, Arthur Cayley, y James Joseph Sylvester, contribuyeron significativamente a la teoría de grafos. Kirchhoff utilizó grafos para resolver problemas de circuitos eléctricos, mientras que Cayley aplicó grafos en la química para estudiar estructuras moleculares. Sylvester, por su parte, introdujo el término "grafo" y desarrolló notaciones y terminologías que aún se utilizan hoy en día.

A lo largo del siglo XX, la teoría de grafos experimentó un crecimiento exponencial,

encontrando aplicaciones en diversas áreas como la biología, la informática, las ciencias sociales y la ingeniería. El desarrollo de algoritmos eficientes para la manipulación y análisis de grafos permitió resolver problemas complejos en redes de comunicación, transporte, y datos.

Uno de los avances más significativos fue la introducción de algoritmos de búsqueda de rutas, entre los cuales destaca el algoritmo de Dijkstra, desarrollado por el científico informático Edsger Dijkstra en 1956. Este algoritmo se utiliza para encontrar las rutas más cortas en grafos ponderados y ha tenido un impacto duradero en la teoría de grafos y sus aplicaciones prácticas.

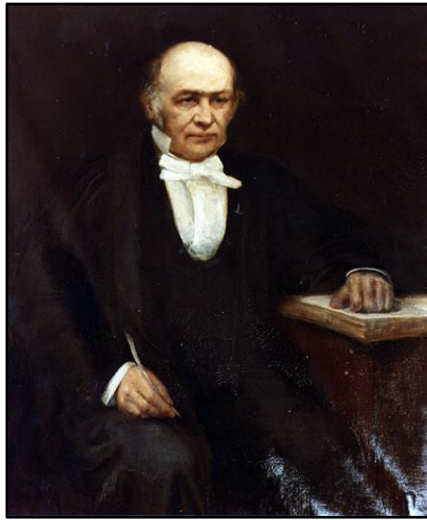
En resumen, la teoría de grafos ha evolucionado desde sus humildes comienzos con Euler hasta convertirse en una herramienta indispensable en múltiples disciplinas. Su capacidad para modelar y resolver problemas complejos continúa expandiéndose, demostrando su relevancia en la ciencia y la tecnología contemporáneas.

2.6.2.2. El Juego Icosiano

El Juego Icosiano fue creado en 1859 por el prodigioso irlandés William Rowan Hamilton (1805 - 1865). El irlandés fue un prodigio para las ciencias pues era físico, matemático y astrónomo que contribuyó en los diferentes campos científicos. Una de estas contribuciones fue el planteamiento de los caminos y ciclos hamiltonianos, denominados de ese modo por otros matemáticos en reconocimiento a la contribución que hizo para determinar dichos tipos de recorridos en la estructura de un grafo.

Figura 23

William Rowan Hamilton (1805 – 1865)




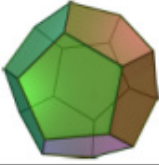
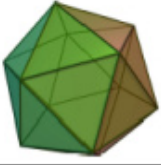


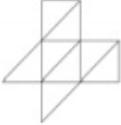




Nota. Fuente externa

El Juego Icosiano, se origina por la fascinación de William desde muy niño por los sólidos platónicos, pues estos sólidos se caracterizan por tener caras poligonales idénticas y ser convexos. Estos sólidos han sido fascinantemente atractivos para diferentes matemáticos y científicos durante siglos por su belleza basada en su simetría única, y el joven prodigio no fue una excepción pues desde los 12 años ya hacía ensayos basados en los sólidos platonianos.

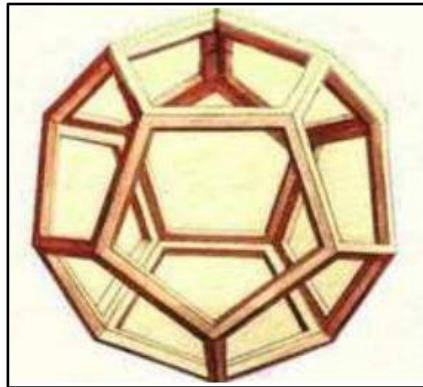
El sólidos platónicos o sólidos regulares son poliedros convexos, donde todas sus caras son polígonos regulares e iguales entre sí. Su nombre se da en honor al filósofo Platón por haberlos estudiado en primera instancia. Estas geometrías son también conocidas como cuerpos cósmicos, poliedros platónicos, sólidos pitagóricos, poliedros regulares convexos. Estos poliedros son cinco: Tetraedro, Hexaedro (cubo), Octaedro, Dodecaedro, Icosaedro.

Figura 24*Los 5 Sólidos Platónicos*

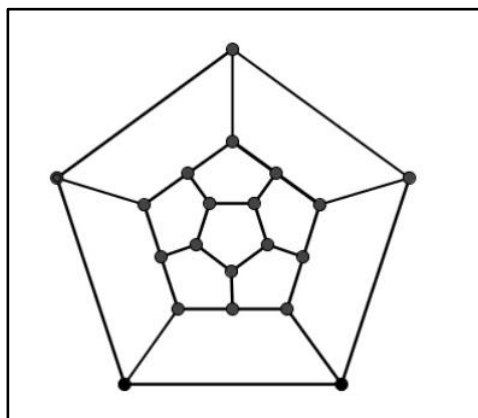
Sólidos Platónicos	Tetraedro	Hexaedro, Cubo	Octaedro	Dodecaedro	Icosaedro
					
Desarrollo					
Número de caras	4	6	8	12	20
Polígonos que forman las caras	Triángulos Equiláteros	Cuadrados	Triángulos Equiláteros	Pentágonos Regulares	Triángulos Equiláteros
Número de aristas	6	12	12	30	30
Número de vértices	4	8	6	20	12
Caras concurrentes en cada vértice	3	3	4	3	5
Vértices contenidos en cada cara	3	4	3	5	3
Grupo de simetría	Tetraédrico (T_d)	Hexaédrico (H_h)	Octaédrico (O_h)	Icosaédrico (I_h)	Icosaédrico (I_h)
Poliedro conjugado	Tetraedro (autoconjugado)	Octaedro	Hexaedro, Cubo	Icosaedro	Dodecaedro
Símbolo de Schläfli	{3,3}	{4,3}	{3,4}	{5,3}	{3,5}

Nota. Recurso externo

El juego planteado por Hamilton buscaba que el jugador encontraría un recorrido a lo largo de un dodecaedro regular, hecho de madera con 20 vértices que representaban las ciudades más emblemáticas de aquel entonces (Bruselas, Cantón, Delhi, Frankfurt y otras más). El camino buscado en el dodecaedro debía permitirle al jugador pasar solamente una vez por cada ciudad (vértice) y volver al punto del que partió (vértice origen).

Figura 25*Dodecaedro de Madera**Nota.* Recurso externo

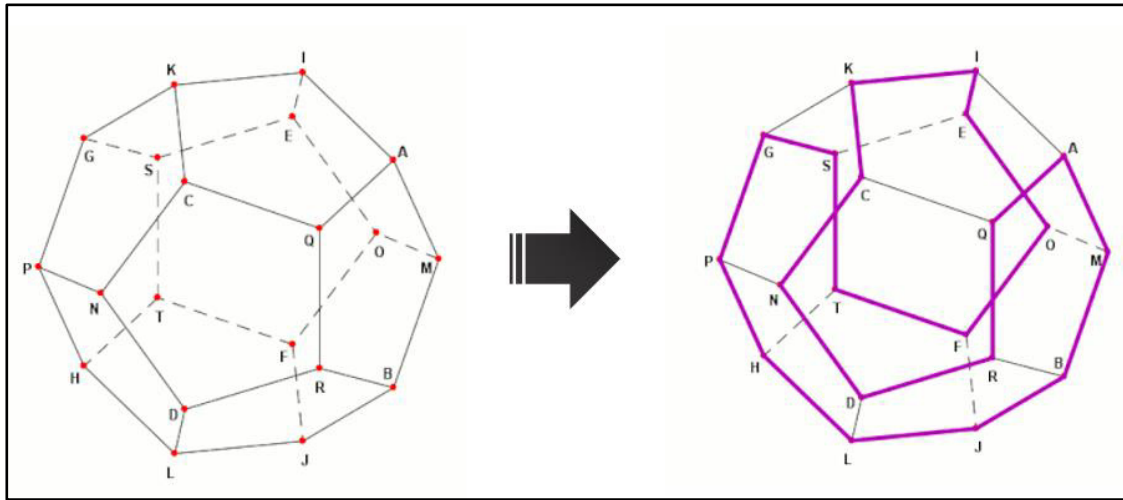
La complejidad matemática para solucionar este enigma cautivó a Hamilton. Por ende, para buscar una solución al juego, el matemático planteó estructurar el dodecaedro como un grafo que le permitiera manipularlo pues por la propia forma del sólido complicaba el análisis del mismo. Se debe recordar que los Grafos fueron acuñados por Leonhard Euler en 1736, pero la teoría planteada por Euler tenía un fin diferente al problema planteado en el juego, sin embargo, William R. Hamilton usó algunos principios de las ideas planteadas por Euler como el uso grafos para buscar solución al juego que el mismo planteó.

Figura 26*Grafo del Dodecaedro de Madera**Nota.* Recurso externo

Las ideas que concibieron la solución al juego del dodecaedro permitieron dar pie años después, mediante la contribución de otros matemáticos en conjunto con Hamilton, a la teoría de caminos y ciclos hamiltonianos.

Figura 27

Solución al Juego Icosiano

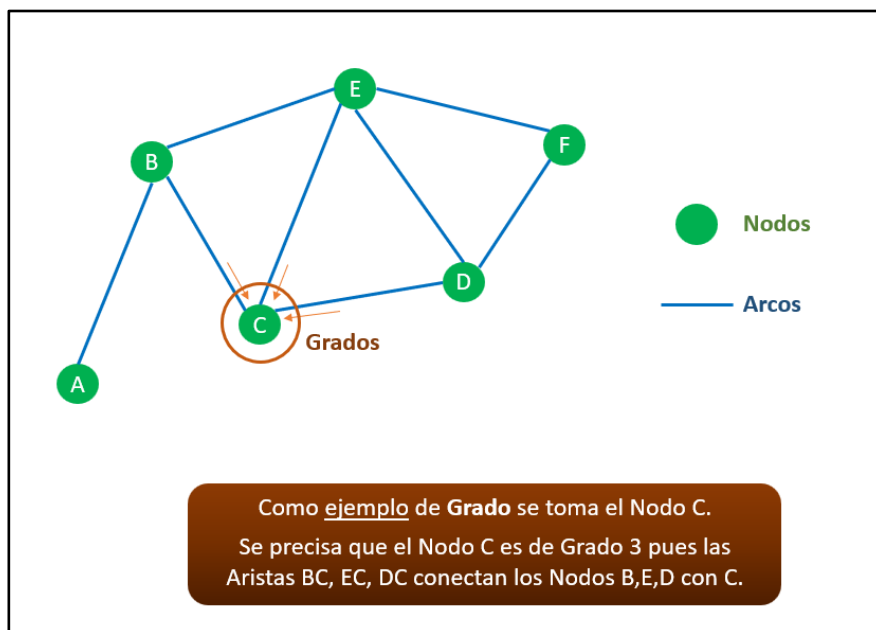


Nota. Recurso externo

2.6.2.3. Fundamentos de la Teoría de Grafos

La Teoría de Grafos busca la forma de representar y modelar relaciones entre los conjuntos de datos.

Para explicar con mayor claridad la teoría, es necesario entender que un Grafo es una estructura interconectada compuesta por vértices denominados nodos, estas conexiones entre los nodos se da por las aristas, que son líneas que conectan los vértices entre sí, a las aristas se les denomina arcos. Finalmente, los grados, o también conocido como el número de orden del nodo, representan la cantidad de arcos que se conectan a este, debemos tener en cuenta que los grados pueden ser negativos si los arcos que salen del nodo o positivos si los arcos llegan hacia el nodo en cuestión, y de ser el caso en el que la arista no cuenta con sentido alguno simplemente el grado será positivo.

Figura 28*Estructura de un Grafo*

Nota. Elaboración propia

Para la aplicación de la Teoría de Grafos en diversas disciplinas científicas, se han desarrollado varias configuraciones fundamentales de grafos. Estas configuraciones se derivan de la tipología básica de los grafos, comúnmente conocidos como grafos simples.

Si bien determinar la cantidad de grafos simples es una tarea matemática extenuante por su amplia variedad de configuraciones, existen ciertos grafos simples que suelen ser los más conocidos entre los que se destacan tales como los Grafos No Dirigidos, Grafos Dirigidos, Grafos Ponderados o también conocidos como Grafos con Pesos, Grafos de Árboles, Grafos Bipartitos, Grafos Acíclicos.

Cada uno de estos tipos ofrece un marco conceptual único para modelar y analizar relaciones y estructuras más complejas. Aclarar que, las estructuras complejas pueden ser modeladas a partir de estructuras más simples.

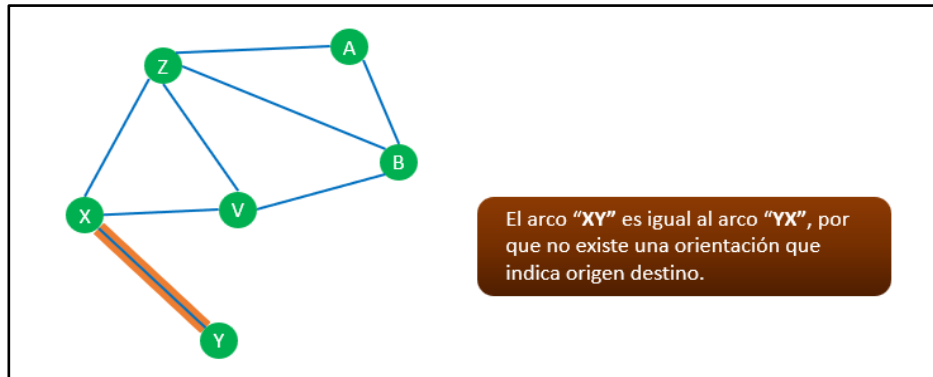
Grafos No Dirigidos:

Es el tipo de grafo más común en el cual los arcos no tienen orientación, es decir no

cuentan con sentido o dirección específica. Por ende, el arco no tiene ida o retorno.

Figura 29

Grafo No Dirigido



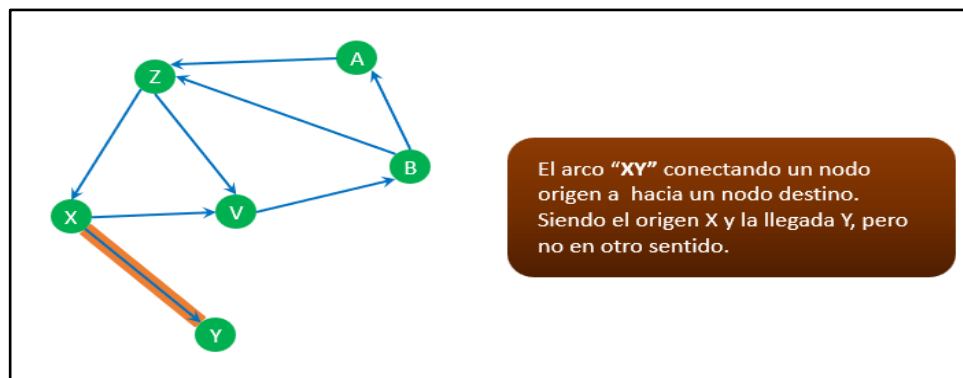
Nota. Elaboración propia

Grafos Dirigidos:

Este tipo de grafo difiere de los No Dirigidos, pues estos sí tienen orientación. Es decir, los arcos conectan un nodo origen con un nodo destino, pero no necesariamente en sentido inverso.

Figura 30

Grafo Dirigido



Nota. Elaboración propia

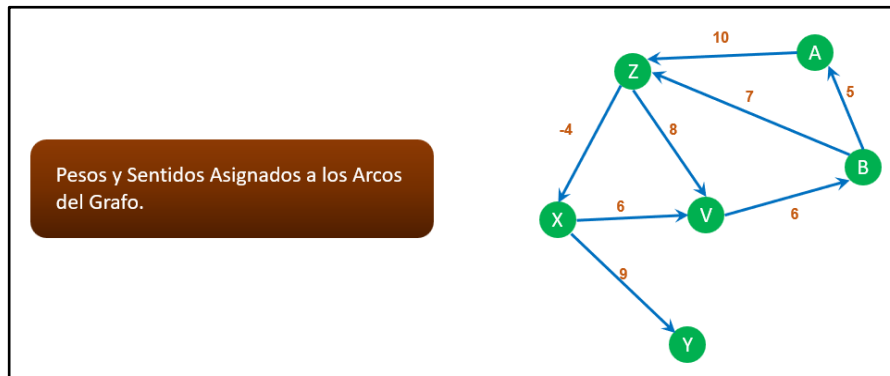
Grafos Ponderados o Grafos con Pesos:

En este tipo de grafos los arcos que conectan los nodos tienen pesos específicos. Los pesos específicos de los arcos pueden hacer referencia a diferentes cosas según el modelo de

negocio en el que se usen, por ejemplo, la distancia entre dos vértices. Este tipo de grafos puede ser Dirigidos como No Dirigidos.

Figura 31

Grafo Ponderado Dirigido



Nota. Elaboración propia

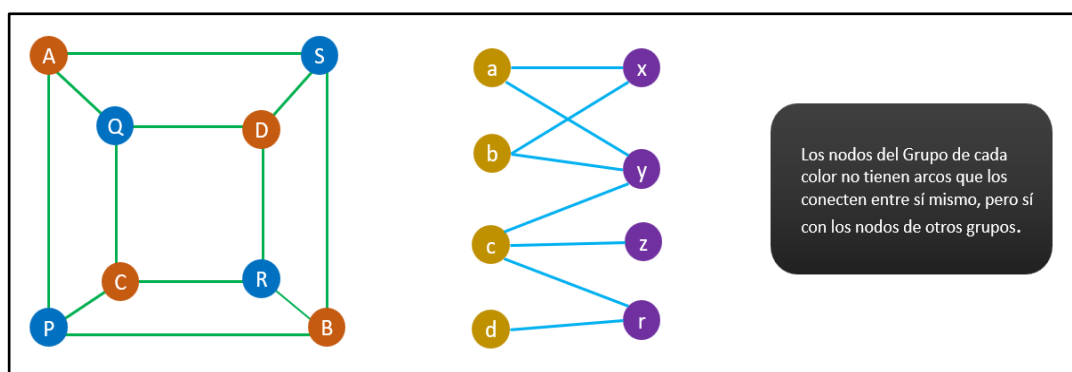
Grafos Bipartitos:

En este tipo grafos los nodos pueden ser separados en dos grupos, lo que permite que cada vértice de un grupo se conecte solamente con uno o más vértices del otro grupo, pero dos o más vértices de un mismo grupo nunca se conectarán directamente.

Los Grafos Bipartitos también se pueden clasificar como Dirigidos y No Dirigidos, y su uso dependerá de la necesidad de representación.

Figura 32

Grafos Ponderados Bipartitos



Nota. Elaboración propia

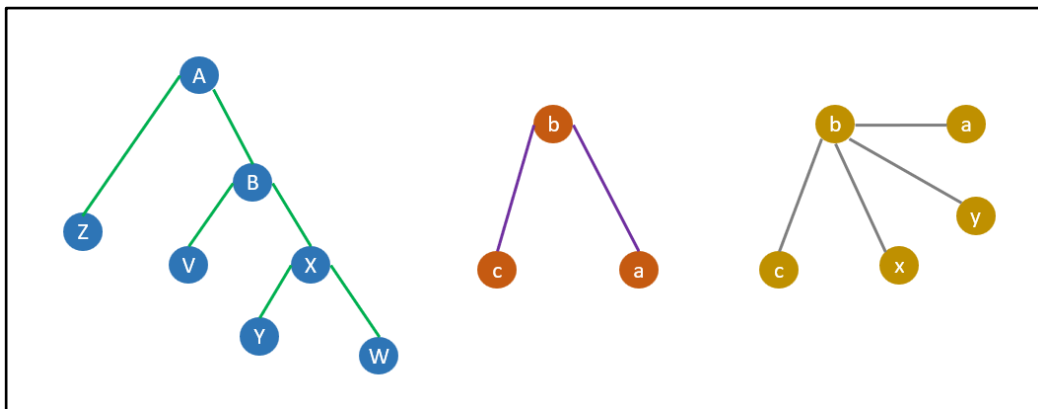
Grafos de Acíclicos

Los Grafos Acíclicos, como su propio nombre los describe se caracterizan por ser grafos que no presentan ciclos, es decir no muestran una configuración de un circuito cerrado.

Tener en cuenta que estos grafos al igual que los anteriores, pueden presentarse como Dirigidos y No Dirigidos.

Figura 33

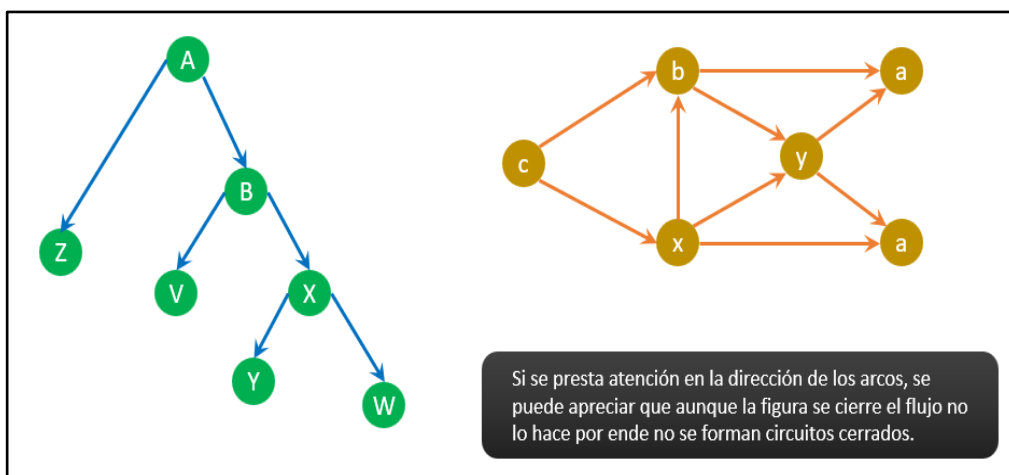
Grafos Acíclicos No Dirigidos



Nota. Elaboración propia

Figura 34

Grafos Acíclicos Dirigidos



Nota. Elaboración propia

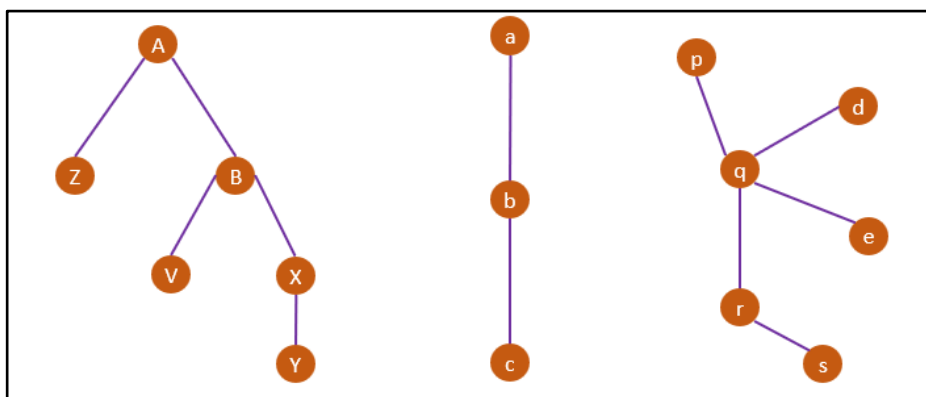
Grafos de Árboles:

Presentan características de los Grafos Acíclicos, y se clasifican como Grafos de Árboles No Dirigidos y Grafos de Árboles con Raíz.

Los Grafos de Árboles No Dirigidos están compuestos por Grafos No Dirigidos y sin ciclos.

Figura 35

Grafos de Árboles No Dirigidos

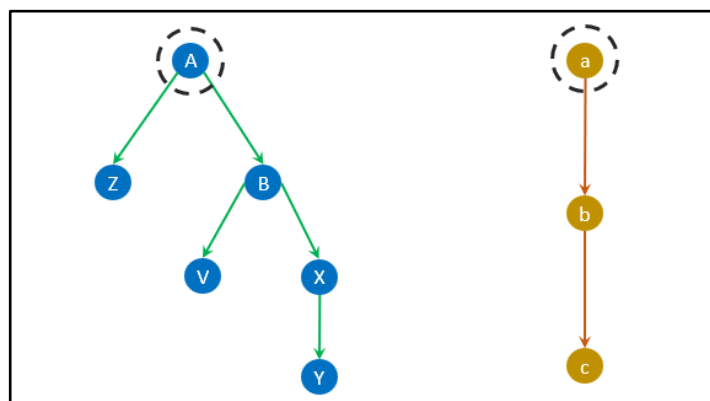


Nota. Elaboración propia

Sin embargo, los Grafos Arboles con Raíz, están compuestos por grafos dirigidos y poseen un vértice desde el cual se puede llegar al resto de vértices del árbol.

Figura 36

Grafos de Árboles con Raíz



Nota. Elaboración propia

Por lo general, cuando se hace referencia a los grafos de árboles, se asume que se está hablando de los Grafos de Árboles con Raíz.

2.6.2.4. Representación y Análisis de los Grafos por Adyacencia

La manera más sencilla de representar los grafos simples y complejos suele ser mediante el uso de gráficos o diagramas, tal como se puede visualizar en el ítem anterior en el que se detalla mediante imágenes la tipología de los grafos simples más conocidos.

Sin embargo, para poder analizar y estudiar el comportamiento de los grafos es necesario estructurarlo más allá de una visualización gráfica, sobre todo para aquellos que sus estructuras suelen ser más complejas.

Por ende, para cumplir el objetivo de estudiarlos se han generado diferentes metodologías, siendo las más usadas y conocidas, la Matriz de Adyacencia y la Lista de Adyacencia.

Matriz de Adyacencia

Estructura de filas y columnas con los nodos y arcos que conforman el grafo. En cada una de las posiciones de la matriz se asigna el peso que posee la arista que conecta a dos vértices.

Lista de Adyacencia

Para generala crea una lista para cada vértice del grafo, donde se almacenan a modalidad de grupos que contengan el nodo de destino y el peso asignado al arco que lleva al mismo.

Figura 37*Matriz y Lista de Adyacencia**Nota.* Recurso externo

2.6.2.5. Ciclos y Caminos Eulerianos

Producto de la estandarización de la solución al problema de los 7 Puentes de Königsberg, dada por el matemático suizo Leonhard Euler, años más tarde estos se volverían teoremas bases para la teoría de los grafos. Planteándose de este modo los conceptos de Circuito y Camino Euleriano.

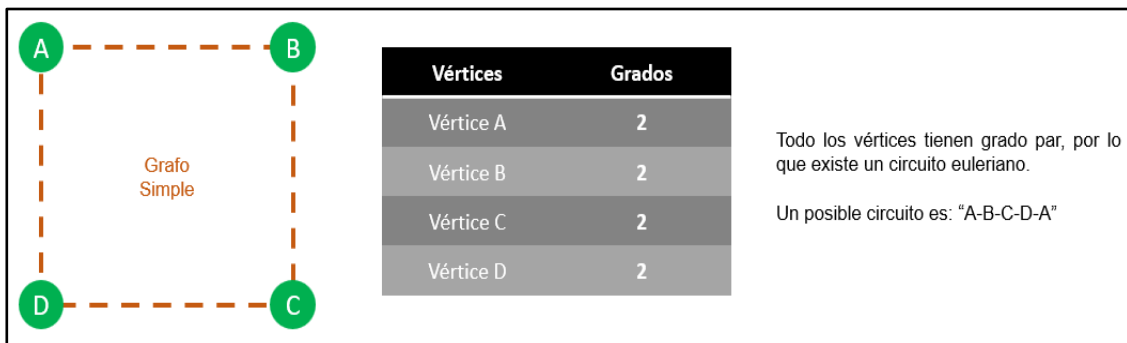
Circuito Euleriano:

Un circuito euleriano es un recorrido en un grafo que:

- Pasa por cada arista exactamente una vez.
- Regresa al vértice inicial, formando un ciclo cerrado.

Un grafo que cumple los siguientes puntos es un circuito euleriano. Condiciones para los circuitos eulerianos:

- Todos los vértices deben tener un grado par, es decir, cada vértice debe tener un número par de aristas conectadas a él.

Figura 38*Circuito Euleriano Simple*

Nota. Elaboración propia

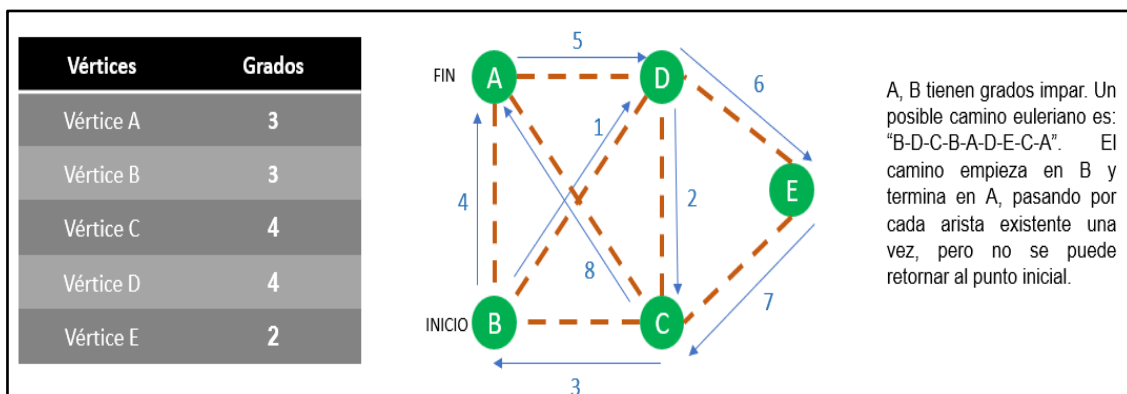
Camino Euleriano:

Un camino euleriano, pero no un circuito euleriano, es un recorrido en un grafo que:

- Pasa por cada arista exactamente una vez.
- No necesariamente regresa al vértice inicial, formando un camino abierto.

Para que un grafo tenga un camino euleriano, debe cumplir las siguientes condiciones:

- Exactamente dos vértices del grafo deben tener un grado impar, estos serán el punto de inicio y el punto de fin del camino.
- Todos los demás vértices deben tener un grado par.

Figura 39*Camino Euleriano Simple*

Nota. Elaboración propia

2.6.2.6. Ciclos y Caminos Hamiltonianos

Considerando los ítems anteriores, los fundamentos para la teoría de los caminos y ciclos Hamiltonianos se originaron producto de plantear soluciones al Juego Icosiano, formulado por el prodigio irlandés William Rowan Hamilton. A partir de estas soluciones diversos matemáticos, incluido William Rowan Hamilton, concibieron la teoría de los caminos y ciclos Hamiltonianos.

Circuito Hamiltoniano:

Un circuito hamiltoniano es un recorrido en un grafo que:

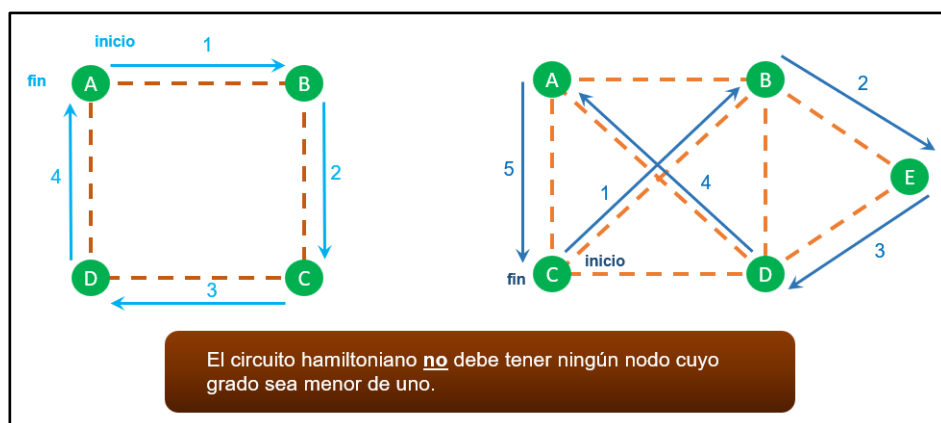
- Regresa al vértice inicial, pasando por cada vértice una única vez, y no necesariamente por todas sus aristas.

Se debe tener claro que, a la fecha no se conocen la cantidad de condiciones necesarias para determinar la existencia de circuitos Hamiltonianos. Sin embargo, sí se puede precisar que para poder analizar si un grafo presenta un circuito Hamiltoniano por lo menos debe cumplir que:

- Aquel grafo que contenga un vértice de grado uno no puede contener un ciclo Hamiltoniano.

Figura 40

Circuitos Hamiltonianos



Nota. Elaboración propia

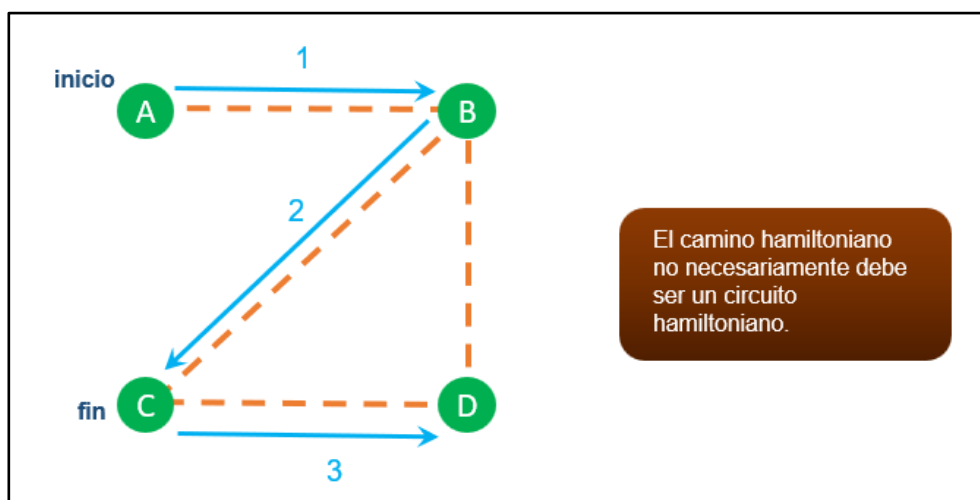
Camino Hamiltoniano:

Un camino hamiltoniano es aquel recorrido sobre un grafo que:

- Pasa por todos los vértices solamente una vez desde un punto inicial hacia un punto final. Por ende, no es necesario que pase por todas las aristas, así como no vuelve a pasar por la misma que ya recorrió más de una vez.

Figura 41

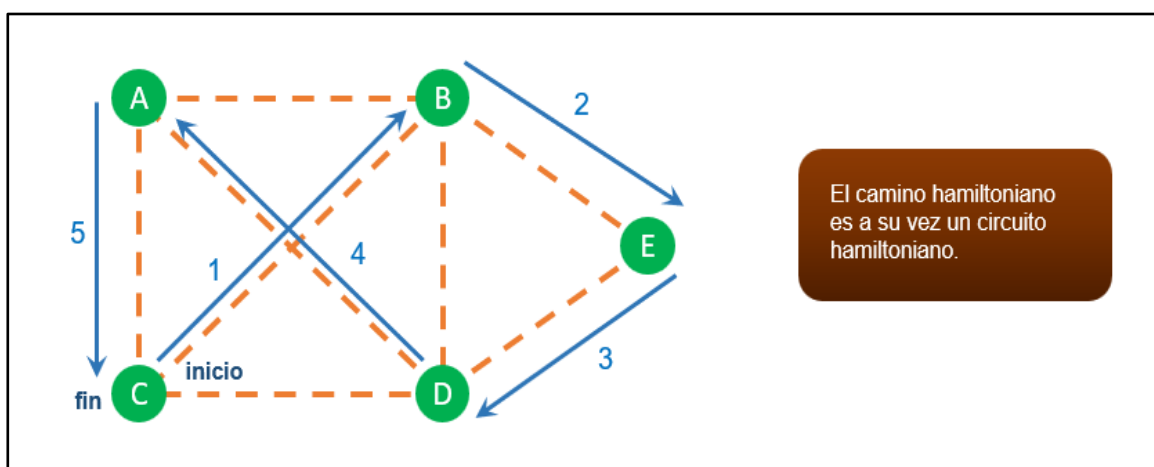
Camino Hamiltoniano Simple



Nota. Elaboración propia

Figura 42

Camino y Circuito Hamiltoniano Simple



Nota. Elaboración propia

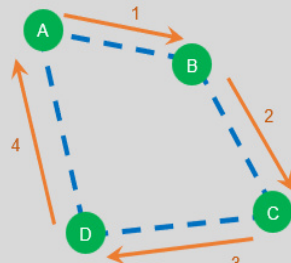
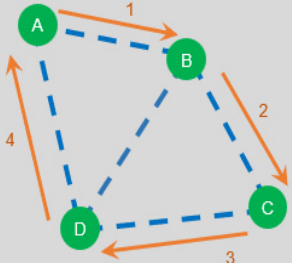
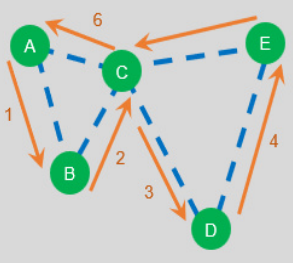
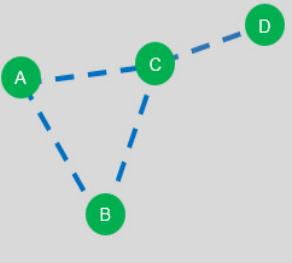
2.6.2.7. Relación de Ciclos y Caminos Eulerianos y Hamiltonianos

Los caminos y circuitos eulerianos se enfocan en recorrer todas las aristas de un grafo solamente una vez. Sin embargo, para el caso de los hamiltonianos estos se enfocan en visitar todos los nodos solamente una vez sin la obligación de recorrer todas las aristas. Por otro lado, los grafos eulerianos tienen condiciones específicas relacionados a los grados de sus nodos para poder identificarlos, mientras que para los grafos hamiltonianos no existen condiciones generales para su identificación, incluso estas condiciones pueden variar de grafo en grafo.

Al analizar los grafos desde el enfoque euleriano el estudio se centra en las aristas mientras que el enfoque hamiltoniano se centra en analizar los nodos de estos. Sin embargo, un grafo puede cumplir tanto propiedades eulerianas como hamiltonianas.

Figura 43

Relación entre Grafos Eulerianos y Hamiltonianos

GRAFOS	Eulerianos	No Eulerianos
Hamiltonianos		
No Hamiltonianos		

Nota. Elaboración propia

Tabla 1*Comparación de Ciclos y Caminos Eulerianos y Hamiltonianos*

Características	Euleriano		Hamiltoniano	
	Camino	Circuito	Camino	Circuito
Definición	Visita cada arista exactamente una vez.	Visita cada arista exactamente una vez y retorna al vértice inicial.	Visita cada vértice exactamente una vez.	Visita cada vértice exactamente una vez y retorna al vértice inicial.
Condición para su existencia	Grafo convexo y vértices con grado par.	Grafo convexo con exactamente dos vértices de grado impar.	No se han determinado las condiciones simples y generales para su identificación. Existen algunos teoremas, pero son aplicables para grafos específicos.	
Dependencia	Depende del grado de los vértices (par o impar).		No determinado, pues no depende de una propiedad simple del grafo. Pudiendo variar según la estructura.	
Problema de Decisión	Determinable en tiempo polinómico.		NP-completo.	

Nota. Elaboración propia

2.6.2.8. Aplicaciones Generales de la Teoría de Grafos

Como ya se ha indicado la aplicación de la Teoría de los Grafos al estudio del as relaciones entre los objetos de estudio es tan versátil, que su aplicación se ha extendido en diferentes campos de la ciencia, partiendo desde las ciencias en computación, pasando por la ingeniería, medicina, biología, química, incluyendo la economía hasta las ciencias sociales.

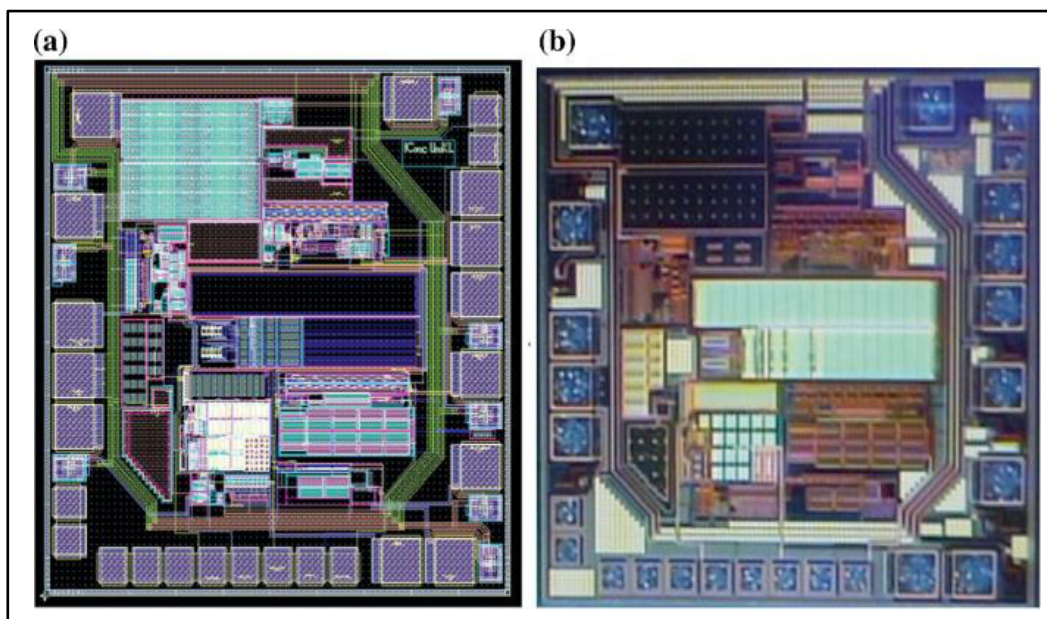
Para contextualizar la aplicación de la Teoría de Grafos, a los diferentes campos científicos, se puede mencionar:

Computación y Algoritmos:

La teoría de grafos ha sido un aspecto fundamental para el desarrollo computacional pues han permitido resolver y superar diferentes aspectos como los problemas de optimización de recursos, rutas más cortas e incluso plantear estructuras de datos eficientes. Como ejemplos tenemos el análisis de circuitos electrónicos, pues mediante el análisis grafos se puede realizar la optimización de los mismos.

Figura 44

Análisis de Conexiones de un Placa Computacional



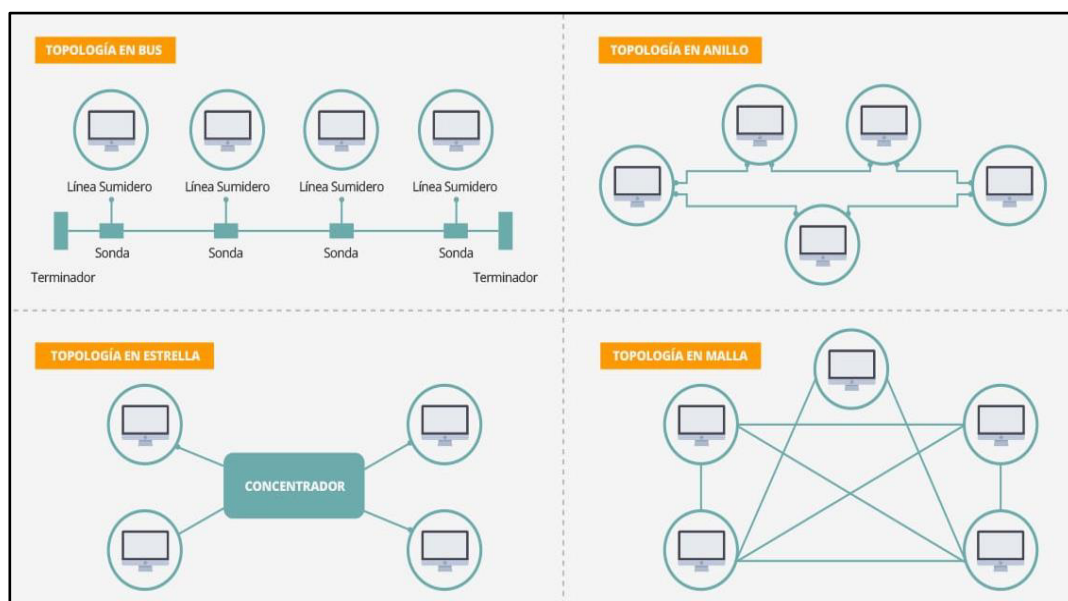
Nota. Recurso externo

Telecomunicaciones:

En las telecomunicaciones la Teoría de Grafos juega un rol muy importante, pues ha permitido generar diseños eficientes y confiables que optimizan la transferencia de los datos. Claro ejemplo son las redes de internet que transfieren de manera óptima los datos de ordenador a ordenador.

Figura 45

Topología de Redes



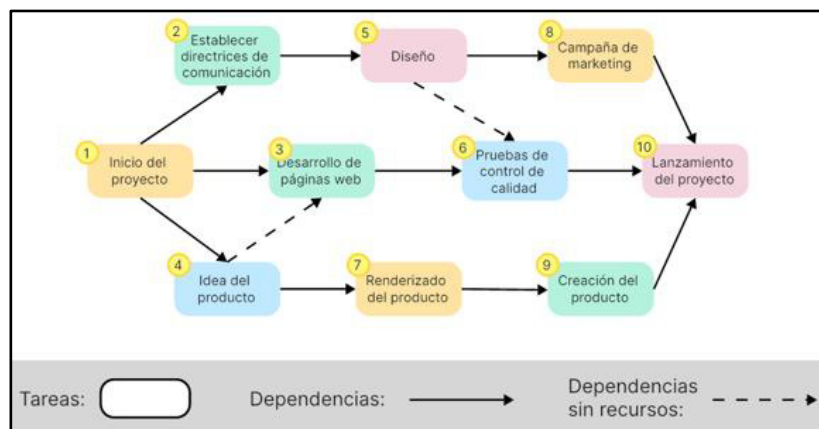
Nota. Recurso externo

Planificación de Proyectos:

La planificación de proyectos no ha sido indiferente en la aplicación de la Teoría de Grafos, pues la misma ha permitido mejorar la gestión y planificación de los proyectos mediante la identificación de dependencias y restricciones de los mismos, permitiendo mejorar la programación y el manejo de recursos. Un claro ejemplo es el diagrama de PERT donde las actividades se presentan como nodos, las dependencias se presentan como aristas, buscando identificar el camino crítico consecuencia de las actividades de mayor duración para identificar la duración mínima del proyecto en su conjunto.

Figura 46

Diagrama de PERT



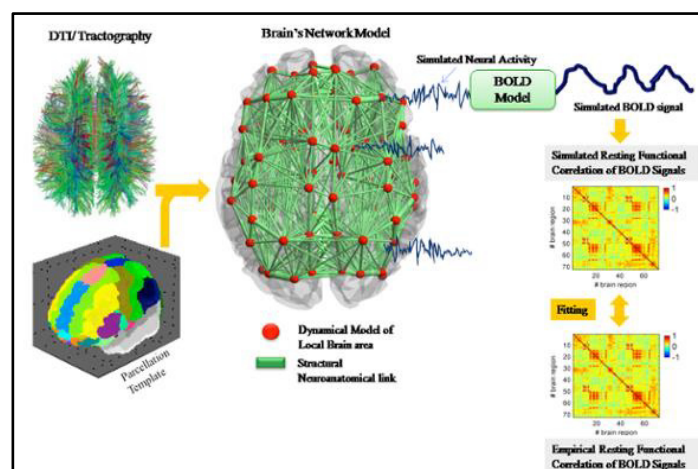
Nota. Recurso externo

Medicina:

En la medicina la teoría de grafos se usa para entender las estructuras y funciones de los componentes orgánicos del cuerpo. Por ejemplo, se usa para analizar la estructura y función de las proteínas ADN y otras moléculas biológicas a través de sus interacciones. Mientras que, en la neurociencia permite el estudio de las conexiones neuronales del cerebro permitiendo analizar su papel en el procesamiento de información y el comportamiento.

Figura 47

Modelo de Estado de Reposo del Cerebro Humano



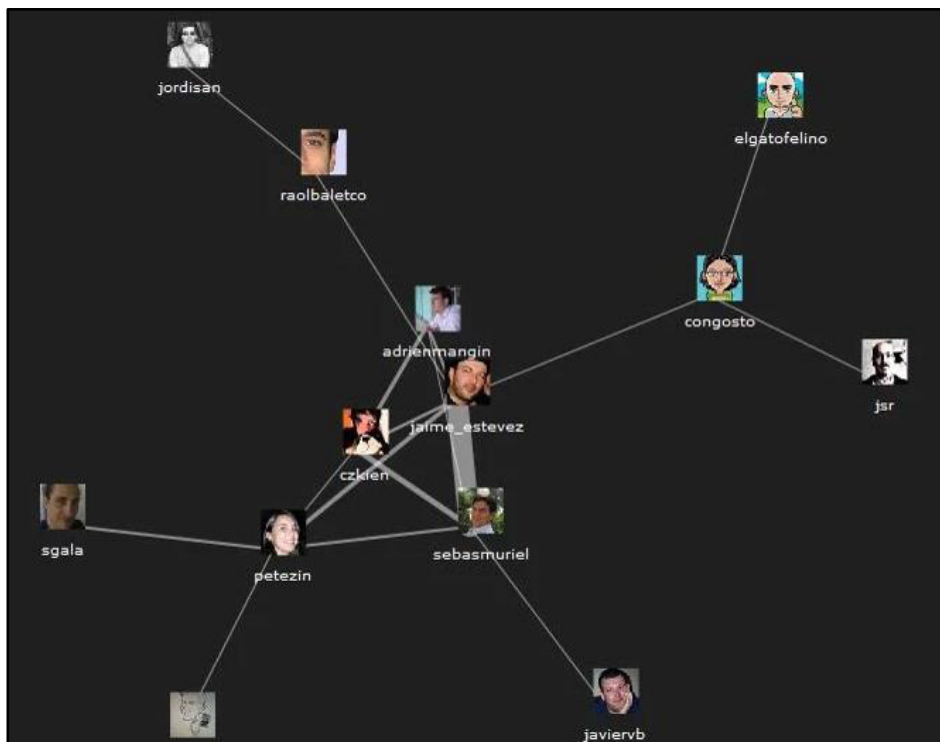
Nota. Recurso externo

Redes Sociales y Comunicación:

Un campo interesante de la aplicación de la teoría de grafos es como esta puede modelar las relaciones entre las personas o entidades con el fin de entender temas tales como los agentes externos del entorno de un individuo influyen en las decisiones de este. Claro ejemplo de esta aplicación es los que hoy conocemos como Redes Sociales Digitales como por ejemplo Facebook que brinda sugerencias de elección según búsquedas, amigos, y otras actividades que realizan los usuarios.

Figura 48

Ejemplo de Red de Contactos de una Red Social



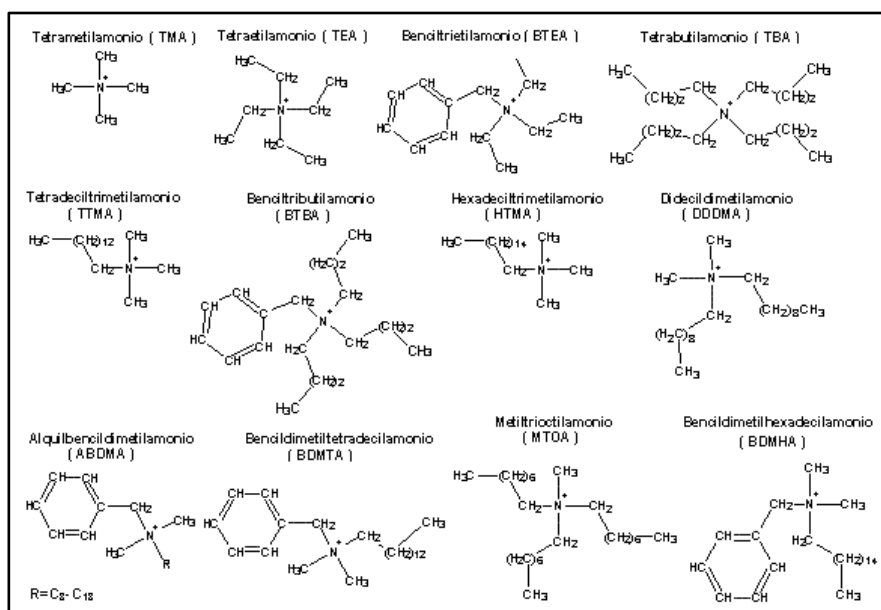
Fuente. Recurso externo

Biología y Química:

Un ejemplo de la aplicación de la Teoría de Grafos en la Biología y Química que ha permitido diversos descubrimientos en beneficio humano es el análisis de la Biología Molecular, donde los átomos asumen los roles de nodos y los enlaces químicos son representados por las aristas.

Figura 49

Estructura Esquemática de Cationes de Amonio Cuaternario



Nota. Recurso externo

Por todo lo descrito, se puede afirmar que la aplicación de los grafos tiene una amplia aplicación a diferentes rubros, e incluso la aplicación es tan variada que se han usado incluso para el análisis de diferentes juegos, siendo los más conocidos los de mesa como el ajedrez, las damas, el go y otros más. Entre los juegos de mesa más populares que han usado grafo tenemos:

Ajedrez: Usando la Teoría de Grafos se han podido desarrollar programas de computadora que puedan evaluar las posiciones de las fichas en el tablero y tomar decisiones estratégicas a partir de la ubicación de las mismas.

Damas: Mediante la teoría de grafos se puede analizar la complejidad del juego y determinar el número mínimo de movimientos necesarios para ganar desde una posición determinada.

Go: Es importante enfatizar que Go es una de los juegos de mesa de mayor complejidad, siendo una complejidad plantearlo bajo la teoría de grafos, sin embargo, no ha sido impedimento para desarrollar algoritmos que pueden emular jugadas que ofrezcan niveles competitivos de respuesta.

2.6.2.9. Teoría de Grafos en la Planificación y Desarrollo Territorial

La Teoría de Grafos tiene un fuerte impacto positivo para la planificación y al desarrollo territorial, pues mediante la representación de los objetos en estudio como nodos y las interconexiones de los mismos mediante el uso de arcos permite generar modelos matemáticos que permitan estudiar la forma en como estos objetos en estudio interactúan con el fin realizar una gestión eficiente de los recursos. La aplicación de los grafos a estas ramas de la geografía permita generar herramientas innovadoras con un enfoque invaluable. Por ejemplo, para la:

Toma de decisiones territoriales:

Mediante la representación de las ciudades, centros poblados o recursos naturales como nodos y las interdependencias de estos mediante arcos, los planificadores pueden tomar decisiones para desarrollar mejoramientos, rehabilitaciones y/o nuevas infraestructuras con el fin de dar el mejor manejo posible a los recursos.

Planificación de Servicios Urbanos:

En la planificación los grafos permiten mejorar los servicios públicos. Pues, mediante técnicas de análisis de flujos y algoritmos de optimización, los planificadores pueden determinar dónde colocar a centros de salud, centros educativos u otros servicios de manera que maximicen la cobertura poblacional y los tiempos de accesos sean los mínimos posibles, por ende, se asegura la eficiencia del servicio y a su vez permite una distribución más equitativa de los recursos.

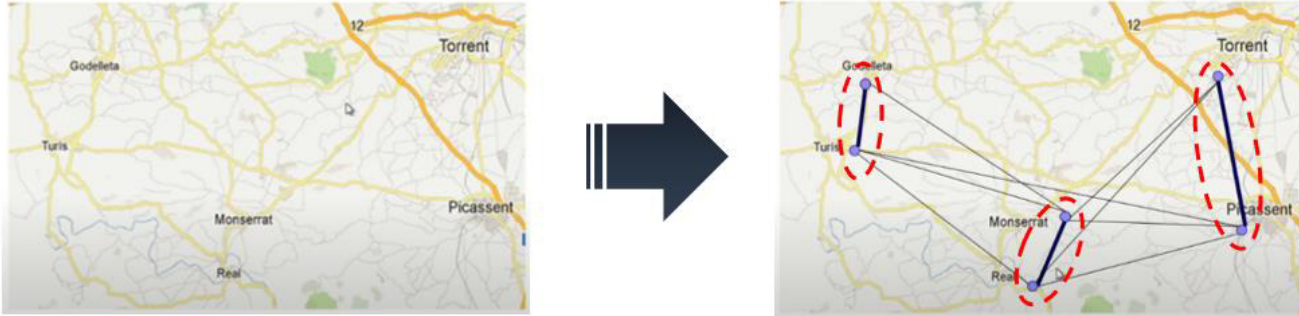
Desarrollo Territorial:

Para el desarrollo territorial los grafos ayudan a modelar la expansión de las ciudades, permitiendo cumplir con los criterios y fines de dicha expansión tales como la preservación de las áreas verdes, cuidado de espacios naturales y dar calidad de vida a sus residentes, mediante donde es necesario evaluar el impacto de las decisiones. Lo cual es de necesaria aplicación teniendo en cuenta el contexto actual respecto al crecimiento urbano acelerado y al impacto del cambio climático, siendo crucial minimizar la huella ecológica de las nuevas infraestructuras.

Figura 50

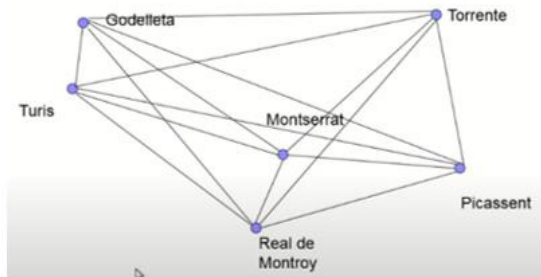
Ejemplo de Aplicación de Grafos para la asignación de lugares de reciclaje

Van a activar el reciclaje mediante la instalación de ecoparques (lugares de reciclaje), el tamaño de estos será proporcional a la población que atenderá. Se requiere sugerir la ubicación de estos teniendo que no es recomendable que los pueblos disten más de 30 km a la zona de reciclaje.



Distancia entre ciudades.

	Torrente	Picassent	Montserrat	Turis	Real de Montroy	Godelleta
Torrente		23	20	32	25	37
Picassent	23		14	26	19	31
Montserrat	20	14		24	5	29
Turis	32	26	24		29	5
Real de Montroy	25	19	5	29		34
Godelleta	37	31	29	5	34	



Modelización

$G=(V,E)$, donde
 $V = \{poblaciones\}$
 $E = \{(u,v) / u, v \in V, u \neq v \text{ y } p(u,v) \leq 30 \text{ km.}\}$
 G es un grafo no dirigido.
 $p(u,v)$ = distancia en km. de u a v

	Torrente	Picassent	Montserrat	Turis	R. de Montroy	Godelleta
Torrente		23	20	32	25	37
Picassent	23		14	26	19	31
Montserrat	20	14		24	5	29
Turis	32	26	24		29	5
Real de Montroy	25	19	5	29		34
Godelleta	37	31	29	5	34	

Resultados de Posibles Relaciones luego de Modelización

	Torrente	Picassent	Montserrat	Turis	R. de Montroy	Godelleta
Torrente		23	20	26	25	
Picassent	23		14	26	19	
Montserrat	20	14		24	5	29
Turis		26	24		29	5
Real de Montroy	25	19	5	29		
Godelleta			29	5		

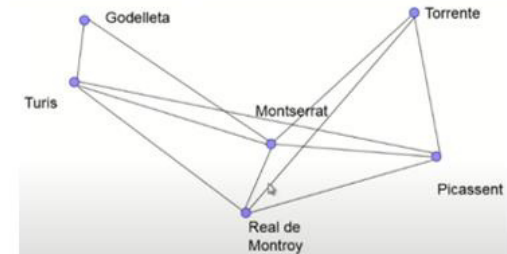
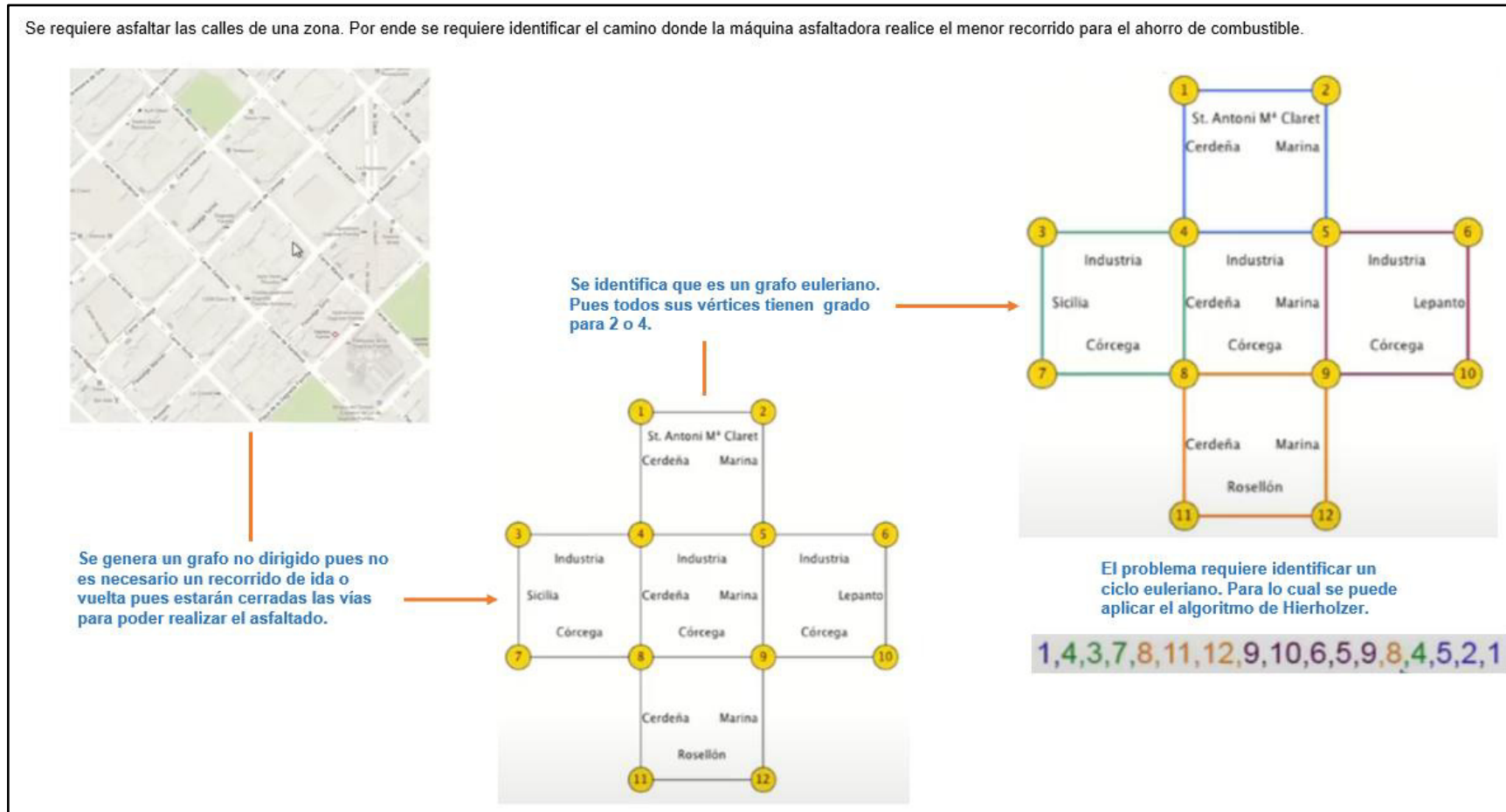


Figura 51

Ejemplo de Aplicación de Grafos para realizar la planificación de asfaltado de calles



Nota. Universidad Politécnica de Valencia

2.6.3. Teoría de Grafos en el Transporte

La teoría de grafos se concibió como una respuesta a enigmas de recorrido, de ahí su desprendida aplicación a diferentes ramas de las ciencias. Por lo mismo, su uso ha sido, es y seguirá siendo una sólida base de conocimiento para las respuestas de enigmas referentes a recorridos, por ende, su aprovechamiento para las ciencias de transporte es intrínseco por la propia naturaleza de la Teoría de Grafos. Como razones que justifiquen lo precisado se puede mencionar:

Representación natural de redes de transporte:

Pues las redes de transporte como carreteras, ferrocarriles, redes aéreas se pueden modelar de manera natural como grafos donde las ciudades (destino/origen) se representan mediante nodos mientras que las conexiones entre ellas son representadas como arcos.

Análisis y Optimización de Redes:

La teoría de grafos es fundamental para la optimización del tráfico y la planificación de infraestructuras que permitan optimizar los tiempos de transporte. Por ejemplo, los nodos representan las intersecciones y los arcos las vías que existentes entre dichas intersecciones.

Algunos ejemplos de la aplicación de la Teoría de Grafos en el campo del transporte son para la:

- Logística y Distribución, permitiendo planificar rutas de reparto, optimizando la carga de los vehículos y con ello optimizando la gestión de flotas.
- Gestión del Tráfico, mediante el control e implementación de señales de tráfico para optimizar el flujo vehicular permitiendo reducir la congestión.
- Infraestructura de Transporte, por apoyar en la planificación de proyectos para el desarrollo de nuevas vías de comunicación sea terrestre, acuático o aéreo evaluando el impacto de estos proyectos.

Referente a la planificación de rutas existen diferentes métodos fundamentados en la Teoría de Grafos para el cálculo de rutas, las cuales varían según el alcance de la necesidad. Sin embargo, existen dos planteamientos que han marcado hitos en el proceso de planificación de rutas. Siendo los siguientes:

La Teoría fundamentada en el Problema del Viajante de Comercio (TSP):

El cual busca que un recorrido retorne a su punto de partida visitando de manera óptima una serie de nodos definidos.

El Algoritmo de Dijkstra:

Este algoritmo tiene como fin buscar la ruta más corta, por ende, óptima entre dos nodos específicos en un grafo con pesos asignados a sus arcos.

2.6.3.1. Teoría de Grafos en el Transporte: TSP

El Problema del Viajero, conocido como TSP por su nombre en inglés el cual es Traveling Salesman Problem, tiene como objetivo buscar la ruta más corta visitando un conjunto de nodos una única vez retornando al punto de partida inicial. Este enigma surge producto de un desafío real propio de la distribución y planificación de rutas, el cual por su propia naturaleza de carácter cotidiano para la búsqueda de recorridos se ha vuelto un problema central en la teoría de grafos y en la ciencia computacional.

Historia del TSP:

El origen del Problema del Viajero de Comercio no tiene un momento exacto de origen, sin embargo, se ha podido rastrear la evolución de esta incógnita desde el siglo XIX, cuando los vendedores y comerciantes buscaban encontrar la ruta más eficiente para visitar múltiples destinos y regresar al punto desde el que partieron.

Durante la década de 1800 el término "Viajero de Comercio" es atribuido al matemático austriaco Karl Menger, al ser uno de los pioneros en formular el TSP como un problema matemático.

Luego durante 1930 el Problema del Viajero gana mayor relevancia por la investigación de varios matemáticos de diferentes países, incluyendo entidades académicas como la Universidad de Harvard y la Universidad de Viena, sobre el mismo.

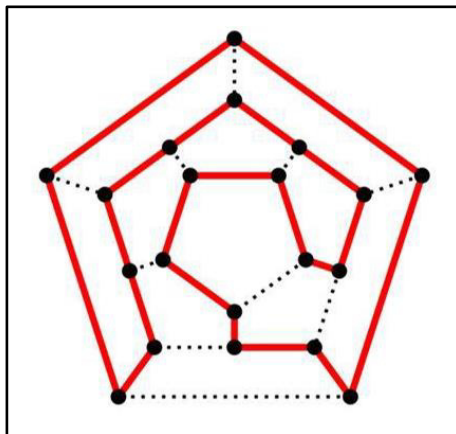
Y para 1950 se introducen los primeros algoritmos diseñados para resolver el TSP, tales como la búsqueda exhaustiva y el algoritmo del vecino más cercano.

Planteamiento Matemático del TSP:

La formulación matemática del TSP se basa en representar las ciudad o puntos de interés como nodos en un grafo y las rutas entre ellos como los arcos ponderados cuyo valor hace referencia a la distancia. El fin de este planteamiento es determinar el ciclo hamiltoniano de peso mínimo, es decir, identificar la ruta mínima cerrada que visite todos los nodos exactamente una vez y retorne al nodo inicial del cual partió.

Figura 52

Representación de Ciclo Hamiltoniano en un Grafo



Nota. Recurso externo

Es preciso indicar que producto de la búsqueda de la resolución al planteamiento matemático del Problema del Viajero se han desarrollado diferentes métodos que se han usado para resolver problemas derivados del mismo. Entre los enfoques más relevantes se pueden mencionar los siguientes:

- Método de Enumeración Completa o Búsqueda Exhaustiva o por Fuerza Bruta

- Programación Dinámica
- Algoritmo de Ramificación y Poda
- Vecino Más Cercano

Se debe tener en cuenta que estos enfoques no son de uso exclusivo para el análisis de problemas TSP, pues pueden ser usados también para la resolución de otro tipo de problemas.

Método de Enumeración Completa o Búsqueda Exhaustiva o por Fuerza Bruta:

El nombre de este método, es producto de su metodología, pues explora todas las rutas posibles en un grafo. Visitando cada nodo exactamente una vez y finalmente regresando al inicio. Para cada ruta, el método calcula la longitud total y selecciona la ruta con la longitud mínima como la solución óptima.

Ventajas:

Este método garantiza encontrar la ruta óptima para cualquier problema del Viajero (TSP).

Desventajas:

Al realizar iteraciones entre todos los nodos evaluando todas las posibles rutas, el nivel de evaluación lo hace de manera exponencial. Por lo mismo, el costo procesamiento de este método es muy elevado pues a mayores nodos tenga el grafo más tiempo y mayor demanda de procesamiento requerirá.

Por la misma demanda de recursos que este método requiere para procesar y por el tiempo que toma en ejecutarse, lo sesga a ser usado para para casos pequeños con número reducido de nodos.

Ejemplo de uso:

Este método puede ayudar a identificar rutas de viaje entre ciudades evaluando diferentes escalas y finalmente retornar a un mismo punto donde los pesos de los arcos son los costos de pasajes y tiempo de llegada entre cada ciudad.

Programación Dinámica:

Este método plantea descomponer el problema por etapas. Por ende, divide el problema en subproblemas más pequeños y manejables, resolviendo cada uno de ellos de manera óptima y almacenando los resultados para evitar recalcularlos. Luego, utiliza estos resultados almacenados para construir la ruta óptima completa.

Ventajas:

Ofrece una mejor eficiencia computacional que la enumeración completa, especialmente para problemas de mayor tamaño, pues al descomponer el problema en subproblemas más pequeños lo hace más manejable permitiendo técnicas de optimización más eficientes.

Desventajas:

Este método también es de costo significativo pues al hacer iteraciones entre cada subproblema, por ende, si la cantidad de subproblemas es de gran tamaño el costo también ser exponencial por lo mismo requiere una definición cuidados de la cantidad de subproblemas a plantear con el fin de garantizar la optimalidad de la solución.

Ejemplo de uso:

Planificación de recorridos para Técnicos de Servicio. Pues Al considerar múltiples ubicaciones de clientes se debe realizar una programación dinámica puede ayudar a optimizar las rutas de los técnicos para los tiempos de viaje y maximizar la eficiencia del servicio.

Algoritmo de Ramificación y Poda:

Este método divide el problema en subproblemas más pequeños de manera similar a la programación dinámica, pero en lugar de almacenar soluciones óptimas para cada subproblema, utiliza reglas de ramificación para explorar solo las ramas del árbol de búsqueda que tienen potencial para conducir a la solución óptima. Las ramas que no cumplen con ciertos criterios de promesa son podadas, eliminadas, para evitar exploraciones innecesarias.

Ventajas:

La eficiencia de su procesamiento es similar a la programación dinámica en términos de eficiencia computacional, especialmente para problemas de gran tamaño. Pues mediante la poda, eliminación, de ramas que no conducen a soluciones eficientes reduce el costo de búsqueda mejorando la eficiencia de este algoritmo.

Desventajas:

Para eliminar las ramas no eficientes, se requiere tener un criterio de análisis muy cuidadoso con el fin de definir estas reglas de eliminación, pues de no tener claro las reglas podría eliminar posibles soluciones que son eficientes para su uso. Por todo ello requiere conocer todas las casuísticas que hacer que la selección de una posibilidad de ruta no sea eficiente.

Ejemplo de uso:

Se puede aplicar para la planificación de recorridos turísticos donde se debe definir el orden de las ciudades a visitar considerando las restricciones de tiempo, presupuesto y distancia.

Vecino Más Cercano:

Este método fundamentado en la heurística iterativa se encarga de construir una ruta de manera incremental, comenzando desde un nodo inicial seleccionado aleatoriamente. En cada iteración, se selecciona un nodo no visitado siendo más cercana al nodo actual en el que se encuentre para agregar una ruta. El proceso continúa hasta que se hayan visitado todos los nodos y se regrese al punto de partida.

Ventajas:

Es un método simple de implementar y computacionalmente eficiente, haciéndolo adecuado para resolver problemas TSP de manera rápida y aproximada, siendo un método excelente para problemas de pequeña y mediana escala, especialmente cuando la distribución

de los nodos es de manera uniforme.

Desventajas:

Este método no garantiza encontrar la ruta óptima, pues las iteraciones son solo de nodos a nodos y no contempla un análisis del grafo en su totalidad, por lo que lo hace poco confiable en búsqueda de rutas para estructuras complejas. Adicionalmente, se debe considerar que su desempeño y resultado estará en función de una distribución uniforme de los nodos, sesgándolo para casos puntuales.

Ejemplo de uso:

Se puede usar para el caso de una distribución de alimentos entre ubicaciones que son próximas entre sí, pues se considera que la generación de la ruta se hará analizando de punto de entrega a punto de entrega por ende la ruta planteada será producto de encontrar la ruta más próxima entre un nodo origen hacia un nodo más cercano.

2.6.3.2. Teoría de Grafos en el Transporte: Dijkstra

El algoritmo Dijkstra o también conocido como Algoritmo de Caminos Mínimos es una herramienta fundamental para encontrar el camino más corto entre un conjunto de nodos cuyas aristas tienen asignados pesos no negativos. Este algoritmo es un punto central en la teoría de grafos con una amplia y variada aplicación desde los sistemas de navegación, sistemas de información geográfico y optimización en la distribución de redes de telecomunicaciones.

El Algoritmo de Dijkstra fue desarrollado por el científico en computación holandés Edsger Wybe Dijkstra en 1956 y publicando en 1959 mediante un artículo denominado “Un algoritmo para la solución del problema del árbol corto”. Considerar que el algoritmo fue concebido durante la época donde las computadoras se encontraban en una tierna etapa, por ende, la tecnología de aquel entonces aún era incipiente.

Figura 53

Edsger Wybe Dijkstra (1930 – 2002)



Nota. Recurso externo

El Origen de Dijkstra:

Es preciso aclarar, que no existe evidencia oficial de que la idea del algoritmo llegó a Edsger Wybe Dijkstra mientras daba un paseo con su prometida María Debets, pues bajo esta especulación se dice que la idea se concibió de manera tan clara para el matemático que decidió anotarla en una servilleta.

Sin embargo, lo que sí es conocido es que la teoría es producto de un proceso de estudio y análisis riguroso producto, motivado por la necesidad de encontrar caminos cortos en grafos en la década de 1950. Esta necesidad surge por el interés del matemático por optimizar las conexiones en las redes telefónicas. Finalmente, luego de años de investigación, presentó su trabajo en 1959 mediante un artículo académico titulado: "A Note on Two Problems in Connexion with Graphs".

2.6.3.3. Relaciones entre el Problema del Viajero y Dijkstra

A pesar de que el Problema del Viajero (TSP) y el Algoritmo de Dijkstra son herramientas usadas para la optimización de rutas, cada uno tiene enfoques y aplicaciones diferentes.

Tabla 2

Comparación de Ciclos y Caminos Eulerianos y Hamiltonianos

Aspecto	Problema del Viajero (TSP)	Algoritmo de Dijkstra
Objetivo	Es un problema de optimización combinatoria que busca encontrar el ciclo hamiltoniano más corto.	Busca encontrar el camino más corto desde un nodo inicial hacia todos los demás nodos en un grafo con pesos no negativos.
Tipo de Problema	Problema de optimización combinatoria NP-completo.	Algoritmo de búsqueda de caminos más cortos en un grafo con pesos no negativos, con tiempos de procesamiento polinomial.
Complejidad Computacional	NP-completo, pues no hay algoritmos eficientes para todas sus casuísticas.	$O((V+E)\log V)$ utilizando un montículo binario, donde V es el número de nodos y E es el número de arcos.
Aplicabilidad	Aplicar cuando se requiere encontrar el camino más corto que visite todos los nodos únicamente una vez y regrese al punto de inicio.	Aplicar cuando se necesita encontrar el camino más corto desde un nodo inicial hacia todos los demás nodos en un grafo con pesos no negativos.
Ejemplos de Uso	Logística (rutas de entrega), turismo (rutas turísticas), diseño de circuitos integrados.	Redes de telecomunicaciones, enrutamiento de paquetes, sistemas de navegación.

Nota. Elaboración propia

2.6.3.4. Algoritmo de Dijkstra

Como se ha indicado, el Algoritmo Dijkstra es una herramienta clave en la teoría de grafos, y se utiliza ampliamente para determinar el camino más óptimo entre dos nodos (origen – destino) de un grafo cuyas aristas deben estar ponderadas. Su eficiencia y simplicidad lo hacen aplicable para diversas áreas incluso es una herramienta destacada en la actualidad por uso aplicación en la Inteligencia Artificial y la Bioinformática. Para entenderlo es necesario conocer los principios, conceptos e implicancias claves que dan lugar al mismo.

Principios Fundamentales:

- **Optimalidad de Bellman-Ford:** Aunque el algoritmo de Dijkstra no emplea directamente el principio de Bellman-Ford, comparte la idea de construir soluciones parciales óptimas progresivamente. Este principio sugiere que una solución óptima global se obtiene combinando soluciones óptimas locales. Dijkstra asegura en cada paso que el camino más corto hacia cada nodo conocido sea el óptimo localmente.
- **Estructuras de Datos Eficientes:** El uso de estructuras de datos como las colas de prioridad (heaps) permite al algoritmo seleccionar y actualizar eficientemente los nodos según sus distancias actuales desde el nodo de origen. Esto se basa en principios de teoría de algoritmos y estructuras de datos, asegurando un tiempo de ejecución eficiente durante la búsqueda del camino más corto.
- **Conceptos de Grafos Ponderados:** El algoritmo de Dijkstra se aplica a grafos ponderados con aristas de pesos no negativos, garantizando que las distancias calculadas no disminuyan a medida que se expande la frontera de nodos procesados.
- **Algoritmos Voraces (Greedy):** Dijkstra es un algoritmo voraz que toma

decisiones locales óptimas en cada etapa, con la esperanza de que estas decisiones conduzcan a una solución global óptima. Este enfoque está alineado con los principios de diseño de algoritmos eficientes para problemas combinatorios.

Conceptos Matemáticos:

- **Teoría de Grafos:** La teoría de grafos proporciona el marco fundamental para el algoritmo de Dijkstra, representando la red como un grafo con nodos y aristas y definiendo conceptos esenciales como caminos, ciclos y árboles de expansión mínima.
- **Matemática Discreta:** Las herramientas de la matemática discreta son cruciales para manejar conceptos de nodos, aristas, pesos y distancias dentro del contexto del algoritmo de Dijkstra.
- **Algoritmos de Búsqueda:** Dijkstra utiliza técnicas de búsqueda eficientes, como la búsqueda en anchura o profundidad, para explorar el grafo de manera sistemática y ordenada.

Implicaciones:

- **Apreciación de Simplicidad y Eficiencia:** El algoritmo de Dijkstra se basa en principios fundamentales que lo hacen fácil de entender e implementar, manteniendo una alta eficiencia.
- **Versatilidad:** Su capacidad para resolver una amplia gama de problemas de búsqueda de caminos cortos lo hace aplicable en diversos campos y escenarios.
- **Análisis del Comportamiento:** La comprensión de los principios subyacentes permite analizar y evaluar el rendimiento del algoritmo en diferentes contextos y escenarios.
- **Desarrollo de Nuevos Algoritmos:** El conocimiento profundo de estos

principios puede inspirar el desarrollo de nuevos algoritmos para resolver problemas relacionados con la optimización de caminos en grafos.

2.6.3.5. Algoritmo de Dijkstra Mejorado

Antes de comentar las mejoras del algoritmo, se debe tener en cuenta que este es aplicable para grafos con las siguientes características:

- Grafos dirigidos y no dirigidos.
- Grafos ponderados, que no tenga pesos negativos.

Durante los años siguientes al planteamiento del Algoritmo de Dijkstra, se han planteado varias mejoras que buscan optimizar el rendimiento del mismo, especialmente para su aplicación en grafos grandes. Algunas de las más conocidas se describen a continuación:

Usar una Cola de Prioridad con un Montículo (Heap):

- En lugar de una simple lista o arreglo, se utiliza un montículo binario (binary heap) o un montículo de Fibonacci (Fibonacci heap) para implementar la cola de prioridad. Esto reduce el tiempo de actualización y selección del siguiente nodo más cercano.
- Utilizando un montículo binario, la complejidad es $O((E + V) \log V)$, donde E es el número de aristas y V es el número de nodos.

Etiqueta de Distancia y Relajación Optimizada:

- Algunas versiones optimizadas del algoritmo de Dijkstra aplican técnicas de etiqueta de distancia y relajación que pueden mejorar el rendimiento en ciertos tipos de grafos.

Dijkstra Bidireccional:

- Esta versión del algoritmo de Dijkstra realiza una búsqueda simultánea desde el nodo de origen y el nodo de destino. La búsqueda se detiene cuando ambas búsquedas se encuentran, lo que puede reducir significativamente el número de

nodos explorados.

- Aplicación: Esto es útil en casos donde se conoce tanto el nodo de inicio como el nodo de destino.

A (A-Star) Algoritmo*:

- Aunque no es una mejora directa del algoritmo de Dijkstra, el algoritmo A* es una variación que utiliza una heurística para guiar la búsqueda, reduciendo así el número de nodos explorados. Combina la distancia desde el nodo inicial con una estimación heurística de la distancia al nodo objetivo.

2.6.3.6. Aplicación del Algoritmo de Dijkstra

En esta sección se detalla cómo se aplica el algoritmo de Dijkstra. Para determinar los caminos óptimos a los diferentes nodos de un grafo desde un nodo inicial.

Ejemplo 01: Aplicación de Dijkstra con Grafo No Dirigido

- El Algoritmo de Dijkstra no soporta pesos negativos.
- Permite retorna el camino de menor costo desde un nodo origen hacia un nodo cualquiera de los nodos que tiene conexión.
- Incorpora etiquetas en sus nodos:
 - Estructura de Etiqueta [Nodo Predecesor, Costo Mínimo desde Nodo Origen]
 - Marca de Nodo Visitado / Analizado, para evitar bucles
 - Se puede mantener la información de etiquetas y nodos marcados en una tabla, arreglos, diccionarios.
- Planteamiento de Algoritmo:
 - Costo mínimo de nodos = infinito (o máximo de entero).
 - Predecesores de nodo = nulo (o a sí mismo).
 - Costo de nodo inicial = 0.

- Todos los nodos sin marcar.

Figura 54

Algoritmo para ejemplo

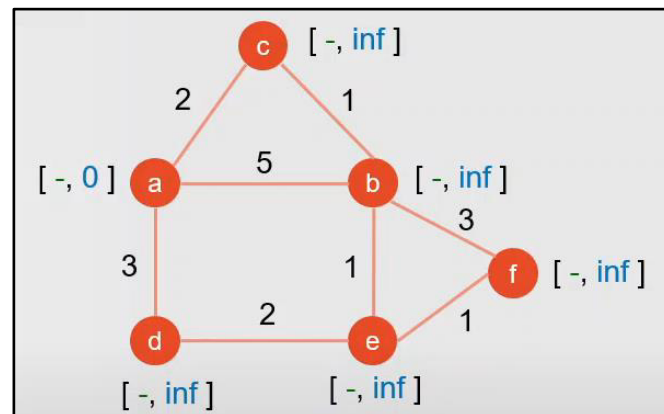


Nota. Recurso externo

- Grafo: Se busca identificar la mejor ruta desde el nodo A hacia el nodo F.

Figura 55

Ejercicio 01



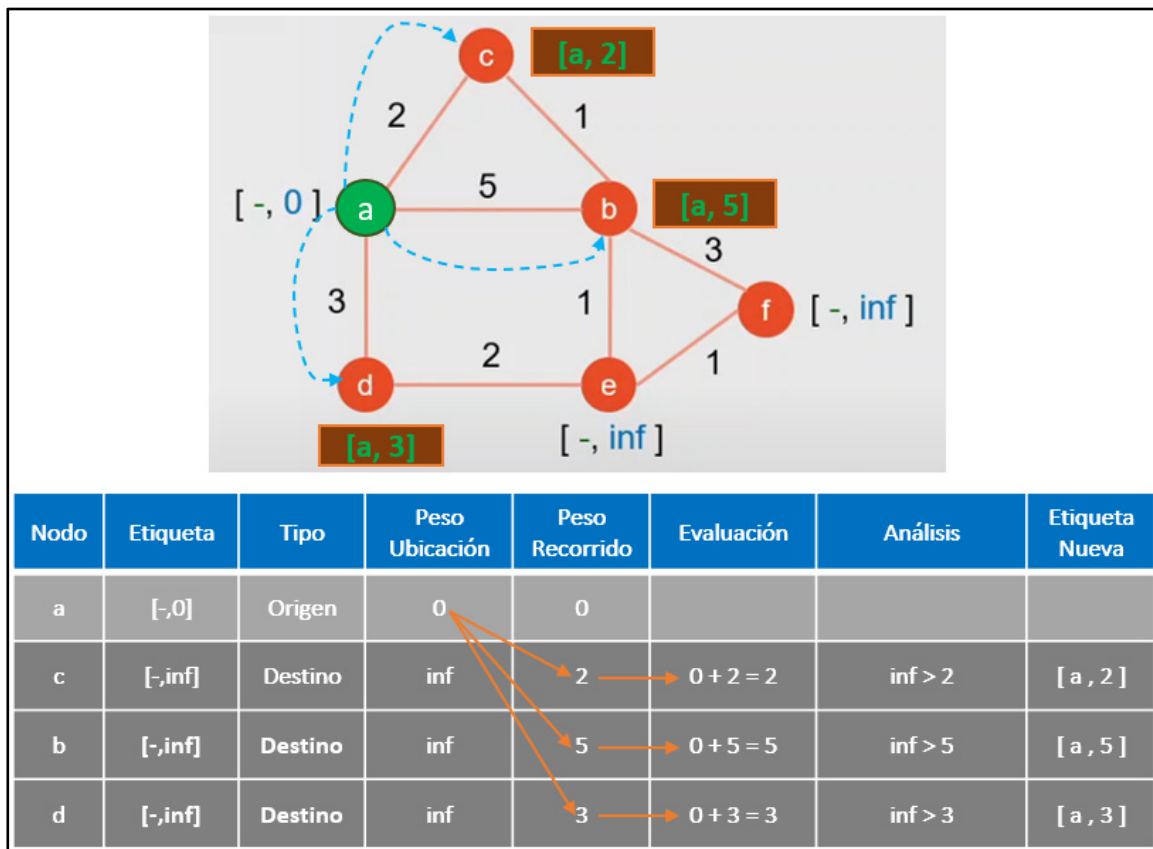
Nota. Recurso externo

- Resolución:

Paso 01: Se evalúan los nodos B, C, D respecto al nodo A, asignándole los pesos de sus aristas que lo llevan hacia cada nodo respectivo. Y el nodo A se marca como ya visitado pues de ahí se partió.

Figura 56

Ejercicio 01 - Paso 01



Nota. Recurso externo personalizado

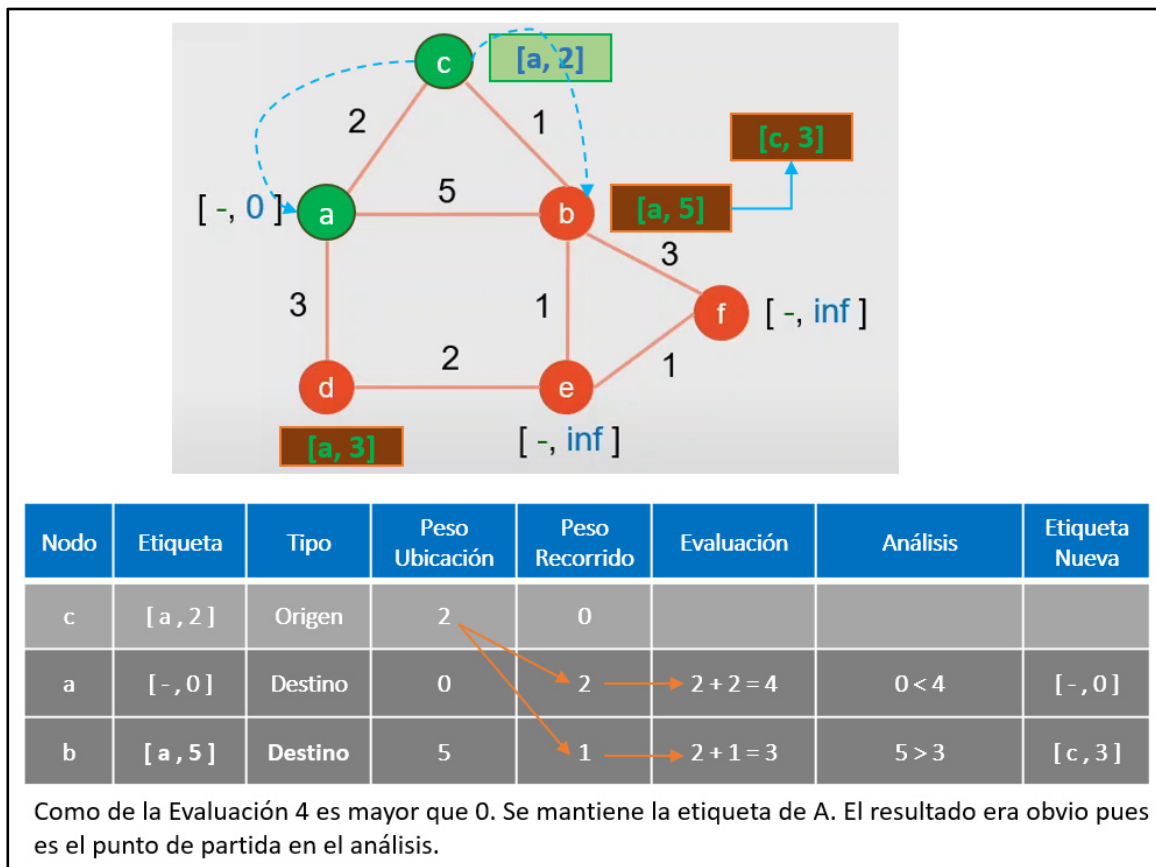
Paso 02: Se evalúa el nodo C respecto a sus conexiones A y B.

Analizamos desde el nodo C hacia el nodo A, comparamos que al sumar 2 del nodo C más 2 de la arista que lleva de C hacia A sale 4 y 4 no es menor que el valor 0 del nodo que lleva hacia A por ende mantenemos el valor de 2 desde A en la etiqueta del nodo C.

Y seguido analizamos del nodo C hacia el nodo B y realizamos el mismo análisis, sumamos 2 que es el valor en C más el peso de la arista que lleva hacia B que es 1 y nos da 3 de lo cual podemos decir que 3 es menor a 5 por ende una llegada más rápida hacia B es desde C, en lugar de ir de A hacia B por ende se asigna una nueva etiqueta para el nodo B siendo esta [C, 3].

Figura 57

Ejercicio 01 - Paso 02



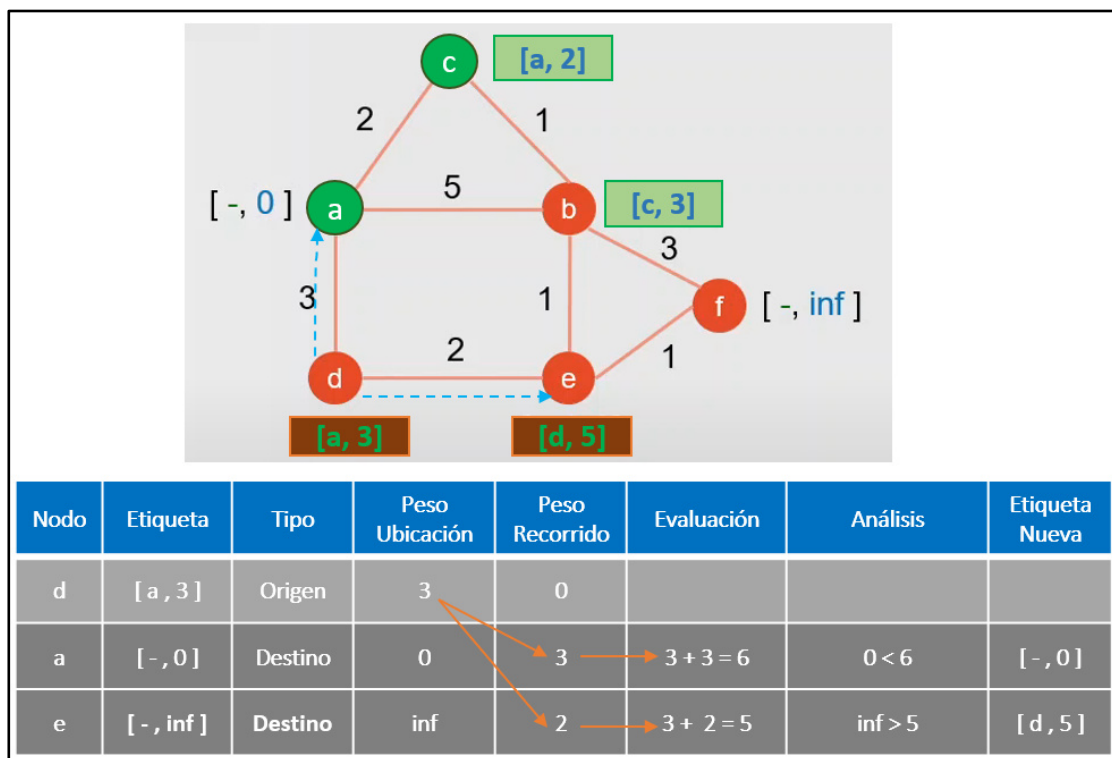
Nota. Recurso externo personalizado

Paso 03: Se evalúa el nodo D respecto a sus conexiones A y E.

Analizamos desde el nodo D hacia el nodo A, comparamos que al sumar 3 del nodo D más 3 de la arista que lleva de D hacia A sale 6, y 6 no es menor que el valor 0 del nodo que lleva hacia A por ende mantenemos el valor de la etiqueta del nodo A. Seguido analizamos del nodo D hacia el nodo E y realizamos el mismo análisis, sumamos 3 que es el valor en D más el peso de la arista que lleva hacia E que es 2 y nos da 5 de lo cual podemos decir que 5 es menor a cualquier valor no conocido por ende un el valor del nodo E en este momento del análisis tiende al infinito y por lo mismo 5 es menor a cualquier valor desconocido en el nodo E, finalmente se actualiza la etiqueta para el nodo E siendo esta [D, 5].

Figura 58

Ejercicio 01 - Paso 03



Nota. Recurso externo personalizado

Paso 04: Se evalúa el nodo B, cuyo peso es de 3, respecto a sus conexiones A, C, E y F. Se prioriza B en lugar de E, pues entre E y B el peso menor del nodo que le antecede está en C pues tiene como etiqueta [A, 2] mientras que D que antecede a E tiene como etiqueta [A, 3]; por ende, el menor peso está en C y por lo mismo el nodo que le precede tiene mayor relevancia.

Analizamos desde el nodo B hacia el nodo C, y comparamos que al sumar el peso del nodo más el peso de la arista que lleva de B hacia C sale 4, y 4 es mayor que el valor 2 del nodo C por ende mantenemos el valor de la etiqueta del nodo C ([A, 2]). Seguido analizamos del nodo B hacia el nodo A, y del mismo modo se suma el peso del nodo B más el peso de la arista que conecta B hacia A dando como valor 8, y 8 es mayor que el valor de A, con mayor razón por ser el punto origen del recorrido

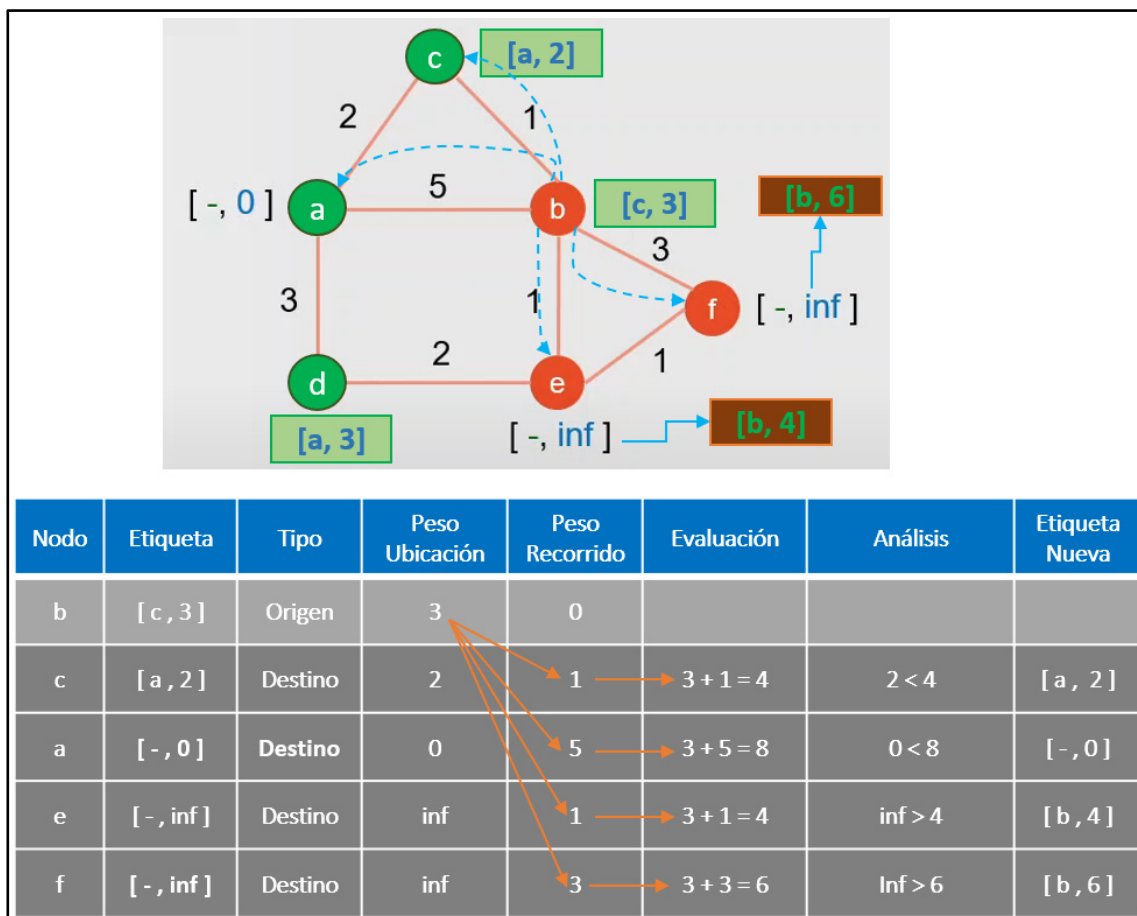
por ende se mantiene la etiqueta de A $[-, 0]$.

Luego se analiza la conexión de B hacia E, para lo cual sumamos el peso de E más el recorrido que lleva de B hacia E dando un resultado igual a 4, en este caso E no tiene definido un valor aún por ende tiende al infinito por ende 4 es menor que el infinito y se define una etiqueta propuesta para E siendo esta $[B, 4]$.

Y como la última conexión de B se analiza de B hacia F, realizando el mismo proceso de suma se identifica que tiene un valor de 6, y como F no tiene definido un peso aún este también tiende al infinito y se le asigna a F una nueva etiqueta propuesta siendo esta $[B, 6]$.

Figura 59

Ejercicio 01 - Paso 04



Nota. Recurso externo personalizado

Paso 05: Se evalúa el nodo E, cuyo peso es de 4, respecto a sus conexiones D, B y F.

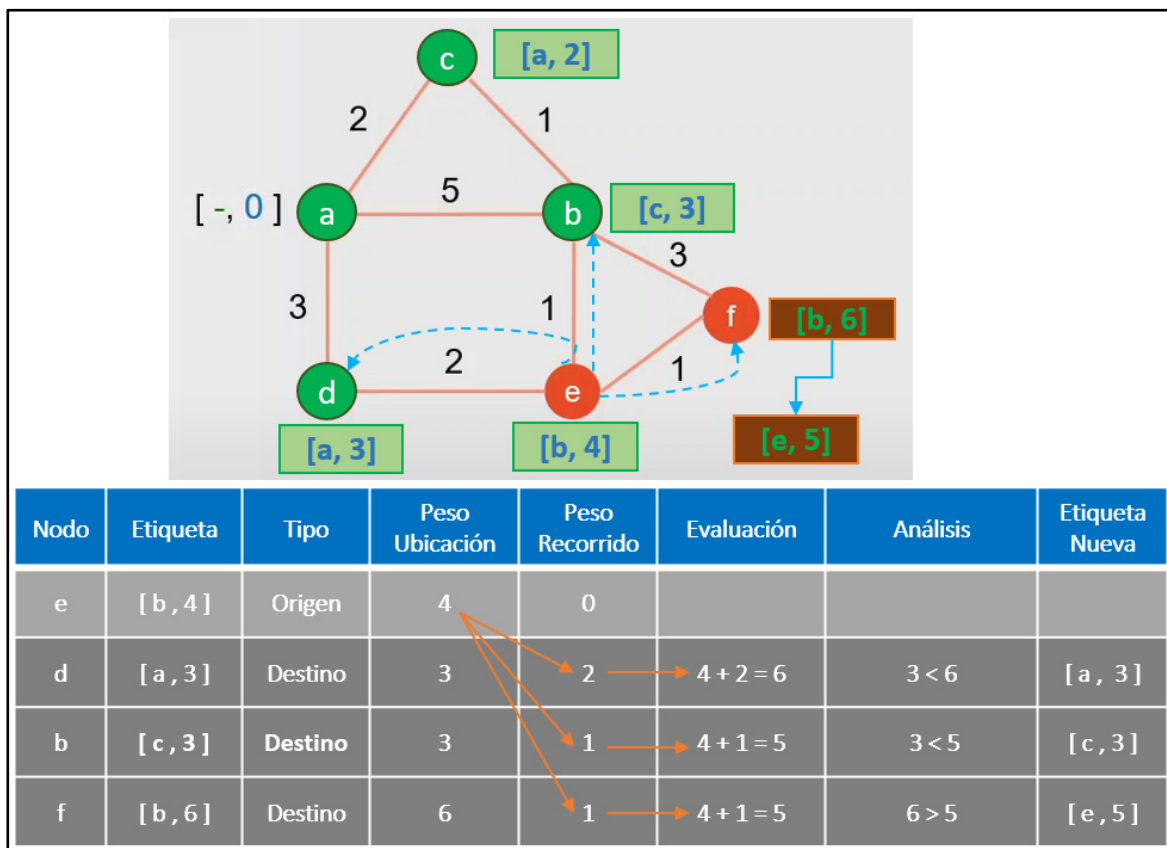
Analizamos desde el nodo E hacia el nodo D, y vemos que de la suma de su peso más la arista es 6, y 6 es mayor que el valor 3 del nodo D por ende mantenemos el valor de la etiqueta del nodo D ([A, 3]). Y del mismo modo entre la conexión del nodo E y B.

Sin embargo, al realizar el análisis entre la conexión entre E y F, podemos ver que el valor de la suma entre el peso del nodo E más el valor de la arista que lleva hacia F da como valor 5, siendo este valor menor que el peso asignado a F, menor a 6.

Por ende, la etiqueta del vértice F se actualiza con el valor de: [E, 5].

Figura 60

Ejercicio 01 - Paso 05

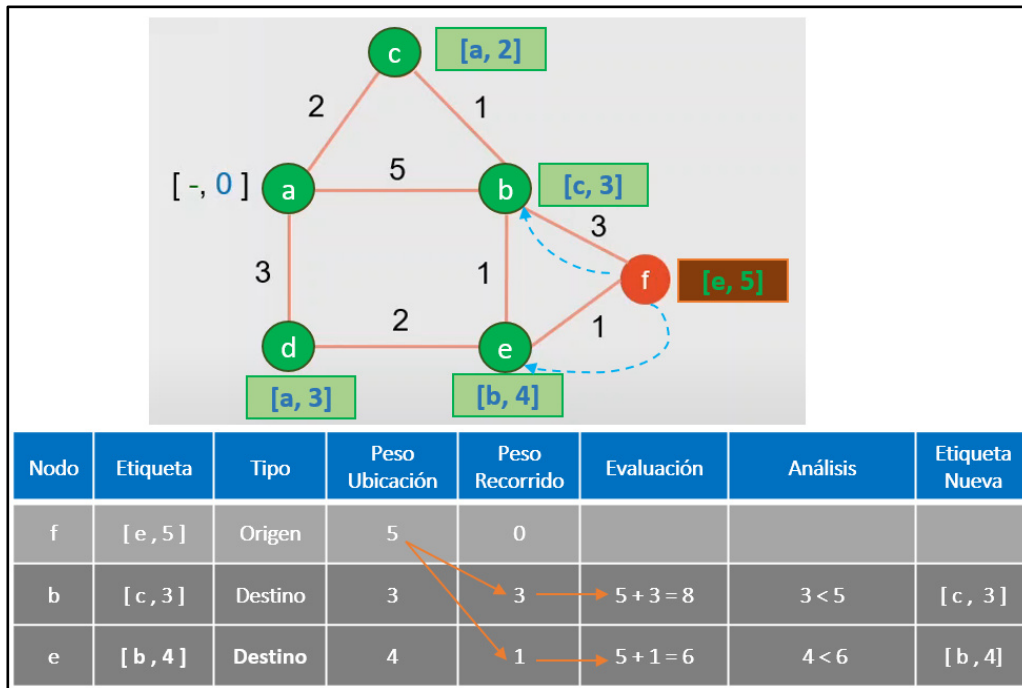


Nota. Recurso externo personalizado

Paso 06: Se evalúa el nodo F, cuyo peso es de 5, respecto a sus conexiones B y E.
 Al realizar el análisis para ambos nodos nos damos cuenta que los valores superan sus pesos asignados por ende se mantienen las etiquetas asignadas.

Figura 61

Ejercicio 01 - Paso 06

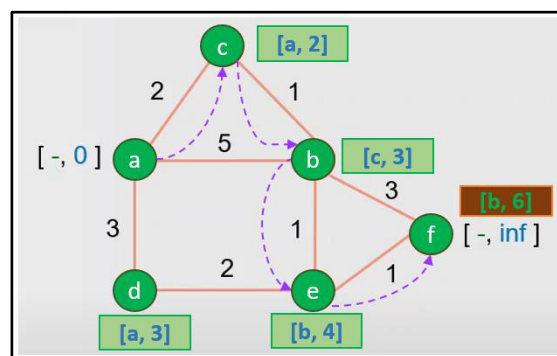


Nota. Recurso externo personalizado

Paso 07: El recorrido más corto es: “A - C - B - E - F”. Con un costo de “5”.

Figura 62

Ejercicio 01 - Paso 07



Nota. Recurso externo personalizado

2.6.4. Base Técnica

Considerando la necesidad de implementar una herramienta de análisis de rutas usando el sistema de transporte regular de Lima y Callao. Esta sección pretende explicar la relación que existe entre los GIS y la tecnología, poniendo énfasis en los componentes tecnológicos usados a nivel de software para cumplir con el alcance del proyecto.

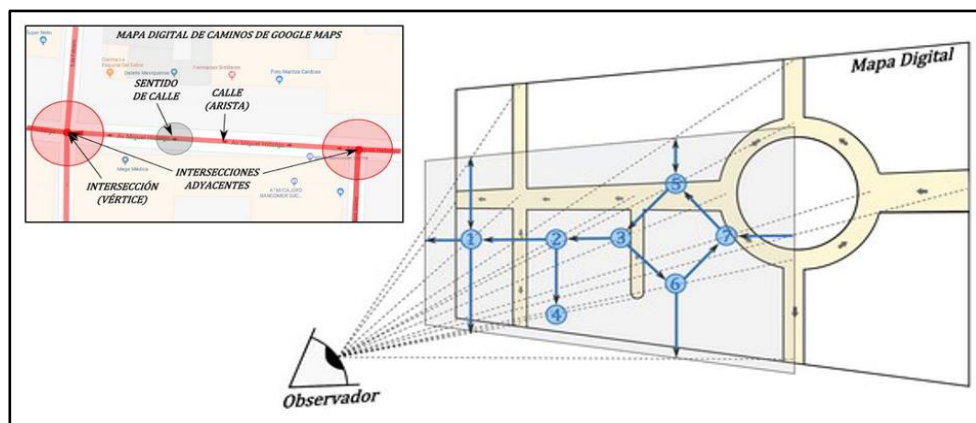
2.6.4.1. Teoría de Grafos en la Programación

Como se ha explicado, los grafos son estructuras de nodos (vértices) conectados a través de arcos (aristas). Los cuales se usan para representar diferentes estructuras de datos con el fin de solucionar problemas de la vida real.

Uno de los ejemplos más intuitivos de la aplicación de grafos en el mundo real es el modelado de redes de transporte. En este caso, los nodos pueden representar ciudades, intersecciones o puntos de interés, mientras que las aristas simbolizan las carreteras, vías férreas o rutas aéreas que conectan estos puntos. Los pesos asociados a las aristas pueden reflejar diferentes aspectos relevantes, como la distancia entre dos ubicaciones, el tiempo estimado de viaje o el costo del transporte. Esta representación gráfica permite abordar problemas cruciales en la planificación y gestión del transporte.

Figura 63

Representación de Mapa en un Grafo



Nota. Recurso externo personalizado

Por ejemplo, el algoritmo de Dijkstra, que encuentra el camino más corto entre dos nodos en un grafo ponderado, puede utilizarse para calcular la ruta óptima para una persona que desea atravesar la ciudad usando un servicio de transporte público, minimizando la distancia recorrida y el tiempo recorrido. Además, los grafos pueden emplearse para analizar la conectividad de la red, identificar puntos críticos de congestión o evaluar el impacto de cierres de carreteras.

Figura 64

Sección del Grafo formado entre Paraderos y Ruta de Servicios



Nota. Recurso externo personalizado

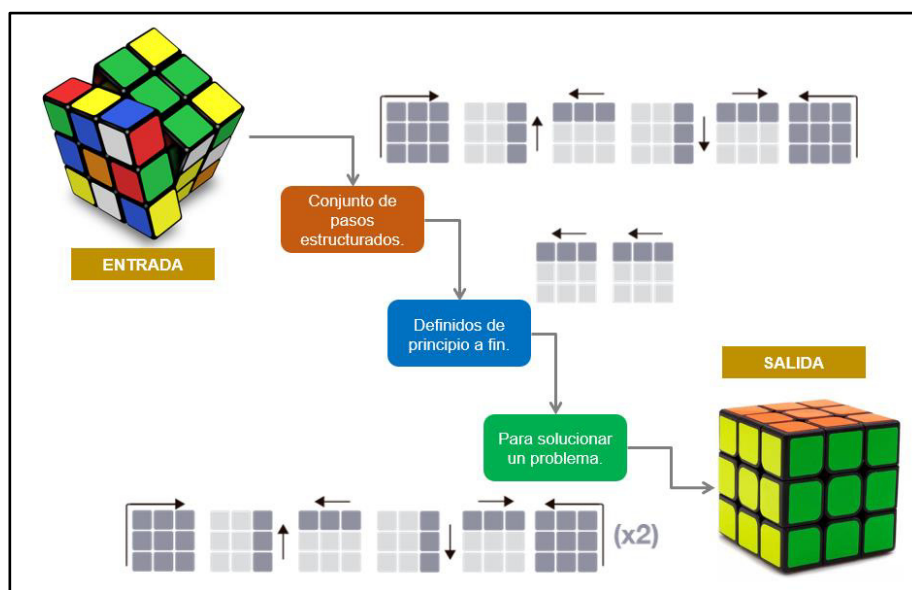
2.6.4.2. Algoritmos en la Programación

La programación permite diseñar soluciones a problemas referentes a la gestión de datos hasta la optimización de procesos. En este proceso, que los algoritmos emergen como los pilares fundamentales sobre los cuales se construyen estas soluciones.

Un algoritmo se puede definir como una secuencia finita de pasos lógicos y bien definidos, que al ser seguido de manera estricta conducen a la resolución de un problema en específico. Por ende, la importancia de un algoritmo radica en su capacidad para transformar ideas abstractas en acciones concretas, optimizando el uso de recursos y garantizando la obtención de resultados correctos.

Figura 65

Ejemplo de Algoritmo con Cubo Rubik



Nota. Recurso externo personalizado

La historia de la informática está compuesta por la concepción e implementación de los algoritmos, pues estos han permitido revolucionar a la tecnología partiendo desde aquellos que permiten organizar grandes volúmenes de datos de manera eficiente hasta aquellos tan sofisticados que han dado lugar al aprendizaje automático impulsando la inteligencia artificial a partir del reconocimiento de patrones. Es en este contexto que el algoritmo de Dijkstra toma lugar como una joya invaluable en la teoría de grafos, por buscar determinar los caminos más cortos entre nodos de un grafo ponderado.

Se podría mencionar de ejemplo los algoritmos para generar el grafo, así como los respectivos pasos para ejecutar Dijkstra.

Algoritmo Crear Grafo:

Estructuración del Grafo mediante la implementación de Lista de Adyacencia.

Paso 1: Definición de Estructuras de Datos

- Definir una clase Nodo para representar cada nodo del grafo.
- Definir una clase Arista para representar las conexiones entre nodos con pesos.
- Definir una clase Grafo para almacenar nodos y aristas.

Paso 2: Agregar Nodos

- Crear nodos y añadirlos al grafo.
- Cada nodo puede tener atributos como identificador, nombre, y coordenadas.

Paso 3: Agregar Aristas:

- Establecer conexiones entre nodos mediante aristas.
- Cada arista debe tener un peso asociado, que puede representar distancia, tiempo, o costo.

Paso 4: Representación del Grafo:

- Utilizar una lista de adyacencia o una matriz de adyacencia para representar las conexiones entre nodos.

Figura 66*Seudocódigo para Crear el Grafo*

```

1  función CrearGrafo():
2  |   grafo = nuevo Grafo
3  |   retornar grafo
4
5  función AgregarNodo(grafo, id):
6  |   nodo = nuevo Nodo(id)
7  |   grafo.agregar_nodo(nodo)
8
9  función AgregarArista(grafo, origen_id, destino_id, peso):
10 |   grafo.agregar_arista(origen_id, destino_id, peso)
11
12 función RepresentarGrafo(grafo):
13 |   para cada nodo en grafo.nodos:
14 |       imprimir nodo.id
15 |       para cada arista en nodo.aristas:
16 |           imprimir "Conecta a:", arista.destino.id, "con peso:", arista.peso
17

```

Nota. Elaboración propia**Algoritmo Dijkstra:**

Secuencia para implementar Dijkstra.

Paso 1: Inicialización

- Asigna Distancias Tentativas:
 - o Asigna a cada nodo una distancia tentativa: 0 para el nodo inicial y ∞ (infinito) para todos los demás nodos.
- Marca Nodos:
 - o Marca el nodo inicial como visitado y los demás nodos como no visitados.
- Crea un Conjunto de Nodos Visitados:
 - o Inicialmente, el conjunto de nodos visitados está vacío.

Paso 2: Selección del Nodo Más Cercano

Mientras haya Nodos No Visitados:

- Selecciona el nodo no visitado con la distancia tentativa más pequeña.

- Márcalo como visitado y añádelo al conjunto de nodos visitados.

Paso 3: Actualización de Distancias Tentativas

Para cada Vecino No Visitado del Nodo Seleccionado:

- Calcula la distancia desde el nodo inicial hasta el vecino pasando por el nodo seleccionado.
- Si esta distancia es menor que la distancia tentativa actual del vecino:
 - o Actualiza la distancia tentativa del vecino.
 - o Registra el nodo seleccionado como el predecesor del vecino en el camino más corto.

Paso 4: Repetición

Repite los Pasos 2 y 3:

- Continúa con la selección del nodo más cercano y la actualización de distancias tentativas hasta que todos los nodos hayan sido visitados.
- Alternativamente, si solo se busca el camino más corto a un nodo específico, repite hasta alcanzar el nodo destino.

Paso 5: Reconstrucción del Camino

Para Encontrar el Camino Más Corto Completo:

- Reconstruye el camino desde el nodo destino hasta el nodo inicial siguiendo los predecesores.

Figura 67*Seudocódigo del Algoritmo de Dijkstra*

```

17
18 función Dijkstra(Grafo G, nodo_inicio s):
19     // Paso 1: Inicialización
20     para cada nodo v en G:
21         si v ≠ s:
22             distancias[v] = ∞ // Inicializa la distancia a infinito
23         sino:
24             distancias[v] = 0 // La distancia al nodo inicial es 0
25             previo[v] = null // Inicializa los predecesores como null
26             no_visitados.agregar(v) // Agrega el nodo al conjunto de no visitados
27
28     // Paso 2: Selección del Nodo Más Cercano
29     mientras no_visitados no esté vacío:
30         u = nodo con la distancia tentativa más pequeña en no_visitados
31         no_visitados.eliminar(u) // Marca el nodo como visitado
32
33     // Paso 3: Actualización de Distancias Tentativas
34     para cada vecino v de u:
35         si v está en no_visitados:
36             distancia_alternativa = distancias[u] + peso(u, v)
37
38             si distancia_alternativa < distancias[v]:
39                 distancias[v] = distancia_alternativa
40                 previo[v] = u
41                 // Actualiza la prioridad del vecino en la cola de prioridad
42                 actualizar_prioridad(Q, v)
43
44     // Paso 4: Repetición
45     // El proceso de repetición está implícito en el bucle while de selección del nodo más cercano
46
47     // Paso 5: Reconstrucción del Camino
48     retornar distancias, previo

```

Nota. Elaboración propia

2.6.4.3. GIS y TIC

En la era digital, las Tecnologías de la Información y la Comunicación (TIC) han permeado cada aspecto de nuestra vida, revolucionando la forma en que interactuamos, trabajamos y nos relacionamos con el entorno. Y dentro de este vasto universo tecnológico, los Sistemas de Información Geográfica (GIS) emergen como una herramienta poderosa, capaz de cumplir con acciones propias de las TIC, tales como capturar, analizar y visualizar datos pero desde un enfoque espacial. Es justamente que, en la convergencia entre TIC y GIS se ha desencadenado un torrente de innovaciones que mejoran la eficiencia, accesibilidad y sostenibilidad de los sistemas de transporte, con un impacto directo en la calidad de vida de los ciudadanos.

Tecnología de la Información y la Comunicación (TIC):

Las TIC están constituidas por una amplia gama de herramientas y técnicas para la creación, almacenamiento, procesamiento y transmisión de información, compuesta por hardware, software, redes y los dispositivos que facilitan la comunicación y gestión de los datos. Todo ello permite, tejer una red invisible que conecta individuos, organizaciones y gobiernos en todo el mundo. El sector transporte, no ha sido ajeno a esta realidad, pues para el mismo existe una gran variedad de aplicaciones móviles, sistemas de pagos electrónicos y otras plataformas como algunas de las herramientas de las TIC que influyen en como los usuarios interactúan con los sistemas de transporte.

Sistemas de Información Geográficos (GIS):

Por su parte, los GIS proporcionan un marco espacial esencial para analizar y visualizar los datos generados por las TIC. Estos datos, que pueden proceder de diversas fuentes, contienen información sobre la ubicación y las características de objetos y fenómenos en el espacio, lo cual es crucial para entender patrones, relaciones y tendencias espaciales. Es esta capacidad de los GIS para integrar y representar datos en mapas que los convierte en herramientas invaluable para la toma de decisiones.

En el ámbito del transporte, los GIS desempeñan un papel crucial en la planificación y gestión de redes de transporte. Al analizar datos espaciales sobre flujos de tráfico, demanda de transporte, infraestructura existente y patrones de movilidad, los GIS permiten optimizar rutas, horarios y frecuencias de servicios, mejorando la eficiencia del transporte público y reduciendo costos operativos. Además, al modelar escenarios de transporte y evaluar el impacto de diferentes políticas, los GIS contribuyen a la reducción de emisiones de carbono y a la promoción de un transporte más sostenible.

Convergencia TIC y GIS en el Transporte Público:

La convergencia entre TIC y GIS ha dado lugar a una sinergia poderosa que está transformando la forma en que se abordan los desafíos de planificación urbana y la misma gestión del transporte. La capacidad de las TIC para recopilar, transmitir y procesar grandes volúmenes de datos en tiempo real se complementa con la capacidad de los GIS para analizar y visualizar estos datos en un contexto espacial permitiendo un análisis más profundo y contextualizado de la información.

Entre algunos casos de éxitos producto de la búsqueda de soluciones innovadoras, se pueden mencionar:

- Sistemas de información al usuario en tiempo real: Aplicaciones móviles como "Moovit" o "Citymapper" utilizan datos GIS y TIC para proporcionar a los usuarios información actualizada sobre horarios, rutas, tiempos de espera y alertas de servicio, mejorando la experiencia del usuario y fomentando el uso del transporte público.
- Planificación de rutas y horarios optimizados: Empresas de transporte como "Google Transit" o "Transit App" utilizan algoritmos GIS y datos de tráfico en tiempo real para optimizar las rutas y horarios de los servicios de transporte público, reduciendo los tiempos de viaje y mejorando la eficiencia.
- Sistemas de pago electrónico integrados: Tarjetas inteligentes como la "Tarjeta bip!" en Santiago de Chile o la "Tarjeta SUBE" en Buenos Aires utilizan tecnología TIC y GIS para facilitar el pago de pasajes y agilizar el acceso al transporte público, mejorando la experiencia del usuario y reduciendo los costos operativos.
- Análisis de datos para la toma de decisiones: Plataformas de análisis de datos como "Remix" o "TransLoc" utilizan datos GIS y TIC para analizar patrones de

movilidad, demanda y satisfacción del usuario, proporcionando información valiosa para la toma de decisiones estratégicas y la planificación de servicios de transporte más eficientes y equitativos.

2.6.4.4. Base de Datos (BD)

Las bases de datos son herramientas fundamentales para el almacenamiento, gestión, procesamiento y análisis de los datos en una variedad de aplicaciones.

Hoy en día, en la era de la información, los datos pueden ser capturados por diferentes fuentes mediante las TIC, y los datos geoespaciales no son la excepción. Los GIS al permitir integrar diferentes orígenes de datos usan estos repositorios como insumo para analizar desde un enfoque cartográfico la información, por este hecho las Bases de Datos juegan un rol importante para los GIS. El sector transporte aprovecha estas ventajas para mejorar la planificación y operación de sus servicios, usando este componente para poder tratar los datos, con el fin de realizar:

- Planificación de rutas.
- Gestión de flotas.
- Análisis de accidentes.
- Modelización del Transporte.

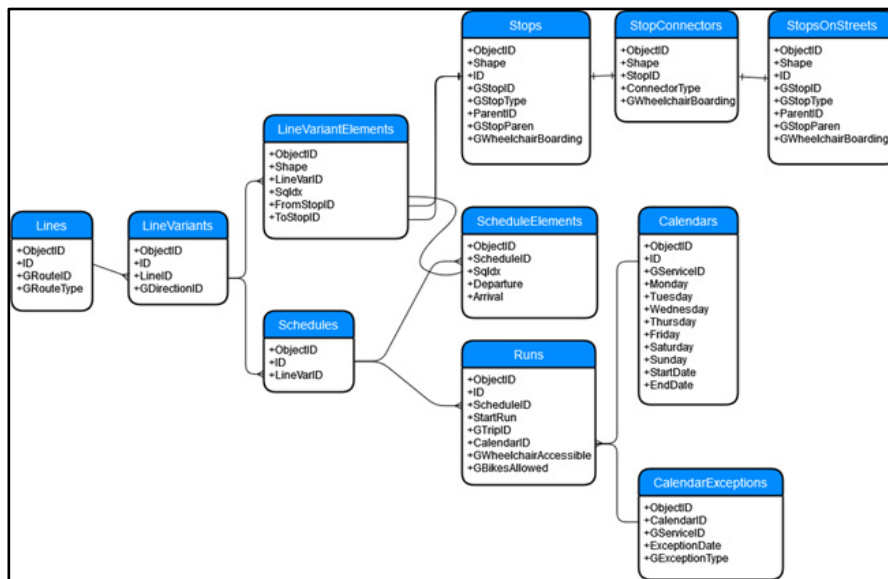
Las bases de datos se clasifican principalmente en dos tipos relacionales y no relacionales.

Base de Datos Relacionales:

Estas BD organizan los datos en tablas con filas y columnas, utilizando un sistema de gestión de bases de datos relacional (RDBMS), es decir, su estructura se basa en tablas interconectadas mediante relaciones predefinidas esto permite realizar consultas complejas y mantener una integridad de los datos.

Figura 68

Ejemplo de Estructura de BD Relacional orientado al Análisis de Redes de Transporte



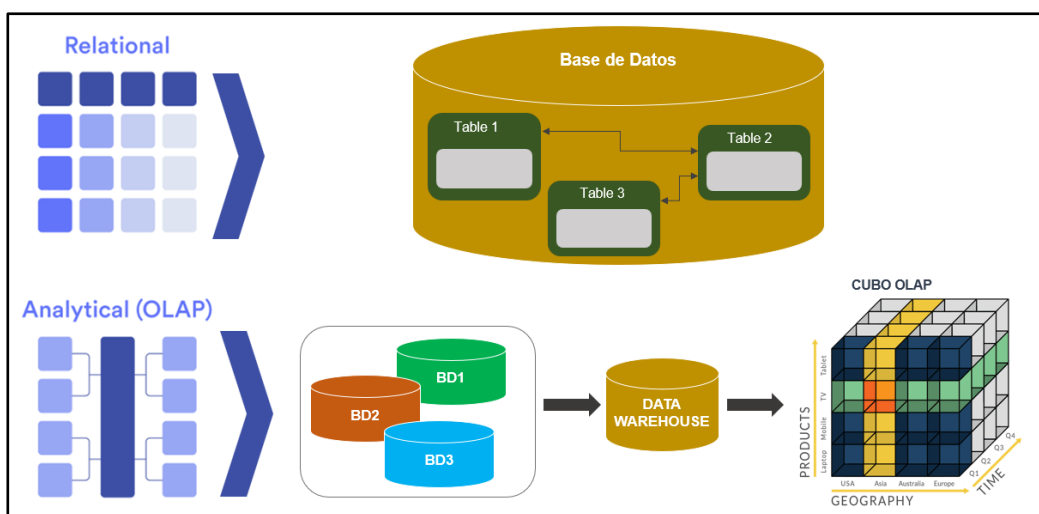
Nota. Recurso externo

Para los GIS, son ideales pues permiten almacenar información geoespacial estructurada como atributos de entidades geográficas y relaciones topológicas.

aplicaciones diferentes.

Figura 69

Base de Datos Relacional



Nota. Elaboración propia

Tabla 3*Algunas Tecnologías de Bases de Datos Relacionales y Aplicaciones GIS*

Tecnología de BD Relacional	Requisitos	Aplicación GIS
MySQL	Conocimientos en optimización combinatoria y algoritmos de grafos. Licenciamiento: Open Source (GPL)	Resolución de problemas de optimización de rutas, como encontrar el camino más corto (problema del viajante).
PostgreSQL (PostGIS)	Conocimientos en modelado de datos espaciales y lenguaje SQL. Licenciamiento: Open Source (GPL)	Almacenamiento, análisis y consulta de datos geoespaciales estructurados (puntos, líneas, polígonos), gestión de topología de redes, soporte para estándares OGC.
Oracle Spatial	Infraestructura empresarial robusta, conocimientos en administración de bases de datos Oracle y lenguaje SQL. Licenciamiento: Comercial	Gestión de grandes volúmenes de datos geoespaciales en entornos empresariales, análisis espacial complejo, soporte para estándares OGC.
MySQL Spatial	Conocimientos básicos en modelado de datos espaciales y lenguaje SQL. Licenciamiento: Open Source (GPL)	Almacenamiento y análisis de datos geoespaciales estructurados, similar a PostgreSQL pero con menos funcionalidades avanzadas.
SQL Server	Conocimientos en administración de bases de datos SQL Server y lenguaje SQL, así como en extensiones espaciales como SQL Server Spatial o terceros. Licenciamiento: Comercial	Almacenamiento, análisis y consulta de datos geoespaciales estructurados, integración con herramientas de Microsoft y soporte para estándares OGC.

Nota. Elaboración propia

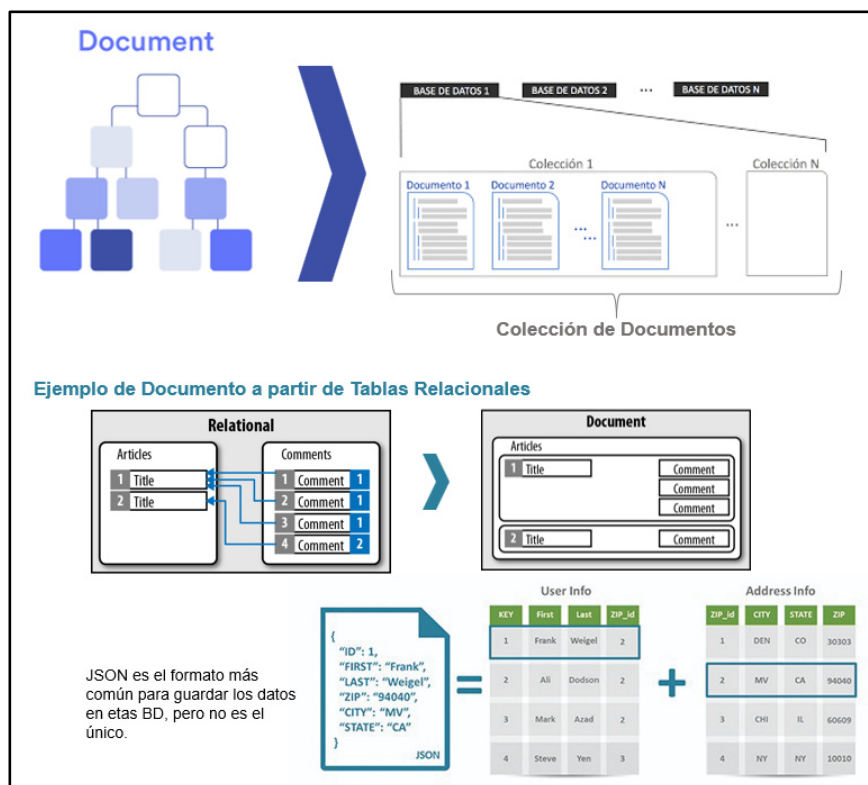
Base de Datos No Relacionales (NoSQL):

Las Bases de Datos No Relacionales, son conocidas comúnmente como Bases de Datos NoSQL. Su nombre hace referencia a que no se estructuran del mismo modo que una BD Relacional, pues su enfoque es la flexibilidad y el manejo de grandes volúmenes de datos no estructurados y semi-estructurados con el fin de brindar escalabilidad y alto rendimiento. Las Bases de Datos NoSQL se organizan de diversas maneras según su aplicación, teniendo como principales categorías:

- Documentales: Permiten almacenar y procesar grandes volúmenes de datos como las imágenes de drones o datos a tiempo real de equipos de telemetría. Suelen usar el formato JSON para almacenar los datos, aunque no es único permitido por esta categoría de BD No SQL.

Figura 70

BD NoSQL Documental

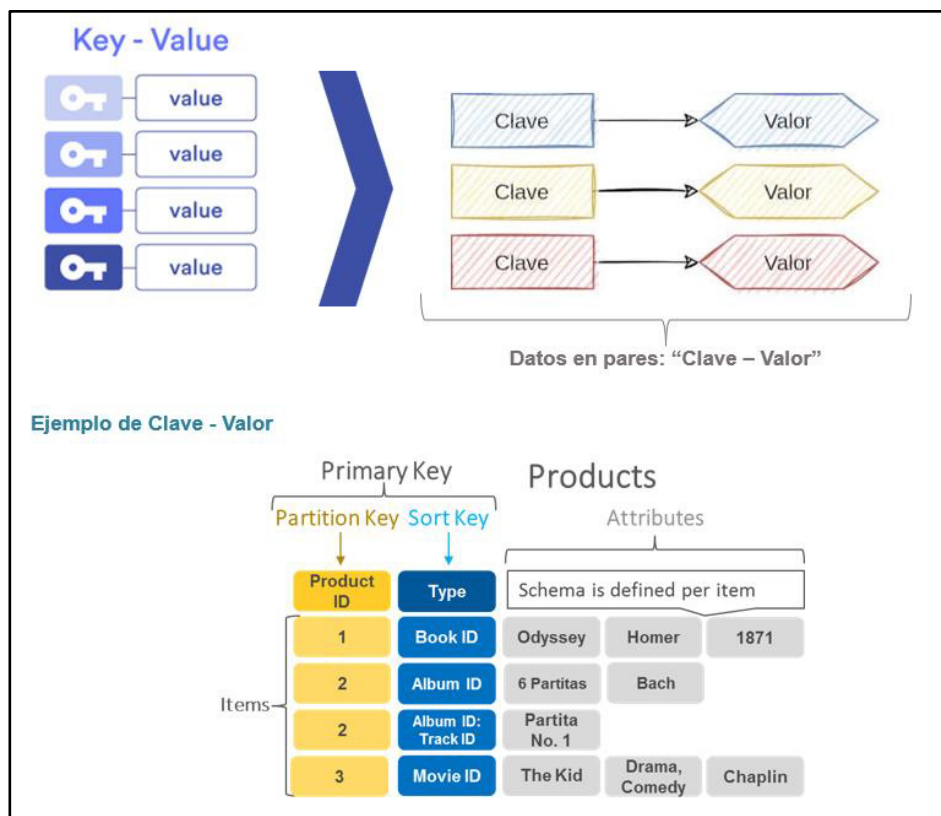


Nota. Recurso externo personalizado

- Clave-Valor: Son ideales para almacenar y recuperar datos mediante consultas simples, donde la relación entre clave y valor es directa y no hay necesidad de realizar consultas complejas. La clave es un identificador único y los valores pueden ser desde un objeto simple hasta estructuras complejas, por lo mismo son altamente divisibles y permiten escalado horizontal. Se suelen usar para almacenar y actualizar rápidamente la ubicación de objetos en movimiento como el tracking de los vehículos.

Figura 71

BD NoSQL Clave - Valor



Nota. Recurso externo personalizado

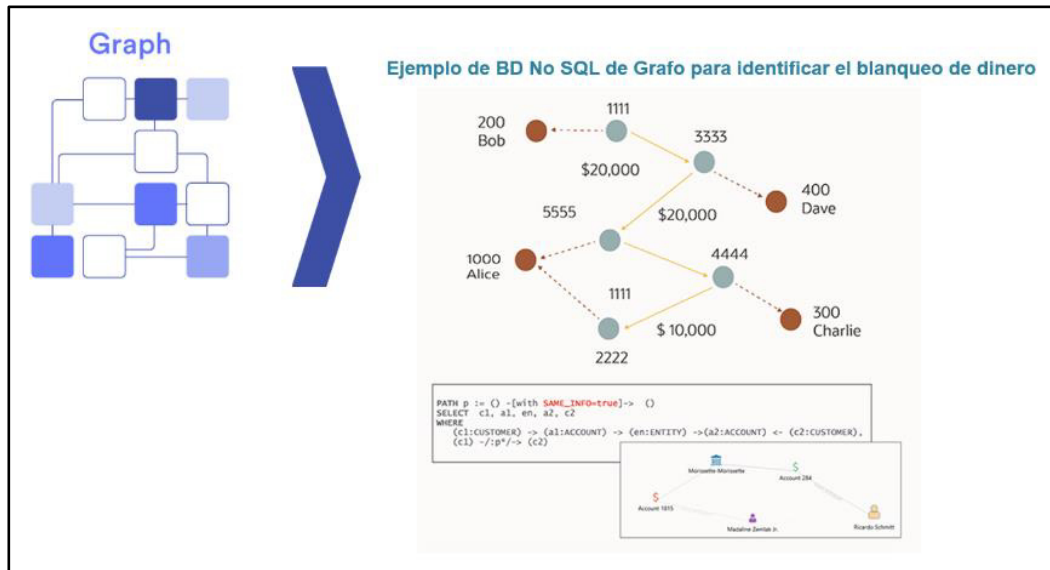
- De Grafos: Son adecuadas para modelar y analizar relaciones complejas entre entidades, donde las conexiones y la estructura de la red es importante. Su capacidad para representar y consultar relaciones de manera eficiente las hace ideales para una variedad de aplicaciones, incluyendo algunas con un enfoque

más científico. Se pueden usar para analizar patrones de interacción como las relaciones sociales, o para buscar rutas óptimas como en una red de transporte.

La tecnología más conocida y usada para este tipo de BD es la Neo4j.

Figura 72

BD NoSQL de Grafos

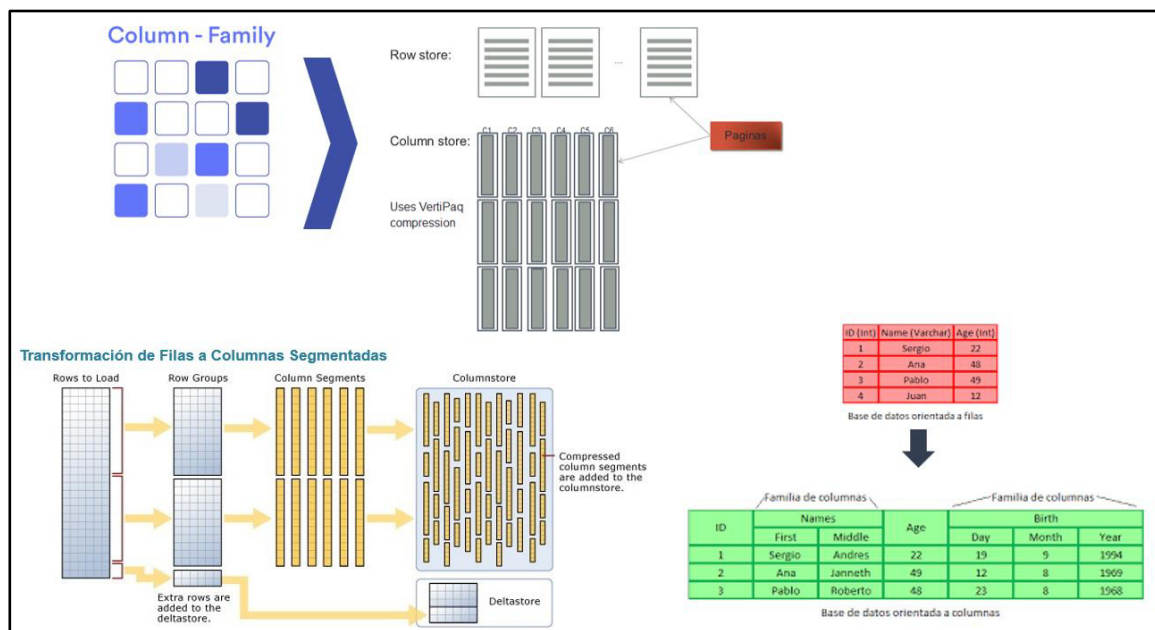


Nota. Recurso externo personalizado

- Orientadas a Columnas: Almacenan los datos en columnas en lugar de filas. Cada columna representa un atributo y se almacena por separado, esto permite acceder a columnas específicas de manera rápida. Se suelen usar para almacenar datos que tienen una gran cantidad de atributos, como los datos de los sensores o series temporales.

Figura 73

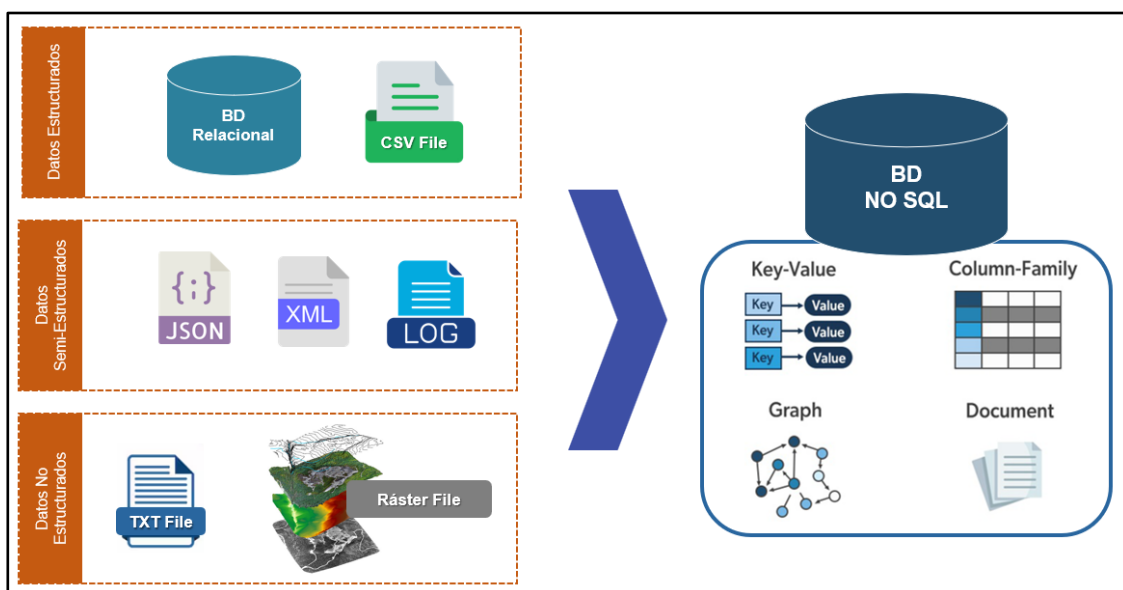
BD NoSQL Orientada a Columnas



Nota. Recurso externo personalizado

Los GIS usan este tipo de repositorios para manejar grandes volúmenes de datos como los datos en tiempo real, también las usan para representar relaciones complejas de redes geospaciales como rutas de transporte, modelar relaciones complejas entre puntos geográficos e incluso implementar algoritmos de rutas y búsqueda de caminos más cortos. No obstante, la implementación y mantenimiento de estas bases de datos requiere demanda mucha disciplina, pues la falta de una estructura relacional es fácil puede llevar a problemas de desorganización, pérdida de integridad de datos y redundancia de información, afectando la calidad la data.

Por esta razón, estas bases de datos suelen ser parte de un ecosistema más amplio, siendo producto de una integración de datos de diferentes fuentes, incluyendo bases de datos relaciones, datos estructurados y/o datos semi-estructurados. En muchas aplicaciones de Big Data, las bases de datos NoSQL no actúan como el repositorio principal, sino como un componente que facilita la explotación de la información producto de integrar diferentes orígenes de datos.

Figura 74*Estructuras de Datos a NoSQL*

Nota. Elaboración propia

Tabla 4*Algunas Tecnologías de Bases de Datos No SQL y Aplicaciones GIS*

Tecnología de BD Relacional	Requisitos	Aplicación GIS
MongoDB	Conocimientos en modelado de datos no estructurados y lenguaje de consulta MongoDB. Licenciamiento: Gratuito. Ofrece una versión licenciada que incluye características de seguridad, auditoría y soporte.	Almacenamiento flexible de datos geoespaciales no estructurados o semiestructurados (GeoJSON), análisis de datos en tiempo real, escalabilidad horizontal.
Cassandra	Conocimientos en modelado de datos distribuidos y lenguaje de consulta CQL. Licenciamiento: Gratuito.	Almacenamiento de grandes volúmenes de datos geoespaciales en tiempo real, alta disponibilidad y tolerancia a fallos, análisis de datos geoespaciales masivos.
Redis	Conocimientos en estructuras de datos en memoria y lenguaje de comandos Redis. Licenciamiento: Gratuito.	Almacenamiento en caché de datos geoespaciales para mejorar el rendimiento de consultas frecuentes, gestión de datos en tiempo real (geolocalización, seguimiento de flotas).
Neo4j	Conocimientos en modelado de grafos y lenguaje de consulta Cypher. Licenciamiento: Gratuito. Ofrece una versión licenciada que incluye clustering, alta disponibilidad y soporte empresarial.	Modelado de relaciones complejas entre entidades geoespaciales (redes de transporte, análisis de rutas), análisis de redes sociales georreferenciadas.
CouchDB	Conocimientos en modelado de datos documentales y lenguaje de consulta CouchDB. Licenciamiento: Gratuito.	Almacenamiento de datos geoespaciales semiestructurados, replicación y sincronización de datos, aplicaciones móviles y offline.

Nota. Elaboración propia

Contraste entre BD Relacional y BD No SQL:

Es importante tener en cuenta que ambas categorías no son mutuamente excluyentes. Sino son de carácter complementario, pues cada una está diseñada para atender diferentes necesidades. Por ende, lo ideal es aprovechar sus fortalezas individuales para crear soluciones de datos más completas y eficientes.

Tabla 5

Cuadro comparativo BD Relacional y BD No SQL

Característica	Base de Dato Relacional (RDMS)	Base de Datos No Relacional (No SQL)
Estructura de Datos	Estructurada, basada en tablas con filas y columnas.	Flexible, puede incluir documentos, grafos, clave-valor, columnas.
Modelo de Datos	Modelo relacional con relaciones y claves primarias/foráneas.	Varios modelos: documental, grafos, clave-valor, orientado a columnas.
Lenguaje de Consulta	SQL (Structured Query Language).	Varios lenguajes, dependiendo del tipo (por ejemplo, JSON para MongoDB).
Escalabilidad	Escalabilidad vertical (aumentar la capacidad del servidor).	Escalabilidad horizontal (añadir más servidores).
Flexibilidad del Esquema	Esquema rígido y predefinido; cambios de esquema costosos.	Esquema flexible; permite cambios dinámicos sin afectar la base de datos.
Rendimiento	Eficiente para consultas complejas y transacciones ACID.	Altamente eficiente para operaciones de lectura/escritura masivas y grandes volúmenes de datos.
Casos de Uso (más frecuentes)	Sistemas financieros, ERP, CRM, aplicaciones con datos estructurados.	Aplicaciones de big data, redes sociales, IoT, aplicaciones con datos no estructurados.
Ventajas	Consistencia y integridad de datos. Soporte para transacciones complejas. Consultas complejas con SQL.	Alta escalabilidad y flexibilidad. Manejo eficiente de grandes volúmenes de datos. Adaptabilidad a diferentes tipos de datos.
Desventajas	Menor flexibilidad para cambios de esquema. Escalabilidad limitada.	Consistencia eventual puede ser un desafío. Falta de estandarización en lenguajes de consulta y características

Nota. Elaboración propia

2.6.4.5. Lenguaje de Programación para Análisis

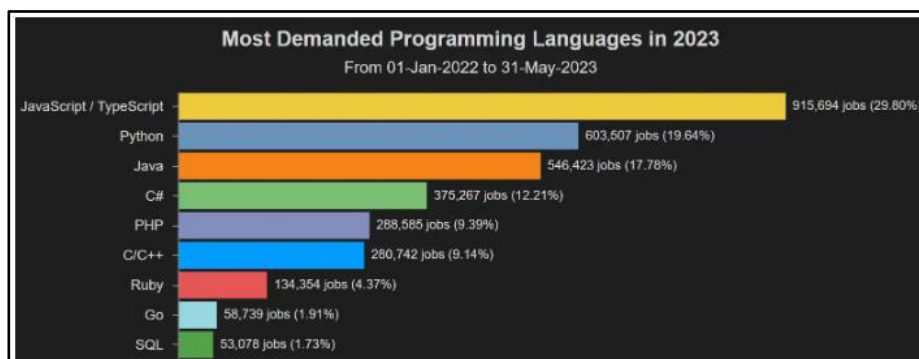
El análisis de datos se ha convertido en una herramienta esencial para la toma de decisiones en diversas áreas. Dentro de este contexto los lenguajes de programación desempeñan un papel crucial, ofreciendo bibliotecas especializadas para la manipulación de estructuras de datos complejas como los grafos y algoritmos avanzados como el de Dijkstra, usados para el análisis de redes.

Lenguaje de Programación orientados al Análisis de Datos:

En el amplio mundo de la tecnología existen diferentes lenguajes de programación según la necesidad que se desee satisfacer. Por ejemplo, en la programación web el lenguaje predominante es JavaScript, sin embargo, existen otros como TypeScript; del mismo modo para realizar análisis de datos existen lenguajes que permiten abordar estos requerimientos de manera eficientemente entre los más populares para estos fines se destacan Python, R y Julia.

Figura 75

Demanda de Lenguajes de Programación 2023



Nota. Recurso externo

Python: Es ampliamente reconocido por su simplicidad y versatilidad. facilita la manipulación de datos, el análisis estadístico y la visualización.

R: Es especialmente popular en la comunidad de estadísticos y científicos de datos por tener un enfoque fuerte en análisis estadístico y visualización de datos.

Julia: Es un lenguaje más reciente que ha ganado atención por su velocidad y eficiencia

en el manejo de cálculos numéricos. Julia combina la facilidad de uso de Python con el rendimiento de lenguajes como C++.

Tabla 6

Ventajas y Desventajas entre Python, R, Julia

Python	R	Julia
VENTAJAS		
Fácil de aprender y leer. Ecosistema de bibliotecas rico (NumPy, Pandas, Scikitlearn, etc.) Gran soporte comunitario.	Capacidades estadísticas extensas. Potente en visualización (ggplot2). Comunidad activa de investigación.	Alto rendimiento gracias a la compilación Just-in-Time (JIT). Buena interoperabilidad con otros lenguajes. Sintaxis limpia y expresiva.
DESVENTAJAS		
Limitaciones de rendimiento debido a su naturaleza interpretada. Global Interpreter Lock (GIL) limita el multithreading. Curva de aprendizaje más pronunciada para técnicas avanzadas.	Curva de aprendizaje más pronunciada para principiantes sin experiencia en programación. Limitaciones en la gestión de memoria con grandes conjuntos de datos. Capacidades limitadas fuera del análisis estadístico	Ecosistema de paquetes aún en crecimiento. Recursos de aprendizaje y documentación limitados en comparación con Python y R.
USO		
Ideal para proyectos de ciencia de datos que requieren una amplia gama de capacidades de manipulación de datos, análisis y aprendizaje automático. Adecuado para principiantes y colaboraciones interdisciplinarias. Comúnmente utilizado en análisis exploratorio de datos, web scraping, procesamiento de lenguaje natural y construcción de modelos de machine learning.	Preferido cuando el análisis estadístico y la visualización son primordiales. Útil en campos como epidemiología, ciencias sociales, econometría y dominios de investigación donde el rigor estadístico es crucial.	Excelente para proyectos donde el rendimiento es crítico y se requiere un procesamiento numérico intenso, simulaciones o investigación algorítmica. Beneficioso para la computación científica, problemas de optimización y proyectos que demandan un equilibrio entre expresividad de alto nivel y eficiencia computacional.

Nota. Elaboración propia

Tabla 7*Análisis de Datos con Python, R, Julia*

Aspecto	Análisis del aspecto para cada Lenguaje de Programación usado en el Análisis de Datos		
	Python	R	Julia
Facilidad de Uso	Alta. Sintaxis simple y fácil de aprender.	Moderada. Sintaxis más técnica y enfocada en estadísticas.	Moderada. Sintaxis limpia, similar a las matemáticas.
Ecosistema para Grafos	Excelente. Librerías como NetworkX, igraph, y graph-tool.	Bueno. igraph y otras librerías especializadas en estadística.	En crecimiento. LightGraphs.jl es una de las principales librerías.
Rendimiento	Adecuado, pero puede ser limitado para tareas muy intensivas en computación debido al Global Interpreter Lock (GIL).	Moderado. Buenas capacidades estadísticas, pero puede enfrentar limitaciones de memoria.	Alto. JIT compilation ofrece rendimiento cercano a lenguajes compilados como C
Documentación y Soporte Comunitario	Extensa y activa. Gran cantidad de recursos, tutoriales y bibliotecas de código abierto.	Amplia en el campo estadístico, con una comunidad investigadora sólida.	Menos extensa, pero en crecimiento. Comunidad activa pero más pequeña.
Aplicaciones Comunes	Análisis de datos, machine learning, aplicaciones web, automatización de tareas.	Análisis estadístico, visualización de datos, investigación académica.	Computación científica, simulaciones numéricas, algoritmos complejos.
Recomendación para grafos y Dijkstra	Altamente recomendado. NetworkX es fácil de usar y eficiente para implementaciones de grafos y algoritmos de rutas.	Menos recomendado. Aunque igraph es útil, R no es tan eficiente en rendimiento para tareas complejas de grafos.	Recomendado si se prioriza el rendimiento. LightGraphs.jl es potente, pero puede requerir más conocimientos para su uso efectivo.

Nota. Elaboración propia

2.6.4.6. Arquitectura Tecnológica del Proyecto

Para el cumplimiento con el proyecto, la elección de los componentes tecnológicos ha sido crucial, pues se debe considerar diferentes aspectos que permitan abordar de manera eficiente el requerimiento, así como considerar los estándares tecnológicos de la entidad y las destrezas técnicas del equipo que dará mantenimiento a la implementación, pero sin dejar de considerar la versatilidad de la tecnología para poder ser migrada de infraestructura, sea on-premise o en la nube.

Es en este contexto que se ha optado por usar PostgreSQL con la extensión PostGIS como base de datos y Python como lenguaje de programación para los procesos de análisis. Y para algunos casos puntuales Q-GIS Desktop y otras herramientas de escritorio como complementos de OSM para el tratamiento y registro de datos.

PostgreSQL con PostGIS: Repositorio Núcleo Geoespacial

PostgreSQL, es un sistema de gestión de bases de datos objeto-relacional de código abierto, destaca por su robustez, escalabilidad y capacidad para manejar grandes volúmenes de datos. Su extensión PostGIS añade un componente geoespacial crucial, permitiendo almacenar, consultar y analizar datos geográficos de manera eficiente. Esta combinación proporciona una base sólida para modelar redes de transporte complejas, incluyendo nodos, rutas, horarios y atributos adicionales.

Para la elección de este motor de base de datos, se consideran los siguientes puntos:

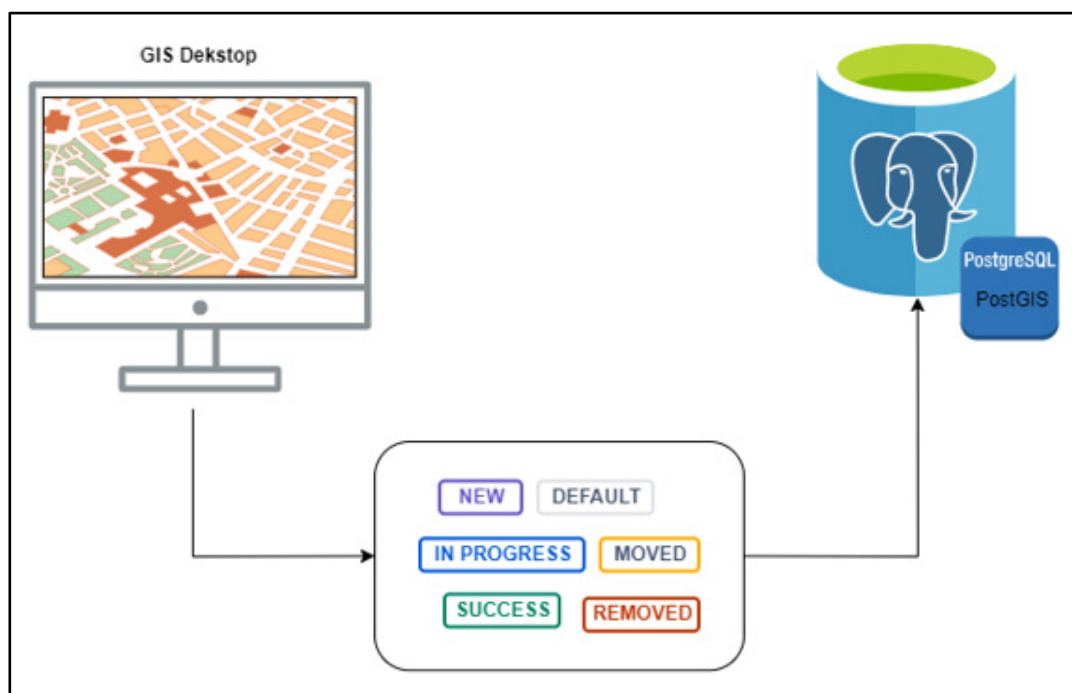
- Licencia y Costos: Al ser de código abierto, elimina los costos de licencias, lo que resulta en un ahorro significativo para el proyecto.
- Escalabilidad: Su arquitectura permite escalar vertical y horizontalmente, adaptándose al crecimiento de los datos y las demandas del sistema.
- Rendimiento: Ofrece un rendimiento excepcional en consultas espaciales complejas, cruciales para el análisis de redes de transporte.

- Soporte y Comunidad: Cuenta con una vasta comunidad de usuarios y desarrolladores, lo que garantiza un soporte técnico sólido y una evolución constante.

Adicionalmente PostGIS presenta gran compatibilidad con las plataformas GIS como Q-GIS, GeoServer y otros programas GIS de licenciamiento libre y/o de pago.

Figura 76

GIS Desktop y BD PostgreSQL



Nota. Elaboración propia

Python: Motor Analítico de Datos

Python, un lenguaje de programación versátil y ampliamente utilizado, es uno de los lenguajes con mejor compatibilidad con PostgreSQL/PostGIS facilitando el desarrollo de scripts para el análisis de datos. Su sintaxis clara y concisa, junto con una amplia gama de bibliotecas especializadas (como NetworkX, Shapely, GeoPandas), facilitan el desarrollo de algoritmos de análisis de redes, optimización de rutas y visualización de resultados.

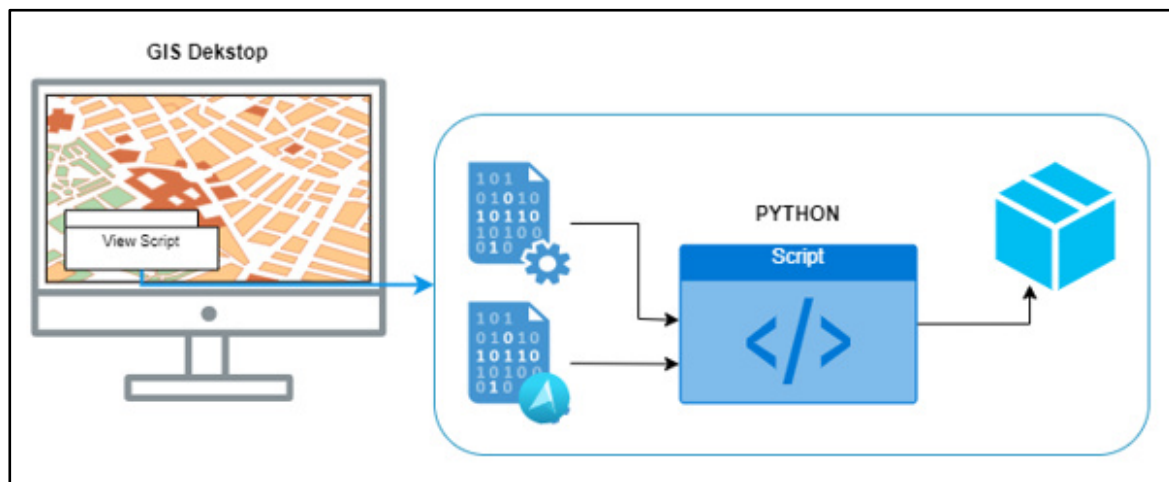
La elección de Python se justifica por:

- Flexibilidad: Permite integrar fácilmente diferentes componentes del sistema, desde la base de datos hasta la interfaz de usuario.
- Productividad: Reduce los tiempos de desarrollo gracias a su enfoque en la legibilidad y la facilidad de uso.
- Comunidad y Documentación: Cuenta con una comunidad activa y una documentación extensa, lo que facilita el aprendizaje y la resolución de problemas.

Python presenta gran compatibilidad con las plataformas GIS como Q-GIS, GeoServer y otros programas GIS de licenciamiento libre y/o de pago como ArcGIS pues estos permiten una integración nativa, propia de fábrica o a través de bibliotecas construidas por los fabricantes para permitir la integración entre sus plataformas y Python.

Figura 77

GIS Desktop y Python



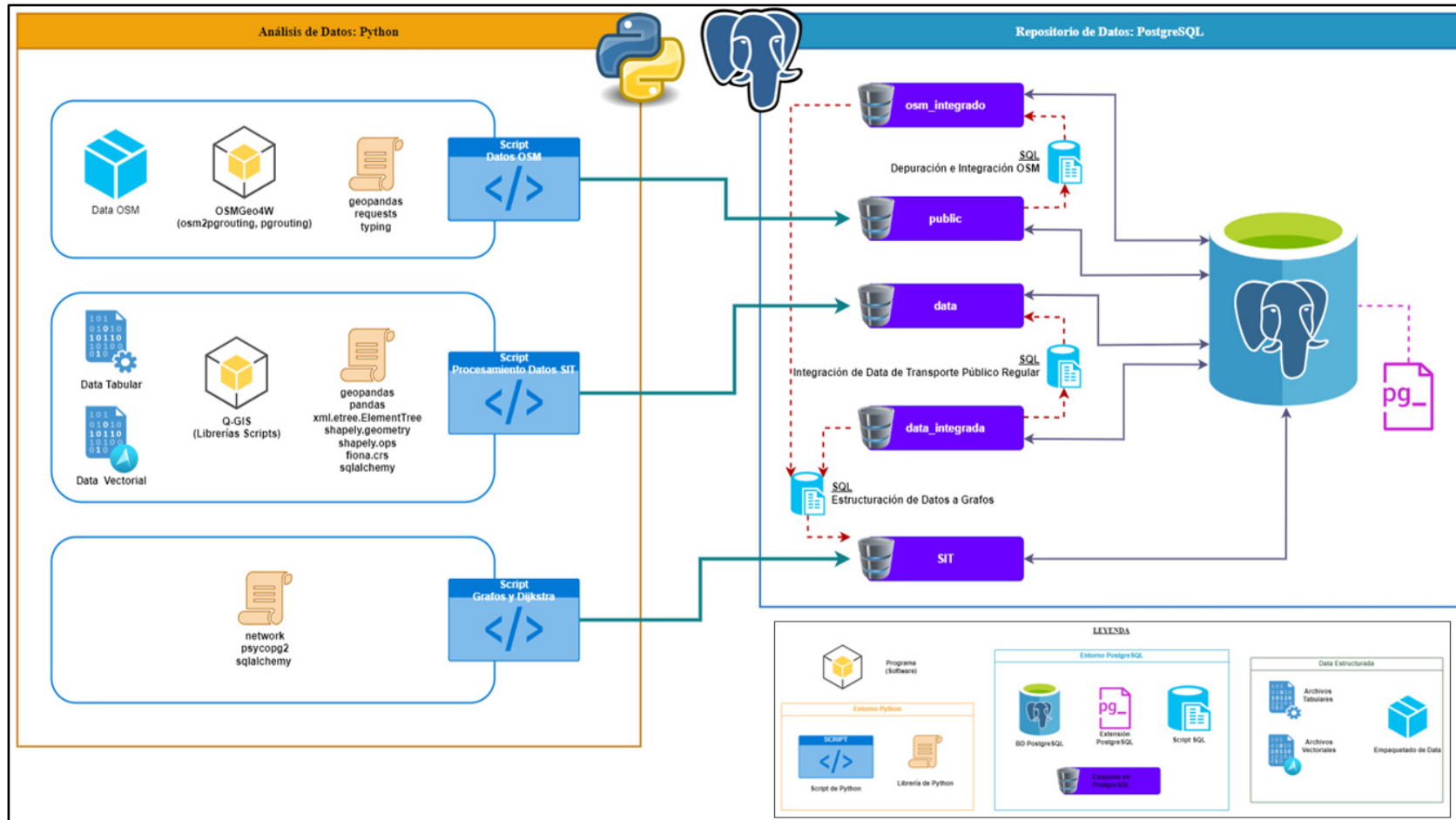
Nota. Elaboración propia

Versatilidad en Infraestructura

La capacidad de PostgreSQL y Python para ser implementados tanto en infraestructuras on-premise como en la nube proporciona una flexibilidad crucial para el proyecto. Esto permite a la organización adaptarse a cambios en la infraestructura tecnológica sin tener que realizar migraciones costosas o disruptivas.

Figura 78

Diagrama de Arquitectura Tecnológica para Implementación de Herramienta de Búsqueda de Rutas



Nota. Elaboración propia

2.7. Requerimientos para Implementar Proyecto

2.7.1. Hardware

Los detalles precisados en esta sección se precisan considerando que el análisis para la generación de rutas se realizará por un usuario a la vez. En caso de requerir que diferentes usuarios ejecuten el procesamiento en simultáneo, en concurrencia, se recomienda realizar un dimensionamiento que contemple la cantidad de usuarios concurrentes, y si la ejecución del proceso de análisis se realizará de manera externa o interna a la red de la entidad, considerando contar con servidores dedicados.

Tabla 8

Detalle de Hardware

Componente	Detalle	Equipo
Disco Duro Sólido	Espacio para Sistema: 250 GB Espacio para Datos: 250 GB	
Procesador RAM	16 GB de RAM	
Procesador	Procesador de 4 núcleos	Case
Tarjeta Gráfica	Compatible con DirectX12	
Complementos	Componentes Complementarios	
Accesorios	Pantalla, Teclado, Mouse	Complementos

Nota. Elaboración propia

2.7.2. Software

Los softwares usados para la generación del proceso análisis, son de plataforma libre, lo cual ha permitido no incurrir en costos de licencias. Acotar, que no se precisa el sistema operativo pues los softwares en mención pueden ejecutarse desde cualquier sistema operativo, sea libre o licenciado.

Tabla 9

Detalle de Software

Componente	Licenciamiento	Uso
Q-GIS 3.x	Libre	Tratamiento de Datos GIS.
OSM Convert	Libre	Herramienta del ecosistema de OSM, para el manejo de archivos PBF.
OSM 2 pgSQL	Libre	Herramienta del ecosistema de OSM, para la carga de archivos OSM a PostgreSQL.
PostgreSQL 13.x	Libre	Repositorio de Datos (entidad-relación), usado para almacenar y automatizar el flujo de información.
Python 3.x	Libre	Lengua de Programación usado para el procesamiento y análisis de dato.
Visual Code 1.x	Libre	Entorno de Desarrollo Integrado para la generación de los Scripts en Python.
DBeaver 24.x	Libre	Entorno de Desarrollo Integrado para la generación de los Scripts en PostgreSQL.

Nota. Elaboración propia

Considerar que los componentes tecnológicos precisados no son de carácter mandatorio para la implementación de los algoritmos, por ende, la tecnología a utilizar queda a elección del lector; sin embargo, se han elegido por su versatilidad para la manipulación y análisis de datos tabulares y georreferenciados, así como el libre licenciamiento de los mismos.

2.7.3. Datos

La fuente de datos del Sistema Integrado de Transporte ha sido provista por las direcciones encargadas de la entidad, y los datos complementarios de las vías de acceso han sido descargados de la Plataforma Libre de Mapas Open Street Map. Todo ello ha permitido que no existan costos para la obtención de la información.

Tabla 10

Fuente de Datos

Dato	Tipo	Fuente	Detalle
Información del SIT Metropolitano	Vectorial Tabular	ATU	Datos de servicios, rutas, horarios y trazos del Sistema Metropolitano: Troncal y Alimentadores.
Información del SIT Corredores Complementarios	Vectorial Tabular	ATU	Datos de servicios, rutas, horarios y trazos del Sistema de Corredores Complementarios: Morado, Rojo, Corredor Azul.
Información del SIT Línea de Metro de Lima y Callao	Vectorial Tabular	ATU	Datos de servicios, rutas, horarios y trazos del Sistema del Líneas de Metro de Lima y Callao Activas: Línea 01 y Sección Línea 02.
Vías de Lima y Callao	Vectorial	Open Street Map	Datos de vías y calles del área urbana de Lima y Callao.

Nota. Elaboración propia.

Se debe precisar que Open Street Map es una plataforma que permite obtener información georreferenciada de manera libre, sin embargo, existe cierto desfase de información respecto a la realidad, la cual es mínima para las zonas urbanas.

2.7.4. Desarrollo del Proyecto

El desarrollo del proyecto contempla la explicación del proceso de Automatización del Análisis de Rutas mediante el algoritmo de Dijkstra fundamentado en la Teoría de Grafos, el cual inicia con el análisis del requerimiento seguido del diseño y planteamiento del Modelo de Datos según el Modelo de Negocio: “Sistema de Transporte Público de Carácter Regular”.

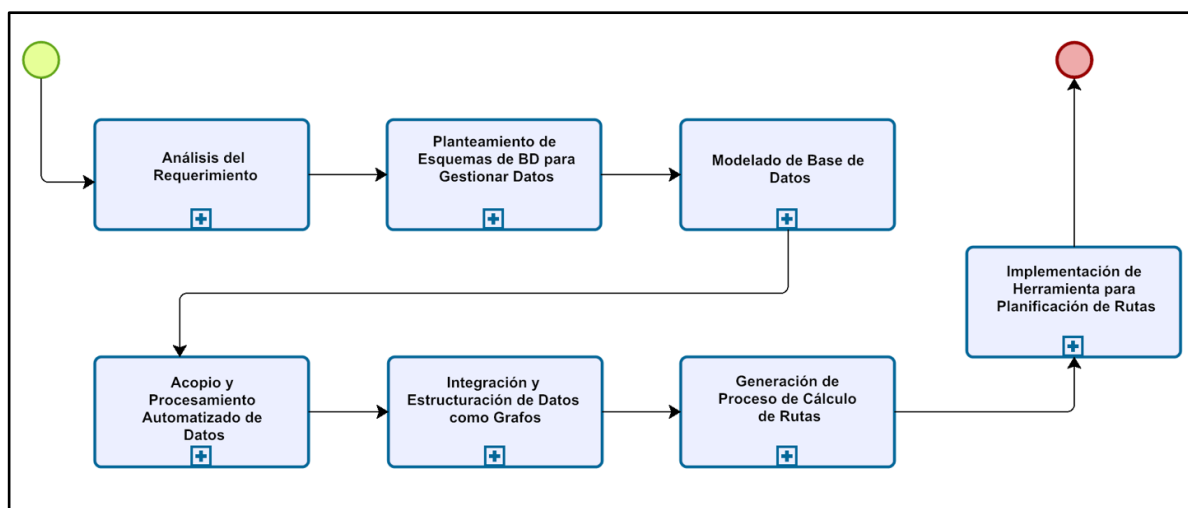
La secuencia de los procesos a implementar se detalla a continuación:

- Análisis del Requerimiento
- Planteamiento de Esquemas en la Base de Datos para la Gestión de los Datos
- Modelado de Base de Datos
- Acopio y Procesamiento Automatizado de Datos
- Integración y Estructuración de Datos como Objetos del Grafo
- Generación de Proceso Análisis para la Planificación de Rutas
- Implementación de Herramienta de Planificación de Rutas

Considerar que los procesos tienen actividades y tareas con mayor detalle.

Figura 79

Flujo para el Desarrollo del Proyecto



Nota. Elaboración propia

2.7.5. *Análisis del Requerimiento*

En esta fase inicial, se hace una evaluación exhaustiva para captar las necesidades, metas y limitaciones del proyecto, garantizando que la solución final sea efectiva y adecuada.

Requerimiento: Contar con un planificador de rutas que permita optimizar el viaje en los servicios del transporte público de carácter regular entre dos puntos.

2.7.5.1. *Análisis de Necesidades y Limitaciones*

Necesidades: Son los requisitos esenciales que el proyecto debe cumplir para ser exitoso. Considerando las funciones y características que permiten encaminar el proyecto a la meta planteada, considerando las expectativas.

Limitaciones: Son las restricciones que podrían afectar el desarrollo del proyecto. Estas pueden incluir aspectos técnicos, presupuestarios, temporales o regulatorios que limitan cómo se pueden cumplir las necesidades.

Tabla 11

Cuadro de Análisis de Necesidades y Limitaciones del Proyecto

Necesidades	Limitaciones
<p>Compilar los datos referentes a las rutas de los servicios de transporte público de carácter regular en un modelo estandarizado que permita gestionarlos.</p>	<p>Usar bases de datos que permitan mantener la integridad y calidad de los mismos.</p>
<p>Integrar los datos del sistema de transporte bajo una estructura que permita dar un óptimo desempeño al proceso de análisis de rutas.</p>	<p>Usar estructuras de programación que permitan ser reutilizable y escalables para otros procesos y proyectos, considerando que su licenciamiento sea gratuito.</p>
<p>Identificar los algoritmos necesarios para implementar el planificador de ruta.</p>	<p>La tecnología a usar para el proyecto debe ser compatible con otras tecnologías asegurando su integración.</p>
<p>El generador de rutas debe evaluar criterios como proximidad a paraderos, para mostrar resultados consistentes respecto a la ubicación de inicio y destino.</p>	

Nota. Elaboración propia

2.7.5.2. Propuesta de Solución

Base de Datos PostgreSQL: Es una base de datos relacional de licenciamiento libre, cuyas características permiten mantener la integridad de los datos para asegurar la calidad de los mismos mediante las reglas de normalización. Adicionalmente cuenta con extensiones gratuitas que permiten la manipulación de información vectorial.

Lenguaje de Programación Python: Es un lenguaje de programación predominante para el procesamiento y análisis de datos tabulares y geoespaciales mediante el uso de sus bibliotecas. Adicionalmente, su sintaxis clara y sencilla permite crear estructuras flexibles de código para poder integrarse con otras tecnologías mediante la implementación de API's.

Puntos adicionales a considerar:

- PostgreSQL se destaca como una base de datos relacional única al ofrecer capacidades avanzadas de análisis geoespacial y manejo de grafos a través de extensiones maduras como pgRouting. Su rendimiento, escalabilidad y adaptabilidad a diversos modelos de datos, incluyendo entidad-relación y grafos, la convierten en una alternativa competitiva a soluciones comerciales como Oracle.
- Python es un lenguaje de programación versátil que ha ganado una gran popularidad en los últimos años, especialmente en el campo del análisis de datos gracias a su amplio ecosistema de bibliotecas especializadas. Sin embargo, su utilidad se extiende más allá del análisis de datos, ya que sus frameworks permiten crear aplicaciones robustas tanto para la integración tecnológica como para el desarrollo de interfaces de usuario.
- La combinación PostgreSQL y Python ofrece una solución potente y flexible para el desarrollo de aplicaciones que requieran un almacenamiento de datos robusto y eficiente, así como capacidades de análisis y procesamiento avanzadas. Su integración es fluida y bien soportada, facilitando la creación de aplicaciones escalables y de alto rendimiento.

Tabla 12*Cuadro de Solución a las Necesidades con Tecnología Planteada*

Necesidades	Tecnología	Justificación
Compilar los datos referentes a las rutas de los servicios de transporte público de carácter regular en un modelo estandarizado que permita gestionarlos.	PostgreSQL	Al ser una base de datos relacional permite mantener la integridad de los datos tanto tabulares como geométricos, garantizando la calidad de los mismos mediante los procesos de normalización. Para lo cual es necesario contar con un modelo entidad-relación consistente y flexible para el modelo de negocio referente al transporte público.
Integrar los datos del sistema de transporte bajo una estructura que permita dar un óptimo desempeño al proceso de análisis de rutas.	PostgreSQL & Python	Mediante el uso de los esquemas en PostgreSQL se puede separar el modelo entidad relación con un esquema orientado a la unificación de la data para ser consumida como una estructura de grafos ya sea desde el mismo módulo de PgRouting propio de la Base de Datos o desde Python previa integración de sus bibliotecas de código abierto.
Identificar los algoritmos necesarios para implementar el planificador de ruta.	Python (Dijkstra)	Python permite implementar Dijkstra mediante sus librerías de código abierto, así mismo permite conectarse a la Base de Datos PostgreSQL, gracias a estas librerías.
El generador de rutas debe evaluar criterios como proximidad a paraderos, para mostrar resultados consistentes respecto a la ubicación de inicio y destino.	Python	El algoritmo de análisis deberá incluir procesos de validación de proximidad para las ubicaciones de salida y llegada respecto a los paraderos para asegurar que haya consistencia en los análisis y resultados mostrados. Proponiéndose un rango de 0 a 1000 metros de distancia como máximo.

Nota. Elaboración propia

Tabla 13

Cuadro de Solución a las Limitaciones con Tecnología Planteada

Limitaciones	Tecnología	Justificación
Usar bases de datos que permitan mantener la integridad y calidad de los mismos.	PostgreSQL	La naturaleza de las bases de datos relacionales prima en buscar mantener la integridad de los datos mediante los niveles de normalización normalización, ello asegura la calidad de la data contenida en este tipo de repositorios. Siendo PostgreSQL una base de datos relacional.
Usar estructuras de programación que permitan ser reutilizable y escalables para otros procesos y proyectos, considerando que su licenciamiento sea gratuito.	Python	Muchos lenguajes de programación se orientan a crear estructuras de código reutilizables mediante el enfoque de la Programación Orientada a Objetos (POO). Este enfoque permite encapsular el comportamiento de los objetos mediante atributos y métodos (funciones), asegurando la herencia para reutilizar estos comportamientos en nuevas clases (subclases) y brindando flexibilidad mediante el polimorfismo el cual consiste en llamar métodos de diferentes maneras en nuevos objetos. Python permite implementar este enfoque de programación, adicionalmente es el lenguaje más usado para el procesamiento y análisis de datos permitiéndose crear estructuras de código reutilizables.
La tecnología a usar para el proyecto debe ser compatible con otras tecnologías asegurando su integración.	PostgreSQL & Python	PostgreSQL permite la integración con otras bases de datos de la misma tecnología mediante extensiones propias (postgres_fdw), y con otras bases de datos mediante el linqueo de bases de datos (dblink) o sus propias extensiones (fdw). Así mismo, entre los procesos más conocidos se conoce el uso de herramientas externas como las Herramientas de Extracción Transformación y Carga (ETL: Extract, Transform, Load) para la integración de datos y tecnologías. Python permite integrarse de diferentes maneras a otras tecnologías, siendo las más eficientes el uso de conectores (ORM o ODM según sea el caso BD relacional y no relacional respectivamente) para el caso de bases de datos y la implementación de API para la integración con otros lenguajes de programación.

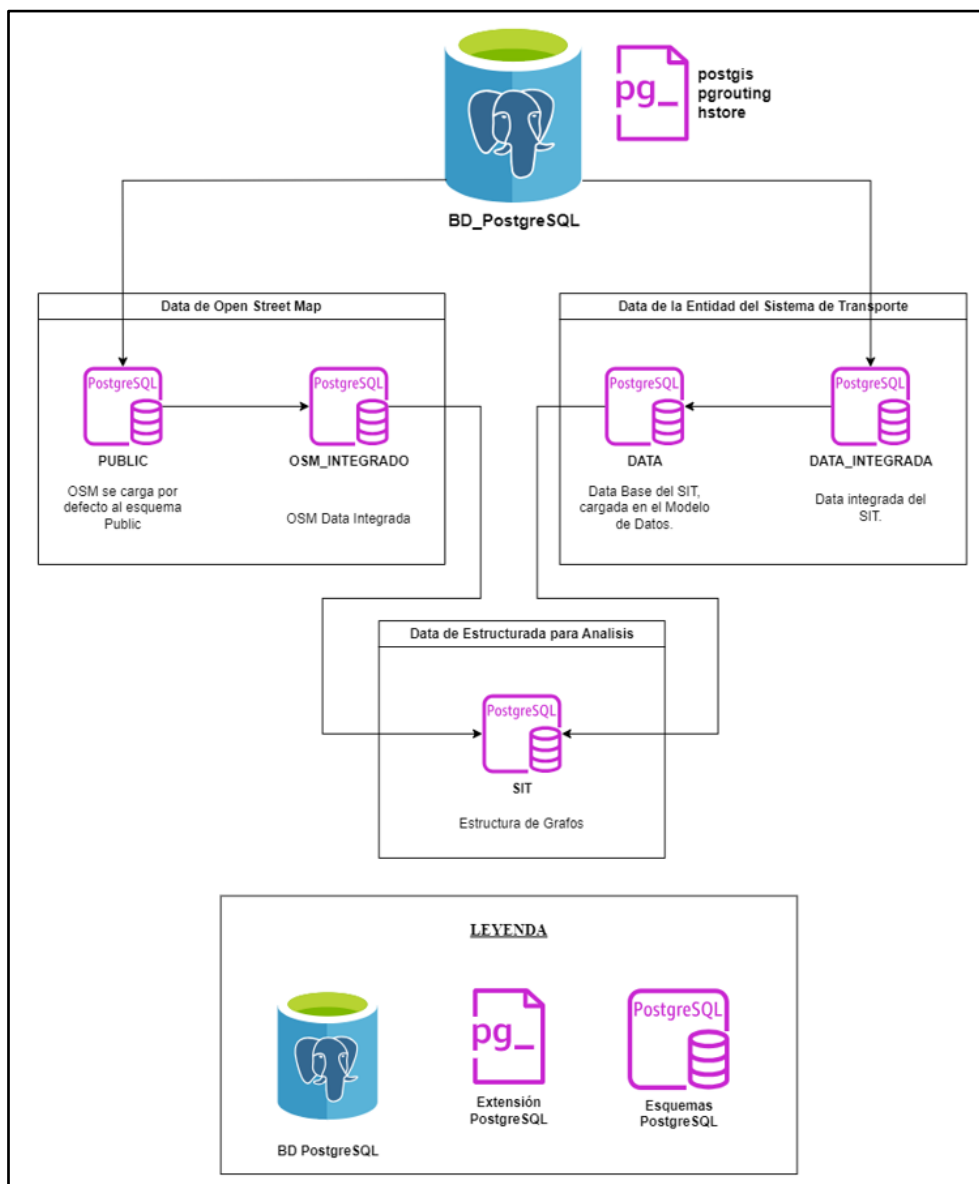
Nota. Elaboración propia

2.7.6. Planteamiento de Esquemas en la Base de Datos para la Gestión de Datos

Antes de iniciar la carga de datos, estructuración y procesamiento de datos, es necesario organizar el repositorio que los contendrá. Para esto la base de datos relacional, manejará diferentes esquemas con el fin de mantener la calidad e integridad de los datos, así como asegurar un proceso de análisis de rutas ordenado y eficiente.

Figura 80

Distribución de Base de Datos



Nota. Elaboración propia

Tabla 14*Distribución de Esquemas de Base de Datos*

Esquema	Uso
PUBLIC	Este esquema contendrá las extensiones habilitadas de la base de datos PostgreSQL, pues al ejecutarlas para su instalación se suelen instalar por defecto en este esquema. Adicionalmente almacenará la data descargada de OSM, pues al migrar los datos desde un repositorio externo hacia la base de datos, sucede lo mismo que con las extensiones se migran al esquema por defecto, para este caso PUBLIC.
OSM_INTEGRADO	Este esquema tiene como fin, albergar la información depurada de OSM la cual será usada para generar las estructuras de datos finales a utilizar en los procesos de análisis en el cálculo de rutas.
DATA	El esquema DATA, está destinado para contener la información procesada e integrada de los archivos vectoriales y alfanuméricos referentes al sistema integrado de transporte público de carácter regular, bajo una estructura entidad-relación.
DATA_INTEGRADA	El esquema DATA_INTEGRADA, contiene las estructuras integradas de los datos provenientes de las tablas pertenecientes al esquema DATA, para facilitar los procesos de integración con la data del esquema OSM_INTGRADO.
SIT	Este esquema integra los datos de los esquemas DATA_INTEGRADA y OSM_INTEGRADO con el fin de crear la estructura de grafos para luego ser usados en los procesos de análisis para el cálculo de rutas mediante el Algoritmo de Dijkstra.

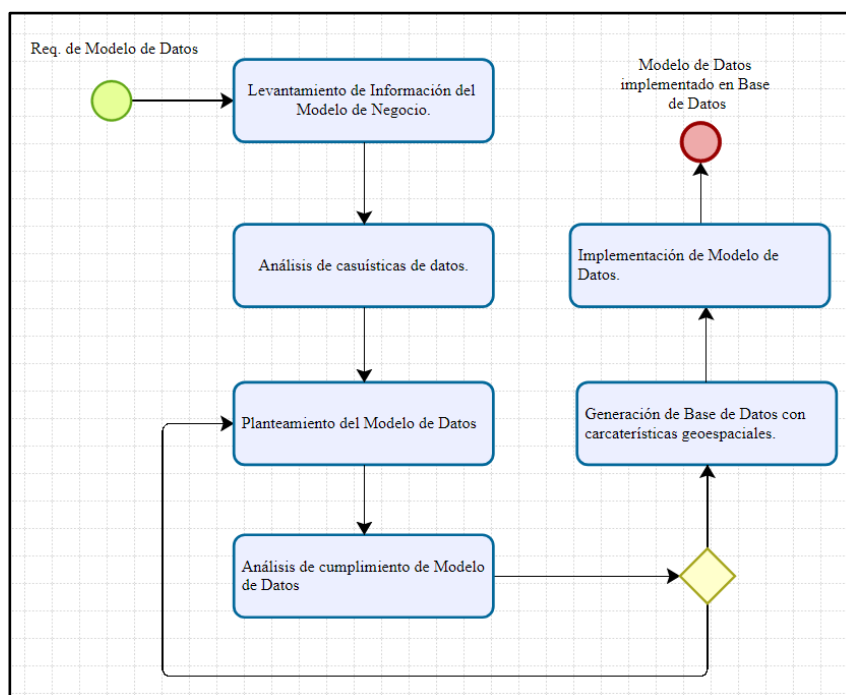
Nota. Elaboración propia

2.7.7. Modelado de Datos

El flujo para determinar el modelo de datos suele iniciar con el entendimiento del modelo de negocio. Para los modelos de datos que interactúan con el ámbito geográfico es necesario contemplar entidades de carácter espacial que permitan almacenar las coordenadas de dichas entidades.

Figura 81

Flujo para la generación del Modelo y Base de Datos



Nota. Elaboración propia

El entendimiento del modelo de negocio permite contemplar las casuísticas de los datos, pues el modelo de datos debe almacenar la información de manera eficiente evitando redundancias y manteniendo la calidad e integridad de estos, con el fin asegurar consultas y análisis eficientes sobre los mismos. El modelo de negocio a modelar, requiere entender la jerarquización del Sistema de Transporte Regular, para ello se define brevemente cada elemento que interactúa en el modelo de negocio:

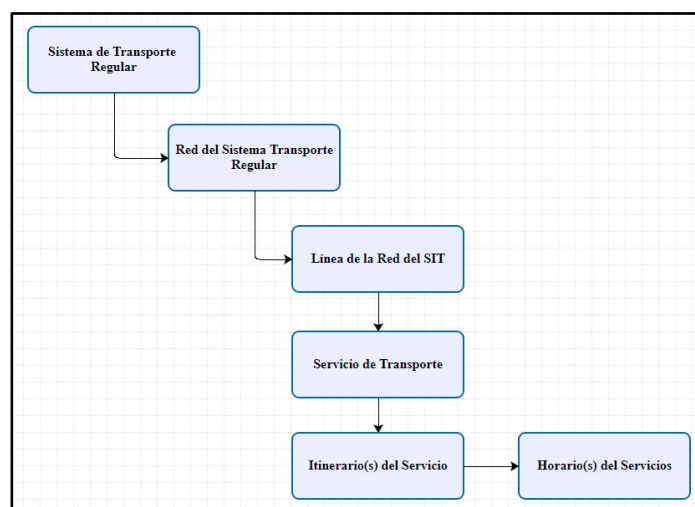
- El Sistema de Transporte es el conjunto de infraestructuras, tecnologías y

equipamiento incluido vehículo que permiten brindar un servicio de movilidad de personas u objetos.

- La Red de Transporte hace referencia al sistema interconectado de diversas rutas y servicios de transporte.
- La Línea de Transporte hace referencia al eje por el cual transitan las rutas, sin embargo, estas están asociados a servicios y horarios.
- El Servicio de Transporte hace referencia a las rutas que transitan por una línea de transporte, las cuales pueden variar según el itinerario y horarios de las mismas.
- La Ruta de Transporte hace referencia al recorrido, sea de ida y/o retorno el cual está asociado a un conjunto de lugares de subida y/o bajada de pasajeros.
- El Horario hace referencia a la programación de días y rango horario en el cual transita una ruta.
- La Infraestructura Vial hace referencia a las paradas de las rutas, la cual puede ser una terminal, una estación o un paradero.

Figura 82

Jerarquía del Sistema de Transporte Regular



Nota. Elaboración propia

Tabla 15*Composición del Sistema Integrado de Transporte*

Tipos de Sistemas	Sistemas	Categoría de Transporte	Detalle
	Sistema Metropolitano	BRT: Bus Rapid Transit Buses de Transporte Rápido	Los sistemas BRT se caracterizan por ofrecer un servicio de alta capacidad y frecuencia, con buses articulados o biarticulados que circulan por carriles exclusivos o segregados.
Transporte Masivo (Regular)	Sistema de Corredores Complementarios	BRT: Bus Rapid Transit Buses de Transporte Rápido	Comparten características similares a los sistemas BRT, como carriles exclusivos y alta frecuencia. Sin embargo, si su capacidad y nivel de servicio son inferiores, podrían clasificarse como un tipo de BRT más básico.
	Sistema de Líneas de Metro	Metro	Los sistemas de metro son sistemas de transporte ferroviario subterráneo o elevado, con alta capacidad y frecuencia.
Transporte Regular No Convencional (Por Cable)	Sistema de Funiculares	Ferrocarril	Aunque los funiculares utilizan cables para desplazarse, su funcionamiento es similar al de un tren, por lo que pueden clasificarse como un tipo de ferrocarril.
	Sistema de Teleféricos	Transporte por Cable	Los teleféricos son sistemas de transporte que utilizan cables para transportar personas o mercancías en cabinas suspendidas. Forman una categoría propia dentro del transporte por cable.
Transporte Especial	Sistema Regular Especial	Transporte Especial	Esta categoría engloba todos los servicios de transporte que atienden a necesidades específicas de determinados grupos de usuarios (trabajadores, escolares, turistas, taxi.)
Transporte Regular Convencional	Sistema Regular Convencional	Autobuses Urbanos y Rurales	Esta categoría engloba todos los servicios de transporte público que utilizan autobuses, microbuses u ómnibus como medio de transporte.

Nota. Elaboración propia

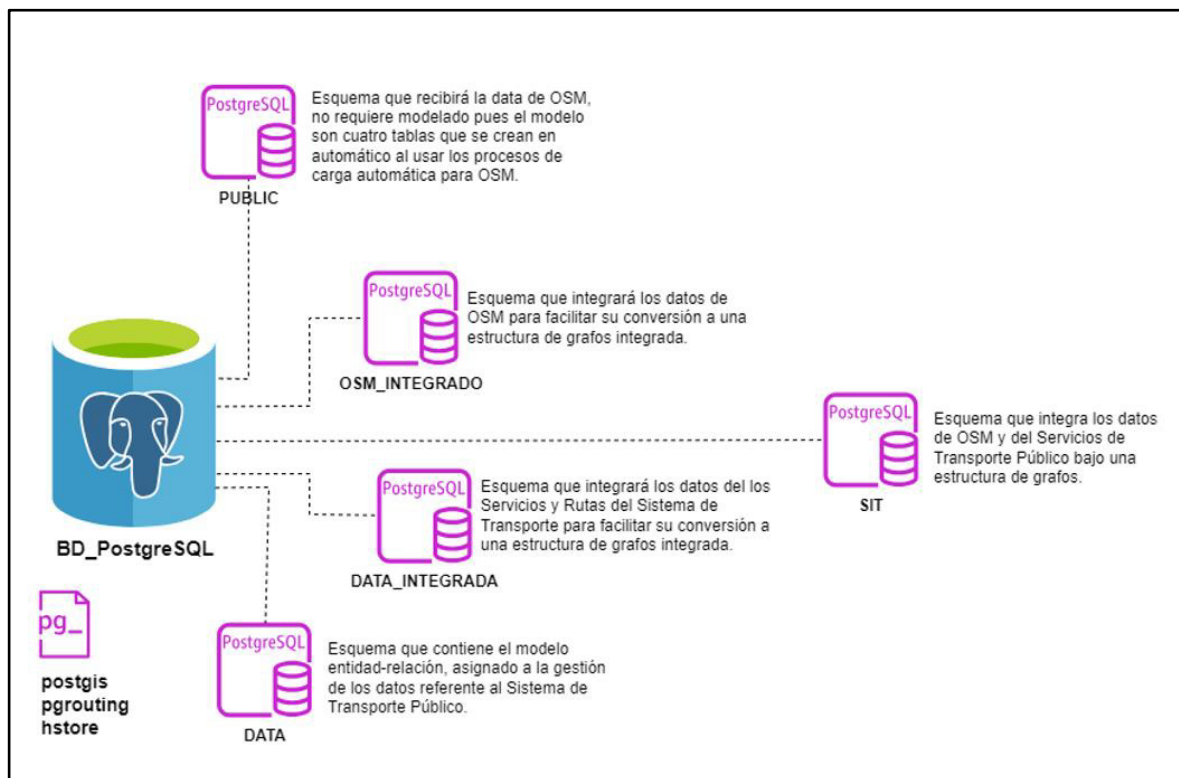
2.7.7.1. Modelo de Base de Datos

El Modelo de Base de Datos está diseñado para gestionar la información de los servicios de transporte públicos brindados por el SIT de Carácter Regular. Su normalización permite integrar los datos de los otros tipos de Sistemas de Transporte Público perteneciente al SIT, manteniendo la calidad e integridad de los datos. Además, su estructura flexible facilita la adaptación a futuros requerimientos, permitiendo la integración de procesos referente al modelo de negocio del SIT, tales como la gestión de proyectos de inversión, la gestión de flotas.

Es importante, considerar que la clasificación de los sistemas de transporte es un tema complejo y en constante evolución, ya que depende de diversos factores como la tecnología utilizada, la capacidad, la infraestructura y el tipo de servicio ofrecido.

Figura 83

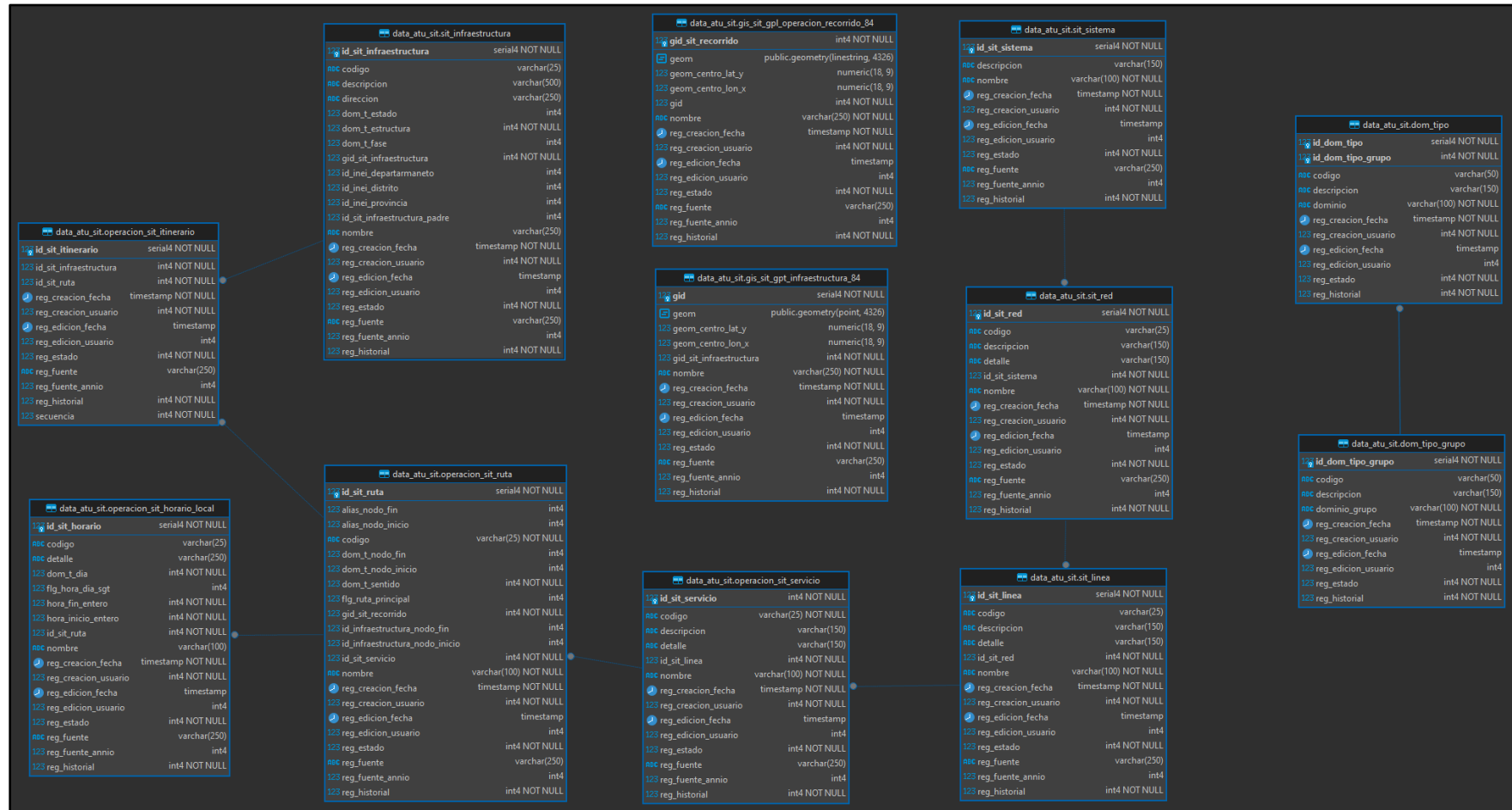
Diagrama de Esquemas y Modelo de Base de Datos



Nota. Elaboración propia

Figura 84

Modelo de Datos del Sistema de Transporte Regular



Nota. Elaboración propia

Tabla 16*Descripción de Objetos de la Base de Datos*

Objeto de Base de Datos	Tipo	Descripción
SIT_SISTEMA	Tabla	Contiene los datos que caracterizan los Tipos de Sistema de Transporte Urbano: Metropolitano, Corredores, Metro.
SIT_RED	Tabla	Contiene los datos que describen y caracterizan las Redes de Transporte asociadas a cada Sistema de Transporte Regular.
SIT_LINEA	Tabla	Contiene los datos que describen las diferentes Líneas de cada Sistema de Transporte Regular.
SIT_SERVICIO	Tabla	Contiene los datos que detallan las características de los Servicios que brinda cada Línea de Transporte.
SIT_RUTA	Tabla	Contiene los datos que detallan las propiedades de las Rutas, tales como sus sentidos.
SIT_HORARIO_LOCAL	Tabla	Contiene la información detallada de los horarios en los cuales operan las Rutas de Transporte Urbano Regular.

Objeto de Base de Datos	Tipo	Descripción
SIT_INFRAESTRUCTURA	Tabla Recursiva	Contiene información de la infraestructura vial usada como puntos de parada de las rutas, las cuales están clasificadas como terminales, estaciones y paraderos.
OPERACION_SIT_ITINERARIO	Tabla	Registra la relación entre las rutas y las infraestructuras (paradas), detallando la secuencia de las paradas respecto a cada ruta.
GIS_SIT_GPT_INFRAESTRUCTURA_84	Tabla Geométrica de Puntos	Contiene la información georreferenciada de las infraestructuras, y está representada por entidades de puntos.
GIS_SIT_GPL_OPERACION_RECORRIDO_84	Tabla Geométrica de Polilíneas	Contiene la información georreferenciada de las rutas, y cuenta con el atributo geoespacial que permite almacenar entidades de tipo polilíneas.
DOM_TIPO_GRUPO	Tabla Maestra de Grupos de Dominios	Clasifica y detalla el uso de los grupos, constituidos por los valores usados en los campos de los diferentes objetos de la base de datos.
DOM_TIPO	Tabla Maestra de Dominios	Agrupa y codifica los valores usados consecutivamente, por los atributos que constituyen los diferentes objetos. Cuyo fin es mantener la integridad de la información. Por ejemplo, los estados de las rutas e infraestructuras que pueden ser activos e inactivos.

Nota. Elaboración propia

2.7.8. Acopio y Procesamiento Automatizado de Datos

La información ha sido recopilada y procesada de dos fuentes de datos, ATU y Open Street Map. Cada grupo de datos presente la necesidad de un proceso de procesamiento particular considerando los formatos de archivos y tipos de datos.

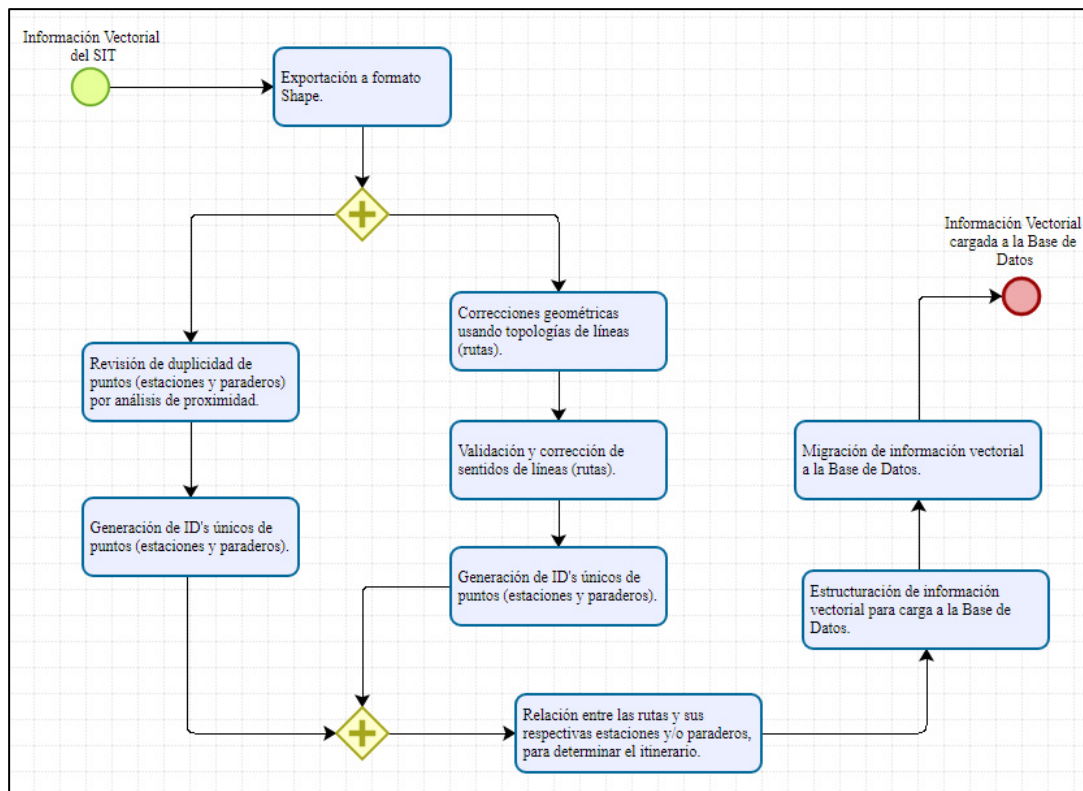
2.7.8.1. Flujos para datos propios de la entidad (ATU)

La ATU, a través de sus direcciones especializadas, provee información del Sistema Integrado del Transporte Regular. El flujo para el tratamiento de los datos del SIT, inicia identificando el estado y formato de la información provista, realizar las correcciones topológicas geométricas requeridas para estructurar los datos vectoriales y tabulares bajo el formato requerido para su migración a la Base de Datos.

Tratamiento de Datos Vectoriales:

Figura 85

Flujo de Tratamiento de Datos Vectoriales

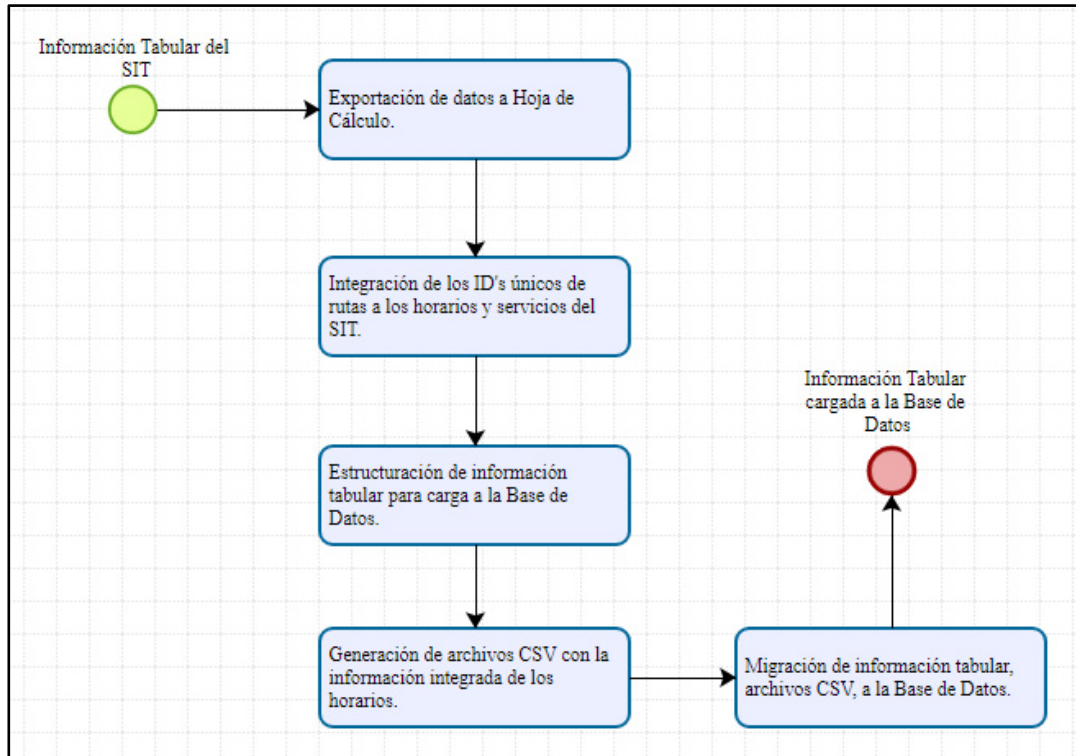


Nota. Elaboración propia

Tratamiento de Datos Tabulares:

Figura 86

Flujo de Tratamiento de Datos Tabulares

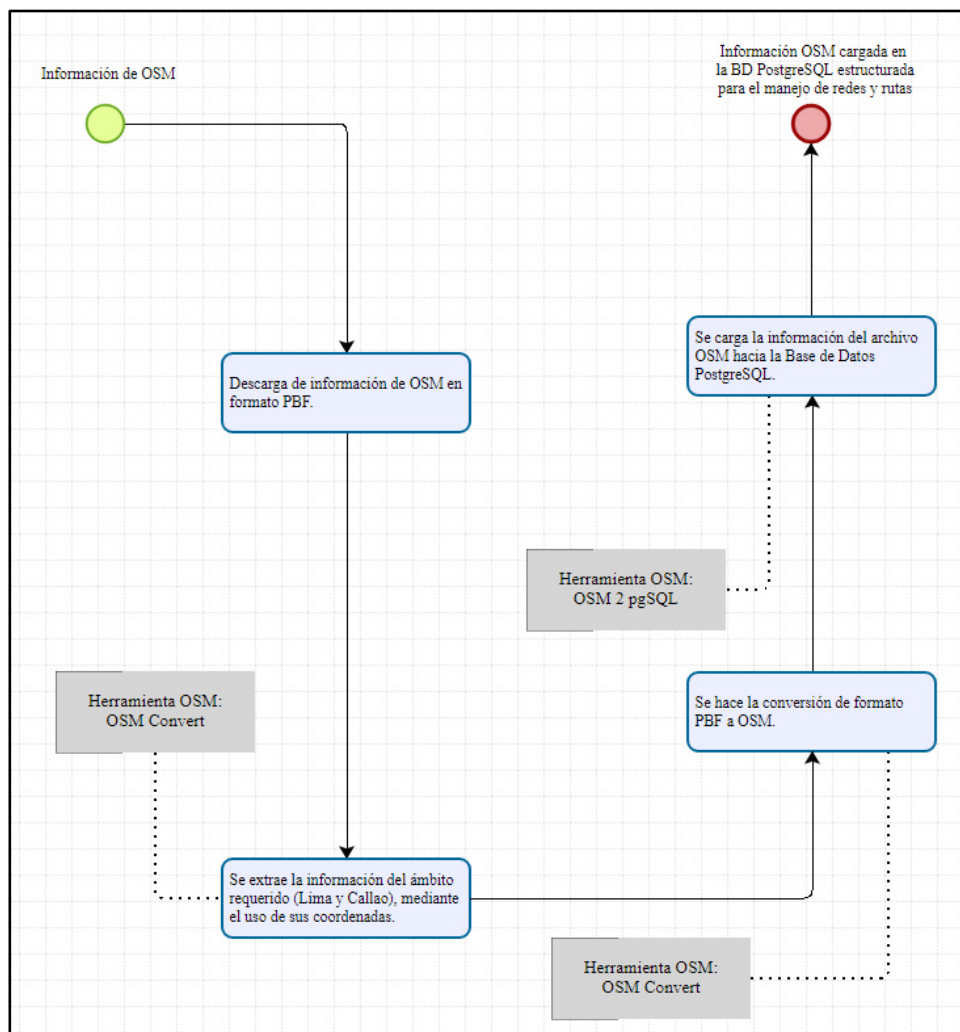


Nota. Elaboración propia

2.7.8.2. Flujos para datos complementarios de Open Street Map (OSM)

Como complemento a los datos del SIT Regular, se integra la información de calles y vías; la cual es descargada, preprocesada y cargada a la Base de Datos mediante herramientas de licenciamiento libre, pertenecientes al ecosistema de la plataforma Open Street Map.

Esta información se usará para determinar las rutas de la caminata entre los paraderos del SIT, en caso de ser necesario, al ejecutar el algoritmo de la generación de rutas.

Figura 87*Flujo de Preprocesamiento y Carga de Datos OSM*

Nota. Elaboración propia

2.7.8.3. Detalle de Procesos Automatizados

Con fin de optimizar los flujos de análisis de los datos es necesario implementar procesos automatizados que puedan ser reutilizados constantemente por la gran cantidad de datos a procesar. Para ello, es necesario desarrollar scripts en Python que tengan un enfoque modular y reutilizable, permitiendo cumplir con los procesamientos de la data de manera constante, sostenible y eficiente.

OSM descarga de datos:

Scripts para descargar información de OSM, usando como área de delimitación de

descarga una capa vectorial de tipo polígono.

Tabla 17

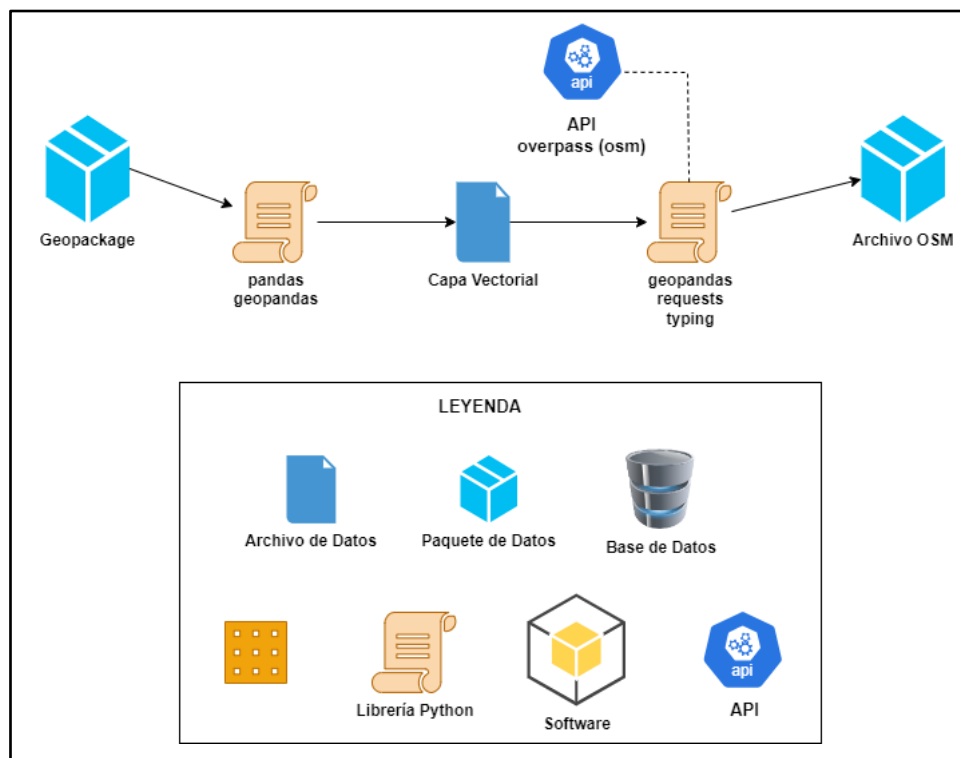
Secciones OSM Descarga de Datos

Ingreso	Salida	Requerimientos
Geopackage	Paquete OSM	Librerías Externas: <ul style="list-style-type: none"> • Pandas • GeoPandas • Requests • Typing API: <ul style="list-style-type: none"> • Overpass OSM

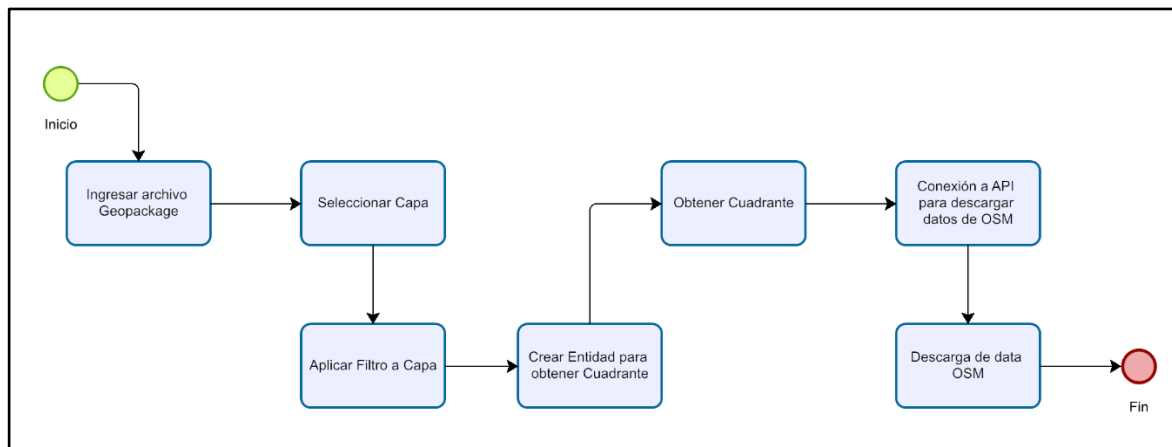
Nota. Elaboración propia

Figura 88

Componentes OSM Descarga de Datos



Nota. Elaboración propia

Figura 89*Flujo OSM Descarga de Datos*

Nota. Elaboración propia

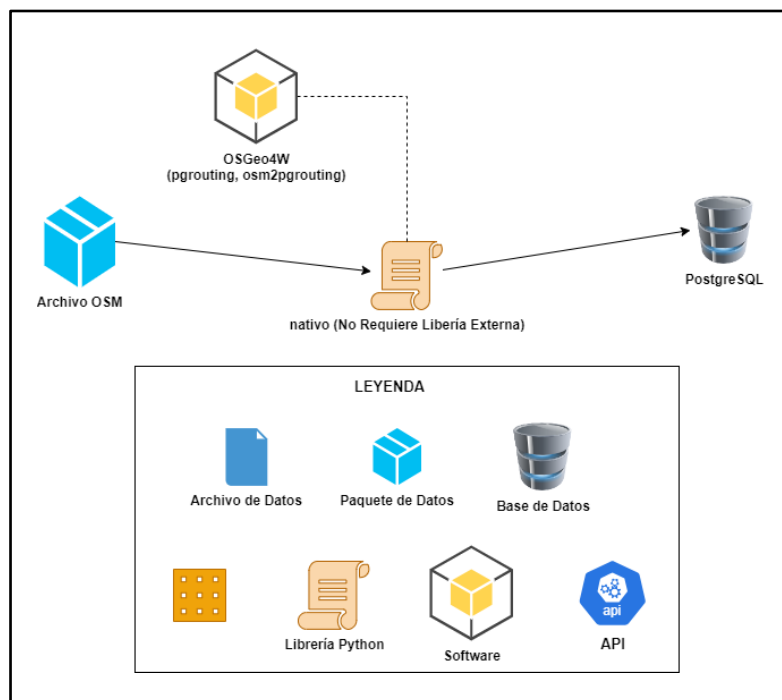
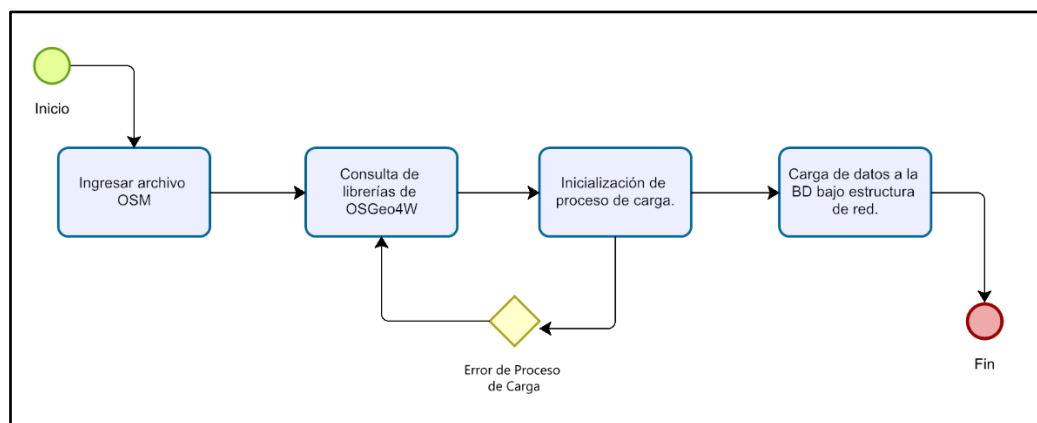
OSM carga de archivo a Base de Datos:

Scripts para registrar los datos descargados de OSM a la Base de Datos en formato de estructura de red.

Tabla 18*Secciones OSM Carga de Datos a BD*

Ingreso	Salida	Requerimientos
Archivo OSM	Tablas con Modelo de Red PostgreSQL	Software: <ul style="list-style-type: none"> • OSGeo4W, con extensiones pgrouting y osm2pgroutin.

Nota. Elaboración propia

Figura 90*Componentes OSM Carga de Datos a BD**Nota.* Elaboración propia**Figura 91***Flujo OSM Carga de Datos a BD**Nota.* Elaboración propia**KML/KMZ exportación de capas a GeoPackage:**

Scripts para exportar los datos de un KML/KMZ a capas integradas, según su tipo de geometría sea punto, línea o polígono, almacenadas dentro de un GeoPackage.

Tabla 19

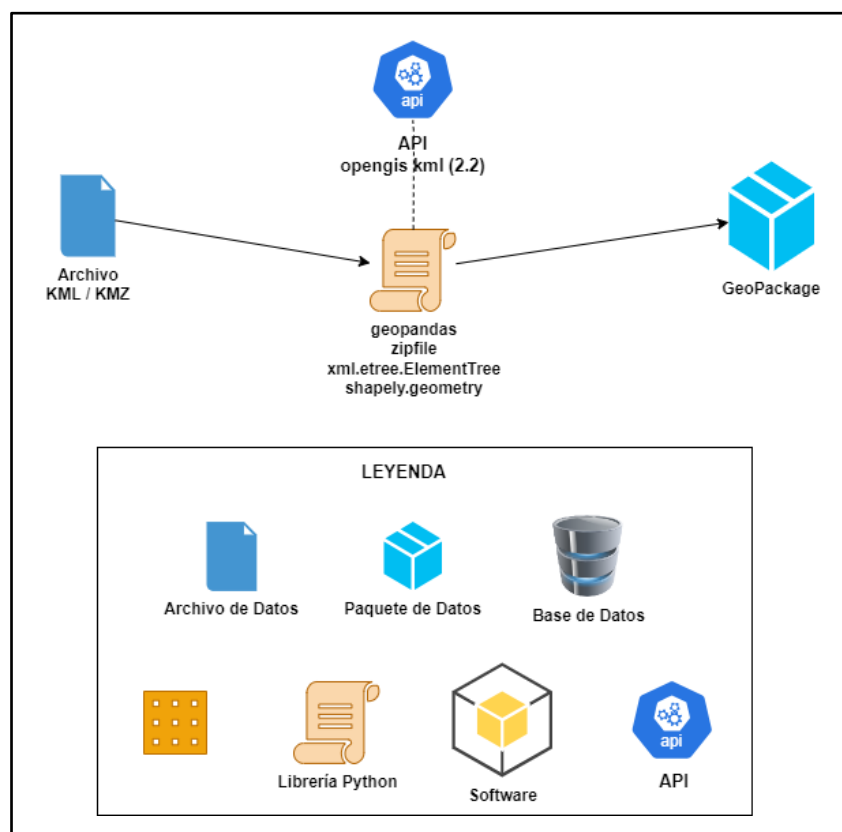
Secciones KML/KMZ exportación a GeoPackage

Ingreso	Salida	Requerimientos
Archivo KMZ/KML	Paquete GeoPackage	<p>Librerías Externas:</p> <ul style="list-style-type: none"> • Pandas • GeoPandas • ZipFile • Xml.etree.ElementTree • shapely.geometry <p>API:</p> <ul style="list-style-type: none"> • OpenGIS KML

Nota. Elaboración propia

Figura 92

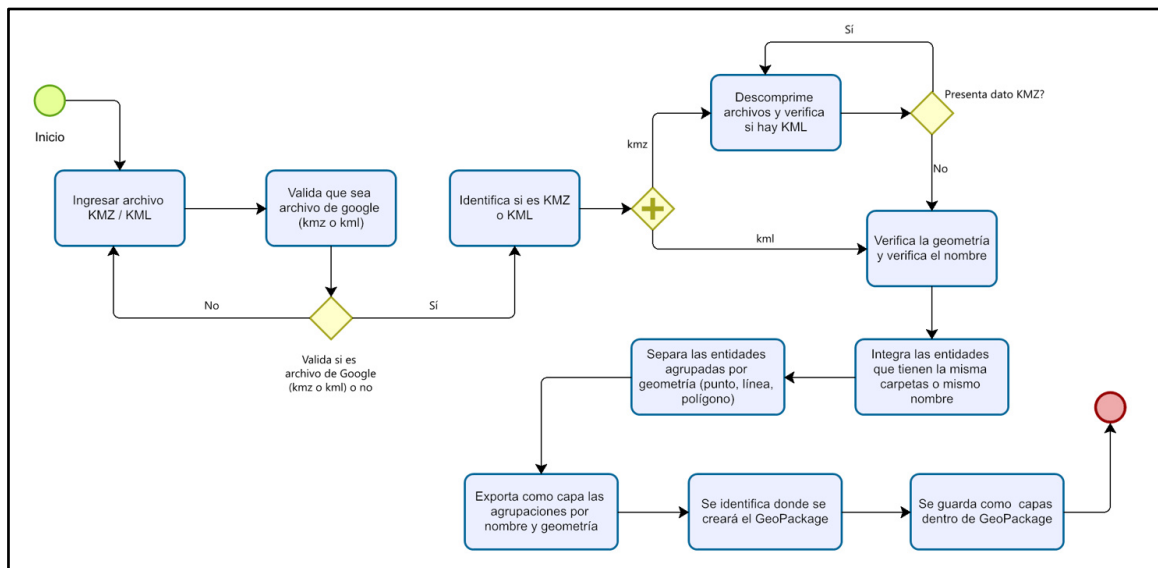
Componentes KML/KMZ exportación a GeoPackage



Nota. Elaboración propia

Figura 93

Flujo OSM Carga de Datos a BD



Nota. Elaboración propia

GeoPackage Capa de Puntos Integridad de Datos:

Scripts para mantener la integridad de los datos a nivel de posición de capas de geometría de tipo punto almacenadas dentro de un GeoPackage.

Tabla 20

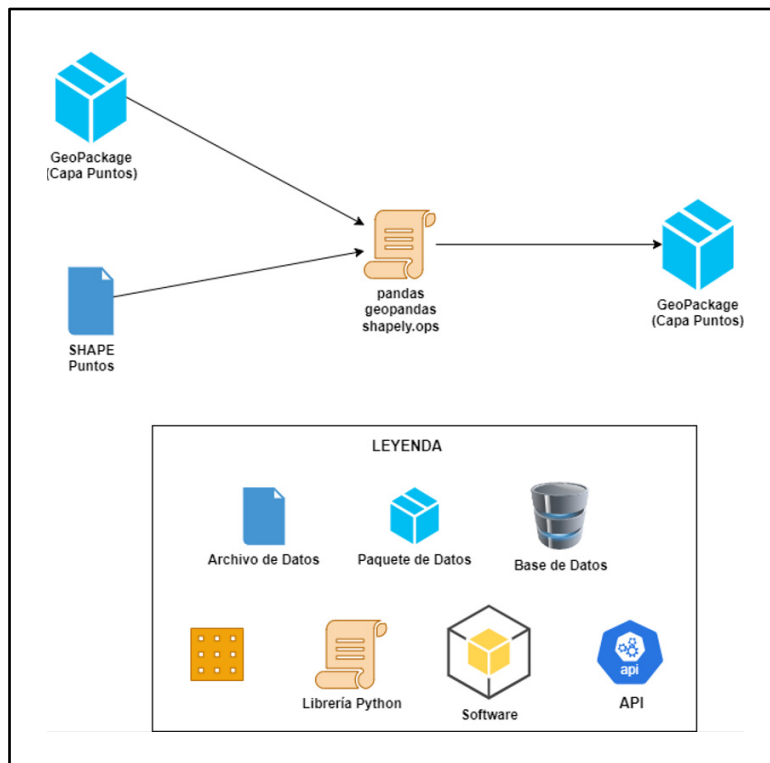
Secciones Integridad de Puntos

Ingreso	Salida	Requerimientos
Paquete GeoPacakage Archivo Shapefile	Paquete GeoPackage	Librerías Externas: <ul style="list-style-type: none"> • Pandas • GeoPandas • shapely.ops

Nota. Elaboración propia

Figura 94

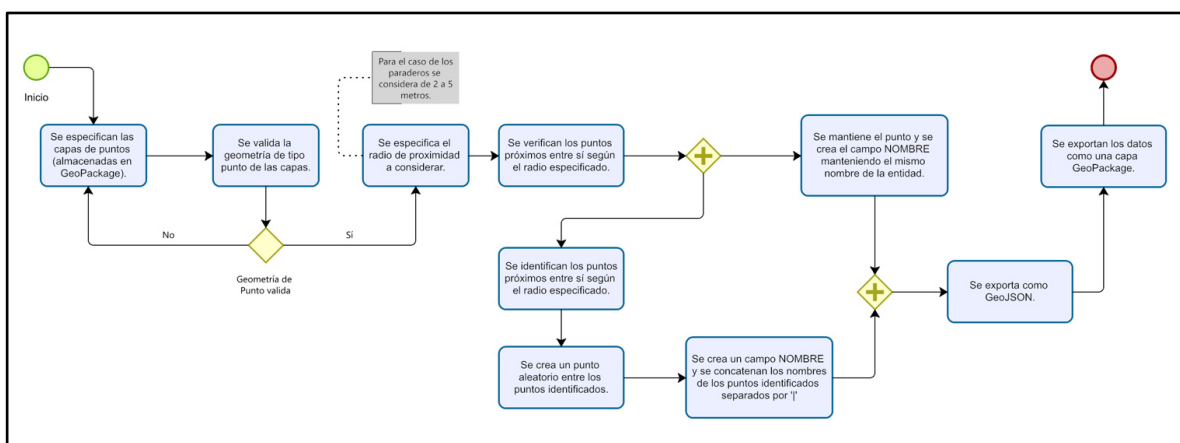
Componentes Integridad de Puntos



Nota. Elaboración propia

Figura 95

Flujo Integridad de Puntos



Nota. Elaboración propia

GeoPackage Capa de Polilíneas Integridad de Datos:

Scripts para mantener la integridad de los datos de polilíneas, enfocado en prevenir la

duplicidad de trayectorias, considerar que puede ser necesario un proceso manual para mejorar la visualización de la geometría.

Tabla 21

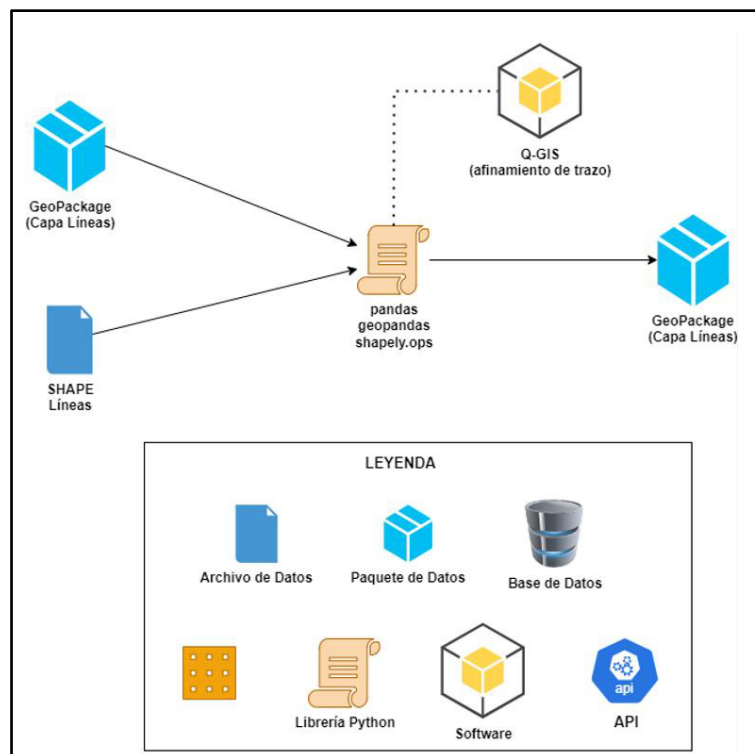
Secciones Integridad de Puntos

Ingreso	Salida	Requerimientos
Paquete GeoPacakage Archivo Shapefile	Paquete GeoPackage	<p>Librerías Externas:</p> <ul style="list-style-type: none"> • Pandas • GeoPandas • shapely.geometry • shapely.ops <p>Software:</p> <ul style="list-style-type: none"> • Q-GIS (para afinamiento manual de geometría)

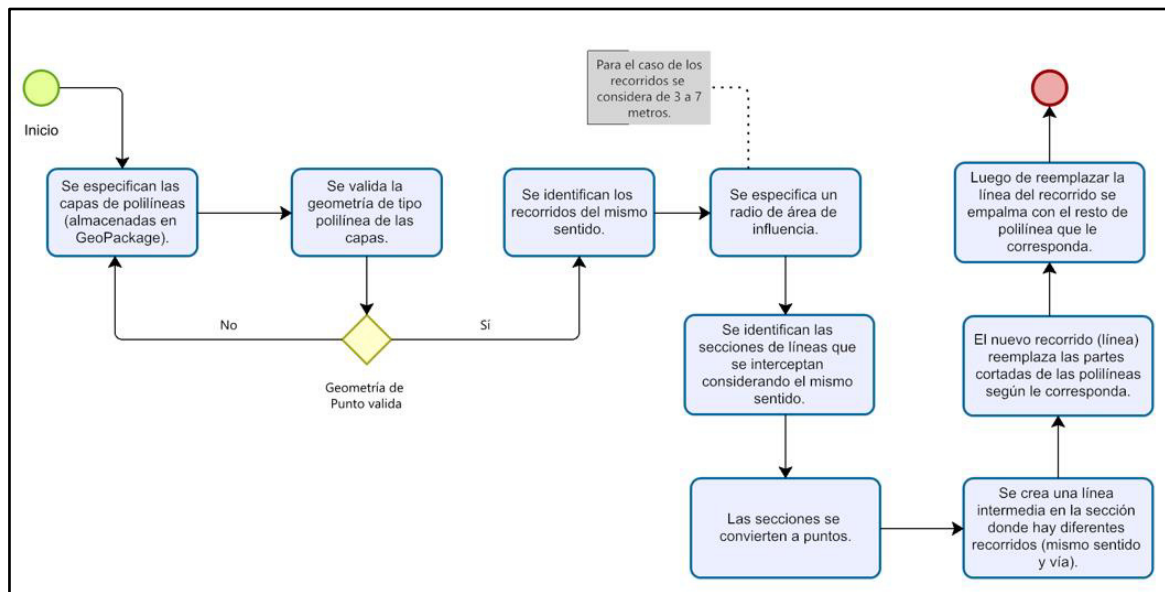
Nota. Elaboración propia

Figura 96

Componentes Integridad de Polilíneas



Nota. Elaboración propia

Figura 97*Flujo Integridad de Polilíneas*

Nota. Elaboración propia

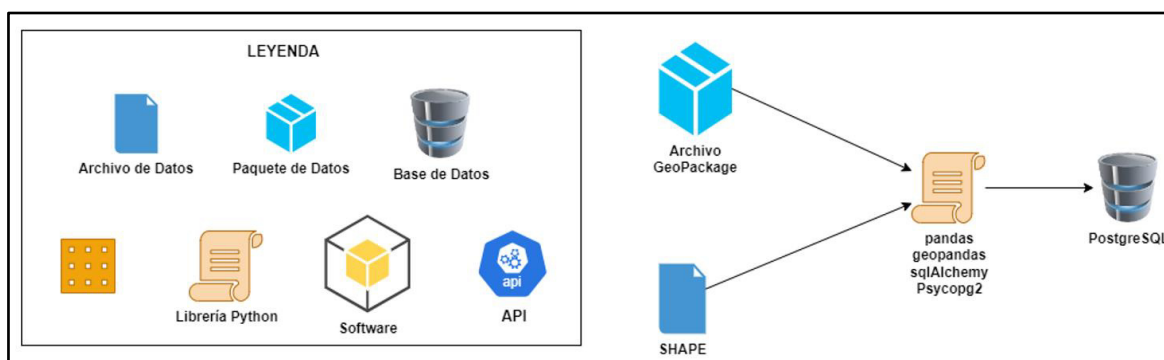
Capas a PostgreSQL:

Scripts para cargar datos vectoriales a la Base de Datos PostgreSQL.

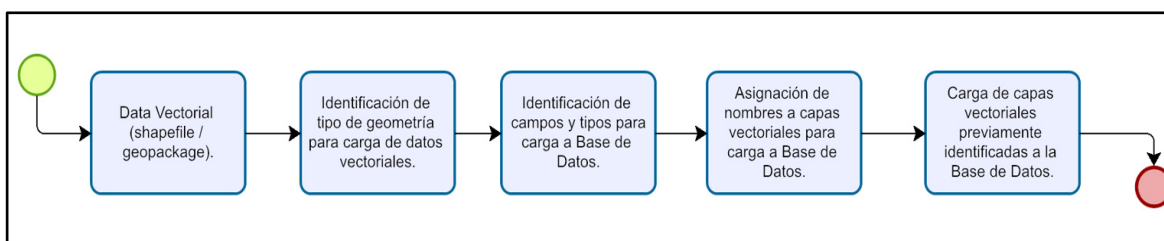
Tabla 22*Secciones Capas a PostgreSQL*

Ingreso	Salida	Requerimientos
Paquete GeoPacakage Archivo Shapefile	Tablas Geométricas PostgreSQL	Librerías Externas: <ul style="list-style-type: none"> • Pandas • GeoPandas • sqlalchemy • Psycopg2

Nota. Elaboración propia

Figura 98*Componentes Capas a PostgreSQL*

Nota. Elaboración propia

Figura 99*Flujo Capas a PostgreSQL*

Nota. Elaboración propia

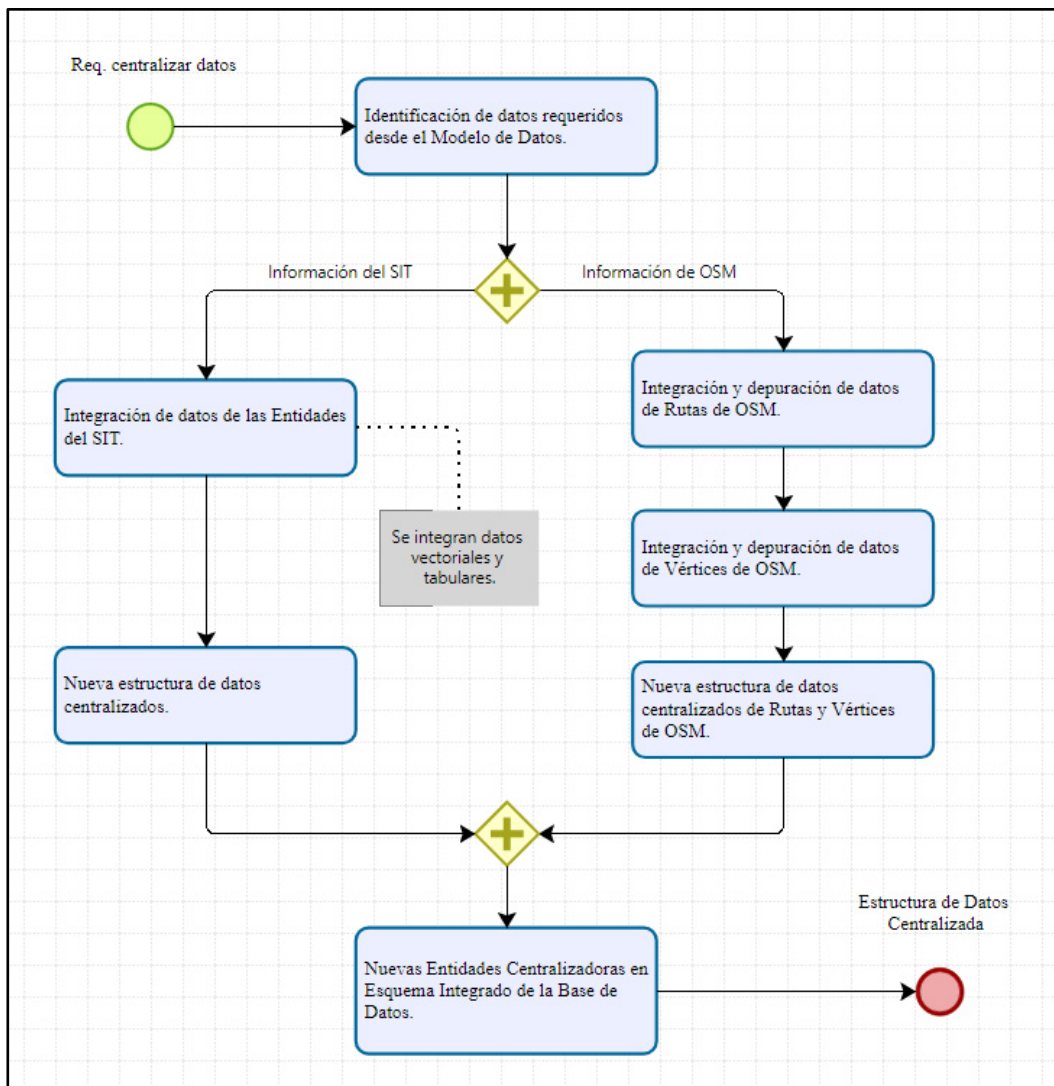
2.7.9. Integración y Estructuración de Datos como Grafos para proceso de Análisis de Ruta

Esta etapa se centra en combinar y organizar los datos provenientes de la entidad ATU y los datos complementarios de OSM, con el fin de unificarlos bajo una estructura de grafos. Esto permitirá abordar el análisis de rutas mediante algoritmos especializados.

Considerar que el enfoque de manejar los datos bajo una estructura grafos es clave para poder aplicar los algoritmos para el cálculo de las rutas óptimas de manera eficiente.

Figura 100

Flujo para la Integración y Estructuración de Datos



Nota. Elaboración propia

2.7.9.1. Integración de datos de la entidad (ATU)

Los datos del SIT, son integrados con el fin de centralizar la información mediante nuevos objetos de base de datos, en este proceso se integran los datos tabulares con los datos geométricos (datos georreferenciados).

Primero: Se identifican los datos a centralizar.

Tabla 23

Identificación de objetos de Base de Datos referentes al SIT para Centralizar

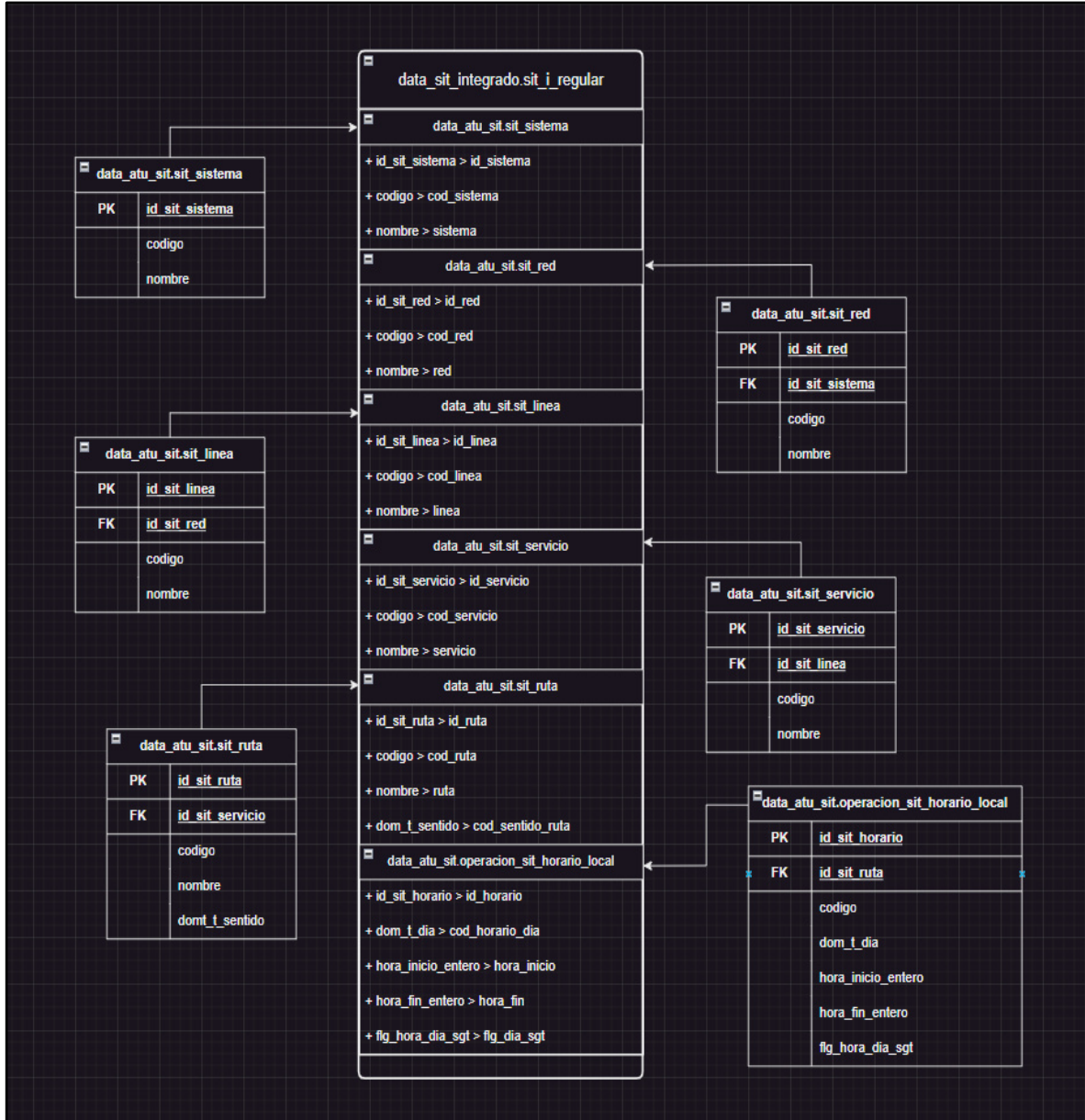
Objeto de Base de Datos del SIT	Objeto de Base de Dato Centralizador
SIT_SISTEMA SIT_RED SIT_LINEA SIT_SERVICIO SIT_RUTA SIT_HORARIO_LOCAL	SIT_I_REGULAR
SIT_RUTA SIT_RECORRIDO_GIS (líneas) SIT_HORARIO_LOCAL	SIT_I_RUTA
SIT_SISTEMA SIT_INFRAESTRUCTURA SIT_INFRAESTRUCTURA_GIS (puntos)	SIT_I_PARADA
SIT_RUTA SIT_RECORRIDO_GIS (líneas) SIT_INFRAESTRUCTURA SIT_INFRAESTRUCTURA_GIS (puntos) SIT_ITINERARIO	SIT_I_RUTA_PARADA

Nota. Elaboración propia

- **Segundo:** Se crean los procesos de integración para centralizar los datos.

Figura 101

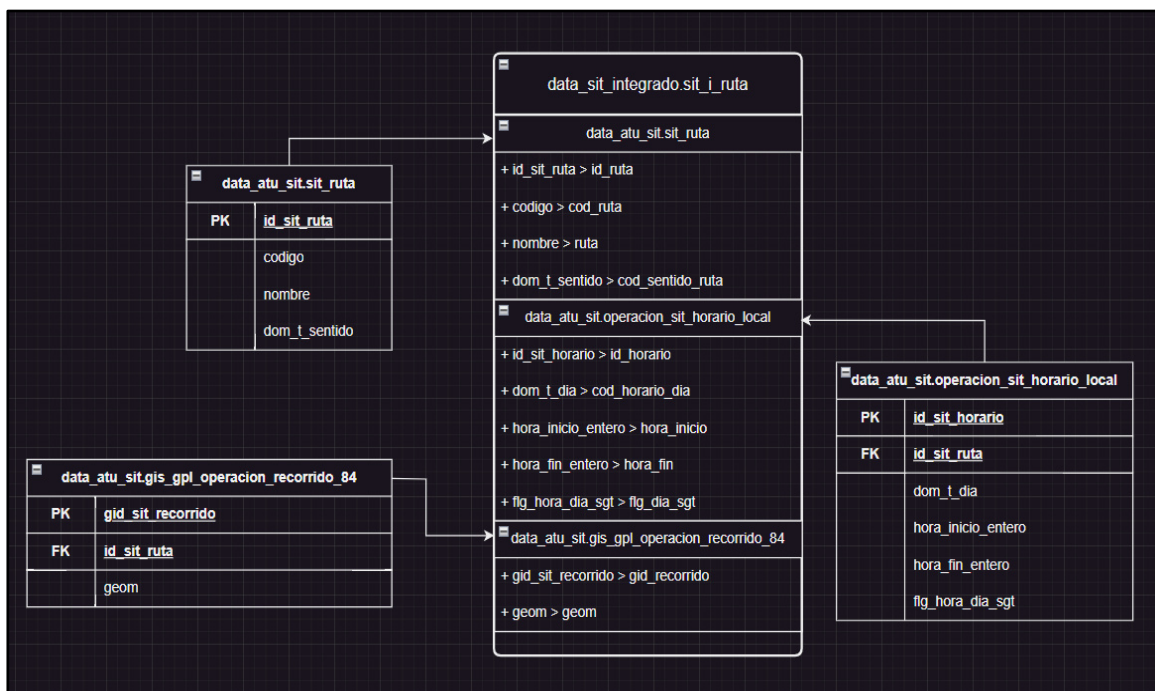
Proceso de centralización del Objeto: SIT_I_REGULAR



Nota. Elaboración propia

Figura 102

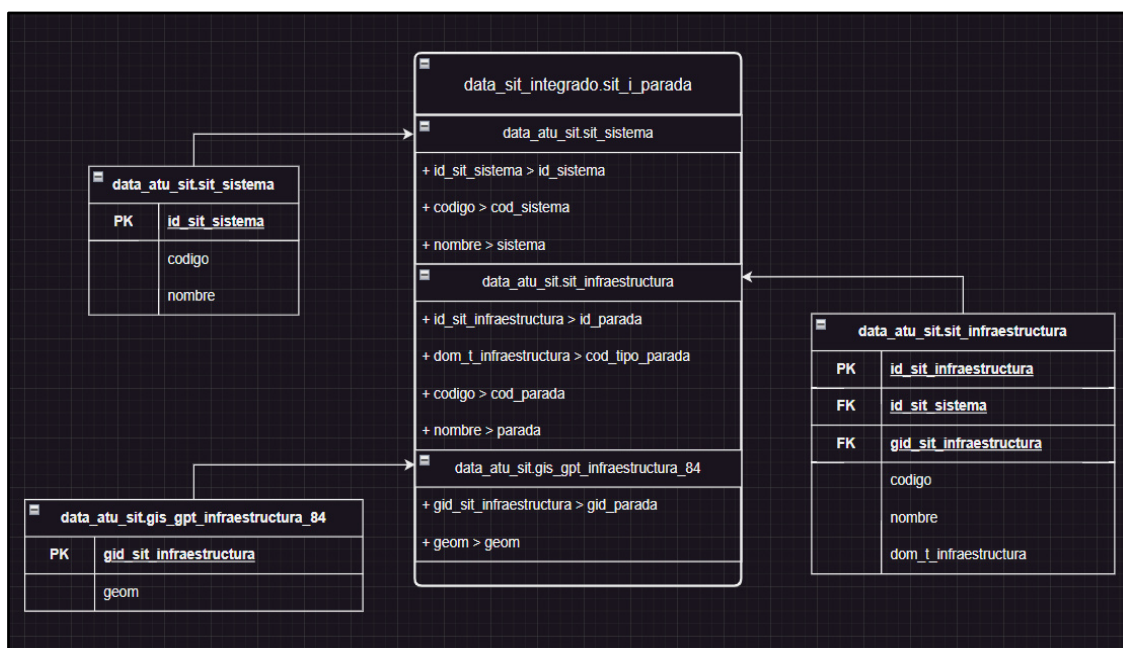
Proceso de centralización del Objeto: SIT_I_RUTA



Nota. Elaboración propia

Figura 103

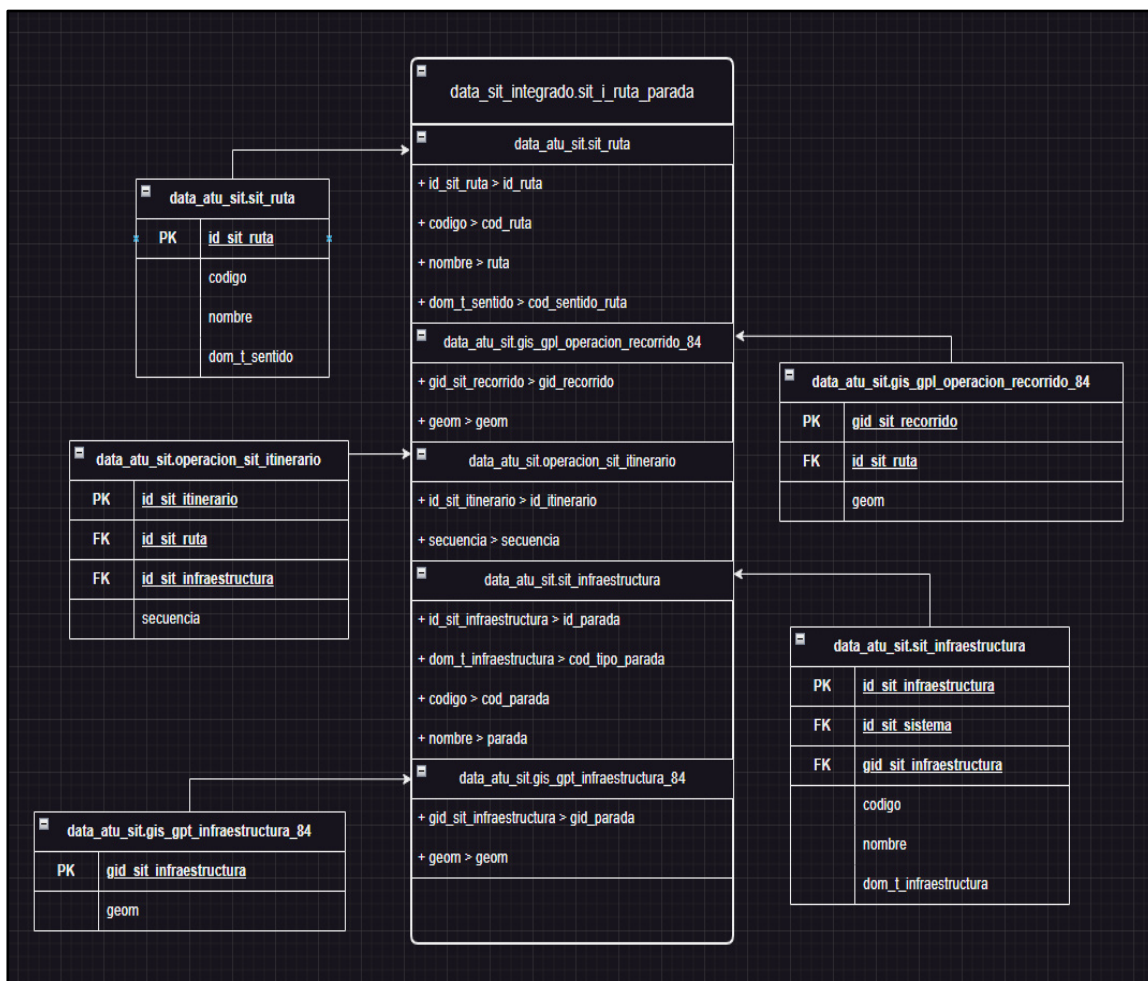
Proceso de centralización del Objeto: SIT_I_PARADA



Nota. Elaboración propia

Figura 104

Proceso de centralización del Objeto: SIT_I_RUTA_PARADA



Nota. Elaboración propia

2.7.9.2. Integración de datos de complementarios de Open Street Map (OSM)

Los datos a requerir de OSM (Open Street Map), se centralizan para poder hacer uso eficiente de las entidades durante los procesos de análisis, cabe precisar que al ejecutar el proceso de carga de los datos de OSM a la Base de Datos se crean las entidades de manera automática, por ende, se los datos tabulares y los datos georreferenciados se registran ya integrados.

- **Primero:** Se identifican los datos a centralizar.

Tabla 24

Identificación de objetos de Base de Datos de OSM (Open Street Map) a Centralizar

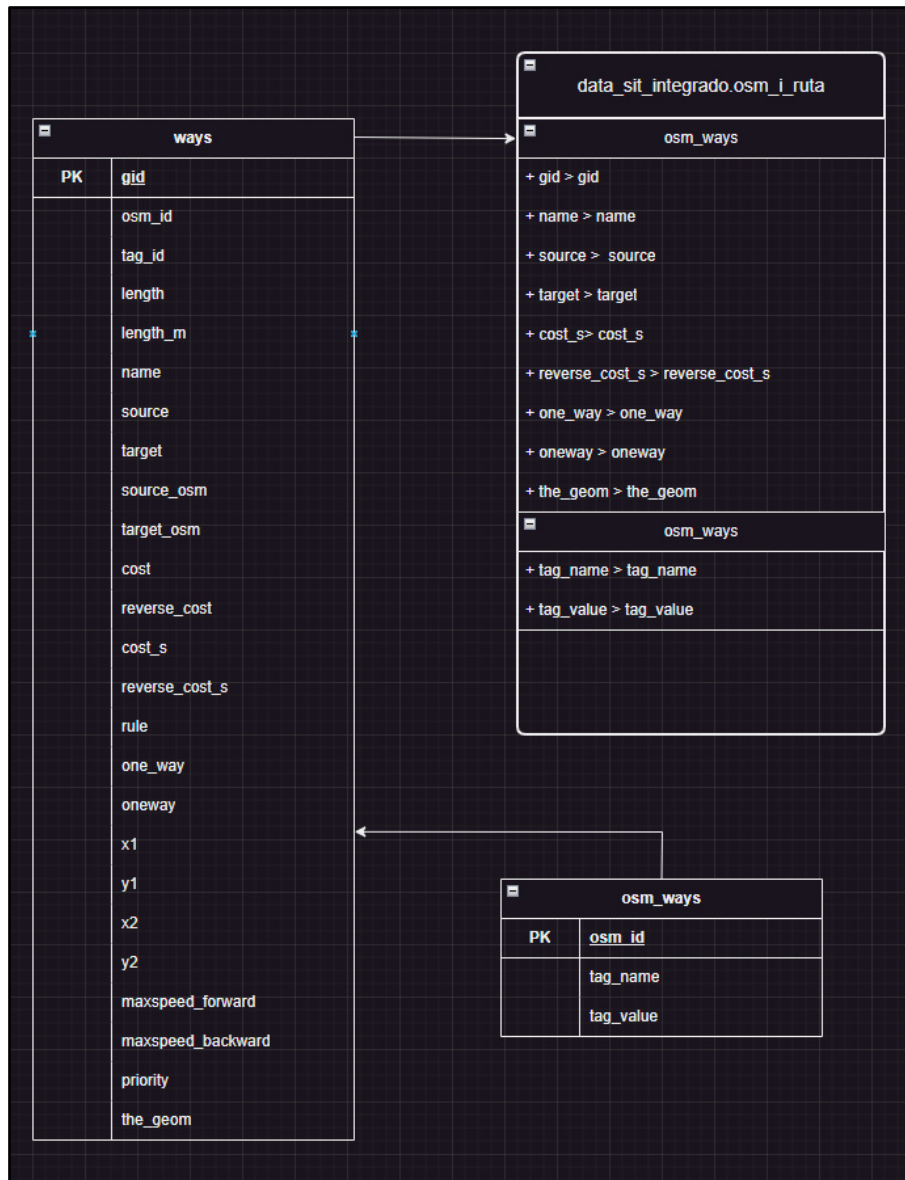
Objeto de Base de Datos de OSM	Objeto de Base de Dato Centralizador
WAYS OSM_WAYS	OSM_I_RUTA
WAYS_VERTICES_PGR	OSM_I_RUTA_VERTICE

Nota. Elaboración propia

- **Segundo:** Para el caso de las Rutas se depuran los datos y se exportan como un nuevo objeto de base de datos: OSM_I_RUTA.

Figura 105

Integración de entidades para el Objeto: OSM_I_RUTA

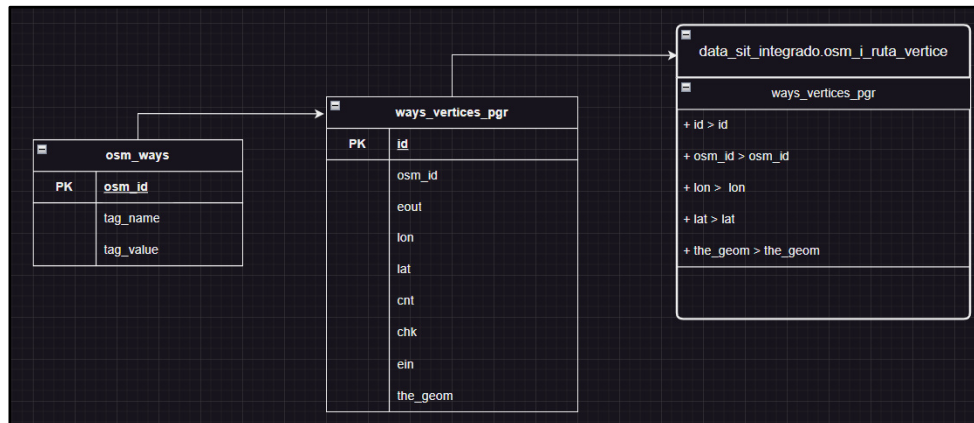


Nota. Elaboración propia

- **Tercero:** Para el caso de los Vértices, estos son depurados partiendo de la data de Rutas previamente depurada, y seguido se exportan como un nuevo objeto de base de datos: OSM_I_VERTICE.

Figura 106

Integración de entidades para el Objeto: SIT_I_VERTICE



Nota. Elaboración propia

2.7.9.3. Estructuración de Datos como Objetos del Grafo

Para generar la estructura de grafos en la base de datos es necesario tener en cuenta, la estructura básica de un grafo compuesta por dos objetos:

- **Nodos:** Representado por los vértices producto de la conexión de un conjunto de líneas. Para el proyecto los vértices son representados por los paraderos existentes y en estado operativo de los servicios de transporte.
- **Arcos:** Hace referencia a las aristas o lados, representando a las líneas que conectan dos o más vértices. Para su aplicación en el proyecto los Arcos están conformados por las rutas de los servicios de transporte que conectan los Paraderos, y a su vez una ruta a nivel de camina entre paradero y paradero previamente identificada gracias a la data vectorial de OSM.

Luego de mencionar las secciones que componen un grafo y cómo éstas son representadas por la data del proyecto, no se puede dejar de lado datos importantes pero

sesgados a los grafos ponderados dirigidos, los cuales son valores clave para el cálculo de rutas óptimas. Estos son:

- **Grados:** El concepto de "grado" de un vértice, que indica el número de arcos que convergen en él, es especialmente útil en grafos dirigidos. En el contexto de nuestro proyecto, el grado de un paradero refleja su importancia en la red de transporte, ya que un paradero con muchas conexiones a diferentes rutas puede ser crucial para encontrar trayectos óptimos.
- **Peso o Ponderado:** Concepto propio de los grafos con pesos o también conocidos como ponderados, donde cada arco posee un valor numérico llamado "peso" que influye en la selección de la ruta óptima. Este peso puede representar diferentes criterios según la aplicación, priorizando arcos con valores mayores o menores. Por ejemplo, en grafos donde el peso indica distancia o tiempo, se prefieren arcos con valores bajos para minimizar el recorrido. En cambio, si el peso representa velocidad, los arcos con valores altos son preferibles para maximizar la rapidez del trayecto. En este proyecto, el peso de los arcos se calculará como el tiempo de viaje, resultado de dividir la distancia entre la velocidad de desplazamiento. Se considerarán distintos medios de transporte del sistema público (metro, buses troncales, alimentadores, corredores) con sus velocidades correspondientes. Para tramos a pie (datos OSM), se usará una velocidad promedio de caminata.

Así mismo, para cumplir con el propósito de la herramienta es necesario integrar datos que no son intrínsecos a los grafos, sin embargo, permiten satisfacer las necesidades de la herramienta. Estos datos son:

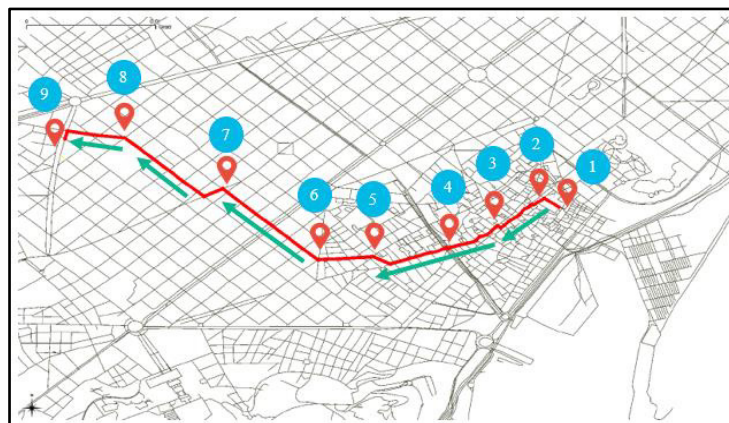
- **Tipo de Sistema de Transporte:** Para los casos que se desee realizar el cálculo o la planificación de una ruta óptima por determinados tipos de sistemas de transporte es necesario contar con dicho valor para poder filtrarlos.

- **Horario de Servicio:** El horario de funcionamiento de los servicios de transporte es fundamental para determinar que rutas deben ingresar al proceso de análisis según el estado de las mismas, pues el análisis de rutas solo deberá contemplar las rutas que se encuentren operativas según el horario de viaje del usuario.
- **Coordenadas:** Aunque pueda parecer evidente al hablar de nodos (paraderos) y arcos (rutas), es crucial destacar que las coordenadas geográficas son fundamentales para el análisis. Estas permiten visualizar la interconexión entre rutas y paraderos, facilitando la comprensión y la toma de decisiones basadas en la ubicación real de los elementos del sistema de transporte.

En conclusión, la estructura de grafos para este proyecto se define como un grafo dirigido y ponderado. Su complejidad radica en la representación de la red de transporte público, donde los nodos son las paradas operativas y los arcos son las rutas en servicio, incluyendo tramos peatonales entre paradas si es necesario. Para asegurar análisis precisos, se incorporan atributos como el tipo de transporte y los horarios de servicio a los elementos del grafo. La visualización de esta red se logrará mediante la representación vectorial de las coordenadas geográficas de los nodos y arcos.

Figura 107

Estructura de Ruta y Paradas



Nota. Elaboración propia

Tabla 25*Estructuración de los Nodos para los Grafos*

Nodos			
Representa las paradas asociadas a las rutas operativas de los servicios de transporte público de carácter regular. Las paradas pueden ser paraderos o estaciones según el tipo de sistema de transporte público regular, y se asocian al as rutas mediante un “ID”.			
Dato	Uso	Cálculo	Origen del Dato
Coordenada (Ubicación)	Representa a la ubicación del paradero o estación en estado operativo y asociados a las rutas operativas de los servicios de transporte público de carácter regular.	Para evitar redundancia se toma un solo punto como parada cuando hay redundancia de puntos en la misma ubicación considerando una proximidad entre ellos de 3 a 5 metros y que sean del mismo tipo paradero o estación.	SIT_I_PARADA OSM_I_RUTA_VERTICE
Grados	Permite identificar el nivel de priorización del nodo mediante la identificación de cantidad de rutas que llegan hacia él.	Se determinar por la cantidad de rutas que llegan hacia el nodo en un determinado horario. Contemplando diferentes grados del mismo nodo, según el rango horario.	SIT_I_RUTA_PARADA
Tipología del SIT	Permite identificar a que tipo de Sistema de Transporte Público pertenece el nodo, y es necesario en caso de realizar un filtro sobre los sistemas y por ende nodos a considerar en el análisis.	No requiere, por ser extraído del Modelo Entidad – Relación.	SIT_I_REGULAR
Horarios de Servicio	Permite identificar si el nodo debe ser evaluado o no según el momento horario en el que se ejecute el procedimiento de análisis.	No requiere, por ser extraído del Modelo Entidad – Relación.	SIT_I_REGULAR

Nodos

Representa las paradas asociadas a las rutas operativas de los servicios de transporte público de carácter regular. Las paradas pueden ser paraderos o estaciones según el tipo de sistema de transporte público regular, y se asocian al as rutas mediante un “ID”.

Dato	Uso	Cálculo	Origen del Dato
Nombre	Dato descriptivo del nombre del nodo.	No requiere, por ser extraído del Modelo Entidad – Relación.	SIT_I_PARADA OSM_I_RUTA_VERTICE
Código	Dato descriptivo del código del nodo.	No requiere, por ser extraído del Modelo Entidad – Relación.	SIT_I_PARADA OSM_I_RUTA_VERTICE
Dirección	Dato descriptivo de la dirección del nodo.	No requiere, por ser extraído del Modelo Entidad – Relación.	SIT_I_PARADA OSM_I_RUTA_VERTICE
Tipo de Parada	Permite identificar si el nodo hace referencia a un paradero o a una estación, pues son dos infraestructuras diferentes en el sistema de transporte público.	No requiere, por ser extraído del Modelo Entidad – Relación.	SIT_I_PARADA

Nota. Elaboración propia

Tabla 26*Estructuración de los Arcos para los Grafos*

Arcos			
Representa a las rutas operativas de los servicios de transporte público y los tramos de caminata entre parada y parada, de ser necesaria. Estas rutas se encuentran segmentadas de parada a parada para darle la estructura de un grafo a la red conformada por el servicio de transporte público regular en estado de operación, permitiendo facilitar los procesos de análisis.			
Dato	Uso	Cálculo	Origen del Dato
Coordenada (Recorrido)	El conjunto de coordenadas representa el recorrido del segmento. Para el caso de los datos de: - El SIT hace referencia a las rutas. - OSM hace referencia a los tramos de caminata	Se crea un segmento el cual tiene como origen y fin un nodo, para el caso de: - Las rutas el segmento está determinado por la secuencialidad de sus paradas, sea paraderos o estación según sea el caso. - Los tramos de caminata, representan un segmento, el cual vincula paraderos y estaciones o paradas del mismo tipo, que no se conectan mediante una ruta.	OSM_I_RUTA SIT_I_RUTA
Sentido (Grafo Dirigido)	Permite identificar la dirección del segmento y por ende saber si es un valor de entrada o salida del nodo. Para el caso de los segmentos de: - Las rutas, depende de su sentido (origen – destino). - Los tramos de caminata, puede asumir el mismo valor de ida y retorno entre paradores por ser de doble sentido.	Se determina mediante la dirección del segmento, para el caso de: - Las rutas, la dirección depende del origen y destino del recorrido, siendo este siempre de un solo sentido (salida – llegada). - Los tramos de caminata, son de doble sentido por ende son segmentos bidireccionales, porque pueden	SIT_I_RUTA

comportarse como conexiones de ida y retorno según sea la necesidad del análisis.

<p>Peso / Ponderación</p>	<p>Permite identificar a que tipo de Sistema de Transporte Público pertenece el nodo, y es necesario en caso de realizar un filtro sobre los sistemas y por ende nodos a considerar en el análisis.</p>	<p>Se determina mediante el cálculo de tiempo de viaje que tiene el segmento, el cual se determina mediante la división de la distancia sobre la velocidad. (Tiempo = Distancia / Velocidad)</p> <p>Considerar que las velocidades varían según el tipo de transporte, excepto para segmentos de los tramos de caminata cuya velocidad es constante.</p>	<p>Velocidad según medio de desplazamiento:</p> <p>Metro: - Vagón: 40 km/h</p> <p>Metropolitano: - Bus Troncal: 40 km/h - Bus Alimentador: 10 km/h</p> <p>Corredores: - Bus de Corredor: 10 km/h</p> <p>Persona Promedio: - Velocidad de Caminata: 5 km/h</p>
<p>Tipología del SIT</p>	<p>Permite identificar a que tipo de Sistema de Transporte Público pertenece la ruta, y es necesario en caso de realizar un filtro sobre los sistemas y por ende entre las rutas a considerar en el análisis.</p>	<p>No requiere, por ser extraído del Modelo Entidad – Relación.</p>	<p>SIT_I_REGULAR</p>
<p>Horarios de Servicio</p>	<p>Permite identificar si el arco, representado por el segmento de ruta, debe ser evaluado o no según el momento horario en el que se ejecute el procedimiento de análisis.</p>	<p>No requiere, por ser extraído del Modelo Entidad – Relación.</p>	<p>SIT_I_REGULAR</p>

Código	Dato descriptivo del código de la ruta.	No requiere, por ser extraído del Modelo Entidad – Relación.	SIT_I_RUTA
Nombre	Dato descriptivo del nombre de la ruta.	No requiere, por ser extraído del Modelo Entidad – Relación.	SIT_I_RUTA

Nota. Elaboración propia

2.7.10. Generación de Procesos de Análisis para la Planificación de Rutas

Luego de determinar la estructura final en la base de datos, en el esquema correspondiente, para optimizar el proceso de análisis. Se procede a implementarlo, este se describe en tres secciones.

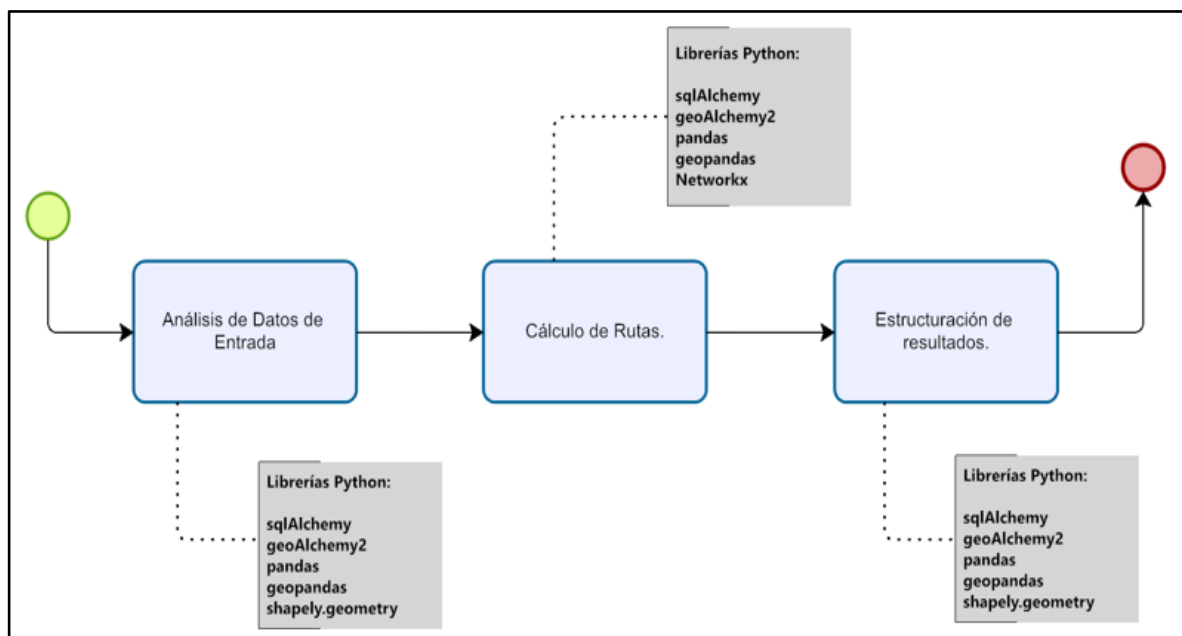
Análisis de Datos de Entrada: En esta fase se evalúan los criterios y filtros de búsqueda, determinando su validez respecto a la data estructura sobre el SIT.

Cálculo de Rutas: En este parte se toman los datos de entrada ya validados para realizar el cálculo de rutas, usando librerías que implementan Dijkstra. Siendo necesario acceder a la base de datos para leer los objetos que se usarán para estructurar el grafo.

Estructuración de Resultados: Con las rutas ya determinadas, previo cálculo, se procede a estructurar la información con sus respectivas coordenadas y colores que definirán cada recorrido, para ser consumidas en la aplicación.

Figura 108

Procesos de Análisis para la Planificación de Rutas



Nota. Elaboración propia

2.7.10.1. Análisis de Datos de Entrada

Analizar los datos de entrada consiste en determinar la validez de la información ingresada según los criterios previamente establecidos y considerando la existencia de información respecto a los filtros determinados al momento de ingresar la información.

Datos a Ingresar:

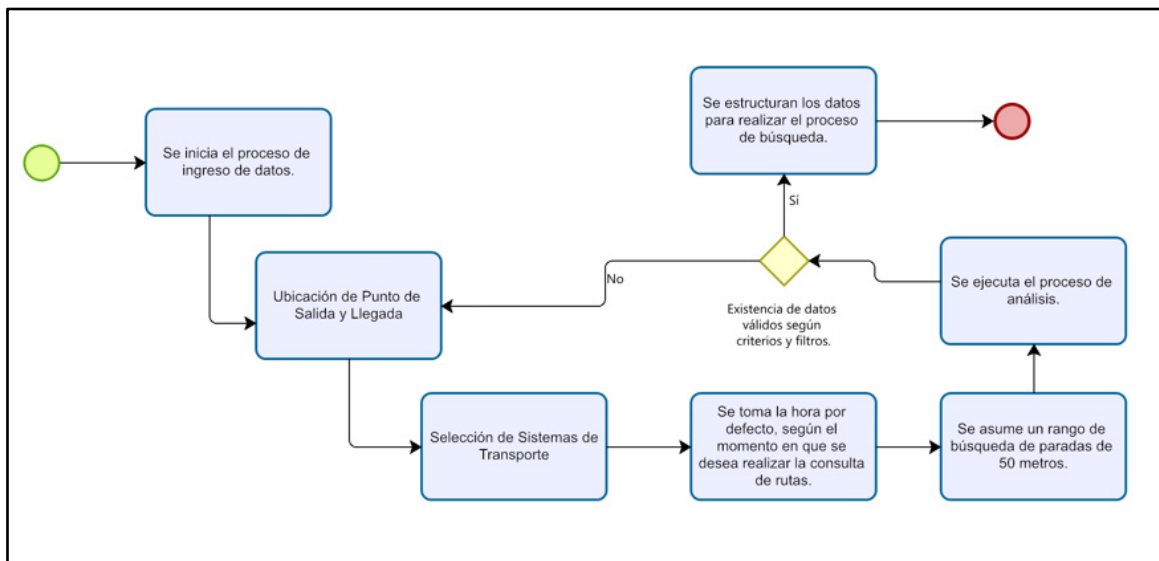
- Punto de Partida: Identifica la ubicación desde la cual se pretende partir hacia un determinado destino, siendo los datos a ingresar las coordenadas de la ubicación.
- Punto de Llegada: Identifica la ubicación a la cual se necesita llegar, siendo los datos a ingresar las coordenadas de su ubicación.

Filtros de Información:

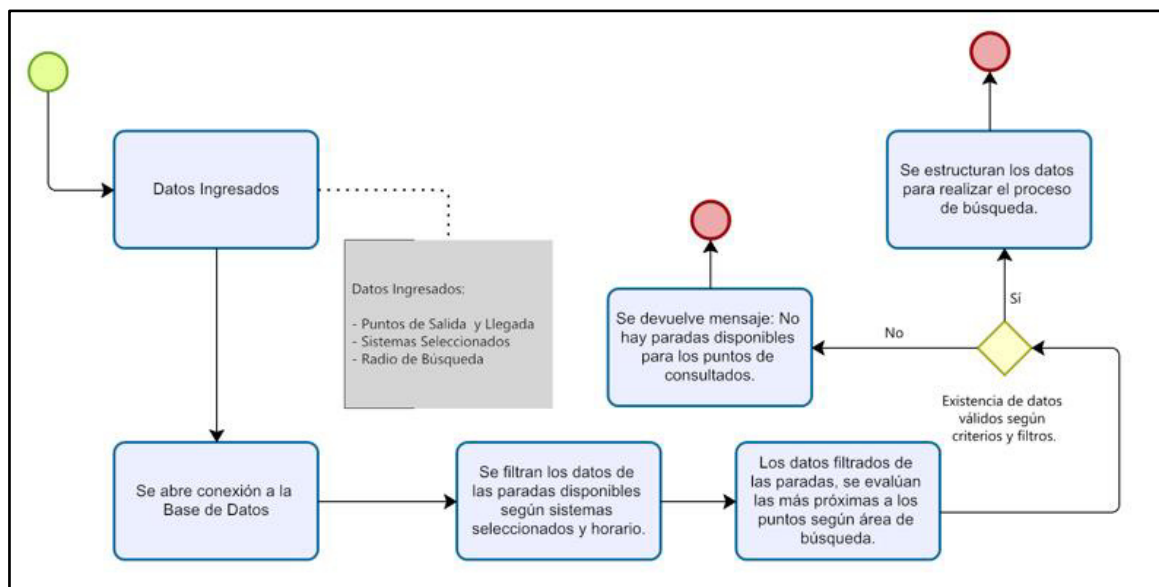
- Horario de Búsqueda de Ruta: Es necesario identificar la hora en la que se desea realizar la planificación del viaje, con el fin de buscar las rutas considerando los servicios de transporte disponibles según el horario. En el caso de la aplicación se toma como hora por defecto, el instante en que se hace la consulta de las posibles rutas óptimas.
- Sistemas de Transporte Público Regular: Este filtro permite precisar sobre si se desea realizar el análisis de búsqueda de rutas entre todos los servicios pertenecientes a los sistemas de transporte o solo en algunos según los sistemas seleccionados, siendo estos: Metro, Metropolitano, Corredores Complementarios. O si es que se desea usar los tres medios,

Criterios de Validación:

- Rango de Búsqueda: Este criterio permite identificar los paraderos o estaciones, según corresponda, pertenecientes a los sistemas previamente seleccionados y según los servicios disponibles considerando su horario, más cercanos a los puntos salida y llegada ingresados. Para la aplicación este criterio tiene un alcance por defecto de 50 metros de radio de búsqueda.

Figura 109*Procesos de Ingreso de Datos para Validación*

Nota. Elaboración propia

Figura 110*Procesos de Validación de Datos*

Nota. Elaboración propia

Tabla 27

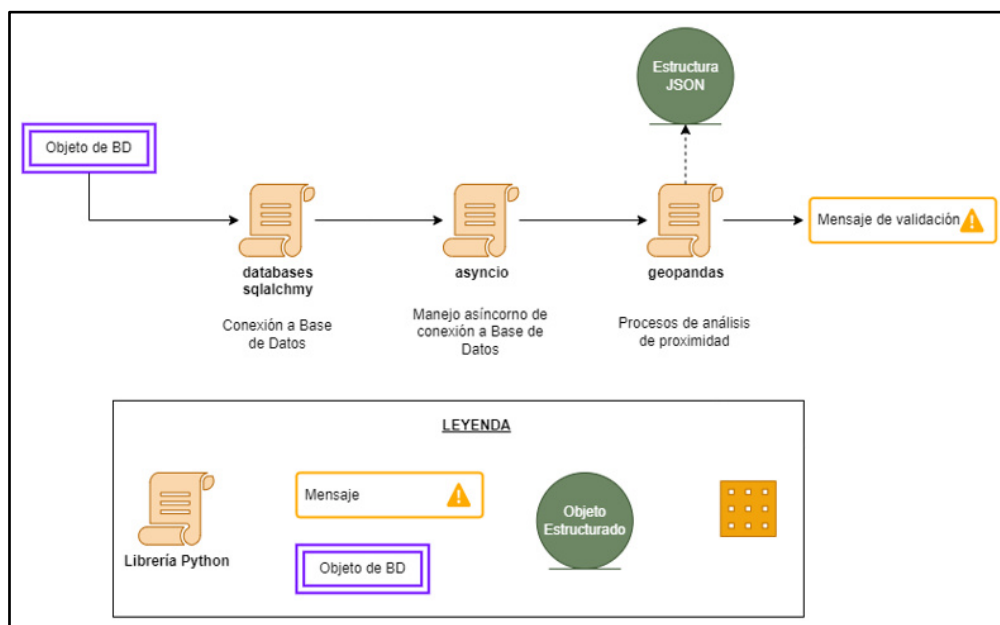
Secciones de la Validación de Datos

Ingreso	Salida	Requerimientos
Punto de Salida (Coordenadas)	Información Válida:	
Punto de Llegada (Coordenada)	- Estructura de Datos para proceder con el análisis de rutas.	
Sistemas de Transporte Seleccionados (Metro, Metropolitano, Corredores)	- Mensaje: “Inicio de Búsqueda de Rutas.”	Librerías Externas:
Área de Búsqueda de Paraderos o Estaciones por defecto 50 metros.	Información Inválida:	<ul style="list-style-type: none"> • databases • sqlalchemy • asyncio • geopandas
Horario de Búsqueda, por defecto es determinado por el momento en que se hace la ejecución del análisis.	- Mensaje: “No hay estaciones y/o paraderos próximos a los puntos consultados”	

Nota. Elaboración propia

Figura 111

Componentes de la Validación de Datos



Nota. Elaboración propia

2.7.10.2. Cálculo de Rutas

Con los datos de entrada validados se procede a realizar el cálculo de rutas óptimas. Para esto es necesario usar los objetos de base de datos determinados como nodos y arcos para estructurar el grafo, el cual se usará para ejecutar el proceso de cálculo mediante el uso de librerías que ya implementan el algoritmo de Dijkstra y que requieren un grafo para su ejecución.

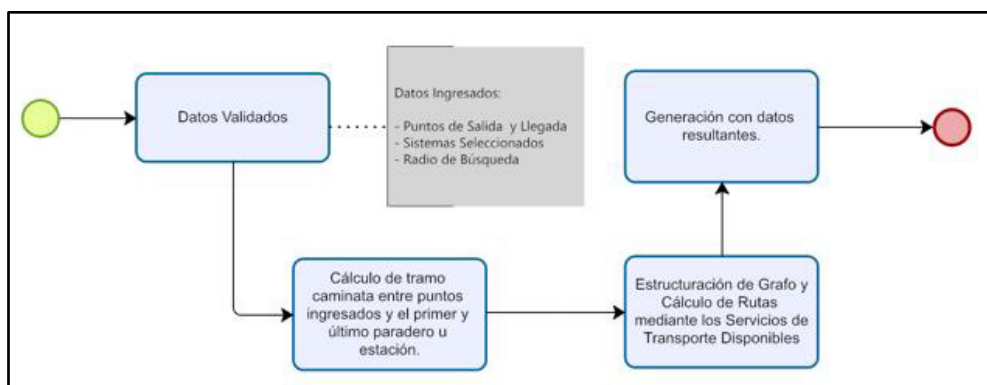
Y para complementar las rutas analizadas, se deberá determinar el tramo de caminata desde los puntos determinados como salida y llegada hacia el paradero o estación determinados como punto de abordaje inicial y desabordaje final, en caso de no existir se determinará una línea recta hacia estos.

Características del Grafo:

- Dirigido: Con arcos direccionales por los sentidos de los recorridos de las rutas, y bidireccionales para secciones de camina entre paraderos y/o estaciones.
- Ponderado: Con valores de tiempo asociados a los arcos, priorizando los arcos cuya estimación de tiempo sea menor.

Figura 112

Proceso de Análisis de Rutas



Nota. Elaboración propia

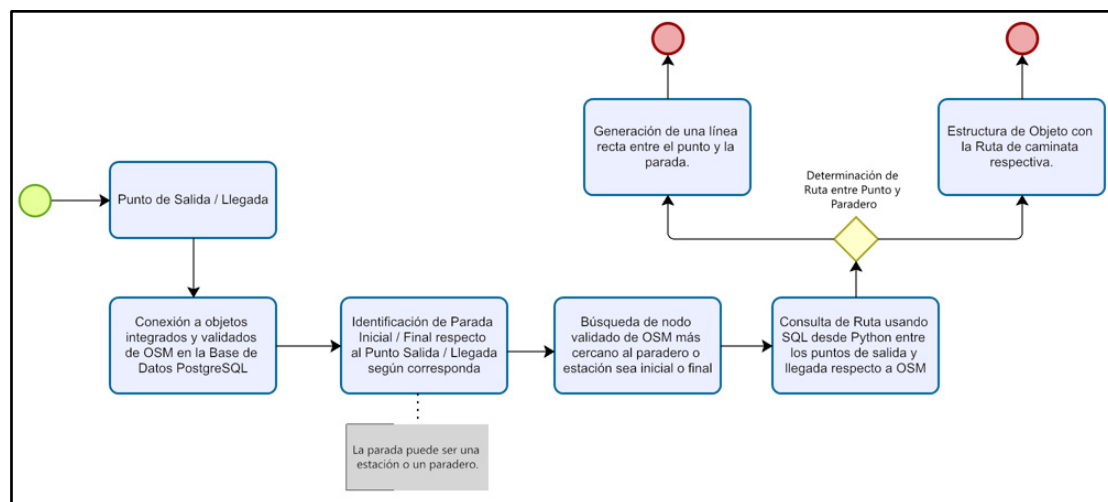
Cálculo del tramo de caminata inicial y final.

Este cálculo se encarga de determinar el tramo de caminata que debe seguirse desde la

ubicación de salida hasta la primera estación o paradero para iniciar el recorrido y del mismo modo para la ubicación del punto de llegada respecto a la última estación o paradero del recorrido. Este cálculo se hace de manera dinámica con los datos integrados y validados de OSM, pues al ser las ubicaciones de salida y llegada siempre diferentes no pueden estar pre definidas dentro de la estructura integral de los objetos definidos como nodos y arcos. En caso de no encontrar un camino para estos tramos se procederá a crear una línea recta entre el punto y su respectiva parada.

Figura 113

Proceso de Cálculo de Caminata para el Punto de Llegada y Salida



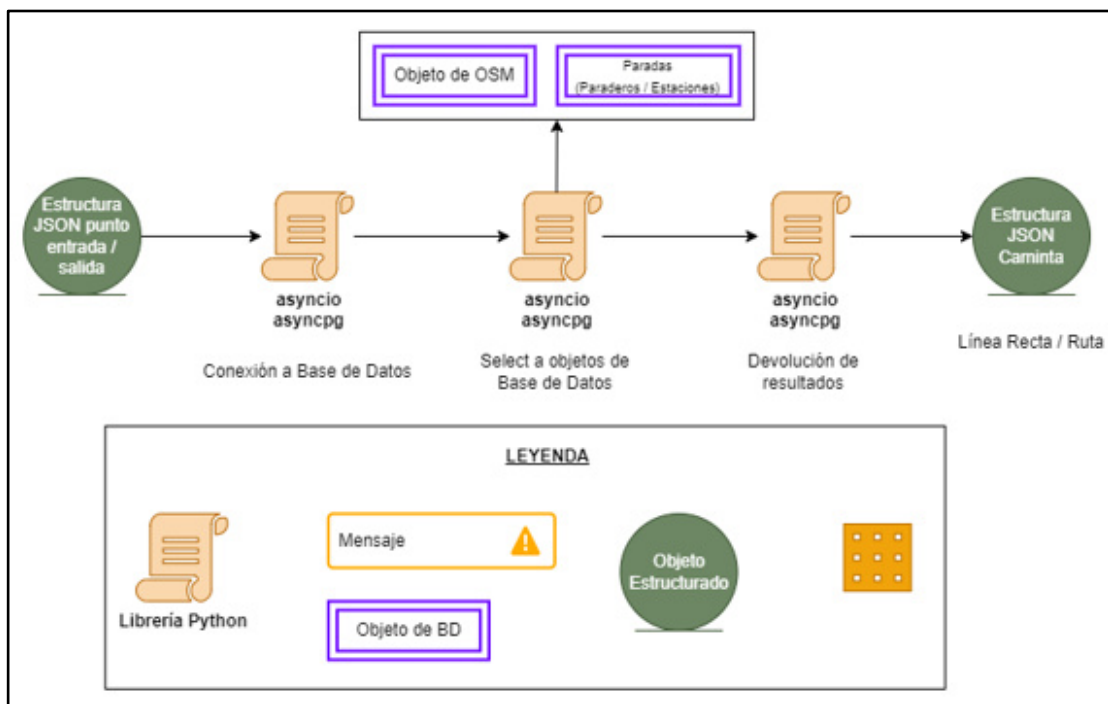
Nota. Elaboración propia

Tabla 28

Secciones del Cálculo de Caminata Inicial y Final

Ingreso	Salida	Requerimientos
Punto de Entrada / Salida.	El objeto resultante entre el punto de salida o llegada hacia el la parada inicial o final según corresponda, puede ser una de las siguientes opciones: <ul style="list-style-type: none"> - Ruta de caminata con los datos de OSM cargados a la BD. - Línea recta. 	Librerías Externas: <ul style="list-style-type: none"> • asyncio • asyncpg

Nota. Elaboración propia

Figura 114*Componentes del Cálculo de Caminata Inicial y Final*

Nota. Elaboración propia

Cálculo de Ruta entre la primera parada y la última usando el Grafo y Dijkstra.

Para realizar el cálculo de la ruta óptima mediante los servicios pertenecientes a los sistemas de transporte previamente seleccionados es necesario estructurar el grafo con los objetos de la base de datos denominados nodos y arcos en el esquema correspondiente, para lo cual se usarán librerías que permiten implementar de manera dinámica el algoritmo de Dijkstra. Tener en cuenta que los datos a nivel de segmentos ya se encuentran estructurados tanto a nivel de recorrido como a nivel de camina entre paraderos y/o estaciones, si fuese necesario integrarlo para determinar una ruta óptima usando múltiples servicios de transporte.

El grafo a estructurar debe ser dirigido y ponderado. Siendo los tramos de caminata bidireccionales por ser posibles de recorrer tanto de ida como de retorno, así como direccionales para los sentidos de los recorridos de los servicios de transporte disponibles por solo ser posible un recorrido sobre ellos. Y el valor de la ponderación de los arcos asociado al

tiempo de viaje sobre ellos según el tipo de servicio.

Figura 115

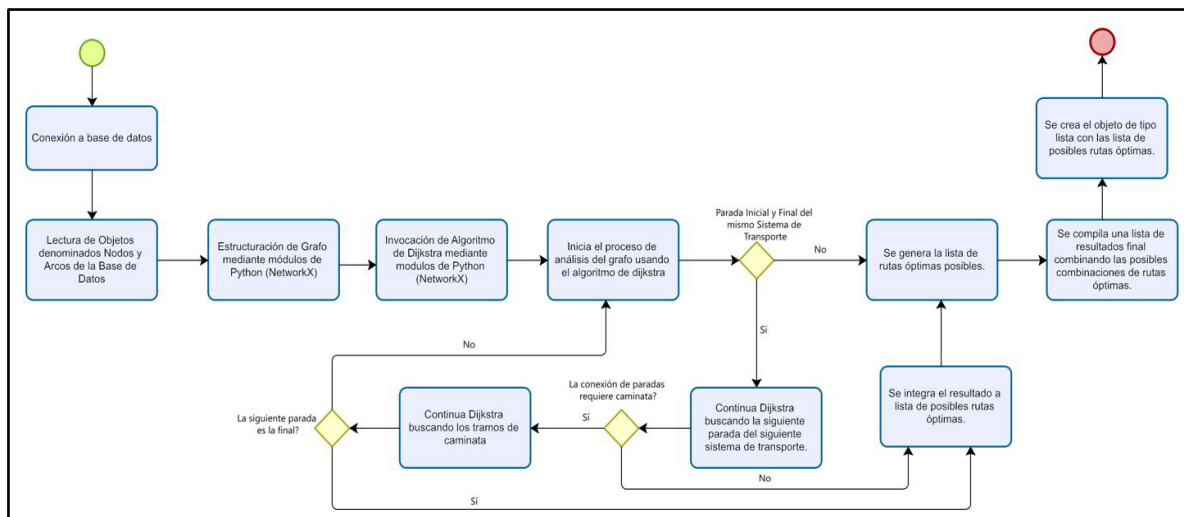
Ejemplo de Estructuración de Grafo



Nota. Elaboración propia

Figura 116

Proceso de Cálculo de Rutas



Nota. Elaboración propia

Tabla 29

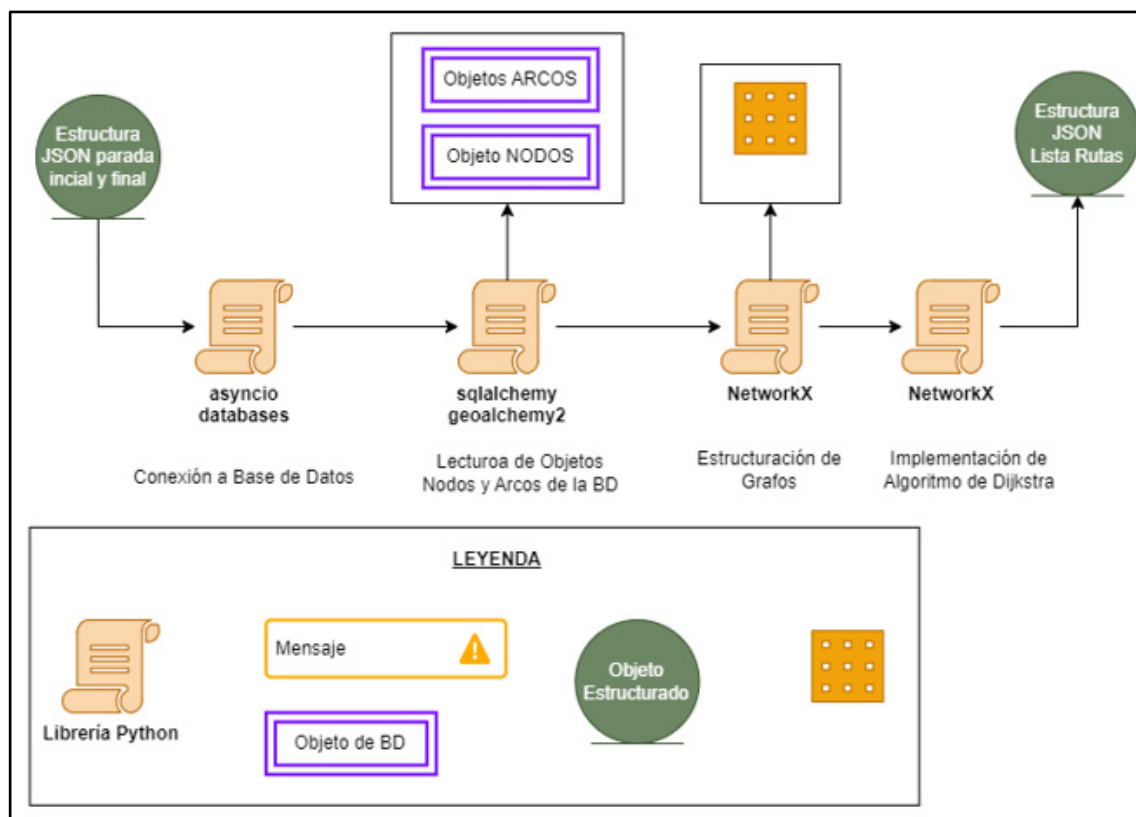
Secciones del Cálculo de Rutas

Ingreso	Salida	Requerimientos
Parada Inicial.		
Parada Final.		
Sistemas de Transporte (previamente seleccionados).	- Objeto tipo lista, que contiene las posibles rutas de viaje mediante el o los servicios de transporte incluido las secciones de caminata de ser el caso.	Librerías Externas: <ul style="list-style-type: none"> • asyncio • databases • sqlalchemy2 • NetworkX
Horario (siendo un valor por defecto determinado por el momento donde se inicia el análisis de cálculo de rutas)		

Nota. Elaboración propia

Figura 117

Componentes del Cálculo de Rutas



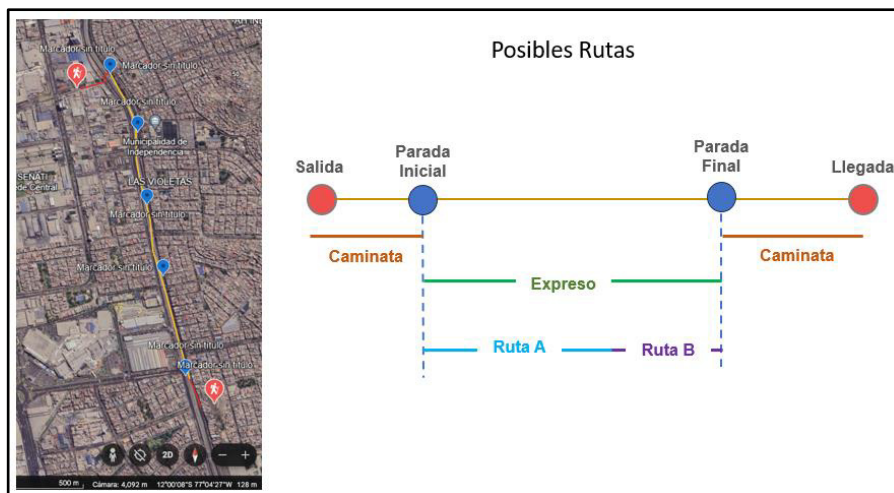
Nota. Elaboración propia

2.7.10.3. Estructuración de Resultados

Como se ha determinado en el punto anterior el proceso de cálculo de rutas genera dos objetos resultantes, la data de caminata que son las rutas calculadas directamente con OSM y la data de las posibles rutas óptimas. Por ende, esta sección del proceso de Análisis para el Planificador de Rutas se encarga de estructurar la información en un objeto que contenga las posibles rutas en modo de una lista y bajo un formato de GeoJson que incluya las coordenadas, así como la diferenciación por colores de los recorridos a realizar según lo servicios del sistema de transporte y tramos a caminar propuestos producto del análisis.

Figura 118

Ejemplo de Integración Caminata (Inicial y Final) con Posibles rutas Óptimas



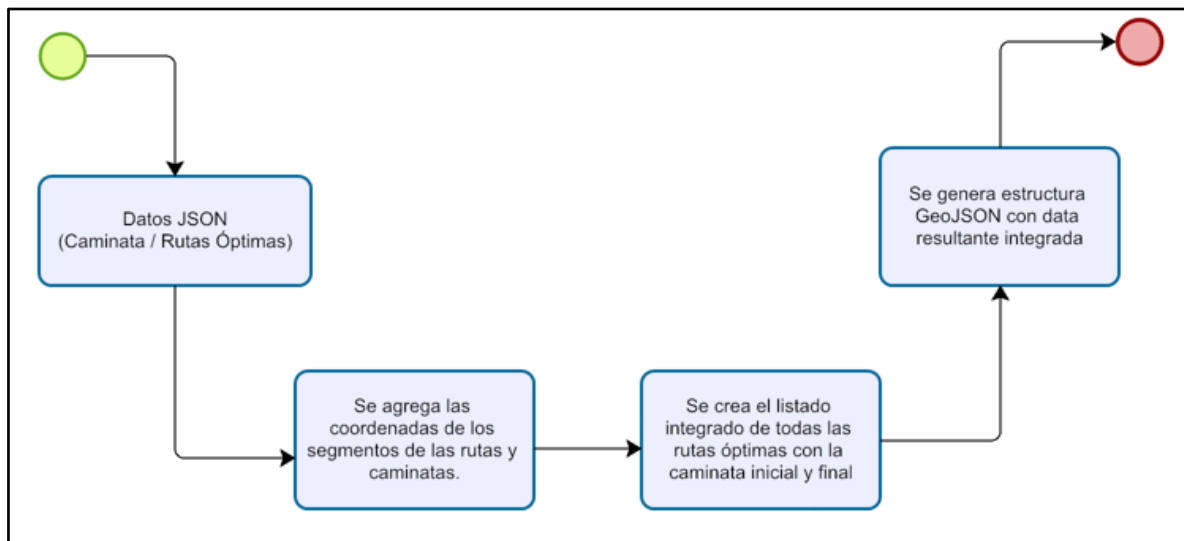
Nota. Elaboración propia

Tabla 30

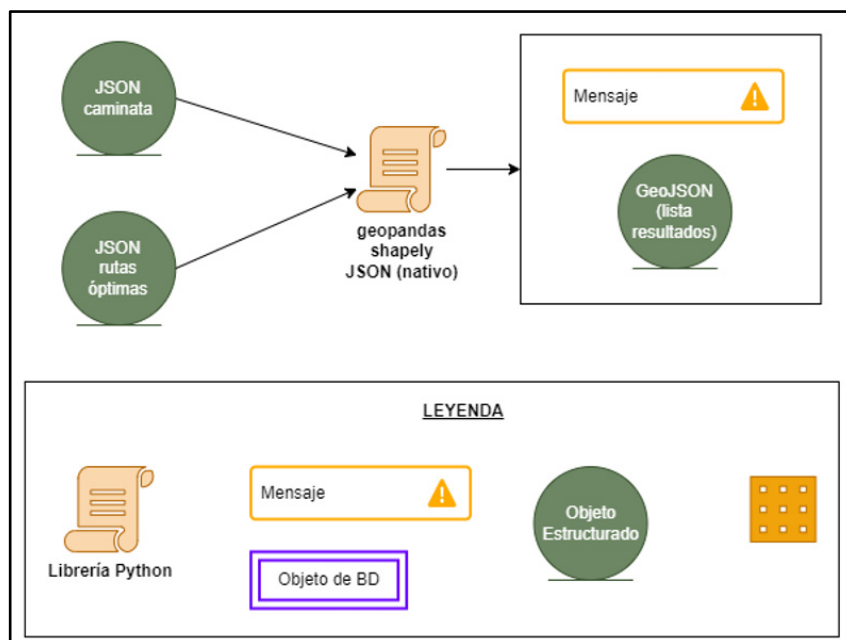
Secciones del Cálculo de Rutas

Ingreso	Salida	Requerimientos
<ul style="list-style-type: none"> - JSON Caminata Inicial - JSON Caminata Final - JSON Rutas Óptimas 	GeoJSON con resultados integrados como lista	Componente Nativo: <ul style="list-style-type: none"> • Json Librerías Externas: <ul style="list-style-type: none"> • GeoPandas • Shapely.geometry

Nota. Elaboración propia

Figura 119*Proceso de Integración de Resultados*

Nota. Elaboración propia

Figura 120*Componentes de Integración de Resultados*

Nota. Elaboración propia

Figura 121

Ejemplo de Integración Caminata (Inicial y Final) con Posibles rutas Óptimas

```

JSON

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [longitudo_salida, latitudo_salida],
          [longitudo_parada_inicial, latitudo_parada_inicial]
        ]
      },
      "properties": {
        "tipo": "caminata",
        "color": "brown" // Or any suitable color code for walking segment
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [longitudo_parada_inicial, latitudo_parada_inicial],
          [longitudo_cambio_ruta, latitudo_cambio_ruta]
        ]
      },
      "properties": {
        "tipo": "Expreso",
        "color": "green"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [longitudo_cambio_ruta, latitudo_cambio_ruta],
          [longitudo_parada_final, latitudo_parada_final]
        ]
      },
      "properties": {
        "tipo": "Ruta A",
        "color": "cyan"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [longitudo_parada_final, latitudo_parada_final],
          [longitudo_llegada, latitudo_llegada]
        ]
      },
      "properties": {
        "tipo": "caminata",
        "color": "brown"
      }
    }
  ]
}

```

Nota. Elaboración propia

2.7.11. Implementación de Herramienta de Planificación de Rutas

Para poder entregar la información a la aplicación con el fin de ser visualizada desde esta, se ha adoptado un estándar de intercambio de datos, en este caso GeoJSON. Para lo cual se ha implementado una API en Python que se integra con la tecnología de la aplicación, concretando así la herramienta de planificación de rutas.

2.7.11.1. Implementación de API Python

Una API, conocida por sus siglas en inglés Application Programming Interface (Interfaz de Programación de Aplicaciones) es un conjunto de reglas y definiciones que funciona como un puente de comunicación entre diferentes sistemas y/o aplicaciones. A través de la API, un desarrollador puede acceder a la funcionalidad de otro programa sin requerir conocer su código interno.

Figura 122

Representación de API

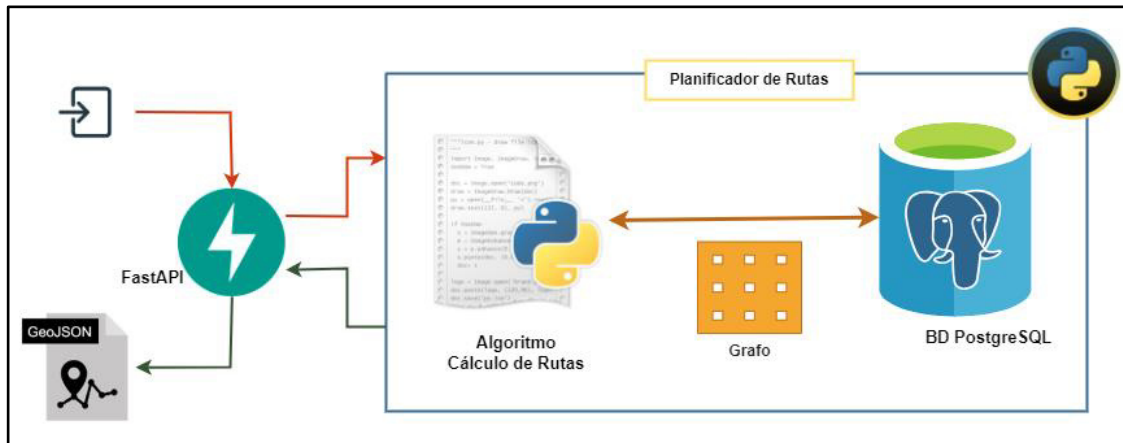


Nota. Recurso externo

Para realizar la implementación de la API de Python se ha hecho uso del framework FastAPI, el cual recibe los datos de la aplicación, seguido hace llamado a los algoritmos de generación de rutas y devuelve el objeto resultante denominado GeoJSON.

Figura 123

Diagrama Integración de FastAPI y Planificador



Nota. Elaboración propia

2.7.11.2. Integración de API Python a la Aplicación Móvil

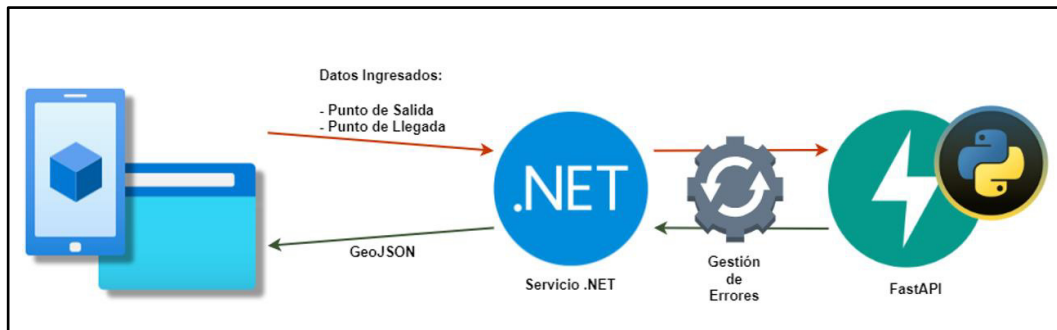
Para integrar la aplicación móvil con la herramienta de análisis de rutas, se hace la integración de FastAPI y la API de la aplicación móvil construida en .NET. Esta integración establece un puente entre ambas tecnologías para brindar los datos de entrada (ubicación final e inicial) al planificador de rutas desarrollado en Python. El análisis de ruta realizado en Python retorna un objeto GeoJSON para ser visualizado en la aplicación móvil mediante la API de .NET.

Es fundamental contar con un manejo eficiente de errores entre ambas tecnologías para gestionar situaciones como ubicaciones inválidas o problemas durante el proceso de cálculo de rutas, de modo que estas situaciones sean notificadas adecuadamente. La integración entre FastAPI y las APIs de .NET permite combinar las capacidades de ambos entornos para crear aplicaciones de análisis de rutas potentes y eficientes. FastAPI actúa como un servicio especializado en el cálculo de rutas, proporcionando resultados en un formato estándar que

puede ser fácilmente consumido y utilizado por las aplicaciones .NET.

Figura 124

Integración API FastAPI y .NET



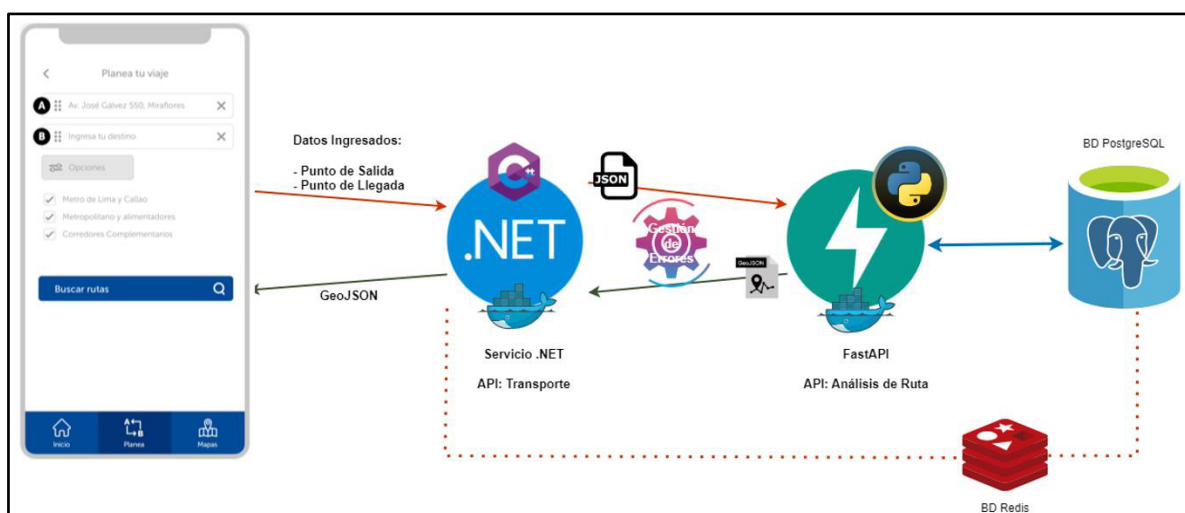
Nota. Elaboración propia

2.7.11.3. Herramienta de Rutas en la Aplicación Móvil

La herramienta de planificación de rutas en la aplicación está compuesta por la integración de las tecnologías con la cual ha sido desarrollada, con el fin de poder concretar su existencia en la aplicación móvil.

Figura 125

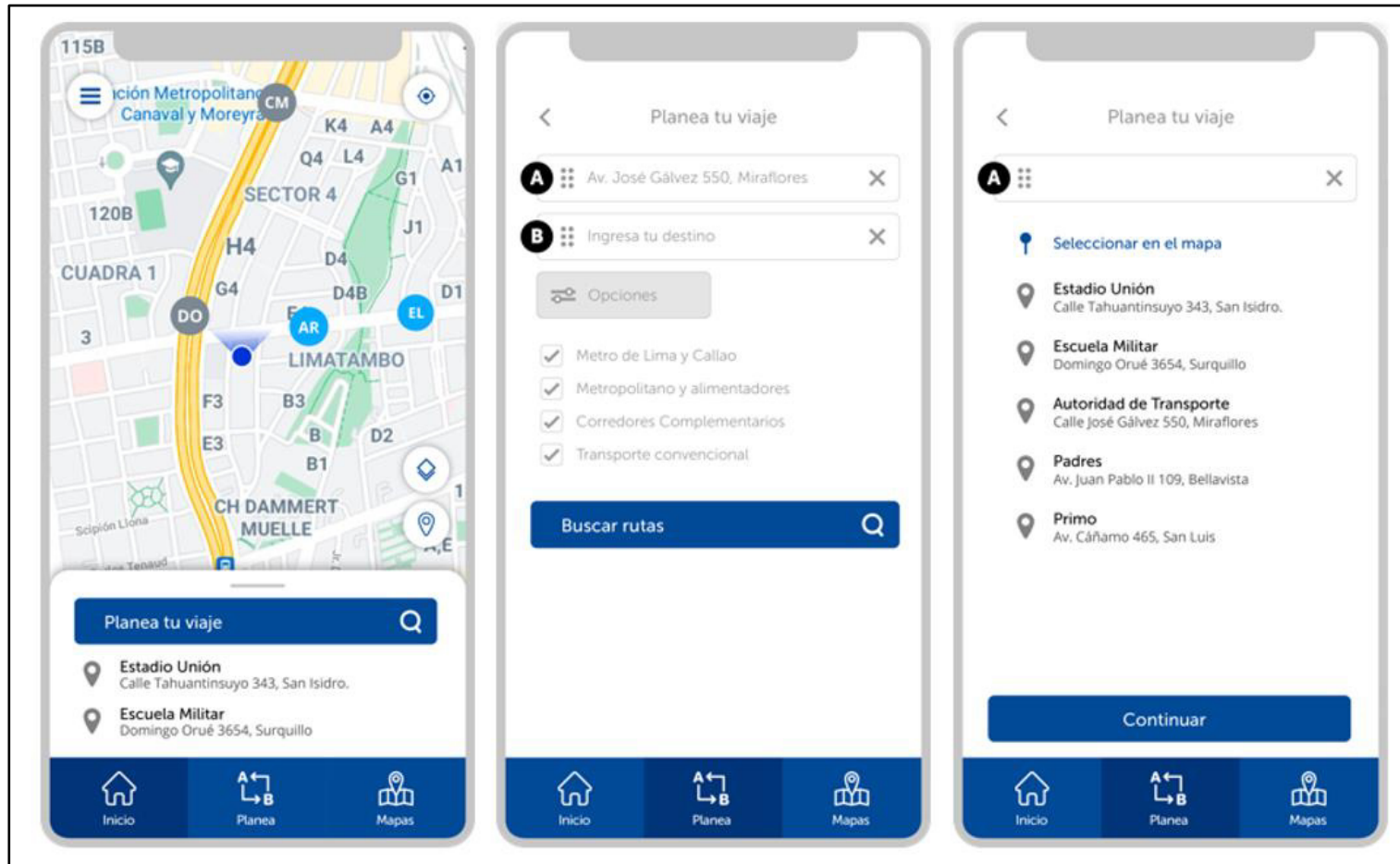
Diagrama Tecnología de Alto Nivel de Aplicación



Nota. Elaboración propia

Figura 126

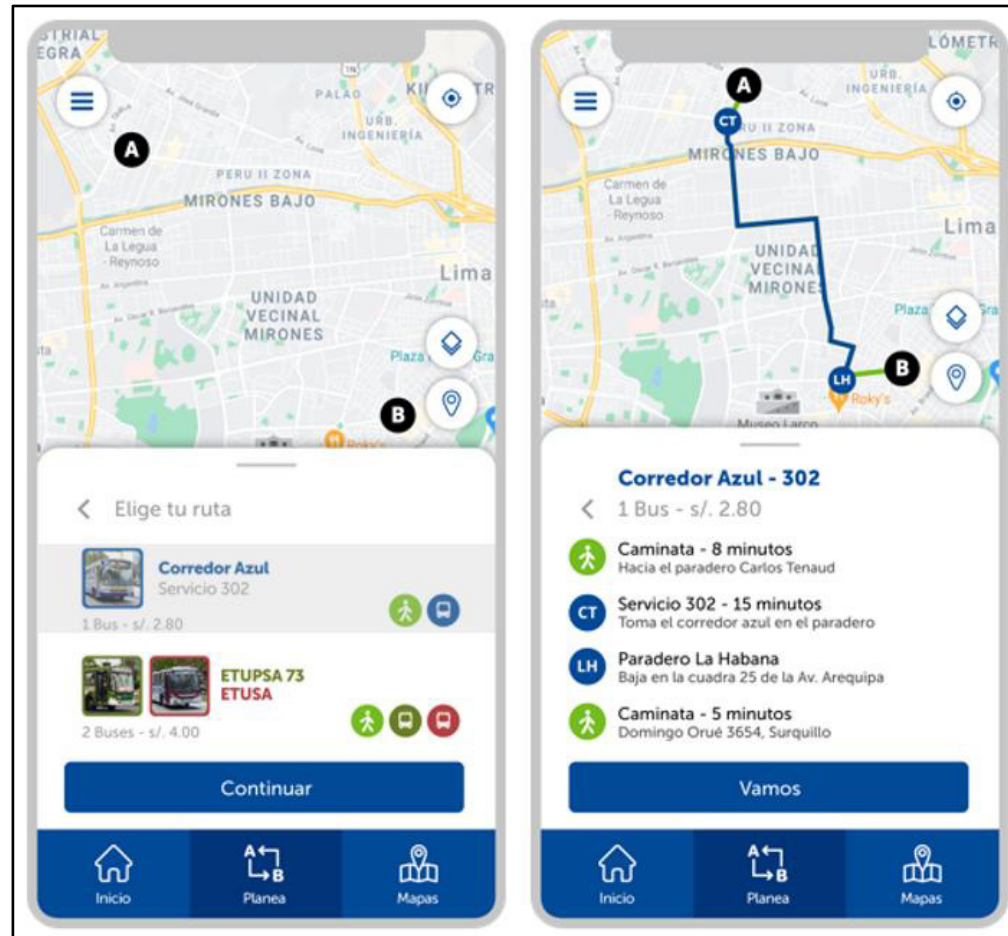
Aplicación: Planificador de Rutas (Ingreso de Datos)



Nota. Recurso externo

Figura 127

Aplicación: Planificador de Rutas (Visualización de Ruta)



Nota. Recurso externo

2.8. Resultados del Proyecto

Los resultados obtenidos tras el desarrollo de la herramienta de análisis de rutas, cuyo insumo son los datos del Sistema de Transporte Público Regular y OpenStreetMap, han sido satisfactorios. El algoritmo de Dijkstra, implementado a través de librería de código abierto, ha permitido generar búsquedas de rutas de manera eficiente, reduciendo significativamente los costos por adquisición de licencias, y ha sentado las bases para futuras aplicaciones, que requiera la entidad, respecto al manejo de rutas. Por ende, el desarrollo del proyecto contempla los objetivos planteados en el presente documento mediante la:

- Construcción de una Base de Datos escalable para el Sistema de Transporte Público.
- Desarrollo de código base para el procesamiento, análisis e integración de información georreferenciada.
- Implementación de una herramienta para la búsqueda de rutas usando los datos del Servicio de Transporte Público mediante Algoritmos de Dijkstra.
- Validación de la factibilidad de integración entre el buscador de rutas con aplicaciones digitales.

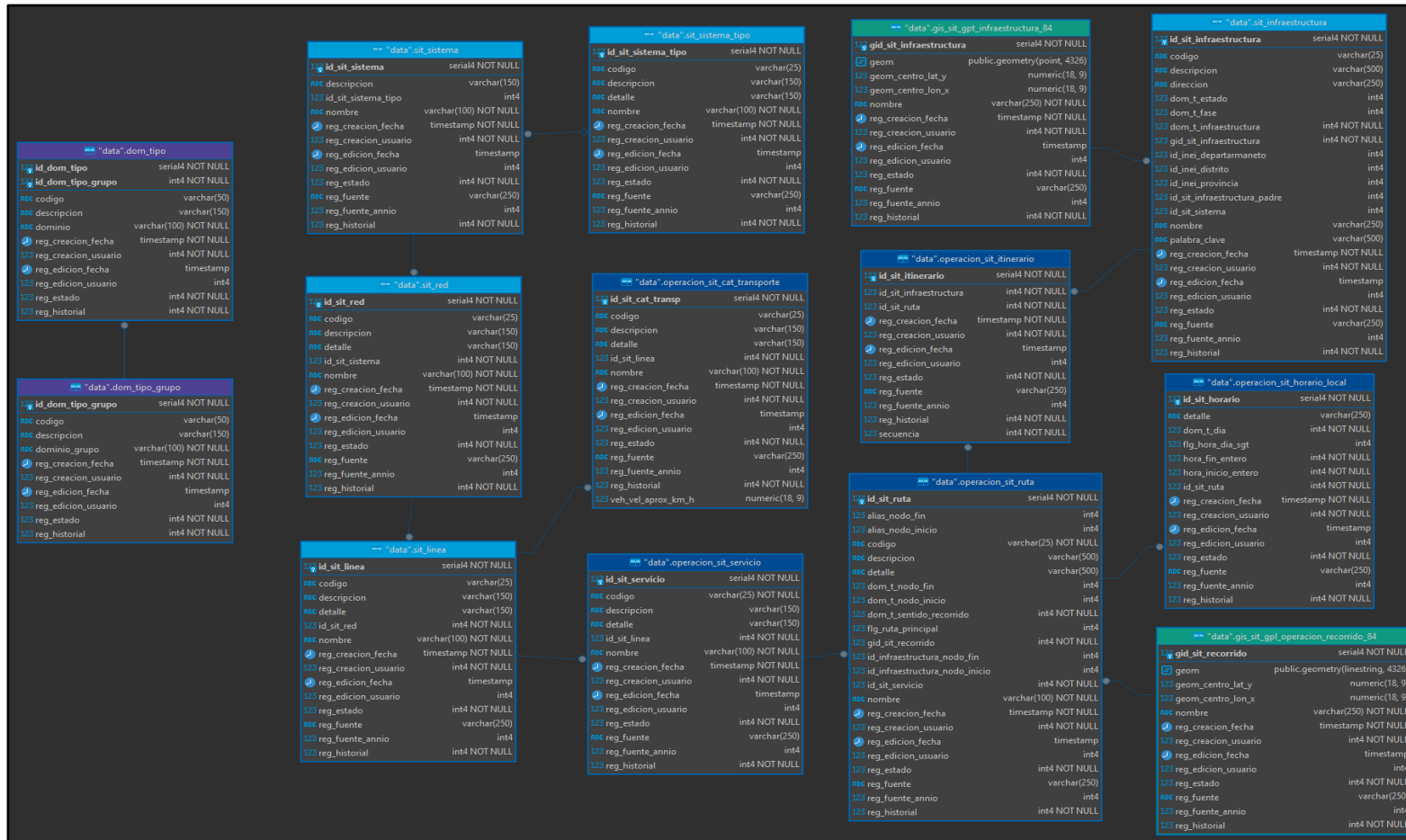
2.8.1. Estructura de Base de Datos para datos del Sistema de Transporte Público

Se ha estructurado el modelo de datos capaz de gestionar la información referente a los servicios de transporte público. Si bien tiene un alcance a nivel de servicios de operación este puede ser escalado para incorporar datos de flotas, proyectos, identificación de incidentes debido a su normalización y flexibilidad. El Modelo de Base de Datos para está compuesto por:

- Cinco (05) tablas de datos que manejan los datos del SIT.
- Cinco (05) tablas de datos que manejan los datos de los Servicios Operativos del SIT.
- Dos (02) tablas que manejan los datos georreferenciados del SIT.
- Dos (02) tablas que manejan la lista de valores reutilizables para el Modelo (dominios).

Figura 128

Modelo de Base de Datos para el SIT



Nota. Elaboración propia

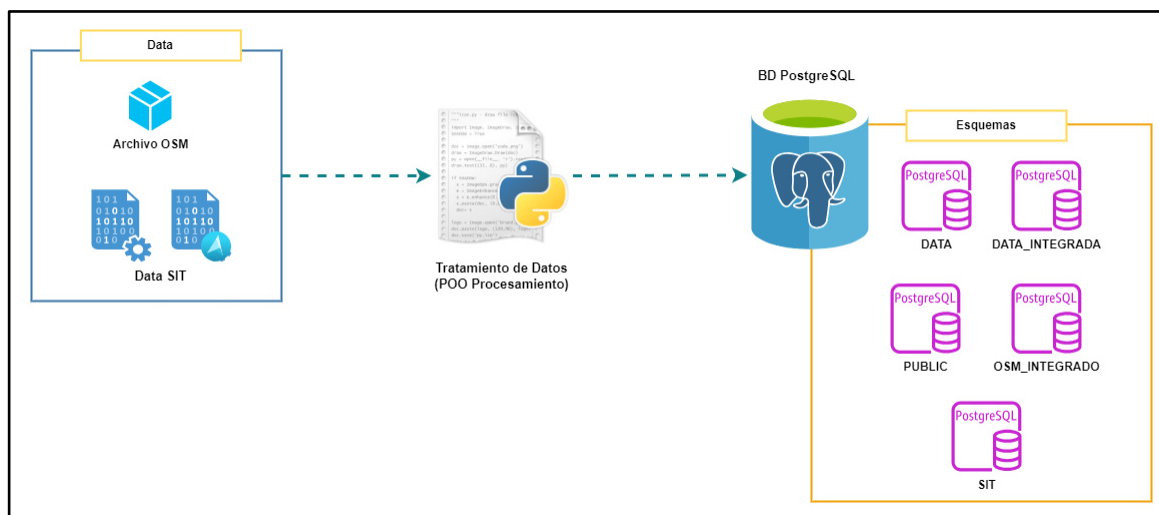
2.8.2. Código Base para los Procesos de Análisis de Datos

Se ha desarrollado scripts en Python, bajo un enfoque de Programación Orientada a Objetos, este enfoque permite hacer que los procesos de análisis puedan ser escalable y reutilizables. Como procesos automatizados se citan:

- OSM: Descarga de datos OSM desde la Web y bajo delimitación de cuadrantes.
- OSM: Tratamiento de archivos OSM para registro en BD PostgreSQL.
- KML/KMZ: Compilacion y migración de datos KML y KMZ a GeoPackage.
- Capa de Puntos: Lectura de archivo de puntos (shape, geopackag, json) y validación de proximidad según tipo de SIT integrando la data en archivo GeoPackage.
- Capa de Polilíneas: Lectura de archivo de polilíneas (shape, geopackag, json) y validación de sobreposición de trayectorias empalmando tramos según los recorridos de los servicios del SIT para integrar la data en archivo GeoPackage.

Figura 129

Flujo de Data y Archivo de Procesamiento de Datos



Nota. Elaboración propia

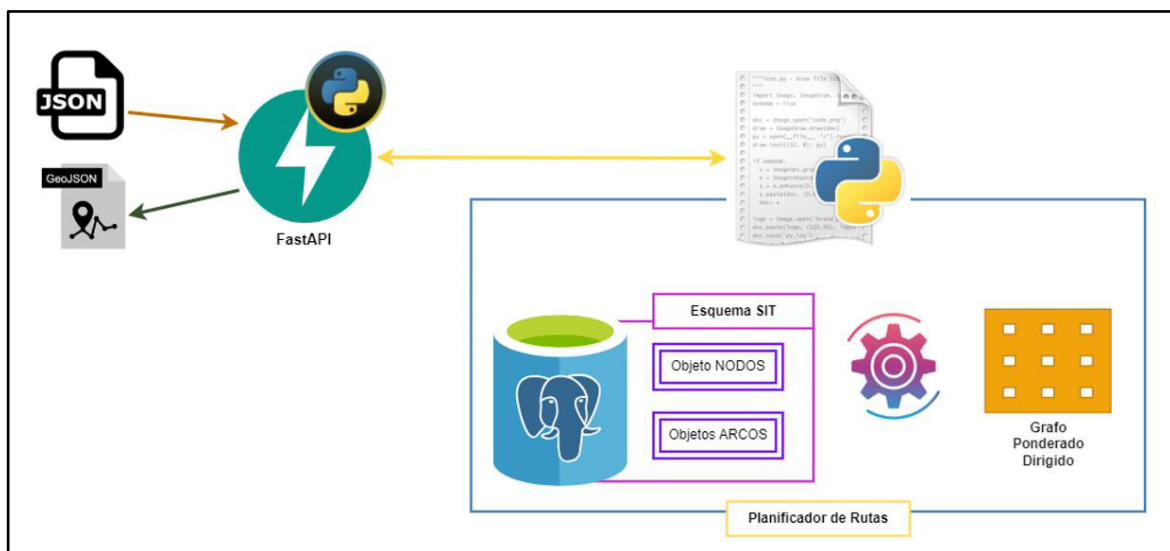
2.8.3. Herramienta para Planificación de Rutas usando el Servicio de Transporte Público

Se ha construido un planificador de rutas usando los algoritmos de Dijkstra, implementados mediante librerías de Python. Como insumo el planificador de rutas tiene la data de los Servicios de Transporte Público Regular y para las secciones de caminata la data de OSM (Open Street Map). Con la data de insumo, se construye dos tablas en base de datos denominadas Nodos y Arcos, los cuales son el insumo para que Dijkstra implementado con Python pueda construir un grafo ponderado dirigido.

Para que la herramienta sea compatible con otras tecnologías se vale del uso de API, siendo el framework FastAPI usado para la implementación de un Rest API por el cual recibo los parámetros de ingreso (punto de salida y llegada) y devuelve la lista de posibles rutas óptimas o mensajes de error en caso de no determinar recorrido. Esto permite que la herramienta sea reutilizable para otros proyectos.

Figura 130

Flujo FastAPI y Análisis de Rutas



Nota. Elaboración propia

2.8.4. Integración de búsqueda de rutas con Aplicación Móvil desarrollada por la Entidad

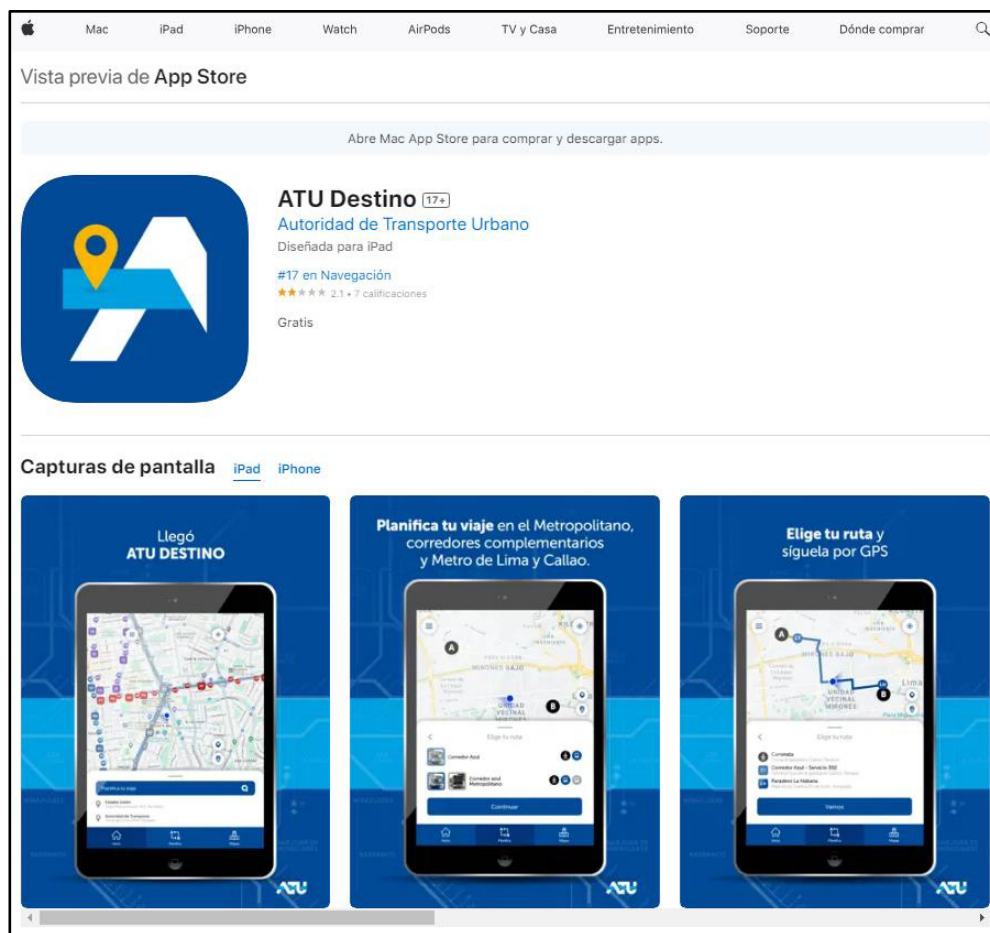
La herramienta desarrollada forma parte de la aplicación creada por la entidad, cuyo fin principal es permitir al usuario planificar un itinerario de viaje desde un origen hacia un destino usando los servicios de transporte público regular.

Aplicación construida por la entidad:

La aplicación móvil construida por la entidad, se lanzó como una versión Beta para funcionar en IOS & Android con el nombre ATU Destino. Se lanzó con esta versión con el fin de capturar las recomendaciones de mejoras de los usuarios.

Figura 131

Aplicación ATU Destino: Plataforma IOS Apple Store



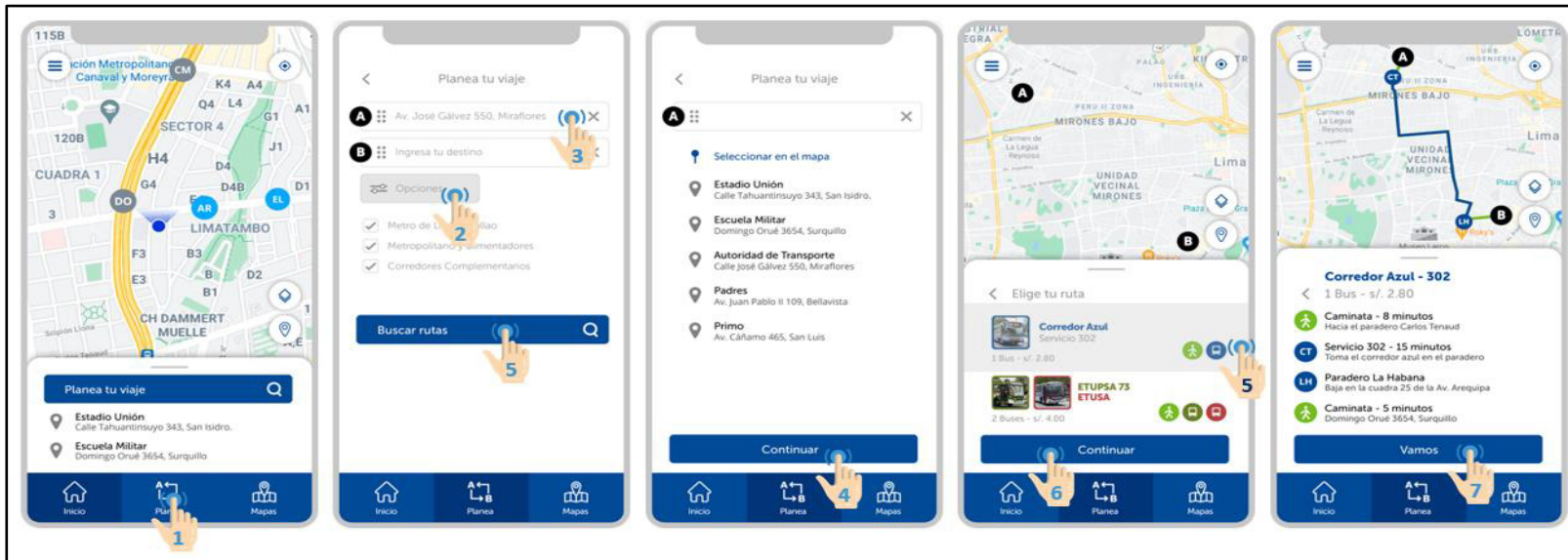
Nota. Recurso externo (<https://apps.apple.com/pe/app/atu-destino/id6462346173>)

Herramienta de Planificación de Rutas en la Aplicación:

La aplicación usa mediante protocolos internos, propios de la arquitectura tecnológica, la herramienta construida en base al algoritmo de Dijkstra para la identificación de itinerarios usando los servicios de transporte de carácter público. Considerar que esta herramienta busca paraderos del sistema regular de transporte a un radio de 50 metros tanto para el punto de salida como también para el punto de llegada, con el fin de brindar una adecuada experiencia de usuario sobre la aplicación.

Figura 132

Aplicación: Búsqueda de Rutas



Nota. Recurso externo

III. APORTES MÁS DESTACABLES A LA INSTITUCIÓN

A la fecha vengo colaborando para la ATU, desde la Dirección de Supervisión de Proyectos. Sin embargo, a la fecha se han realizado diferentes tipos de aportes, entre los cuales se puede destacar:

- Arquitectura: Implementación de una infraestructura empresarial para la Tecnología GIS de la Entidad.
- Proceso: Flujo de trabajo para el registro y visualización de información cartográfica desde la Base de Datos GIS hacia las aplicaciones Web GIS.
- Base de Datos: Generación de base de datos para el Sistema Integrado de Transporte.
- Base de Datos: Modelamiento y generación de modelo de datos para el seguimiento de proyectos referentes al Sistema Integrado de Transporte.
- Análisis: Generación de herramientas para el procesamiento y carga de información masiva hacia los repositorios de datos de la Dirección de Supervisión de Proyectos.
- Análisis: Generación de herramienta para la descarga de datos vectoriales de Servicios GIS de ArcGIS.
- Análisis: Generación de herramientas para la descarga de los Mapas Base de Open Street Map.
- Aplicación: Aplicaciones GIS para la consulta de información del Sistema Integrado de Transporte y sus proyectos.

La herramienta desarrollada beneficia directamente a la aplicación generada por la entidad:

- La aplicación permite consultar rutas óptimas mediante los servicios del sistema transporte público regular, mediante la integración de la herramienta desarrollada.

IV. CONCLUSIONES

- Se integraron y estructuraron los datos cartográficos del Sistema Integrado de Transporte y del Mapa Mundial de Calles Open Street Map para optimizar los procesos de análisis para la consulta de rutas de viaje.
- Se implementó el análisis de rutas usando el algoritmo de Dijkstra, el cual tiene su fundamento en la teoría de grafos. Es importante saber que el algoritmo de Dijkstra se encuentra como módulos y/o librerías gratuitas disponibles para Python y PostgreSQL.
- Con la implementación del análisis de rutas mediante tecnología de licenciamiento gratuito se evitó incurrir en gastos de licencias, teniendo en cuenta que las herramientas de Google facturan según la cantidad de veces que hagan uso de ellas.
- La implementación del algoritmo se encuentra en el lado del servidor, si bien forma parte de la arquitectura de la aplicación requerida de la entidad, esta herramienta puede ser reutilizada para otras aplicaciones con requerimientos similares.
- Es posible implementar generadores de rutas con información cartográfica propia, debidamente estructura, sin necesidad de incurrir en costos de licenciamientos de software.

V. RECOMENDACIONES

- Se recomienda integrar información del Transporte Regular Convencional para ampliar los análisis de rutas sobre la red de servicios de microbuses, minibuses y ómnibus, considerando que la herramienta implementada realiza el análisis consultando información del Transporte Regular de carácter Concesionado (Metropolitano, Corredores Complementarios, Línea 01 del Metro).
- Para mejorar la información de calles, se recomienda realizar actualizaciones detalladas sobre la data vectorial descargada de Open Street Maps.
- Para optimizar la herramienta de búsqueda de rutas, se recomienda integrar información de incidencias para identificar las rutas que deben ser omitidas, durante el proceso de análisis, por presentar algún tipo de incidencia.
- Se recomienda mantener la información cartográfica centralizada y estructurada en base de datos con capacidades para tratar datos georreferenciados, como PostgreSQL, con el fin de facilitar su reutilización e integración con aplicaciones.
- Con el fin de mantener los datos actualizados, se recomienda generar directivas que indiquen que las áreas encargadas de la información remitan los datos vectoriales de los servicios de transporte cada vez que haya actualizaciones en los mismos, o construir plataformas que permitan la actualización automática de los mismos,
- En caso de requerir expandir el uso de la herramienta de análisis, a un alcance masivo y de carácter público, se recomienda destinar recursos informáticos al repositorio con los datos del Sistema de Transporte Público, Servidor de la Base de Datos, y al Servidor donde se realiza la ejecución de la Herramienta de Análisis.
- Se recomienda usar la tecnología de Python para generar Servicios Rest API, y no solo para el procesamiento de datos, para mejorar el desempeño de las plataformas tecnológicas a implementar en la Entidad.

VI. REFERENCIAS

- Bueno, S., Alfonso M. y Núñez, J. (2004). Siete puentes, un camino: Königsberg. Revista sobre Enseñanza y Aprendizaje de las Matemáticas. *SUMA*, 2004, 69-78.
<https://revistasuma.fespm.es/sites/revistasuma.fespm.es/IMG/pdf/45/069-078.pdf>
- Cardozo, O. D., Gómez, E. L., & Parras, M. A. (1). Teoría de Grafos y Sistemas de Información Geográfica aplicados al Transporte Público de Pasajeros en Resistencia (Argentina). *Revista Transporte Y Territorio*, (1), 89-111.
<https://doi.org/10.34096/rtt.i1.223>
- Germán, C. (2002). *Una Introducción Teoría de Grafos*. Universidad Pedagógica Nacional de Bogotá.
<https://core.ac.uk/outputs/33252609/>
- Mikiewicz, D., Mackiewicz, M. y Nycz, T. (2017). *Mastering PostGIS: Crear, Entregar y Consumir datos espaciales utilizando PostGIS. Geospatial Analyst*. Packt Publishing.
<https://github.com/PacktPublishing/Mastering-PostGIS>
- Sánchez, G. y Lozano, V. (2001). Algoritmo de Dijkstra Un Tutorial Interactivo. *VII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2001)*. Universidad de las Islas Baleareas.
<https://rua.ua.es/dspace/handle/10045/128198>

VII. ANEXOS

- Anexo A: Grado Académico de Bachiller
- Anexo B: Vistas de la Aplicación
- Anexo C: Diagrama de Tecnológica de la Aplicación

Anexo A
(Grado Académico de Bachiller)

Republica



del Perú

A nombre de la Nación

El Rector de la Universidad Nacional "Federico Villarreal"

Por cuanto: El Consejo de Facultad de Ingeniería Geográfica, Ambiental y
Ecoturismo con fecha 06 de Febrero del 2017 ha aprobado

el otorgamiento del Grado de Bachiller en
Ingeniería Geográfica

A André Volentín Roque Huamán



Y, el Consejo Universitario con fecha 08 de Mayo del 2017 le ha conferido el Grado correspondiente.

Por tanto: le expido el presente Diploma para que se le reconozca como tal.

Dado en la ciudad de Lima, a los 08 días del mes de Mayo del 2017



INTERESADO



Registrado a fojas 227 del Libro 150 respectivo con el No. 17650

UNIVERSIDAD NACIONAL FEDERICO VILLARREAL

CÓD. UNIV. 021

CONFERIDO POR R.R. N°.....*734*.....-2017-CU-UNFV FECHA: *28-03-17*
TIPO DE DOCUMENTO: *DMZ* N° *76657878* ABREVIATURA G/T: *B*
MODALIDAD DE ESTUDIOS: *P* TIPO DE EMISIÓN DE DIPLOMA: *P*
MODALIDAD DE OBTENCIÓN: *Autómata*

NOMBRES DE AUTORIDADES:

RECTOR : Dr. JUAN OSWALDO ALFARO BERNEDO
SECRETARIO GENERAL : Abog. RAMIRO IGNACIO RUIZ ALMEIDA
DECANO(A) : *Dr. Juan W. Gómez Lora*

Abog. HENRY RONALDO RAVELO HUERTA
JEFE DE LA OFICINA DE GRADOS Y TÍTULOS UNFV



N° 100251 01 - 064 - 8500
21 - 04 - 017



Anexo B
(Vistas de Aplicación)

Anexo B: Vistas de Aplicación

Adicional a la herramienta para la planificación de rutas, la aplicación ofrece opciones adicionales de carácter relevante para la facilidad de viaje por parte de los usuarios de los servicios de transporte público regular.

Entre las opciones adicionales ofrecidas por la aplicación se tiene:

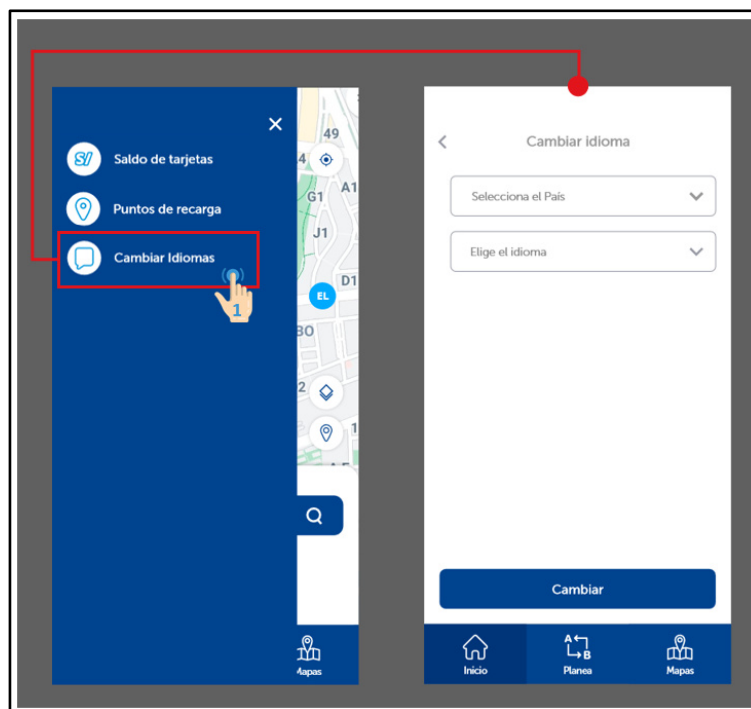
- Cambio de Idioma.
- Consulta de Saldo de Tarjeta.
- Ubicación de Puntos de Recarga de Tarjeta.
- Mapeo de Paraderos y Servicios de Transporte.
- Visualización de Hora de Llegada de Buses.

Es importante mencionar, que la información mostrada en estas secciones se nutre de la misma base de datos construida, lo que demuestra su flexibilidad y potencial para futuras expansiones. Sin embargo, algunas de estas secciones no han sido documentadas en este proyecto por no formar parte del alcance del mismo.

Herramienta de Cambio de Idioma:

La aplicación es de carácter público, y si bien está enfocada a usuarios propios de la metrópoli de Lima y Callao, no desestima que sea útil para usuarios turísticos. Por lo mismo, es de conveniencia para los mismos, contar con una opción que les permita consultar las funcionalidades de la aplicación en otros idiomas como el inglés.

Aplicación: Cambio de Idioma

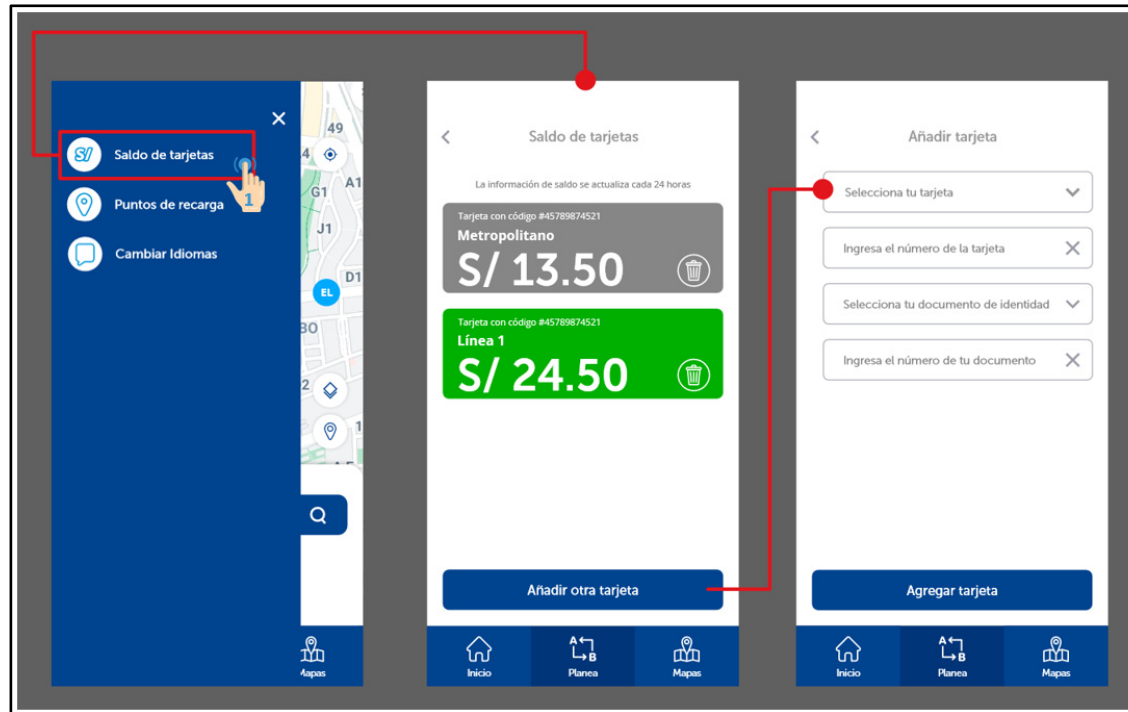


Nota. Recurso aplicación ATU Destino

Herramienta de Consulta de Saldo de Tarjeta:

Esta opción permite consultar el saldo de las tarjetas usadas para los servicios del Metropolitano y de la Línea 1 del Metro de Lima y Callao. Considerando que el saldo visualizado desde la opción tiene una latencia de actualización de 24 horas debido a procesos técnicos de integración.

Aplicación: Consulta de Saldo de Tarjeta

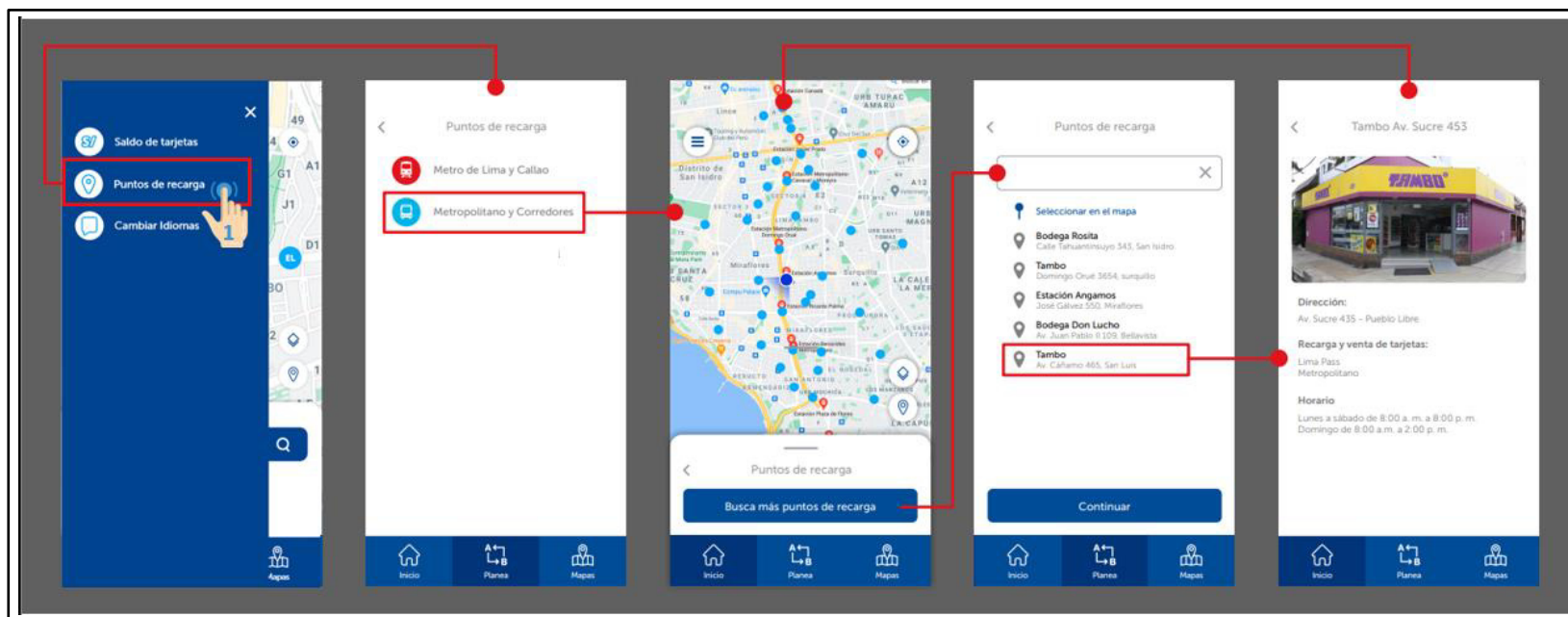


Nota. Recurso aplicación ATU Destino

Herramienta para visualizar la Ubicación de los Puntos de Recarga de Tarjeta:

La aplicación permite ubicar los puntos de recarga de las tarjetas usadas en el Metropolitano como el Metro de la Línea 01 de Lima y Callao, para ello se ha recolectado la información de los lugares de recarga asociados a cada tarjeta brindando información de su ubicación y horarios de funcionamiento. Esta información es de carácter referencial pues los lugares son actualizados de manera mensual y no presentan datos en tiempo real, por lo cual un establecimiento puede estar en funcionamiento o no.

Aplicación: Ubicación de Puntos de Recarga

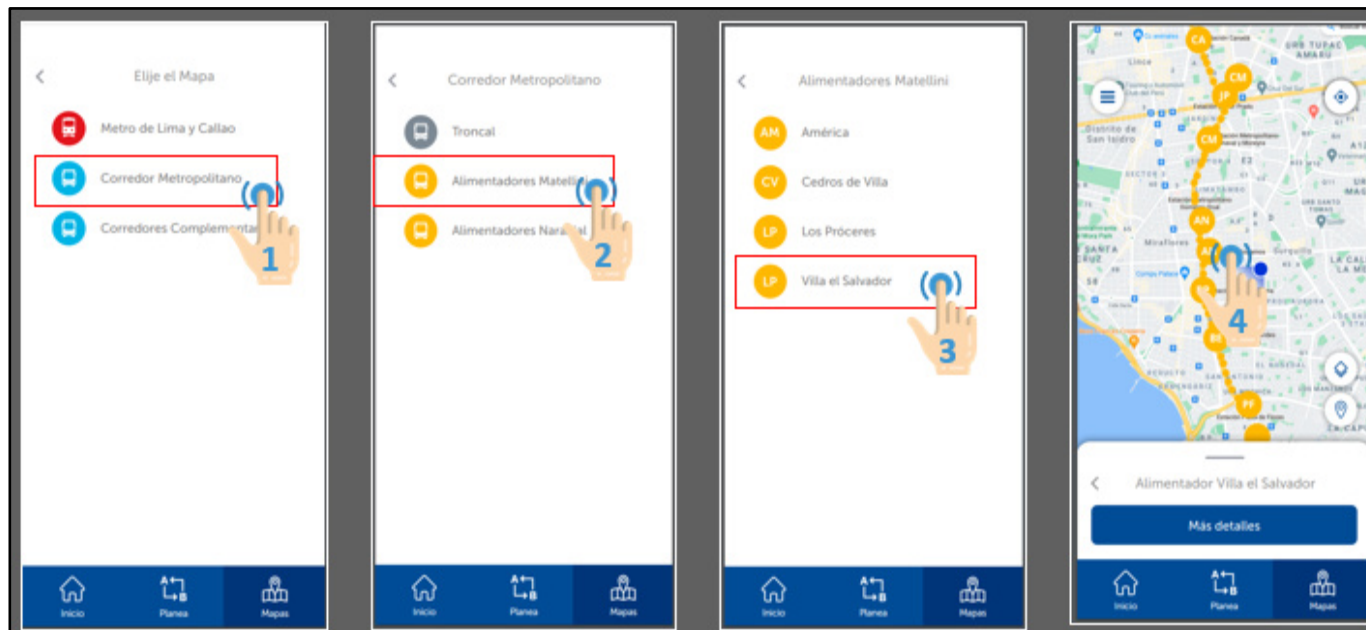


Nota. Recurso aplicación ATU Destino

Herramienta para el Mapeo de Paraderos y Servicios de Transporte:

Esta funcionalidad evidencia la aplicación de los GIS en las plataformas digitales pues permite consultar los recorridos y respectivas paradas de los servicios de transporte mediante el uso de mapas web. Esta funcionalidad evidencia el uso básico de los GIS, pues permite conocer el detalle vectorial de las entidades georreferenciadas facilitando la visualización de las paradas de los servicios de transporte, así como el recorrido de los mismos.

Aplicación: Ubicación de Puntos de Recarga

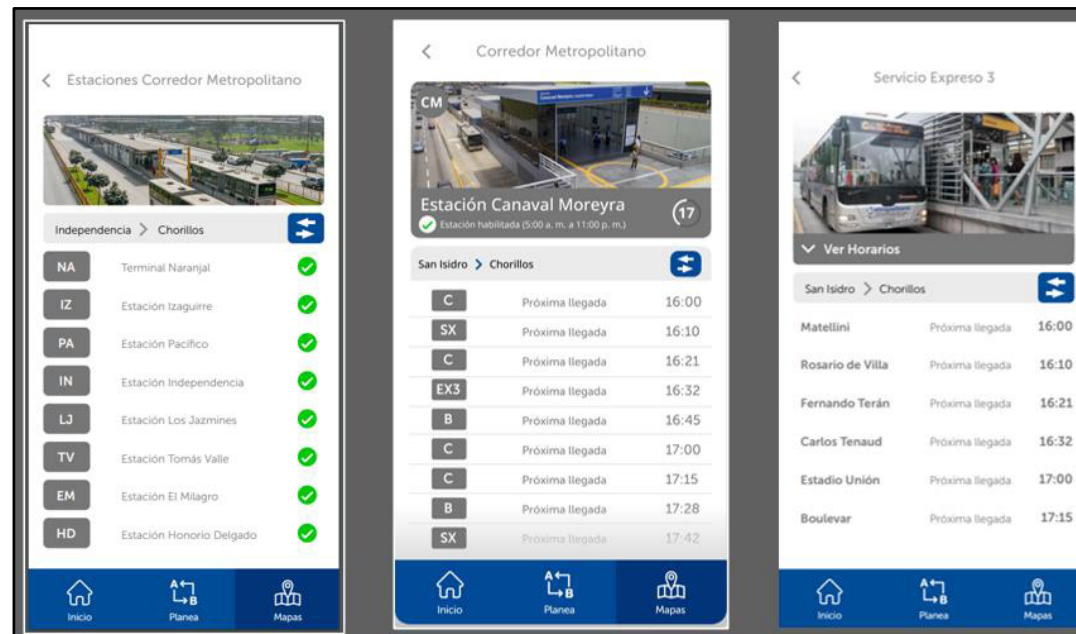


Nota. Recurso aplicación ATU Destino

Herramienta para la Visualización de Hora de Llegada de Buses:

Esta funcionalidad permite evidenciar otro gran aporte de los GIS integrado con las TIC. Pues, mediante esta opción se puede conocer las horas de llegada a cada estación o paradero de los vehículos que brindan servicio de transporte, así como la ubicación precisa de los mismos, sea para el Metropolitano como para los vagones de metro de la Línea 1. El cálculo de horas de llegada se hace a partir de cálculos internos relacionados a la velocidad y ubicación de los vehículos, producto de obtener su ubicación en tiempo real mediante dispositivos de telemetría.

Aplicación: Ubicación de Puntos de Recarga



Nota. Recurso aplicación ATU Destino

Anexo C

(Diagrama de Tecnología de la Aplicación)

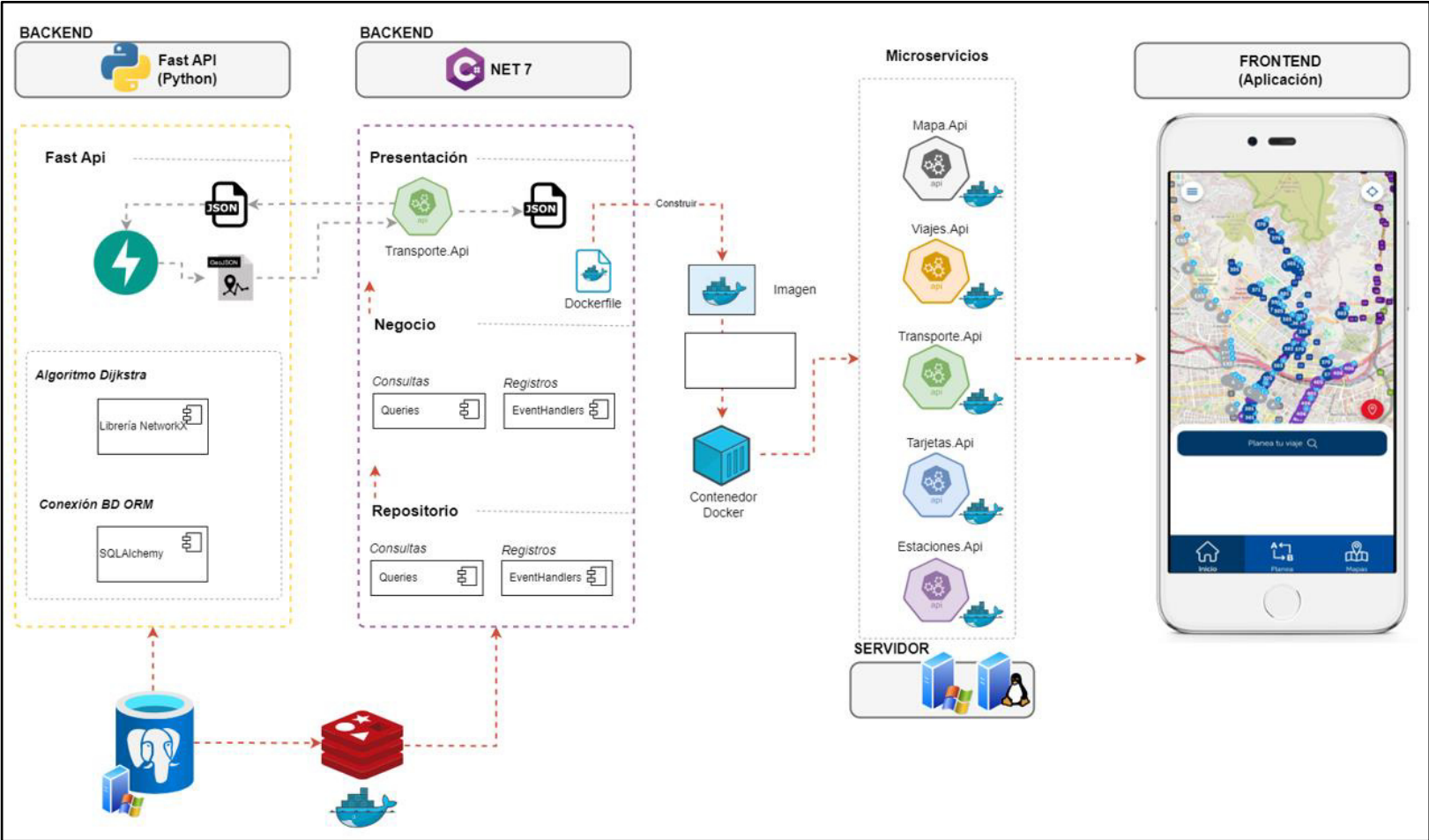
Anexo C: Diagrama de Tecnología de la Aplicación

El presente diagrama presenta la arquitectura tecnológica de la aplicación, exponiendo las tecnologías usadas para la construcción de la misma y como estas se proceden a integrar para la materialización de sus módulos/herramientas.

Sección de Aplicación	Tecnología	Uso
Repositorio de Datos	PostgreSQL	PostgreSQL se utiliza para el almacenamiento seguro y duradero de toda la información crítica que la aplicación maneja, como las rutas, ubicaciones y otros datos esenciales para la planificación de trayectos.
Repositorio Caché	Redis	Redis se implementa para almacenar temporalmente datos que se consultan con frecuencia. Esto mejora el rendimiento del sistema al reducir la necesidad de acceder a la base de datos principal repetidamente.
Back - End (MicroServicios)	.Net	Los servicios construidos en .NET son responsables de gestionar la lógica de negocio y coordinar la comunicación entre las distintas partes del sistema, incluyendo la base de datos, la API de FastAPI, y la aplicación móvil.
Back - End (Planificador de Rutas)	Fast API	FastAPI se encarga de calcular rutas, procesando las ubicaciones de inicio y destino provistas por la aplicación móvil, y devolviendo los resultados en un formato adecuado para ser utilizado en el sistema, como GeoJSON.
Front - End (Aplicación Móvil)	IONIC	IONIC se utiliza para desarrollar la interfaz móvil que los usuarios emplean para interactuar con la herramienta de planificación de rutas, permitiendo la consulta y visualización de las rutas generadas.

Nota. Elaboración propia

Diagrama Alto Nivel Arquitectura de Aplicación



Nota. Elaboración propia