

UNIVERSIDAD NACIONAL DE CAJAMARCA
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA CIVIL



TESIS

**“AUTOMATIZACIÓN Y OPTIMIZACIÓN DEL MODELADO BIM DE TUBERÍAS DE
AGUA FRÍA EN EDIFICACIONES USANDO EL ALGORITMO DIJKSTRA”**

PARA OPTAR AL TÍTULO PROFESIONAL DE INGENIERO CIVIL

AUTOR

Bach. CORTEZ ORTIZ CARLOS JAIR

ASESOR

Dr. JAIME AMORÓS DELGADO

CAJAMARCA - PERÚ

2025

CONSTANCIA DE INFORME DE ORIGINALIDAD

- FACULTAD DE INGENIERÍA -

1. **Investigador:** CARLOS JAIR CORTEZ ORTIZ

DNI: 70615424

Escuela Profesional: Ingeniería Civil

2. **Asesor:** Jaime Octavio Amoros Delgado

Facultad: Ingeniería

3. **Grado académico o título profesional**

☐ Bachiller

☒ Título profesional

☐ Segunda especialidad

☐ Maestro

☐ Doctor

4. **Tipo de Investigación:**

☒ Tesis

☐ Trabajo de investigación

☐ Trabajo de suficiencia profesional

☐ Trabajo académico

5. **Título de Trabajo de Investigación:**

AUTOMATIZACIÓN Y OPTIMIZACIÓN DEL MODELADO BIM DE TUBERÍAS DE AGUA FRÍA EN EDIFICACIONES USANDO EL ALGORITMO DIJKSTRA

6. **Fecha de evaluación:** 11/10/2025

7. **Software antiplagio:**

☒ TURNITIN

☐ URKUND (OURIGINAL) (*)

8. **Porcentaje de Informe de Similitud:** 12 %

9. **Código Documento:** trn:oid:::3117:511607292

10. **Resultado de la Evaluación de Similitud:**

☒ APROBADO ☐ PARA LEVANTAMIENTO DE OBSERVACIONES O DESAPROBADO

Fecha Emisión: 13/10/2025



FIRMA DEL ASESOR

Nombres y Apellidos: Jaime Octavio Amoros Delgado



Firmado digitalmente por:
BAZAN DIAZ Laura Sofia
FAU 20148258801 soft
Motivo: En señal de
conformidad
Fecha: 13/10/2025 20:12:57-0500

UNIDAD DE INVESTIGACIÓN FI



Universidad Nacional de Cajamarca

"Norte de la Universidad Peruana"

Fundada por Ley 14015 del 13 de Febrero de 1962

FACULTAD DE INGENIERÍA

Teléf. N° 365976 Anexo N° 1129-1130



ACTA DE SUSTENTACIÓN PÚBLICA DE TESIS.

TITULO : AUTOMATIZACIÓN Y OPTIMIZACIÓN DEL MODELADO BIM DE TUBERÍAS DE AGUA FRÍA EN EDIFICACIONES USANDO EL ALGORITMO DIJKSTRA

ASESOR : Dr. Ing. Jaime Octavio Amorós Delgado.

En la ciudad de Cajamarca, dando cumplimiento a lo dispuesto por el Oficio Múltiple N° 0723-2025-PUB-SA-FI-UNC, de fecha 30 de octubre de 2025, de la Secretaría Académica de la Facultad de Ingeniería, a los **cinco días del mes de noviembre de 2025**, siendo las ocho horas (08:00 a.m.) en la Sala de Audiovisuales (Edificio 1A – Segundo Piso), de la Facultad de Ingeniería se reunieron los Señores Miembros del Jurado Evaluador:

Presidente : Mag. Ing. Héctor Hugo Miranda Tejada.
Vocal : M.Cs. Ing. Marco Antonio Silva Silva.
Secretario : Ing. Marco Wilder Hoyos Saucedo.

Para proceder a escuchar y evaluar la sustentación pública de la tesis titulada **AUTOMATIZACIÓN Y OPTIMIZACIÓN DEL MODELADO BIM DE TUBERÍAS DE AGUA FRÍA EN EDIFICACIONES USANDO EL ALGORITMO DIJKSTRA** presentado por el Bachiller en Ingeniería Civil **CARLOS JAIR CORTEZ ORTIZ**, asesorado por el Dr. Ing. Jaime Octavio Amorós Delgado, para la obtención del Título Profesional.

Los Señores Miembros del Jurado replicaron al sustentante debatieron entre sí en forma libre y reservada y lo evaluaron de la siguiente manera:

EVALUACIÓN PRIVADA : 7 PTS.
EVALUACIÓN PÚBLICA : 11 PTS.
EVALUACIÓN FINAL : 18 PTS **Dieciocho** (En letras)

En consecuencia, se lo declara **APROBADO** con el calificativo de **18 (Dieciocho)** acto seguido, el presidente del jurado hizo saber el resultado de la sustentación, levantándose la presente a las **9:30** horas del mismo día, con lo cual se dio por terminado el acto, para constancia se firmó por quintuplicado.

Mag. Ing. Héctor Hugo Miranda Tejada.
Presidente

M.Cs. Ing. Marco Antonio Silva Silva.
Vocal

Ing. Marco Wilder Hoyos Saucedo.
Secretario

Dr. Ing. Jaime Octavio Amorós Delgado
Asesor

DEDICATORIA

A mi madre, mi hermana y mi compañera de vida, por brindarme su amor incondicional y su apoyo, sin los cuales no hubiera podido conseguir mis objetivos.

A mi primo Kevin, por su templanza y paciencia para conmigo y a mis tíos y primos que han sido ejemplo y soporte de mi vida.

A la memoria de mi padre el ingeniero Fausto Cortez Vera otrora catedrático de la Universidad Nacional de Cajamarca, por su cariño y fé en mí.

AGRADECIMIENTO

A **Dios** por darme salud a mí y a mi familia, por darme las capacidades para poder contribuir con la sociedad y por darme las oportunidades que me ha dado.

A mi asesor el **Dr. Jaime Octavio Amorós Delgado** por su gran trabajo como docente y por ser inspiración para desarrollar este tema.

A mis **amigos, compañeros, docentes** y demás personas que he conocido en la Universidad Nacional de Cajamarca quienes me han apoyado durante mi etapa académica.

A mis **colegas** del área BIM de quienes he aprendido y gracias a los cuales he ganado más experiencia profesional.

Finalmente, a quien pueda leer este trabajo, espero que sea de ayuda y que puedan seguir contribuyendo con el conocimiento humano.

CONTENIDO

ÍNDICE DE FIGURAS	viii
ÍNDICE DE TABLAS.....	xii
RESUMEN	xiii
ABSTRACT	xiv
RELACIÓN DE ABREVIATURAS	xv
CAPÍTULO I. INTRODUCCIÓN.....	1
1.1. Contexto.....	1
1.2. Problemática	2
1.2.1. Planteamiento del Problema	2
1.2.2. Formulación del Problema	3
1.3. Hipótesis	3
1.4. Justificación de la Investigación	3
1.5. Alcances o Delimitación de la Investigación	4
1.6. Objetivos	4
1.6.1. Objetivo General	4
1.6.2. Objetivos Específicos.....	5
CAPÍTULO II. MARCO TEÓRICO	6
2.1. Antecedentes Teóricos	6
2.1.1. Antecedentes Teóricos Internacionales	6
2.1.2. Antecedentes Teóricos Nacionales.....	6
2.1.3. Antecedentes Teóricos Locales	7
2.2. BIM.....	7
2.2.1. Definición de BIM.....	7
2.2.2. Dimensiones BIM.....	8
2.2.3. Usos BIM	10
2.2.4. Niveles de Información BIM	12
2.2.5. Gestión de la Información	13

2.2.6.	Metodología de Trabajo Colaborativo BIM.....	16
2.2.7.	Plan BIM Perú y Ley General de Contrataciones Públicas	16
2.2.8.	Modelo de Información.....	17
2.3.	IFC	19
2.4.	Automatización en BIM	20
2.4.1.	Automatización de Procesos BIM	20
2.5.	Softwares BIM (Revit)	20
2.5.1.	Entidades de Revit.....	21
2.5.2.	Revit API	25
2.5.3.	pyRevit.....	26
2.5.4.	Ejemplos de Automatización.....	27
2.5.5.	Softwares de Automatización de Modelado de Tuberías.....	28
2.6.	Algoritmo	29
2.6.1.	Definición	29
2.6.2.	Importancia en la Solución de Problemas Complejos.....	29
2.7.	Algoritmo Dijkstra	30
2.7.1.	Historia y Origen	30
2.7.2.	Definición y Funcionamiento.....	30
2.7.3.	Aplicaciones.....	39
2.7.4.	Ventajas y Limitaciones	39
2.8.	Grafos	39
2.8.1.	Definición de Grafos	39
2.8.2.	Aplicaciones.....	40
2.9.	Definición de Términos Básicos	40
2.9.1.	Expediente Técnico	40
2.9.2.	Metrado.....	41
2.9.3.	Presupuesto de Obra.....	42

2.9.4.	Análisis de Precios Unitarios.....	43
2.9.5.	Costo Directo	43
CAPÍTULO III. MATERIALES Y MÉTODOS		44
3.1.	Procedimiento	44
3.2.	Plugin “Dijkstra Tuberías”	45
3.2.1.	Configuración previa (pyRevit).....	46
3.2.2.	Botón “Configuraciones”	48
3.2.3.	Botón “Dijkstra Local”	49
3.2.4.	Cálculo de ruta óptima	62
3.2.5.	Modelado automático de tuberías.....	67
3.3.	Aplicación de Dijkstra por Niveles	69
3.3.1.	Botón “Dijkstra Global”	71
3.4.	Cálculo y Redimensionamiento.....	74
3.5.	Diagrama de flujo	78
CAPÍTULO IV. ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....		79
4.1.	Análisis de automatización.....	79
4.2.	Análisis y discusión de metrados de partidas.....	80
4.3.	Análisis de variación en costos	85
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES		88
5.1.	Conclusiones.....	88
5.2.	Recomendaciones	89
REFERENCIAS BIBLIOGRÁFICAS		90
ANEXOS		96

ÍNDICE DE FIGURAS

Figura 1. Dimensiones del sistema BIM.	9
Figura 2. Dimensiones BIM, propósito de los contenedores de información.	10
Figura 3. Usos BIM nacionales relacionados con las fases del Ciclo de Inversión....	11
Figura 4. Fases del Ciclo de Inversión.....	11
Figura 5. Niveles de Información (LOIN)	13
Figura 6. Grados de progresión de los niveles de gestión de la información BIM	15
Figura 7. Curva Mcleamy sobre el esfuerzo del proceso constructivo.....	19
Figura 8. Organización de elementos de Revit.....	22
Figura 9. Muestra de entidades de tipo "Room" software Revit.....	23
Figura 10. Ejemplo de tubería, accesorio de tubería y unión de tubería en Revit	24
Figura 11. Ejemplo de elementos pertenecientes a la categoría "Aparatos sanitarios" en el software Revit	25
Figura 12. Organización de directorios de pyRevit para la creación de una extensión	27
Figura 13. Ejemplo de aplicación del algoritmo Dijkstra paso 1: inicialización del algoritmo, nodo de origen seleccionado.	32
Figura 14. Ejemplo de aplicación del algoritmo Dijkstra paso 2: selección del nodo con menor distancia provisional.	32
Figura 15. Ejemplo de aplicación del algoritmo Dijkstra paso 3: Actualización de distancias a los vecinos del nodo actual.....	33
Figura 16. Ejemplo de aplicación del algoritmo Dijkstra paso 4: marcado del nodo como visitado.....	33
Figura 17. Ejemplo de aplicación del algoritmo Dijkstra paso 5: selección del siguiente nodo con menor distancia provisional.	34
Figura 18. Ejemplo de aplicación del algoritmo Dijkstra paso 6: actualización de nuevas distancias a los vecinos.	34
Figura 19. Ejemplo de aplicación del algoritmo Dijkstra paso 7: revisión de nodos visitados y pendientes.	35
Figura 20. Ejemplo de aplicación del algoritmo Dijkstra paso 8: continuación del proceso de selección y actualización.	35
Figura 21. Ejemplo de aplicación del algoritmo Dijkstra paso 9: cálculo progresivo de las rutas más cortas.	36

Figura 22. Ejemplo de aplicación del algoritmo Dijkstra paso 10: consolidación de distancias mínimas en nodos intermedios.....	36
Figura 23. Ejemplo de aplicación del algoritmo Dijkstra paso 11: avance hacia la ruta óptima desde el origen.	37
Figura 24. Ejemplo de aplicación del algoritmo Dijkstra paso 12: refinamiento de las distancias más cortas restantes.....	37
Figura 25. Ejemplo de aplicación del algoritmo Dijkstra paso 13: últimas actualizaciones antes del resultado final.	38
Figura 26. Ejemplo de aplicación del algoritmo Dijkstra paso 14: ruta más corta determinada por el algoritmo.	38
Figura 27. Modelo BIM especialidad arquitectura “Gran Mercado Modelo Chota”	44
Figura 28. Modelo BIM especialidad estructuras “Gran Mercado Modelo Chota”	44
Figura 29. Página oficial de pyRevit	46
Figura 30. Instalador de pyRevit.....	46
Figura 31. Pestaña de complemento pyRevit y sus herramientas.....	47
Figura 32. Creación de la extensión "DijkstraTuberías.extension"	47
Figura 33. Plugin creado "Dijkstra Tuberías"	48
Figura 34. Botón "Configuraciones" del plugin "Dijkstra Tuberías"	49
Figura 35. Ventana emergente desde botón "Configuraciones previas".....	49
Figura 36. Función para seleccionar elementos con pyRevit	50
Figura 37. Código para filtrar los elementos seleccionados que pertenecen a la categoría "Aparatos Sanitarios"	50
Figura 38. Código de la función para obtener la geometría de cualquier "room".....	51
Figura 39. Ejemplo de límites de una habitación	52
Figura 40. Habitaciones dentro del modelo BIM especialidad arquitectura “Gran Mercado Modelo Chota”	52
Figura 41. Ejemplo de punto de salida analítico de un inodoro con tanque	56
Figura 42. Ubicación del punto de salida analítico de un inodoro con tanque.....	56
Figura 43. Ejemplo de recta de llegada L1 (90°) a un inodoro con tanque	57
Figura 44. Ejemplo de llegada de recta L1 (45°) a un inodoro con tanque.....	57
Figura 45. Ejemplo de llegada de recta L1 (Paralela) a un inodoro con tanque	58
Figura 46. Tipos de conexiones más comunes para tuberías de agua fría (PVC).....	59
Figura 47. Ejemplo de rectas temporales generadas en un inodoro con tanque.....	60

Figura 48. Ejemplo de rectas temporales generadas para distintos tipos de aparatos sanitarios	60
Figura 49. Generación de malla para aplicación de algoritmo Dijkstra	61
Figura 50. Malla generada para aplicación de algoritmo Dijkstra	61
Figura 51. Pesos iniciales de los segmentos para aplicación de algoritmo Dijkstra ..	62
Figura 52. Representación gráfica de las posibles rutas para obtener la línea matriz.....	63
Figura 53. Ejemplo de posibles rutas matrices en un sistema local	64
Figura 54. Determinación de la ruta matriz óptima	64
Figura 55. Pesos y posibles rutas para el trazo de rutas secundarias hacia la matriz.....	65
Figura 56. Filtrado de rutas secundarias no permitidas	66
Figura 57. Ruta final optimizada para el modelado de tuberías.....	66
Figura 58. Modelado de tuberías con el botón "Dijkstra Local"	67
Figura 59. Comparación de ruta determinada analíticamente y las tuberías modeladas en Revit.....	67
Figura 60. Ejemplo de isométrico de modelado de tuberías automatizado	68
Figura 61. Modelado automático local de tuberías usando plugin "Dijkstra Tuberías"	68
Figura 62. Función creada para el modelado de tuberías con un punto inicial y un punto final.....	69
Figura 63. Código para modelado de tuberías automático siguiendo las líneas determinadas analíticamente.....	69
Figura 64. Ejemplo de "rooms" presentes en una edificación.....	70
Figura 65. Ejemplo de modelado automático con el plugin "Dijkstra Tuberías"	70
Figura 66. Botón "Dijkstra Global"	71
Figura 67. Ventana desplegable "Dijkstra Global"	72
Figura 68. Ejemplo de nodos para aplicación de "Dijkstra Global"	72
Figura 69. Ejemplo Red de tubería modelada automáticamente con "Dijkstra Global".....	73
Figura 70. Aplicación de Dijkstra Global en modelo BIM especialidad instalaciones sanitarias "Gran Mercado Modelo Chota"	73
Figura 71. Plugin de cálculo y redimensionamiento "PLUMBER"	74
Figura 72. Configuración general plugin "PLUMBER"	75

Figura 73. Configuración de familias y unidades Hunter plugin "PLUMBER"	75
Figura 74. Administrador de diámetros del plugin "PLUMBER"	76
Figura 75. Modelado previo (todas las tuberías tienen el diámetro 1/2")	76
Figura 76. Redimensionado de tuberías con plugin "PLUMBER"	77
Figura 77. Diagrama de flujo general automatización y optimización de modelado de tuberías.....	78
Figura 78. Interferencia de tuberías con elemento estructural, área de frutas ("Gran Mercado Modelo Chota").....	79
Figura 79. Variación porcentual de metrados por partida	82
Figura 80. Variación porcentual de metrado de accesorios y uniones de tubería	84
Figura 81. Variación de costo directo por partida	86
Figura 82. Comparación costo directo original vs optimizado.....	87

ÍNDICE DE TABLAS

Tabla 1. Usos BIM de acuerdo con el plan de implementación 2025 ANIN	12
Tabla 2. Habitaciones o "Rooms" obtenidos del modelo BIM especialidad arquitectura "Gran Mercado Modelo Chota"	53
Tabla 3. Ubicación de puntos de salida de cada aparato sanitario	55
Tabla 4. Ubicación de salida de agua fría de acuerdo con el eje de cada aparato sanitario	55
Tabla 5. Comparativo de partidas de "Instalaciones Sanitarias" ("Gran Mercado Modelo Chota").....	80
Tabla 6. Comparación del metrado de accesorios y uniones.....	83
Tabla 7. Comparación de costo directo por partidas.....	85

RESUMEN

El modelado convencional de redes de agua fría en edificaciones, dentro de entornos BIM constituye una tarea de carácter repetitivo y susceptible a inconsistencias, lo que incrementa la cantidad de trabajos, sobrecostos y desviaciones en los plazos de ejecución. De esta forma, la presente tesis tiene como objetivo principal automatizar y evaluar la optimización mediante la implementación del algoritmo Dijkstra y su validación en el expediente técnico del “Gran Mercado Modelo de Chota”. Para llevarlo a cabo, se transformó el espacio de diseño en grafos (nodos y aristas), se asignó pesos según longitud, accesorios y restricciones, y al mismo tiempo, se aplicó el algoritmo de Dijkstra con diversas rutinas en Dynamo y Revit API que automatizó la extracción de datos y el nuevo modelamiento de tuberías para obtener rutas óptimas. Con ello, se completó el proceso con 477 puntos de salida de agua fría y una longitud total del sistema de 2,596.14 m de tuberías redimensionadas. En comparación con los metrados se observó una reducción entre el 14% y 20% en los diámetros más frecuentes, disminución en la cantidad de accesorios como los codos de Ø3/4"×90° de 694 a 653 unidades; y, el presupuesto presentó una reducción de materiales que representó un ahorro cercano al 5.2% del costo directo de las partidas consideradas. En consecuencia, la metodología implementada permitió automatizar el modelado, facilitar el redimensionamiento hidráulico y obtener metrados y presupuestos optimizados, siendo su eficacia dependiente de la calidad del modelo BIM de entrada y de los ajustes manuales requeridos por interferencias.

Palabras clave: BIM, Automatización, Algoritmo Dijkstra, Modelado, Optimización.

ABSTRACT

The manual modeling of cold water networks in buildings within BIM environments is a repetitive and error-prone process, which generates rework, higher costs, and delays. This research is justified by the need for tools that automate pipe routing and reduce inconsistencies, thereby improving productivity in the design stage within the Peruvian context. The proposal consisted of transforming the design space into a graph (nodes and edges), assigning weights according to length, fittings, and restrictions, and applying Dijkstra's algorithm to obtain optimal routes. Routines were developed in Dynamo and the Revit API to automate data extraction, algorithm execution, and pipe recreation in the model, applied to the technical file of the "Gran Mercado Modelo de Chota." The results show that the add-in processed 477 cold water outlet points, generating a network of 2,596.14 m of resized pipes. In the comparison of quantities, reductions between 14% and 20% were observed in the most frequent diameters, as well as a decrease in the number of fittings, such as Ø3/4"×90° elbows, which decreased from 694 to 653 units. In the economic analysis, variations due to material reductions represented savings of approximately 5.2% of the direct cost of the evaluated items. In conclusion, the implemented methodology made it possible to automate modeling, facilitate hydraulic resizing, and obtain optimized quantities and budgets, with its effectiveness depending on the quality of the input BIM model and the manual adjustments required due to interferences.

Keywords: BIM, Automation, Dijkstra Algorithm, Modeling, Optimization.

RELACIÓN DE ABREVIATURAS

AEC:	Architecture, Engineering and Construction (Arquitectura, Ingeniería y Construcción)
ANIN:	Autoridad Nacional de Infraestructura
API:	Application Programming Interface (Interfaz de programación de aplicaciones)
BIM:	Building Information Modeling (Modelado de la información de la construcción)
CAD:	Computer-Aided Design (Diseño asistido por computadora)
CDE:	Common Data Environment (Entorno común de datos)
IFC:	Industry Foundation Classes (Clases fundamentales de la industria)
INEI:	Instituto Nacional de Estadística e Informática
ISO:	International Organization for Standardization (Organización Internacional de Estandarización)
LOD:	Level of Detail (Nivel de detalle)
LOI:	Level of Information (Nivel de información)
LOIN:	Level of Information Need (Nivel de información necesaria)
MEF:	Ministerio de Economía y Finanzas
MEP:	Mechanical, Electrical, and Plumbing (Mecánicas, electricidad y fontanería)
NTP:	Norma Técnica Peruana
PBI:	Producto Bruto Interno
PEB:	Plan de Ejecución BIM
PVC:	Policloruro de vinilo
VDC:	Virtual Design and Construction (Diseño y construcción virtual)

CAPÍTULO I. INTRODUCCIÓN

1.1. Contexto

En las últimas décadas, el sector de la construcción ha jugado un rol fundamental en el desarrollo económico a nivel mundial, especialmente en países en vías de desarrollo como el Perú. Según proyecciones de Global Construction Perspectives y Oxford Economics, la industria constructiva crecerá en un 42% para el año 2030, en respuesta a la creciente demanda de infraestructuras urbanas y de servicios. En Perú, el sector contribuye aproximadamente con un 5.6% al PBI, manteniéndose como un pilar clave para la economía y un agente esencial en la recuperación post-pandemia (INEI, 2023). Sin embargo, la industria enfrenta serios desafíos, como los altos costos derivados de errores de diseño, la falta de coordinación entre disciplinas y las ineficiencias que se generan en las fases de planificación y ejecución (World Economic Forum, 2016).

La metodología BIM (Building Information Modeling) surge como una respuesta a estos desafíos, proporcionando un marco integral que permite la gestión eficiente de proyectos constructivos a lo largo de todo su ciclo de vida. El uso de BIM posibilita la creación de modelos tridimensionales que no solo contienen representaciones visuales de los elementos de construcción, sino que también integran datos críticos que facilitan la toma de decisiones. Esta metodología ha mostrado una notable capacidad para reducir los errores de diseño, mejorar la coordinación entre especialidades y optimizar los costos de ejecución (Eastman et al., 2011).

Pese a sus beneficios, la adopción de BIM en Perú se ha visto obstaculizada por factores como la falta de capacitación técnica en herramientas de modelado y la resistencia al cambio de métodos tradicionales de trabajo (Alfaro, 2019). En este contexto, la automatización de procesos mediante el uso de programación visual y la implementación de herramientas como Dynamo y lenguajes de código han cobrado relevancia, al permitir la optimización de flujos de trabajo y la reducción de errores en el diseño y documentación de proyectos de infraestructura (Dynamo BIM Organization, 2017).

Este estudio se enfoca en la automatización y optimización del modelado de redes de agua fría en edificaciones, utilizando el algoritmo de optimización Dijkstra, que calcula la ruta más corta desde un nodo origen hacia todos los demás en un grafo, explorando de manera iterativa el nodo con la distancia acumulada más baja y actualizando los costos mínimos hacia sus vecinos.

La implementación de este enfoque no solo busca reducir tiempos de diseño y minimizar errores, sino también establecer un marco práctico y replicable que impulse el uso de BIM en la ingeniería civil en Perú, fomentando así una industria constructiva más moderna, eficiente y competitiva.

1.2. Problemática

El proceso de diseño en edificaciones enfrenta diversos desafíos entre los cuales destacan la falta de innovación y adaptación tecnológica, lo que afecta a la eficiencia y precisión de los proyectos. La falta de automatización de algunas tareas implica que los ingenieros deben realizar actividades repetitivas y propensas a errores, estos errores pueden derivar en revisiones costosas y retrasos que comprometen la viabilidad y éxito del proyecto.

BIM ha demostrado ser efectiva para mejorar la coordinación entre disciplinas y reducir errores durante las fases previas a la ejecución física, pero su adopción sigue siendo un reto en muchos contextos debido a la resistencia al cambio de procesos tradicionales y la falta de herramientas específicas para automatizar el modelado, sin embargo, la implementación de este enfoque conlleva desafíos técnicos, como la integración precisa de datos y la utilización efectiva de la programación, para lograr un sistema automatizado eficaz y replicable en la práctica.

Conjuntamente, las características de las edificaciones y los requisitos específicos de cada proyecto complican la creación de soluciones automatizadas adaptativas y eficaces (Sacks et al., 2018).

1.2.1. Planteamiento del Problema

La generación de modelos BIM de redes de agua fría en edificaciones es un proceso complejo que requiere una planificación cuidadosa y detallada, así como un conocimiento técnico especializado, lo que lo convierte en un procedimiento tedioso y propenso a errores.

En este contexto, la automatización del modelado de redes de agua potable en edificaciones puede ser una solución efectiva para mejorar la eficiencia y reducir los errores y el tiempo en el proceso de diseño. Sin embargo, la falta de herramientas automatizadas específicas para esta tarea y la necesidad de adaptar el diseño a las características particulares de cada edificación hace que sea un desafío lograr una automatización completa. Por lo tanto, el problema que se aborda en este proyecto de tesis es cómo desarrollar un sistema automatizado de diseño de redes de agua fría en edificaciones, que tenga en cuenta las características únicas de cada edificación y permita un diseño óptimo de tuberías y accesorios para el sistema.

1.2.2. Formulación del Problema

La investigación resuelve la pregunta:

¿Cómo implementar el algoritmo Dijkstra para automatizar y optimizar el proceso de modelado BIM de redes de agua fría en edificaciones?

1.3. Hipótesis

La implementación del algoritmo Dijkstra en un complemento para Autodesk Revit permite automatizar el modelado BIM de tuberías de agua fría, en el caso de estudio del “Gran Mercado Modelo de la ciudad de Chota” y produce una disminución del recorrido total de tuberías y variaciones en las cantidades de accesorios respecto al expediente técnico original.

1.4. Justificación de la Investigación

El diseño del trazado de la red de distribución constituye una etapa fundamental, ya que una adecuada planificación asegura su correcto funcionamiento, reduce los costos y mejora la eficiencia del sistema.

Si bien el uso de BIM tiene un impacto significativo en la reducción de errores de diseño, en la investigación “Exploring the linkages between the adoption of BIM and design error reduction” (Wong et al., 2018), se determinó que la afirmación de que BIM puede reducir los errores humanos durante la etapa de diseño es errónea, debido a los diversos conjuntos de variables exógenas y endógenas que afectan la cognición y la capacidad de un diseñador para ejecutar tareas.

La investigación contribuirá al avance de las herramientas de modelado automatizado para redes de agua fría en edificios y proporcionará un recurso valioso para los ingenieros civiles y otros profesionales en el campo.

El presente trabajo se enfoca en acelerar y optimizar el proceso de modelado y diseño redes de tuberías de agua fría en edificaciones, como una estrategia para aumentar la productividad en la etapa de diseño de proyectos constructivos.

Mediante la metodología BIM se obtienen datos técnicos de las edificaciones, dichos datos pueden ser usados de manera eficiente para automatizar tareas complejas como el diseño de redes de agua, de esta manera no solo se puede conseguir acelerar los procedimientos de diseño, sino también se minimizan errores y se optimizan los recursos utilizados en las obras (Vargas Ramírez, 2016).

1.5. Alcances o Delimitación de la Investigación

La investigación se enmarca en la automatización y optimización del modelado BIM de tuberías de agua fría, mediante creación de un plugin para el software Revit que permita aplicar el algoritmo Dijkstra usando la información del modelo BIM para determinar las rutas óptimas. Para esto se tomó como base el expediente técnico: “Creación de los servicios de comercialización en el gran mercado modelo de la ciudad de Chota, distrito de Chota - provincia de Chota - departamento de Cajamarca”

Se inicio a partir del modelo BIM del expediente técnico usando el software Revit y mediante la aplicación del algoritmo Dijkstra a través de un complemento para el software Revit, se buscó automatizar el modelado y encontrar la ruta de menor recorrido para las tuberías. Los resultados obtenidos se compararon con los del expediente técnico comparando los metrados para determinar la optimización de recursos.

1.6. Objetivos

1.6.1. Objetivo General

- Automatizar y evaluar la optimización del modelado BIM de redes de tuberías de agua fría en edificaciones mediante la implementación del algoritmo Dijkstra y su validación en el expediente técnico del “Gran Mercado Modelo de la ciudad de Chota”.

1.6.2. Objetivos Específicos

- Desarrollar un complemento para Autodesk Revit que automatice el modelado BIM de tuberías usando el algoritmo Dijkstra.
- Aplicar el complemento y realizar el redimensionamiento hidráulico de la red modelada.
- Comparar cuantitativamente los metrados (longitudes y cantidades de accesorios) entre el expediente técnico original y el diseño generado por el algoritmo, calculando variaciones porcentuales por partida.
- Comparar los presupuestos obtenidos del expediente técnico original y del modelo optimizado, determinando las variaciones en el costo directo de las partidas de tuberías y accesorios.

CAPÍTULO II. MARCO TEÓRICO

2.1. Antecedentes Teóricos

2.1.1. Antecedentes Teóricos Internacionales

Atencio et al. (2022) en el artículo de investigación “Towards the Integration and Automation of the Design Process for Domestic Drinking-Water and Sewerage Systems with BIM”, se desarrolló un flujo de trabajo diseñado basado en BIM para un sistema doméstico de agua potable y alcantarillado mediante un proceso de diseño semiautomatizado, en donde se integró algunos softwares como Autodesk Revit y MS Excel, además la investigación demuestra una posible extensión de aplicaciones más allá de las convencionales.

Zhang et al. (2022), en la investigación “BIM-based automated design of drainage systems for panelized residential buildings”, se desarrolla una aplicación de extensión BIM como un complemento de Autodesk Revit para realizar el diseño y optimización de un sistema de drenaje residencial, el enfoque de su algoritmo se basa en reglas para mejorar la precisión y la eficacia del proceso de diseño. Para el diseño de las tuberías de desagüe, se aplica un enfoque heurístico que integra la consideración de los elementos de entramado de un panel y el recorrido de las tuberías por gravedad, con el fin de optimizar el costo total de la instalación.

Kim et al. (2013), en la investigación “The development of a practical pipe auto-routing system in a shipbuilding CAD environment using network optimization”, propone e implementar un sistema automático de trazado de tuberías del cuarto de máquinas de un buque comercial, que hacen uso de distintos algoritmos como el algoritmo Maze y el algoritmo de escape Hightower (1969).

2.1.2. Antecedentes Teóricos Nacionales

Víctor et al. (2017), en la investigación “Optimización de redes pluviales de desagüe y alcantarillado del distrito de Pucarpata con métodos genéticos”, se logró demostrar la aplicación de los algoritmos genéticos a la hidrología e hidráulica en redes de drenaje urbano, además de demostrar la ventaja del uso de la computación evolutiva, en el uso de los algoritmos genéticos a un problema de drenaje urbano y de diseño en general.

Rojas Córdova et al. (2022), en la investigación “Aplicación de tecnologías 4.0 a proyectos de edificación”, concluye que la industria 4.0 busca una producción más eficiente y que las principales ventajas de la aplicación de las tecnologías 4.0 en el sector de la construcción pueden ser que los procesos se llegan a optimizar correctamente mediante la seguridad, la calidad y la eficiencia.

2.1.3. Antecedentes Teóricos Locales

Cabrera Chávarry (2023), en la tesis denominada “Automatización de procesos en las fases de diseño y documentación de expedientes técnicos utilizando la metodología VDC/BIM y la programación visual”, se centra en el proyecto de la Aldea Infantil San Antonio en Cajamarca, donde se implementó la metodología VDC/BIM junto con la programación visual en Revit y Dynamo. Los resultados mostraron que esta combinación permitió detectar incompatibilidades de diseño entre las especialidades de arquitectura, estructuras e instalaciones, reduciendo significativamente los errores en el diseño y aumentando la precisión en la planificación. Este enfoque automatizado en proyectos locales resalta cómo BIM, mejora la eficiencia en la fase de diseño y minimiza los costos y tiempos asociados a modificaciones en etapas tardías del proyecto. También recomienda, que, para mejorar futuros trabajos, se dé más rigor a la parte informática usando protocolos de programación que permitan el desarrollo de diagramas de flujo y de procesos.

2.2. BIM

2.2.1. Definición de BIM

De acuerdo con la ISO 19650-1, es “El uso de una representación digital compartida de un activo construido, para facilitar los procesos de diseño, construcción y operación, con la finalidad de contar con una base confiable para la toma de decisiones”, la Resolución Ministerial RM-242-2019-MVCS publicada por el Ministerio de Vivienda Construcción y Saneamiento indica que BIM es una “metodología de trabajo colaborativo que utiliza herramientas informáticas para la gestión de un proyecto de obra civil y edificación, a través de una base de datos gráfica que permite crear un modelo tridimensional inteligente de una edificación u obra civil, que además de ser una representación gráfica 3D incluye la información no gráfica, como especificaciones técnicas, estados de avance, entre otros”.

BIM es un proceso revolucionario en la industria de la arquitectura, la ingeniería y la construcción, que implica la creación y gestión de representaciones digitales de proyectos físicos a lo largo de su ciclo de vida. BIM utiliza modelos 3D coordinados e interdisciplinarios que incorporan objetos que se pueden presentar en dibujos 2D sincronizados. Estos modelos están basados en bases de datos, lo que permite contar, ordenar y adjuntar datos de objetos (Baldwin, 2014).

Otra definición la proporciona el Ministerio de Economía y Finanzas mediante el DS -289-2019-EF, 2019 donde se precisa al BIM como: “un conjunto de metodologías, tecnologías y estándares que permiten formular, diseñar, construir, operar y mantener una infraestructura pública de forma colaborativa en un espacio virtual”.

En este marco, el Plan de Ejecución BIM (PEB), conocido internacionalmente como BIM Execution Plan (BEP), se constituye como un instrumento clave para organizar la implementación de BIM en los proyectos, estableciendo objetivos, usos, procesos, roles y responsabilidades de todos los actores involucrados. De forma complementaria, el Entorno Común de Datos (ECD) o Common Data Environment (CDE) actúa como la plataforma centralizada de almacenamiento y gestión de la información del proyecto, asegurando que los datos sean confiables, accesibles y estén siempre actualizados. Ambos elementos, PEB y ECD, resultan fundamentales para garantizar la coordinación, eficiencia y trazabilidad de la información durante todo el ciclo de vida de la infraestructura

2.2.2. Dimensiones BIM

Las dimensiones BIM se refieren a la manera en que diversos tipos de datos se asocian con un modelo informativo. Cada dimensión, junto con los datos que la acompañan, ofrece una visión más completa del proyecto, incluyendo detalles sobre cómo se llevará a cabo la construcción, en qué momento se realizará y cuál será su costo. (Piaseckienė, 2022)

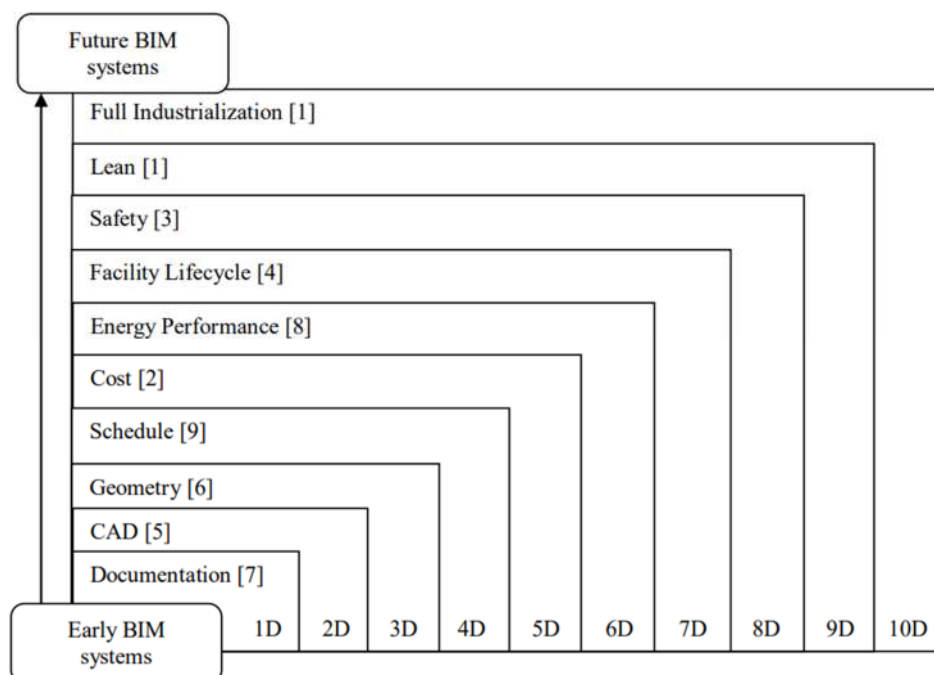
En la (Guía Nacional BIM, 2021) se indica que la elección del software de CDE depende del propósito del contenedor de información; en modelos 3D esto se relaciona con las dimensiones BIM (1D–7D): 3D (geometría), 4D (tiempo), 5D (costos),

6D (sostenibilidad) y 7D (operación). Cada dimensión define cómo se usa el modelo y orienta los flujos de trabajo del CDE.

Las dimensiones BIM representan un enfoque estructurado para integrar diferentes tipos de información dentro de un modelo de construcción digital. Estas dimensiones permiten relacionar datos específicos con elementos del proyecto, lo que contribuye a una mejor planificación y gestión. Por ejemplo, cada dimensión agrega detalles esenciales: cómo se llevará a cabo la construcción, el cronograma asociado, el presupuesto necesario, y otros factores críticos como la sostenibilidad y el mantenimiento a lo largo de la vida útil del edificio. Esta metodología proporciona a los equipos de proyecto una visión integral y facilita la toma de decisiones informadas, optimizando los recursos y minimizando los riesgos durante todo el ciclo de vida del proyecto.

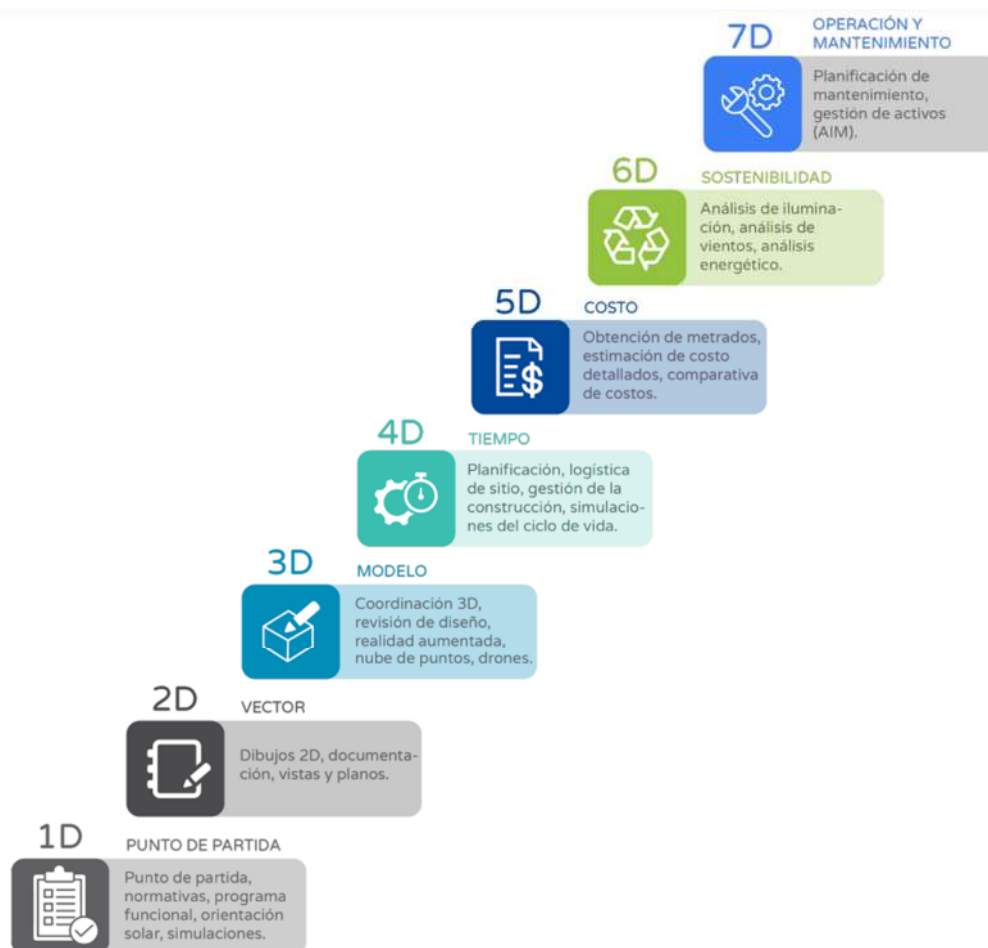
En la publicación: Implementation of Building Information Modelling in infrastructure construction projects: a study of dimensions and strategies (Ershadi et al., 2021), se hace indica que Las dimensiones BIM son los diferentes enfoques (1D a 10D) que caracterizan cómo BIM documenta, integra y gestiona información y procesos a lo largo del ciclo de vida de un proyecto.

Figura 1. Dimensiones del sistema BIM.



Fuente: Tomado de “Implementation of Building Information Modelling in infrastructure construction projects: a study of dimensions and strategies” (Ershadi et al., 2021)

Figura 2. Dimensiones BIM, propósito de los contenedores de información.



Fuente: Guía Nacional BIM 2023

2.2.3. Usos BIM

De acuerdo con la Guía Nacional BIM, los usos BIM hacen referencia a los métodos en lo que se aplica BIM y que se definen de acuerdo con procesos que se pueden ubicar, orientar y relacionar con cada fase del ciclo de inversión de un proyecto, con el fin de alcanzar uno o más objetivos previamente planteados. Estos usos sirven en otras palabras para explicar las diferentes formas en las que las partes que están involucradas pueden hacer uso de BIM en una determinada inversión.

Figura 3. Usos BIM nacionales relacionados con las fases del Ciclo de Inversión



Fuente: Guía Nacional BIM

Figura 4. Fases del Ciclo de Inversión



Fuente: Ministerio de Economía y Finanzas

En el documento presentado por la Autoridad Nacional de Infraestructura (ANIN), denominado “Plan de Implementación BIM 2025”, se hace constar que para

el año 2030 se tiene como objetivo implementar 13 de los 27 usos BIM a nivel nacional y se presenta una tabla donde se precisan los objetivos de los usos BIM en dicha entidad.

Tabla 1. Usos BIM de acuerdo con el plan de implementación 2025 ANIN

Usos BIM	Objetivos de los usos BIM
(1) Levantamiento de Condiciones Existentes	Obtener información precisa y confiable del terreno y la topografía para la toma de decisiones y elaboración de propuestas de diseño en la ubicación óptima con los modelos de información tomando en cuenta la línea base ambiental.
(2) Análisis del entorno físico	
(3) Diseño de Especialidades	Realizar el desarrollo del diseño de las especialidades del proyecto a través del Modelo de Información para su optimización, de manera que se obtenga la mejor alternativa posible tanto a nivel funcional como económico.
(4) Elaboración de Documentación	Obtener documentación de diseño confiable y precisa, que no genere incompatibilidades durante la etapa de construcción, que esté siempre disponible para la toma de decisiones reduciendo el tiempo de producción y revisión de información gráfica y no gráfica, así como asegurando la interoperabilidad de los Contenedores de Información.
(5) Visualización 3D	Favorecer el entendimiento del proyecto utilizando herramientas visuales para facilitar la comunicación entre las partes involucradas con motivos de revisión y validación, así como para las partes externas interesadas según sea el caso.
(6) Coordinación de la información	Resolver las incidencias, interferencias, incompatibilidades y criterios de diseño de forma oportuna, reducir las latencias de respuesta o retrasos, y evitar posibles RFI's en etapas de ejecución física mediante procedimientos de coordinación, integración y compatibilización, asegurando su gestión y resolución por medio de informes y sesiones.
(9) Revisión del diseño	
(16) Detección de interferencias e incompatibilidades	
(8) Estimación de cantidades y costos	Realizar la planificación de la construcción desde etapas tempranas, asegurar la constructibilidad y obtener cantidades precisas para cuantificación de costos, así como simulaciones constructivas, programación de obra o cualquier otra actividad relacionada a los datos de costos y tiempo para la toma de decisiones orientada a la constructibilidad.
(13) Análisis de la capacidad constructiva	
(17) Planificación de la fase de ejecución	
(23) Gestión de activos	El modelo generado en la etapa de la ejecución debe contar con la información necesaria para seguir gestionando el modelo en la etapa de mantenimiento. Realizar la planificación de las actividades de mantenimiento preventivo en los activos generados e incluir todas las acciones realizadas.
(24) Programación del mantenimiento preventivo	Incluir en la información del AIM, la información y acciones realizadas en el mantenimiento correctivo. Implementar una plataforma de gestión del mantenimiento vinculada al BIM, la cual facilite la gestión de la documentación del mantenimiento y control de los activos que requieran reposición en base a sus fechas de garantías y vida útil.

Fuente: Autoridad Nacional de Infraestructura 2025

Nota: (En verde) Se denota en el uso BIM “Diseño de especialidades” que se pretende optimizar el diseño de especialidades haciendo uso del modelo de información

2.2.4. Niveles de Información BIM

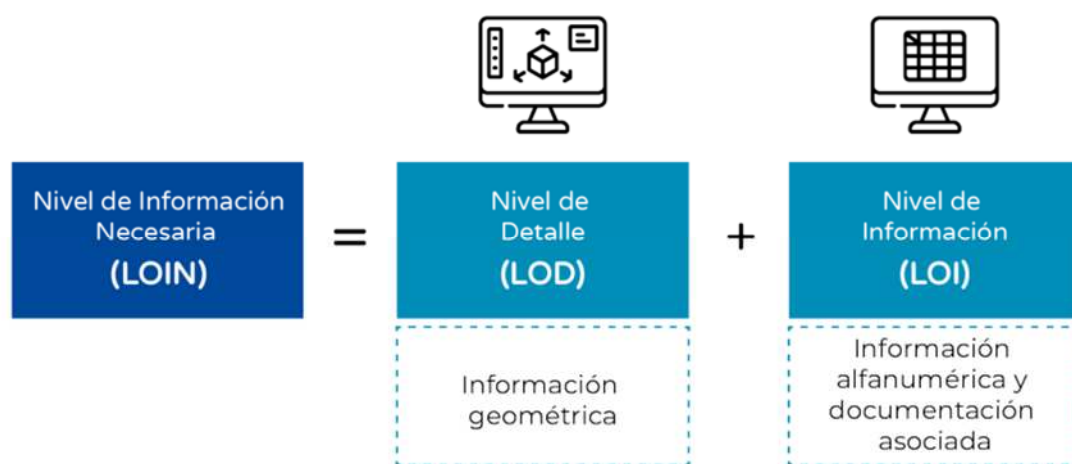
El concepto de nivel de información BIM se refiere a la riqueza y precisión de los datos que se integran en un modelo durante las diferentes etapas de un proyecto. A medida que el proyecto avanza, desde la fase de diseño inicial hasta la construcción y el mantenimiento, el nivel de información evoluciona y se vuelve más detallado y específico. (Paula et al., 2015)

En las primeras etapas, los modelos pueden contener datos básicos sobre la geometría y las principales características estructurales. Sin embargo, a medida que el proyecto progresa, se agregan más detalles, como especificaciones de materiales, datos de costos, cronogramas de construcción, análisis de sostenibilidad, y finalmente, información relacionada con el mantenimiento y la operación del edificio.

Este enfoque escalonado permite a los equipos de proyecto tener acceso a la información necesaria en el momento adecuado, facilitando una mejor coordinación, gestión eficiente y toma de decisiones informadas en cada fase. Además, un nivel de información bien gestionado garantiza que todos los involucrados, desde arquitectos e ingenieros hasta propietarios y gestores de instalaciones, puedan colaborar de manera efectiva, utilizando un modelo que evoluciona junto con las necesidades del proyecto.

Para la guía nacional BIM, el nivel de información necesaria (LOIN), es el nivel de información necesaria para satisfacer los objetivos relacionados a la información de una inversión, en cada proceso de intercambio de información, y está conformada por un nivel de información gráfica (LOD) y un nivel de información no gráfica (LOI).

Figura 5. Niveles de Información (LOIN)



Fuente: Guía Nacional BIM (2021)

2.2.5. Gestión de la Información

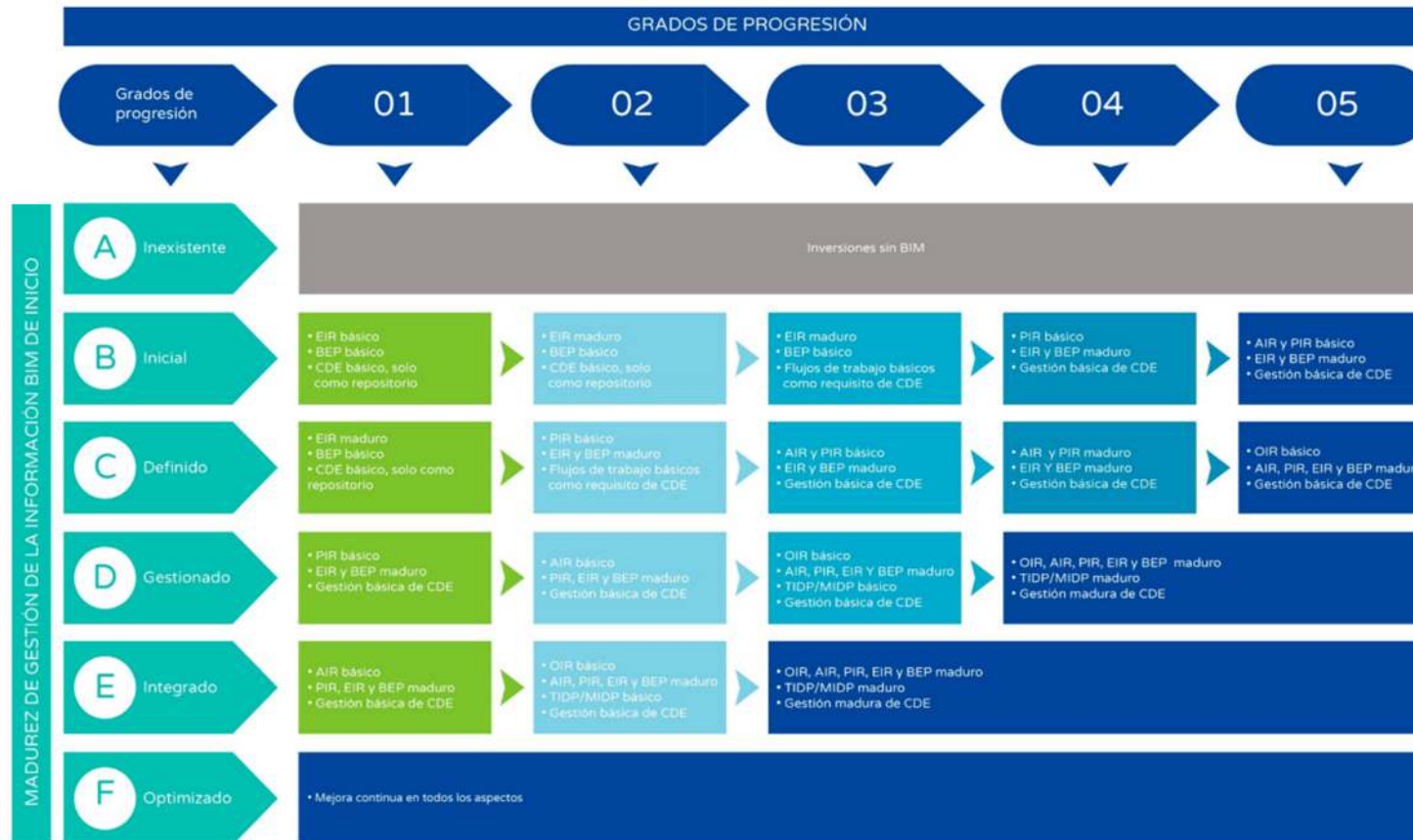
La gestión de la información BIM se refiere al proceso de organizar, almacenar, compartir y mantener la información generada durante el ciclo de vida de un proyecto de construcción utilizando la metodología BIM. Este enfoque garantiza que todos los

datos relevantes estén disponibles y actualizados para los diferentes actores involucrados, facilitando la toma de decisiones y optimizando la eficiencia del proyecto.

La gestión de la información BIM implica coordinar y supervisar la creación, intercambio y almacenamiento de datos relacionados con el proyecto, asegurando que la información sea precisa, coherente y accesible para todos los participantes. Esto incluye la implementación de estándares y protocolos para el intercambio de información, el uso de entornos comunes de datos (ECD) y la aplicación de normas internacionales como la ISO 19650, que establece los principios para la gestión de la información en proyectos de construcción utilizando BIM.

Una gestión eficiente de la información BIM permite mejorar la colaboración entre las partes involucradas, reducir errores y retrabajos, optimizar los procesos de construcción y facilitar el mantenimiento y operación de las infraestructuras a lo largo de su ciclo de vida.

Figura 6. Grados de progresión de los niveles de gestión de la información BIM



Fuente: Guía Nacional BIM

Nota: En la figura se muestra que el grado de madurez finalmente buscado es el “Optimizado”, el cual busca una “Mejora continua en todos los aspectos”

2.2.6. Metodología de Trabajo Colaborativo BIM

El trabajo colaborativo en BIM se basa en la integración de equipos multidisciplinarios que colaboran de manera efectiva utilizando modelos digitales compartidos. Esta forma de trabajo tiene como objetivo optimizar la planificación, diseño, construcción y mantenimiento de proyectos mediante un enfoque coordinado y basado en datos (Lu et al., 2013).

En este contexto, arquitectos, ingenieros, constructores, propietarios y otros actores clave trabajan en un entorno común, conocido como Entorno Común de Datos (CDE), donde se almacenan y gestionan todos los datos del proyecto. Esto permite que todos los involucrados tengan acceso a la misma información actualizada y puedan realizar modificaciones en tiempo real, lo que reduce el riesgo de errores y mejora la toma de decisiones.

El trabajo colaborativo en BIM también fomenta una comunicación constante y fluida entre los participantes del proyecto. A través de reuniones regulares, revisiones conjuntas y la utilización de herramientas digitales avanzadas, los equipos pueden coordinarse mejor y resolver conflictos de diseño de manera eficiente antes de que se conviertan en problemas en el sitio de construcción.

2.2.7. Plan BIM Perú y Ley General de Contrataciones Públicas

La gestión de la información en Building Information Modeling (BIM) es esencial para optimizar la planificación, ejecución y mantenimiento de los proyectos de construcción, al facilitar decisiones basadas en datos confiables y coordinados. En el Perú, el Plan BIM Perú, impulsado por el Ministerio de Economía y Finanzas, constituye la estrategia nacional para la adopción progresiva de esta metodología en la inversión pública, con horizonte al año 2030. Dicho plan busca reducir sobre costos y retrasos, incrementar la eficiencia y la calidad en todas las fases del ciclo de inversión, además, fortalecer la transparencia y sostenibilidad en la gestión de la infraestructura pública.

Entre sus instrumentos destaca el Plan de Ejecución BIM (PEB), que define la metodología de trabajo, procesos, roles, responsabilidades y entregables en cada etapa, en función de un nivel de información necesaria,

compuesto por el detalle gráfico (geométrico) y la información alfanumérica requerida según los objetivos de la fase del proyecto.

En el plano normativo, la Directiva N.º 017-2023-CG/GMPL establece que en las obras de categoría C (mayores a S/ 10 millones) el expediente técnico debe incorporar un modelo BIM 3D con suficiente nivel de detalle que asegure la ausencia de colisiones entre las especialidades básicas. De manera complementaria, la Ley General de Contrataciones Públicas (Ley N.º 32069) y su reglamento (2025) incorporan expresamente la metodología BIM en los procesos de contratación pública, disponiendo su aplicación progresiva en concordancia con el Plan BIM Perú, y encargando al OSCE y al MEF la emisión de lineamientos técnicos y condiciones específicas, priorizando su implementación en proyectos de gran envergadura como parte de la modernización del sistema de inversión pública.

El Plan BIM Perú 2019–2030 se estructura en hitos a corto, mediano y largo plazo: en el corto plazo (2019–2022), el objetivo es sentar las bases institucionales y normativas de la adopción BIM, desarrollando lineamientos y pilotos; en el mediano plazo (2023–2026), se busca la consolidación de la implementación progresiva en sectores estratégicos, ampliando la capacitación y el uso obligatorio en proyectos de mayor envergadura; y en el largo plazo (2027–2030), se proyecta la adopción generalizada de BIM en todo el ciclo de vida de los proyectos de inversión pública, logrando estándares internacionales de eficiencia, calidad y sostenibilidad en la infraestructura nacional.

En conjunto, la integración de BIM en la normativa peruana y en la hoja de ruta del Estado resalta la centralidad de la gestión de la información como herramienta para proyectos más eficientes, coordinados y sostenibles, que permitan mejorar la provisión de infraestructura pública y el uso responsable de los recursos del país.

2.2.8. Modelo de Información

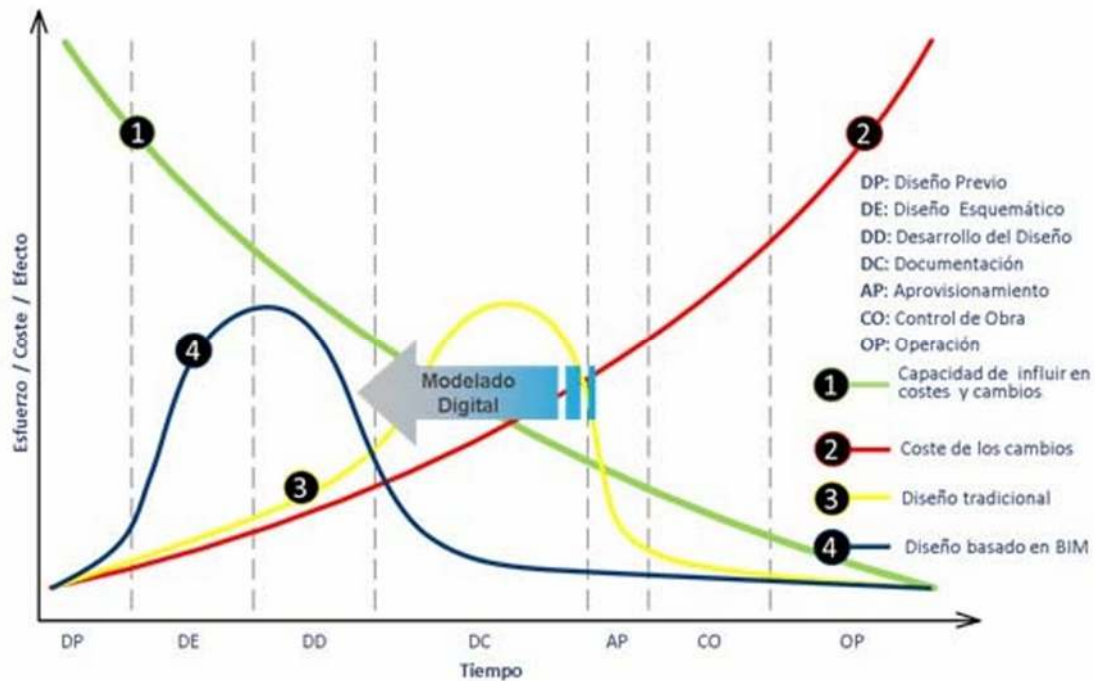
Según el Decreto Supremo N.º 108-2021-EF, que modifica al DS 289-2019-EF se define como un conjunto que agrupa tanto contenedores de información estructurada como no estructurada. Incluye toda la documentación

generada a lo largo de una inversión, diseñada para cumplir con los requisitos de información y almacenada en una base de datos confiable.

El modelo funciona como una base centralizada de información que facilita la planificación, diseño, construcción y gestión de infraestructuras, permitiendo organizar y asociar datos con los elementos del modelo. Además, las Industry Foundation Classes son el estándar más utilizado dentro del esquema BIM (Isikdag, 2015).

El propósito esencial de la metodología BIM es erradicar la pérdida de valor de la información a lo largo del ciclo de vida del proyecto, una deficiencia inherente al método tradicional que exige una producción fragmentada y repetitiva de información en cada fase. En contraste, BIM establece un flujo de trabajo que garantiza un crecimiento sostenido del valor informativo, rompiendo con la desarticulación característica de los enfoques convencionales. Para alcanzar esta meta, es indispensable transformar radicalmente el proceso de toma de decisiones, trasladándolo a las etapas más tempranas del proyecto, cuando la capacidad de incidir en el costo final es máxima y el esfuerzo económico para implementar cambios es mínimo. Este cambio de paradigma se ilustra de forma contundente en la célebre Curva de MacLeamy.

Figura 7. Curva Mcleamy sobre el esfuerzo del proceso constructivo



Fuente: Introducción a la metodología BIM (Jesus González Marquez & Soler, 2014)

2.3. IFC

Las IFC (Industry Foundation Classes) son un esquema de datos abierto y neutral adoptado internacionalmente bajo la norma ISO 16739-1:2024, que describe objetos, propiedades y relaciones dentro de modelos BIM, permitiendo que distintos softwares intercambien de forma interoperable geometría, metadatos y estructura del proyecto completo.

En cuanto a versiones y cómo han cambiado: IFC 2x3 (2006) es muy usado aún por su compatibilidad y amplia implementación internacional, pero tiene limitaciones en geometría avanzada, interoperabilidad de clases e infraestructura, IFC 4 (y sus addendums, hasta el actual IFC 4.3 Addendum 2) corrige muchas de esas limitaciones: mejora la definición de, amplía soporte para infraestructura (vías, puentes, alineamientos), mejora consistencia de esquema, optimización de propiedades y capacidad de exportación/importación más fiable.

Se puede decir que IFC sirve para lograr interoperabilidad en proyectos BIM. Es un estándar abierto, que define cómo representar y compartir información de un modelo digital de construcción de manera **neutral** e independiente del software. Hoy se recomienda usar IFC 4.3.2.0 como estándar

para nuevos proyectos debido a esas mejoras, aunque muchos proyectos siguen en IFC 2x3 por cuestiones de compatibilidad, herramientas disponibles, certificaciones y costumbre.

2.4. Automatización en BIM

2.4.1. Automatización de Procesos BIM

La automatización de procesos BIM se refiere a su integración con varios procesos automatizados para mejorar la eficiencia en los proyectos de construcción (Tyurin, 2020).

Este enfoque busca usar la tecnología para minimizar el tiempo invertido en realizar tareas repetitivas, usando como principal herramienta la información generada en los modelos de información BIM, la automatización BIM es un campo emergente que tiene como objetivo optimizar y mejorar los procesos de construcción mediante la integración de BIM y sistemas automatizados. Ofrece posibles beneficios en la verificación de cumplimiento, reduciendo las inconsistencias causadas por la interpretación humana (Reinhardt & Mathews, 2017).

2.5. Softwares BIM (Revit)

Revit es un software de modelado de información BIM para la construcción desarrollado por Autodesk, se utiliza principalmente para el diseño, la planificación y el análisis de edificaciones y otras estructuras. Permite a los profesionales del sector de la construcción crear modelos precisos y detallados que contienen información sobre cada componente de una construcción, lo que facilita la coordinación entre diferentes disciplinas y reduce los errores durante el proceso de diseño y construcción. (Autodesk, 2024)

Este programa permite a arquitectos, ingenieros y profesionales de la construcción diseñar, modelar y documentar proyectos de edificación en un entorno tridimensional y paramétrico. A diferencia de los sistemas de diseño asistido por computadora (CAD) tradicionales, Revit integra información geométrica y no geométrica, facilitando la coordinación y colaboración entre los distintos actores involucrados en un proyecto. Además, Revit ofrece herramientas para la planificación, análisis y simulación de edificaciones, lo que

contribuye a una gestión más eficiente y precisa durante todo el ciclo de vida del proyecto.

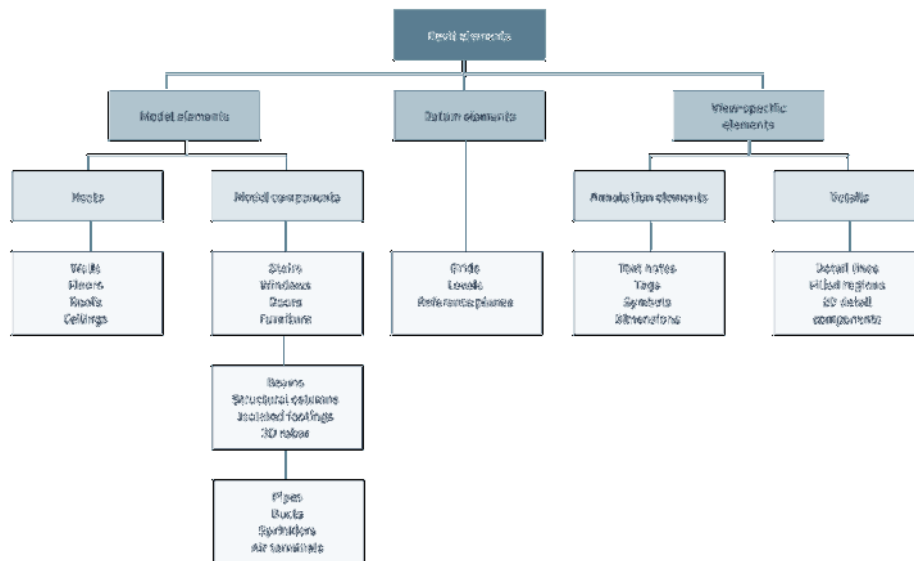
2.5.1. Entidades de Revit

En el contexto de Revit, es fundamental comprender las entidades clave que estructuran el modelo de información del edificio. A continuación, se detallan las principales entidades:

- **Categoría:** Clasificación amplia y predefinida en Revit que agrupa familias de elementos con características y comportamientos comunes. Las categorías determinan reglas y propiedades compartidas por las familias que contienen, ejemplo: "Muros", "Puertas" y "Ventanas" son categorías que agrupan familias relacionadas.
- **Familia:** Conjunto de elementos dentro de una categoría que comparten un conjunto común de propiedades (parámetros) y una representación gráfica relacionada. Las familias pueden ser de sistema, cargables o in situ, ejemplo: dentro de la categoría "Puertas", una familia podría ser "Puerta de madera de una hoja".
- **Tipo:** Variaciones dentro de una familia que tienen un conjunto específico de propiedades. Los tipos permiten definir diferentes configuraciones o tamaños de un elemento sin crear nuevas familias, ejemplo: dentro de la familia "Puerta de madera de una hoja", los tipos podrían ser "Puerta de 0.80 x 2.10 m" y "Puerta de 0.90 x 2.10 m".
- **Elemento (Ejemplar):** Instancia específica de un tipo colocada en el modelo. Los elementos representan objetos físicos o abstractos en el proyecto y pueden ser de modelo, de referencia o específicos de vista, ejemplo: Una "Puerta de 0.80 x 2.10 m" ubicada en la entrada principal es un elemento o ejemplar específico en el modelo.

Estas definiciones y clasificaciones basadas en IFC son esenciales para una comprensión profunda de la estructura y organización de los elementos en Revit, facilitando una gestión eficiente y coherente del modelo de información del edificio (Autodesk, 2024).

Figura 8. Organización de elementos de Revit

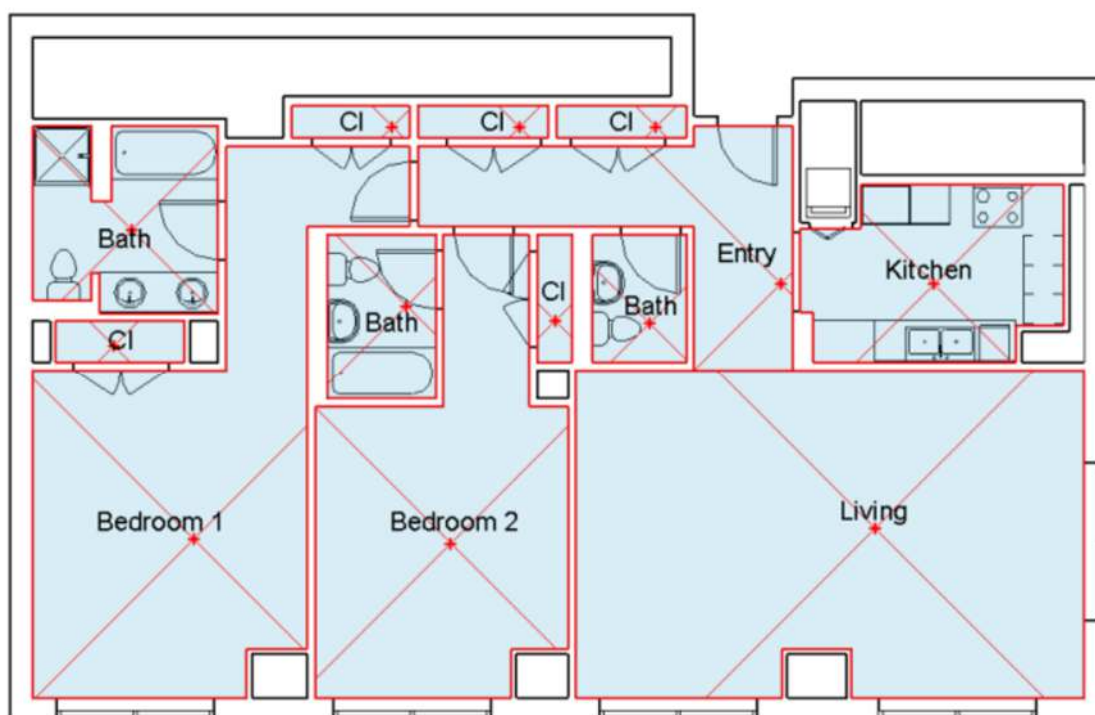


Fuente: Autodesk 2024

2.5.1.1. Room o Habitación

En Revit una de las principales entidades o elementos son los denominados “Room” o “Habitación”, lo cual es una subdivisión del espacio dentro de un modelo de edificio, definida por elementos como paredes, pisos, techos y cielorrasos, los cuales actúan como delimitadores. Revit puede obtener mucha información utilizando estos elementos, así como el perímetro, el área y el volumen.

Figura 9. Muestra de entidades de tipo "Room" software Revit



Fuente: Autodesk 2024

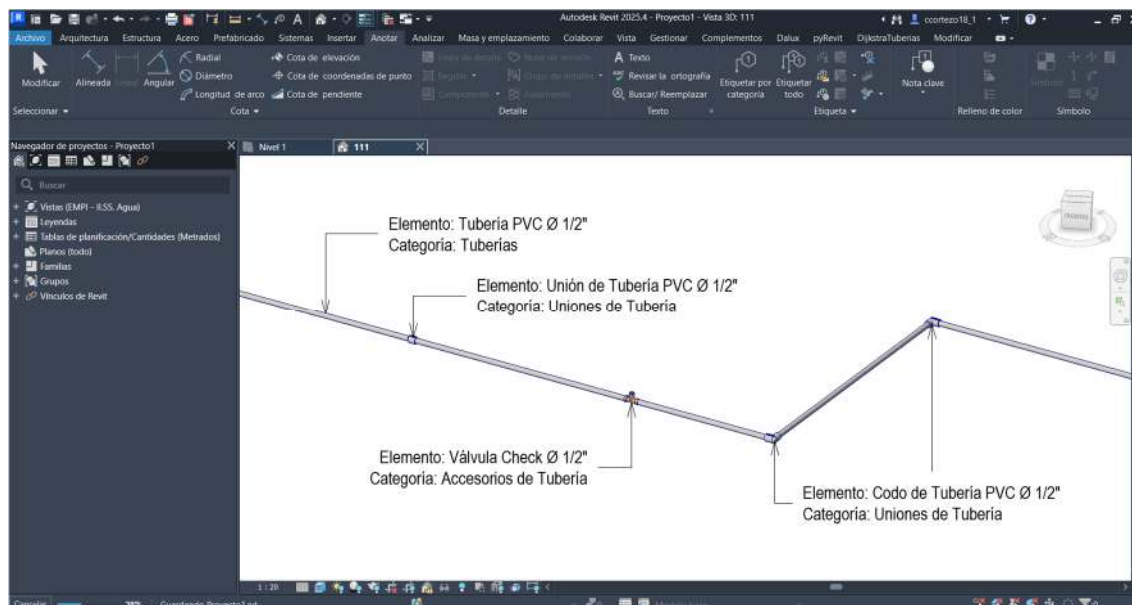
2.5.1.2. Pipe o Tubería

Para modelar tuberías que integran distintos sistemas dentro de una edificación, tales como agua fría, agua caliente, desagüe, agua contra incendios, entre otros, el software Revit emplea la categoría denominada "Pipe" o "Tubería". Esta categoría es fundamental dentro del entorno MEP (Mechanical, Electrical and Plumbing), ya que agrupa todos los elementos lineales que representan los conductos por donde fluye el agua u otro fluido.

Además, esta categoría no actúa de forma aislada, sino que se encuentra estrechamente relacionada con otros componentes necesarios para el ensamblaje y correcto funcionamiento del sistema de tuberías. Entre estos se incluyen las conexiones, uniones, codos, tees, válvulas y otros elementos que permiten dirigir, ramificar o controlar el flujo dentro de la red. Estos componentes adicionales se clasifican en categorías específicas dentro de Revit, como "Uniones de tubería" (Pipe Fittings) y "Accesorios de tubería" (Pipe Accessories), lo que facilita su manipulación paramétrica, conexión lógica y validación geométrica en el entorno de modelado BIM.

Gracias a esta estructura categorizada, Revit permite una representación precisa y coherente de las redes hidráulicas y sanitarias, asegurando su correcta integración dentro del modelo BIM del edificio.

Figura 10. Ejemplo de tubería, accesorio de tubería y unión de tubería en Revit



Fuente: Obtenido del software Revit 2025

2.5.1.3. Aparato Sanitario

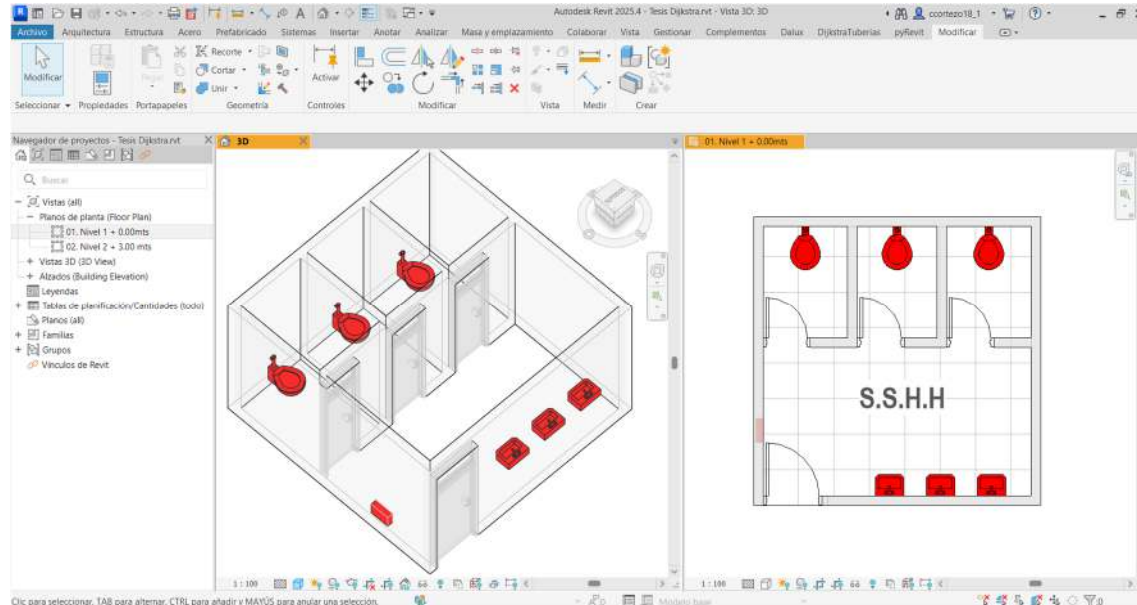
En Revit, los aparatos sanitarios se agrupan dentro de la categoría denominada “Plumbing Fixtures” en inglés. Esta categoría incluye todos aquellos elementos que forman parte del sistema de plomería y que interactúan directamente con el usuario o con el ambiente, como lavatorios, inodoros, urinarios, duchas, lavaderos, entre otros.

Estos elementos son fundamentales en el diseño de instalaciones sanitarias, ya que representan los puntos de consumo o descarga dentro del sistema. En el entorno BIM de Revit, cada aparato sanitario no solo tiene una representación gráfica tridimensional, sino que también contiene información paramétrica relevante, como el tipo de conexión (entrada de agua, salida de desagüe), caudales, presión requerida, alturas de instalación, entre otros datos técnicos que permiten una correcta coordinación y cálculo del sistema.

Además, al estar integrados en el modelo, los aparatos sanitarios pueden conectarse directamente a las redes de tuberías, facilitando el diseño automático

o asistido del sistema hidráulico y permitiendo análisis como la detección de interferencias o el cálculo de caudales y presiones dentro del modelo.

Figura 11. Ejemplo de elementos pertenecientes a la categoría "Aparatos sanitarios" en el software Revit



Fuente: Obtenido del software Revit 2025

Nota. Se muestra una captura de un proyecto de ejemplo de modelado desarrollado en el software Revit, en rojo se ven resaltados los elementos de la categoría "Aparatos Sanitarios", a la izquierda se muestra una vista en Isométrico y a la derecha una vista en planta.

2.5.2. Revit API

API, o interfaz de programación de aplicaciones, es un componente crucial en informática que permite a las aplicaciones solicitar servicios de bibliotecas y sistemas operativos, las API se pueden utilizar para crear aplicaciones innovadoras que mejoren la experiencia del usuario (E. Tollerud, 2016).

Revit API es una interfaz de programación que permite a los desarrolladores y usuarios avanzados interactuar y personalizar el software Autodesk Revit. A través de Revit API, es posible crear aplicaciones o scripts que automatizan tareas repetitivas, mejoran flujos de trabajo específicos, o integran Revit con otros sistemas y bases de datos externas.

Esta API proporciona acceso a las funcionalidades internas de Revit, permitiendo manipular elementos de modelos BIM, realizar cálculos, modificar

propiedades, generar informes, y realizar análisis complejos que no son posibles con las herramientas estándar de la interfaz gráfica. De esta manera, Revit API se convierte en un recurso esencial para extender y personalizar el uso de Revit, facilitando una optimización avanzada del modelado y la gestión de información en proyectos de construcción.

Al aprovechar las capacidades de Revit API, los profesionales pueden desarrollar soluciones automatizadas y personalizadas que mejoran la eficiencia y precisión en el diseño, alineándose con la necesidad de adoptar procesos de construcción más integrados y tecnológicamente avanzados, como lo promueve la metodología BIM.

2.5.3. pyRevit

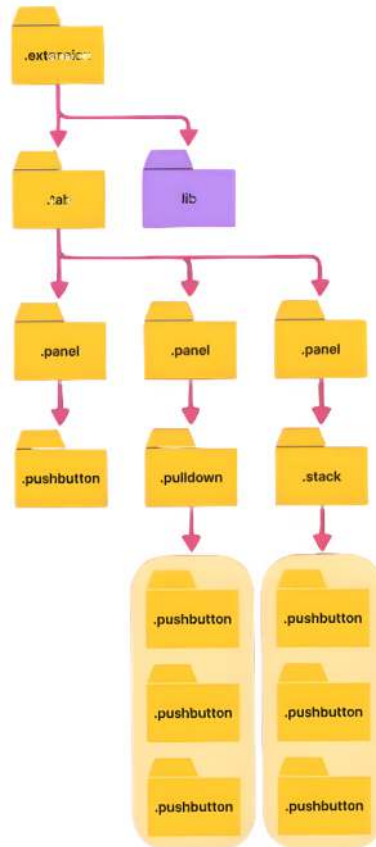
Es una extensión de código abierto para Autodesk Revit que facilita la automatización, creación de scripts y desarrollo de herramientas dentro del software. Su función principal es actuar como un “entorno de Desarrollo Rápido de Aplicaciones”, permitiendo que arquitectos, ingenieros y diseñadores personalicen y optimicen sus flujos de trabajo mediante el uso de lenguajes de programación como Python, C# y VB.Net.

Esta herramienta ofrece una variedad de funcionalidades preconfiguradas, tales como procesamiento masivo de tareas, modificaciones rápidas, mejoras en la visualización y auditoría avanzada de modelos. Gracias a estas características, los usuarios pueden “optimizar su tiempo, estandarizar procesos BIM y mejorar la eficiencia en el desarrollo de proyectos”. Además, pyRevit brinda un entorno de scripting adaptable, lo que permite la creación de comandos y automatizaciones personalizadas sin requerir conocimientos profundos en programación.

Uno de los aspectos más destacados de pyRevit es su integración intuitiva con la interfaz de usuario de Revit, lo que posibilita la personalización de paneles, incluyendo redes de tuberías para sistemas de agua potable, desagües y sistemas contra incendios. Para crear una extensión personalizada sobre la base de pyRevit se tiene que organizar archivos de manera específica e indicar la ruta

de acceso a los mismos. A grandes rasgos la organización de los archivos para la creación de una extensión se ve de la siguiente manera:

Figura 12. Organización de directorios de pyRevit para la creación de una extensión



Fuente: Obtenido de “The Begginers Guide for Revit API” (Erick Frits, 2024).

2.5.4. Ejemplos de Automatización

La implementación madura de BIM en proyectos de construcción ha estandarizado la información, permitiendo un análisis detallado de tiempos, costos e infraestructura. Aunque estos datos son valiosos, su gestión se ha vuelto tediosa y repetitiva. Por ello, la automatización emerge como una solución para mejorar la eficiencia y reducir los costos operativos (Konstruedu, 2021).

Entre las múltiples aplicaciones de automatización BIM, podemos encontrar aplicaciones para la generación automática de la documentación, extracción automatizada de mediciones y metrados, creación de geometrías complejas, corrección automática de interferencias en los modelos, extracción e importación de datos, automatización de concepción arquitectónica; además de

múltiples herramientas que nos permiten automatizar el procedimiento de modelado en áreas específicas como el modelado de refuerzo de acero en elementos de concreto armado o el modelado de encofrados basados en elementos preexistentes.

2.5.5. Softwares de Automatización de Modelado de Tuberías

Existen algunos softwares que hacen un acercamiento al problema desarrollado en esta investigación, entre los cuales tenemos:

- AutoSPRINK VR: Aunque está orientado principalmente al diseño de sistemas contra incendios, AutoSPRINK VR ofrece capacidades para el trazado automático de tuberías en función de puntos de conexión y especificaciones del sistema. Su motor de enrutamiento automático facilita la generación de rutas eficientes, aunque su aplicación en sistemas de agua potable puede requerir adaptaciones específicas.
- Bentley OpenFlows WaterGEMS, este software de Bentley Systems está enfocado en el modelado y simulación hidráulica de redes de distribución de agua. Permite generar automáticamente rutas de tuberías basadas en análisis hidráulicos detallados, optimizando el diseño para cumplir con los requerimientos de presión y caudal. Su integración con plataformas BIM facilita la incorporación de datos hidráulicos en el modelo general del proyecto.
- Hydros, desarrollado por AltoQi, Hydros es un software especializado en proyectos hidráulicos que realiza el modelado automático de redes de agua potable. Calcula el trazado de las tuberías considerando la ubicación de las fuentes de agua y los puntos de consumo, optimizando el diseño según las normas y especificaciones técnicas vigentes.
- Cypecad MEP: Este software, desarrollado por CYPE Ingenieros, está diseñado para la modelación y cálculo de instalaciones de edificios,

2.5.6. HidraSoftware

HidraSoftware se describe como un software desarrollado principalmente para realizar cálculos hidráulicos repetitivos de manera automatizada, cuenta con un apartado que permite instalar un complemento para Autodesk® REVIT

MEP que realiza el cálculo y diseño hidráulico de instalaciones de abastecimiento de agua en edificaciones directamente dentro del entorno BIM. Ofrece opciones para el cálculo de pérdidas por fricción, selección de accesorios y generación de tablas de resultados y vistas anotadas. Toda la información que el software requiere para realizar el cálculo hidráulico, la obtiene del modelo de información BIM.

2.6. Algoritmo

2.6.1. Definición

Un algoritmo es un procedimiento computacional bien definido utilizado para resolver un problema, compuesto por una secuencia finita de instrucciones ejecutables por una computadora. Toma ciertos valores como entrada y, tras realizar una serie de cálculos, produce valores de salida (Manelli, 2020).

Un algoritmo, en su definición moderna, es un conjunto de reglas o procedimientos estructurados que operan como un mecanismo para transformar entradas (datos iniciales) en salidas (resultados deseados), utilizando un flujo secuencial o iterativo de operaciones. Se puede conceptualizar como una combinación de máquinas de estados abstractas y procesos de cálculo, donde cada paso se define con precisión y puede ejecutarse de manera eficiente por un sistema computacional. Los algoritmos están diseñados no solo para resolver problemas específicos, sino también para abordar una amplia gama de situaciones relacionadas, haciendo uso de estructuras como ciclos, condicionales y toma de decisiones para optimizar los resultados.

2.6.2. Importancia en la Solución de Problemas Complejos

Algunos algoritmos, como los algoritmos de optimización metaheurística han surgido como herramientas efectivas para manejar espacios de solución no lineales, no convexos y discontinuos en problemas de ingeniería del mundo real, estos algoritmos se han aplicado con éxito en varios campos de la ingeniería civil, incluido el diseño estructural, la gestión de la construcción y los sistemas de transporte (Kashani et al., 2020).

Los algoritmos son cruciales en la ingeniería civil para abordar y resolver problemas complejos de manera eficiente. Permiten automatizar cálculos

estructurales, optimizar el diseño de redes de infraestructura como tuberías y carreteras, y gestionar grandes volúmenes de datos en proyectos de construcción. Al aplicar algoritmos avanzados, los ingenieros pueden analizar múltiples variables simultáneamente, encontrar soluciones óptimas en tiempo real y minimizar errores, lo que mejora la precisión y eficacia en la planificación y ejecución de obras.

2.7. Algoritmo Dijkstra

2.7.1. Historia y Origen

El algoritmo de Dijkstra, un concepto fundamental en informática fue desarrollado por el matemático holandés Edsger Dijkstra en 1956. El algoritmo encuentra de manera eficiente la ruta más corta entre nodos en un grafo con pesos de arista no negativos. Originalmente propuesto en 1959, también se lo conoce como el "algoritmo de etiqueta" (Reddy, 2013). Este algoritmo trabaja sobre grafos ponderados con aristas de peso no negativo y permite determinar la distancia mínima entre un nodo de origen y todos los demás nodos del grafo. Su funcionamiento se basa en un proceso iterativo: en cada paso selecciona el nodo no visitado con la menor distancia provisional, lo marca como definitivo y actualiza las distancias de sus vecinos si encuentra rutas más cortas a través de él. De esta manera, construye progresivamente un conjunto de nodos con la distancia mínima ya calculada, asegurando en todo momento la corrección del resultado.

Sus aplicaciones son muy extendidas, incluido el enrutamiento de estado de enlace en redes de computadoras. La importancia del algoritmo es evidente en su uso y adaptación continuos, como su extensión para crear algoritmos de búsqueda de rutas más avanzados como A*. El algoritmo de Dijkstra sigue siendo una herramienta crucial en varios campos, lo que demuestra su perdurable relevancia para resolver problemas de ruta más corta.

2.7.2. Definición y Funcionamiento

El algoritmo de Dijkstra es un método ampliamente utilizado para encontrar la ruta más corta entre nodos en un grafo. Funciona seleccionando iterativamente el vértice no visitado con el costo tentativo más bajo, agregándolo

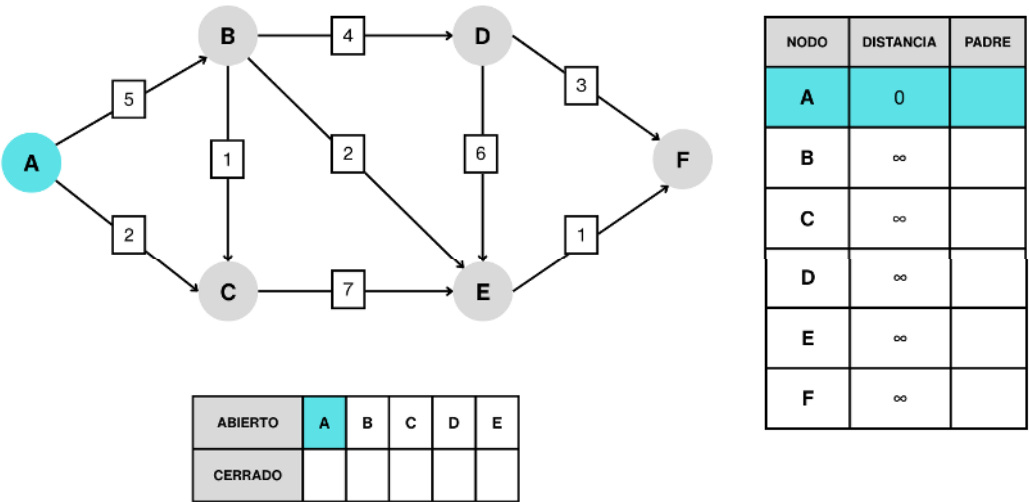
al conjunto visitado y actualizando los costos de los vértices adyacentes. (Javaid, 2013)

Explicándolo paso a paso tenemos:

1. Inicialización: Se asigna una distancia inicial de 0 al nodo de origen (el punto de partida) y una distancia infinita a todos los demás nodos. Se crea un conjunto de nodos no visitados que incluye todos los nodos del grafo.
2. Selección del nodo actual: Se elige el nodo con la distancia mínima que aún no ha sido visitado. Al principio, este será el nodo de origen.
3. Actualización de las distancias: Para cada nodo vecino del nodo actual, se calcula la distancia desde el origen a través del nodo actual. Si esta nueva distancia es menor que la distancia previamente registrada para el nodo vecino, se actualiza la distancia más corta para ese nodo.
4. Marcar el nodo como visitado: Una vez que se han actualizado las distancias de todos los vecinos del nodo actual, este se marca como visitado y se elimina del conjunto de nodos no visitados. Un nodo visitado no se vuelve a revisar.
5. Repetir el proceso: Se repiten los pasos 2 a 4 hasta que todos los nodos hayan sido visitados o se haya alcanzado el nodo de destino (si solo se busca la distancia más corta a un nodo específico).
6. Resultado: Cuando el algoritmo termina, se ha encontrado la distancia más corta desde el nodo de origen a todos los demás nodos, y se puede reconstruir la ruta más corta siguiendo las actualizaciones realizadas.

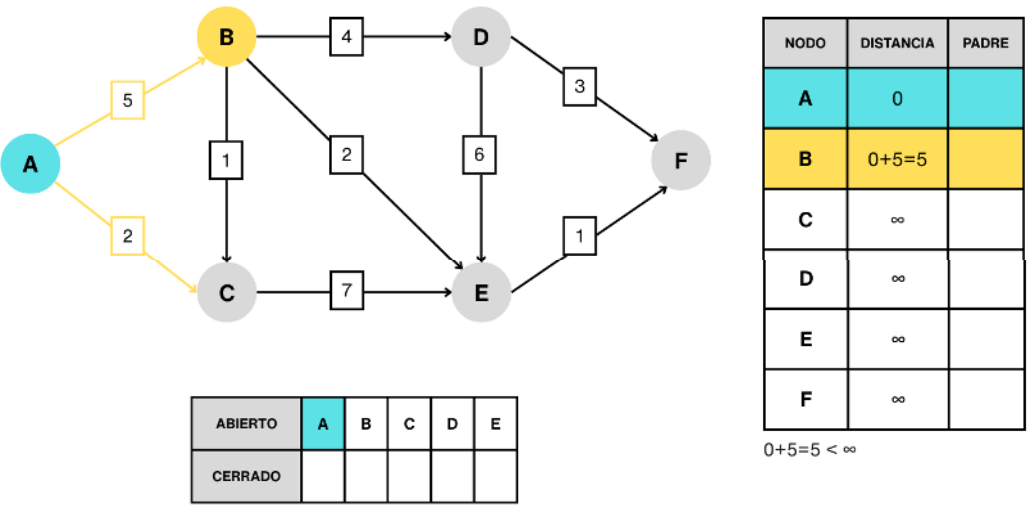
Esto se puede visualizar de mejor manera en el siguiente ejemplo:

Figura 13. Ejemplo de aplicación del algoritmo Dijkstra paso 1: inicialización del algoritmo, nodo de origen seleccionado.



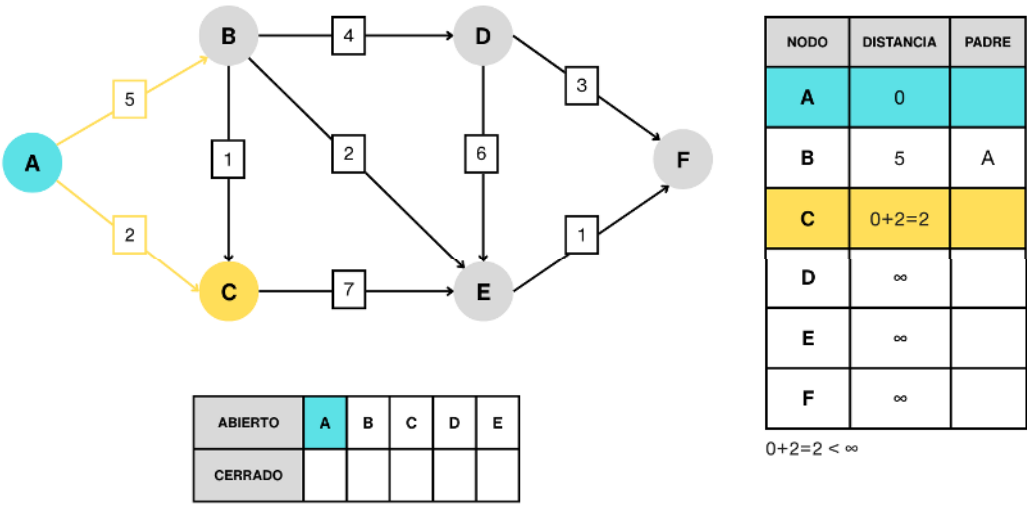
PASO 1

Figura 14. Ejemplo de aplicación del algoritmo Dijkstra paso 2: selección del nodo con menor distancia provisional.



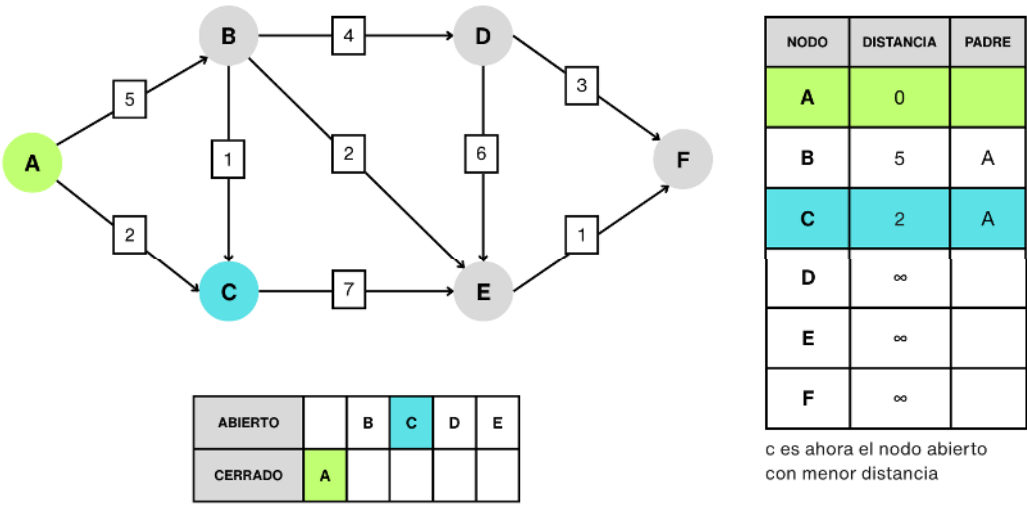
PASO 2

Figura 15. Ejemplo de aplicación del algoritmo Dijkstra paso 3: Actualización de distancias a los vecinos del nodo actual.



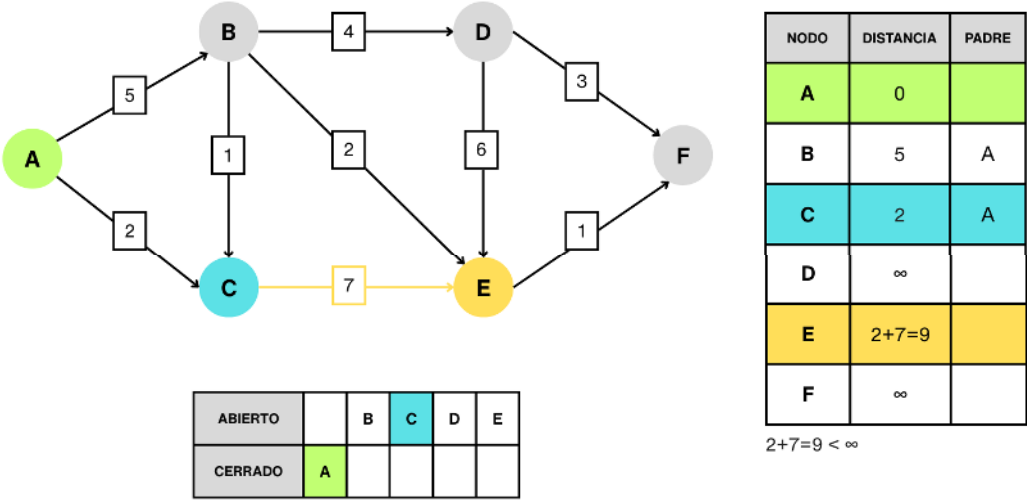
PASO 3

Figura 16. Ejemplo de aplicación del algoritmo Dijkstra paso 4: marcado del nodo como visitado.



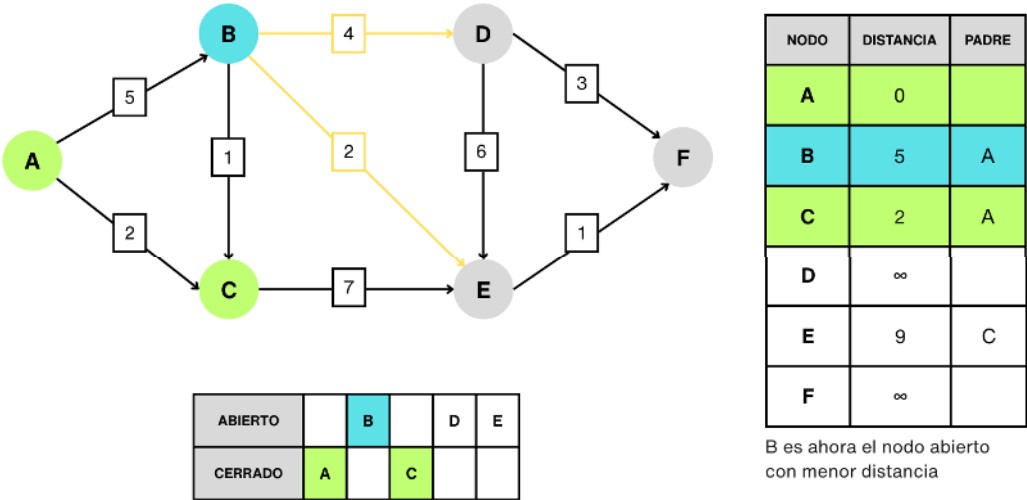
PASO 4

Figura 17. Ejemplo de aplicación del algoritmo Dijkstra paso 5: selección del siguiente nodo con menor distancia provisional.



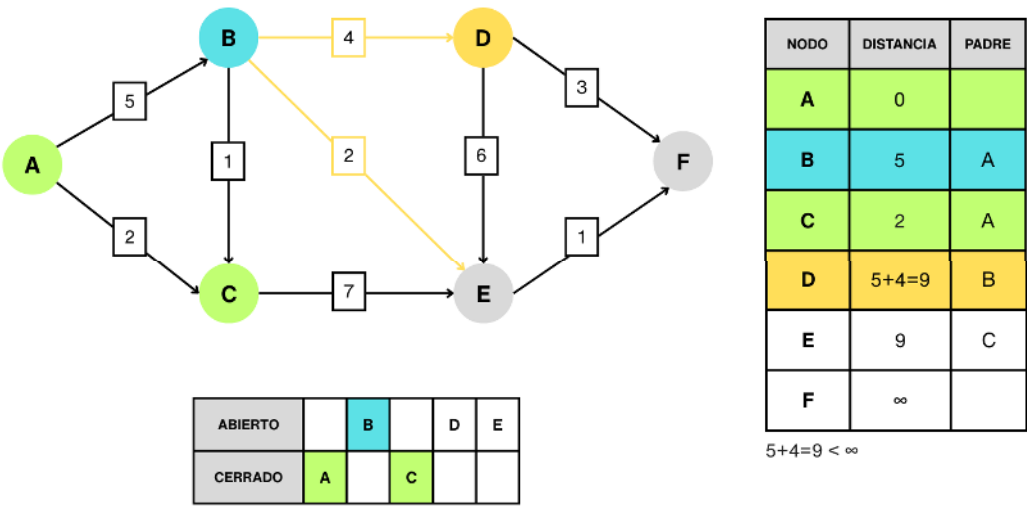
PASO 5

Figura 18. Ejemplo de aplicación del algoritmo Dijkstra paso 6: actualización de nuevas distancias a los vecinos.



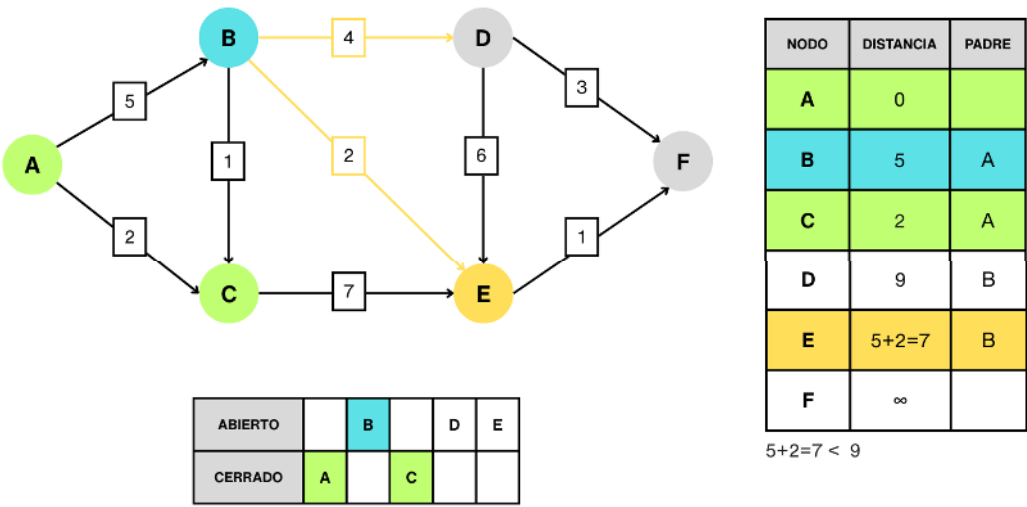
PASO 6

Figura 19. Ejemplo de aplicación del algoritmo Dijkstra paso 7: revisión de nodos visitados y pendientes.



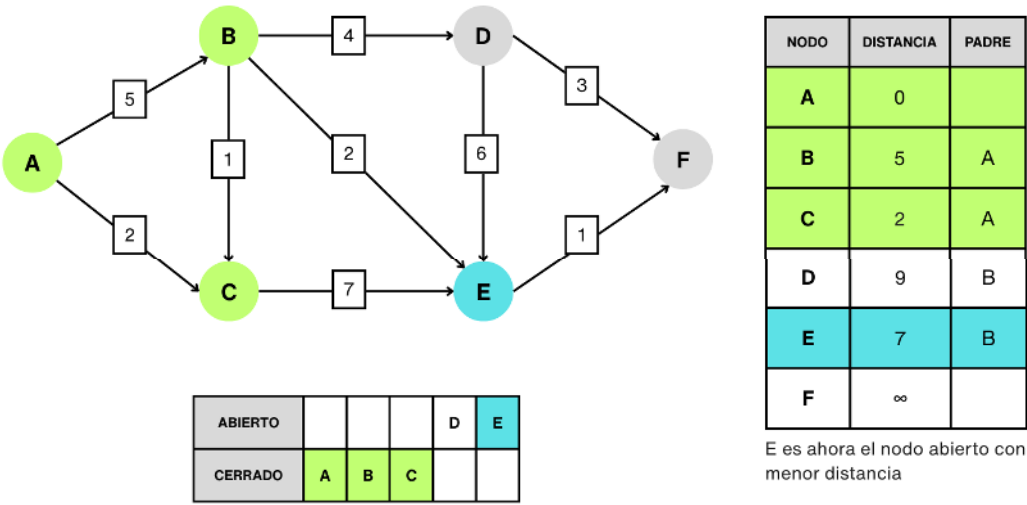
PASO 7

Figura 20. Ejemplo de aplicación del algoritmo Dijkstra paso 8: continuación del proceso de selección y actualización.



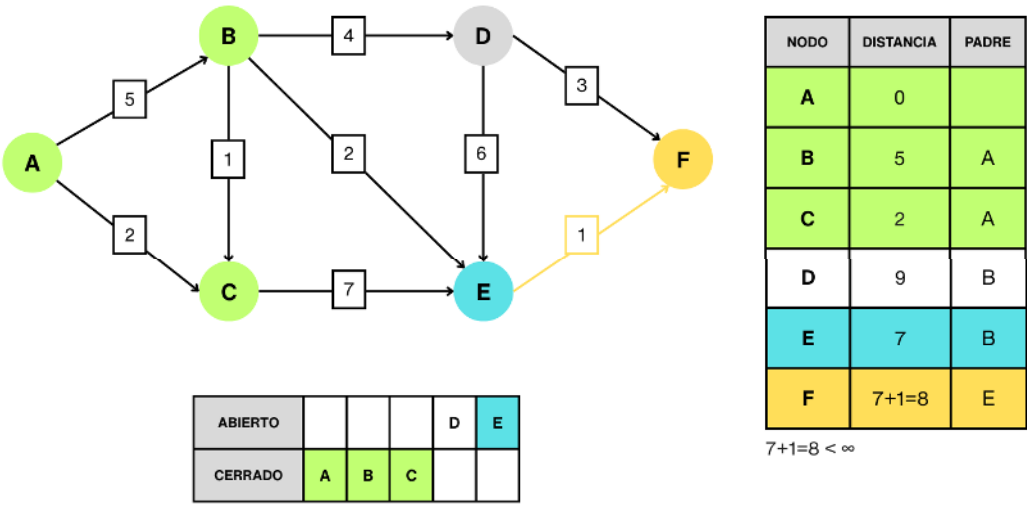
PASO 8

Figura 21. Ejemplo de aplicación del algoritmo Dijkstra paso 9: cálculo progresivo de las rutas más cortas.



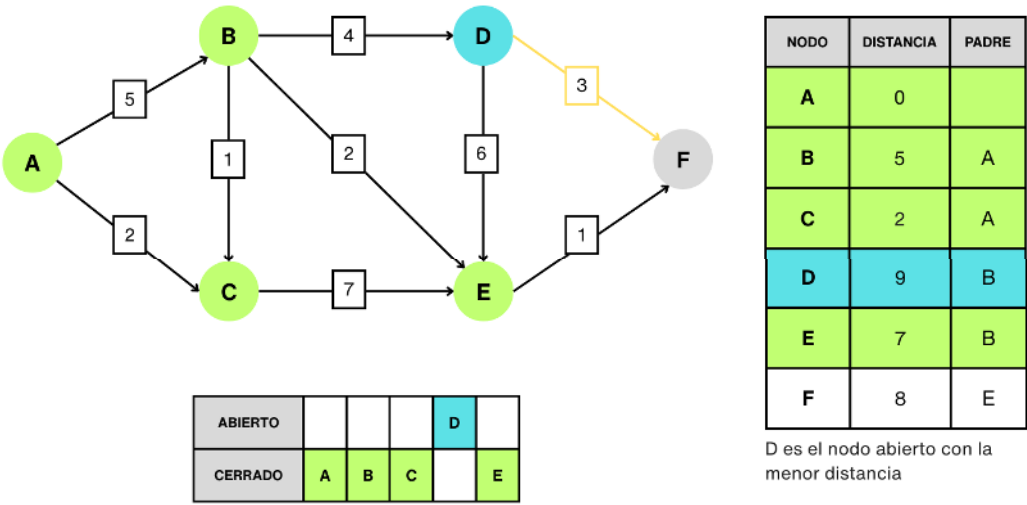
PASO 9

Figura 22. Ejemplo de aplicación del algoritmo Dijkstra paso 10: consolidación de distancias mínimas en nodos intermedios.



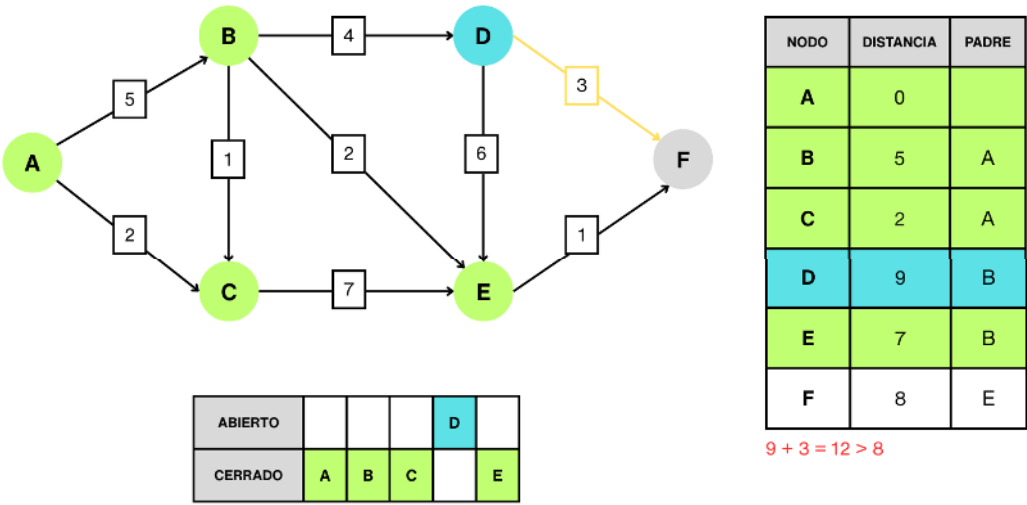
PASO 10

Figura 23. Ejemplo de aplicación del algoritmo Dijkstra paso 11: avance hacia la ruta óptima desde el origen.



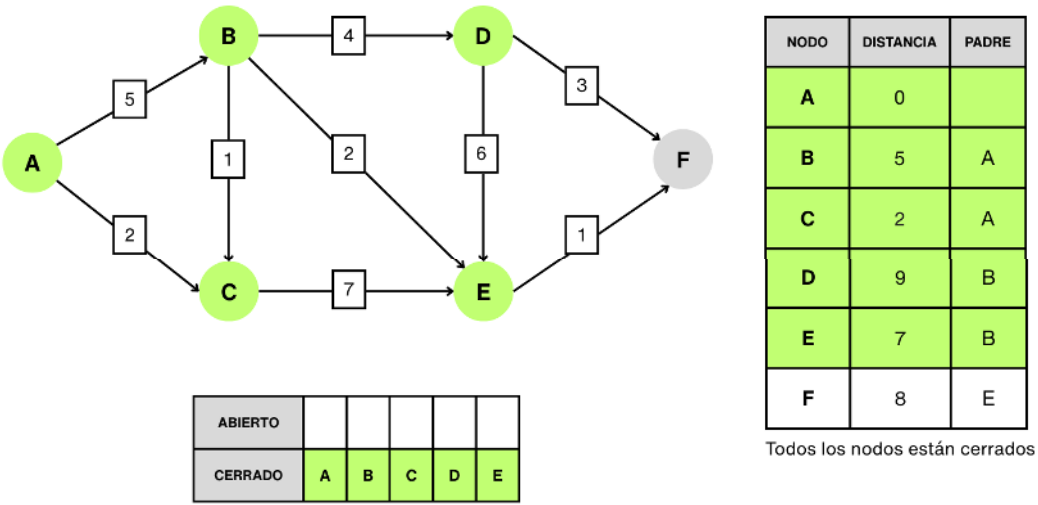
PASO 11

Figura 24. Ejemplo de aplicación del algoritmo Dijkstra paso 12: refinamiento de las distancias más cortas restantes.



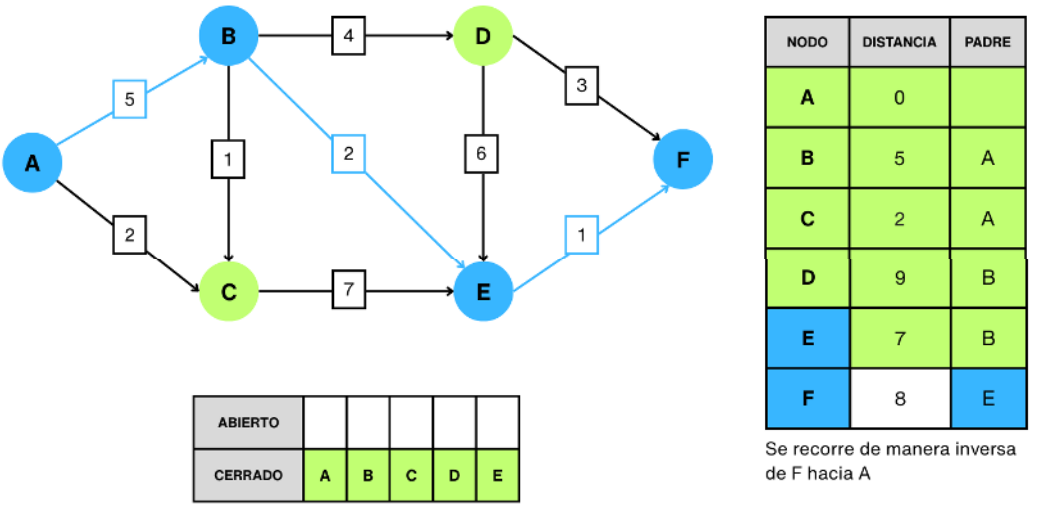
PASO 12

Figura 25. Ejemplo de aplicación del algoritmo Dijkstra paso 13: últimas actualizaciones antes del resultado final.



PASO 13

Figura 26. Ejemplo de aplicación del algoritmo Dijkstra paso 14: ruta más corta determinada por el algoritmo.



PASO 14

2.7.3. Aplicaciones

El algoritmo de Dijkstra es un método ampliamente utilizado en teoría de grafos para resolver problemas de rutas más cortas, y tiene aplicaciones significativas en ingeniería y ciencias de la computación. Se ha empleado en distintos campos, como las comunicaciones, el transporte, la inteligencia artificial y los sistemas de información geográfica. Los investigadores han desarrollado optimizaciones para mejorar su eficiencia, enfocándose en una mejor selección de nodos y en el uso de estructuras de almacenamiento de datos más efectivas.

El algoritmo ha demostrado ser útil en situaciones prácticas, como calcular las distancias más cortas entre edificios y desarrollar sistemas de rastreo de viajes urbanos. Gracias a su simplicidad y versatilidad, se ha convertido en un elemento esencial para abordar problemas complejos en varias disciplinas de la ingeniería (Win & Thein, 2020).

2.7.4. Ventajas y Limitaciones

Los beneficios del algoritmo incluyen su capacidad para encontrar rutas eficientes entre dos puntos y su versatilidad para manejar diferentes tamaños de grafos. Sin embargo, tiene algunas limitaciones, como el requisito de pesos de arista no negativos y dificultades en la paralelización, con solo un 10% de aceleración promedio lograda a través de la ejecución paralela. Los investigadores han propuesto mejoras al algoritmo, incluidos métodos para manejar las penalizaciones por transferencia de vértices sin expansión de grafos y enfoques para aumentar la cantidad de nodos sin degradar el rendimiento. A pesar de sus desafíos, el algoritmo de Dijkstra sigue siendo una herramienta fundamental para resolver problemas de ruta más corta de una sola fuente en grafos (Jaiswal & Chakrawarti, 2013).

2.8. Grafos

2.8.1. Definición de Grafos

Un grafo es una estructura matemática que consiste en vértices (o puntos) y aristas (o líneas) que conectan pares de vértices (Bin Xiong et al., 2020).

Se utiliza para modelar relaciones entre objetos, está compuesto por un conjunto de vértices (o nodos) y un conjunto de aristas (o enlaces) que conectan

pares de vértices. Los grafos se utilizan ampliamente en áreas como la informática, las matemáticas y la ingeniería para representar redes, como redes sociales, mapas de carreteras, circuitos eléctricos y más.

Si las aristas tienen una dirección asignada, se llama grafo dirigido, si las aristas no tienen una dirección específica, se llama grafo no dirigido. El estudio de los grafos se conoce como teoría de grafos y es una rama importante de la matemática discreta.

2.8.2. Aplicaciones

La teoría de grafos tiene diversas aplicaciones en múltiples campos. En informática, se utiliza para la agrupación de documentos web, la criptografía y el análisis de algoritmos (Majeed & Rauf, 2020). Los grafos son esenciales para modelar varias redes, incluidas las redes informáticas, inalámbricas, biológicas y sociales. En ingeniería eléctrica, la teoría de grafos se aplica a las redes de comunicación y la teoría de la codificación. También encuentra aplicaciones en bioquímica para la genómica y en la investigación de operaciones para problemas de programación. Los métodos de teoría de grafos se utilizan cada vez más para representar conjuntos de datos con estructuras irregulares, como los adquiridos por redes de sensores o en el aprendizaje automático (Adalı & Ortega, 2018). Además, la teoría de grafos tiene aplicaciones en la secuenciación de ADN, la seguridad de redes informáticas y la asignación de frecuencias en redes de telefonía móvil GSM (Sachdeva, 2012).

2.9. Definición de Términos Básicos

2.9.1. Expediente Técnico

En el marco normativo peruano, el Reglamento Nacional de Edificaciones (RNE), a través de la Norma Técnica G.040 “Definiciones”, establece que el expediente técnico es el conjunto de documentos que determinan de manera explícita las características y requisitos del proyecto, así como las especificaciones técnicas necesarias para la ejecución de la obra, integrado por planos por especialidades, memorias descriptivas, especificaciones técnicas y estudios técnicos específicos, cuando la naturaleza de la obra así lo requiera (R.M. N°029-2021-VIVIENDA).

Por su parte, el Reglamento de la Ley de Contrataciones del Estado (D.S. N°344-2018-EF) define el Expediente Técnico de Obra como el conjunto de documentos técnicos y económicos que permiten la adecuada ejecución de la obra, comprendiendo memoria descriptiva, especificaciones técnicas, planos de ejecución, metrados, presupuesto de obra, análisis de precios, calendario de avance valorizado, fórmulas polinómicas y, de ser necesario, estudios de suelos, geológicos, ambientales u otros complementarios.

Además, el expediente técnico debe contener un presupuesto que desglosa los costos asociados con cada componente del proyecto, un cronograma que define las etapas y plazos de ejecución, y, si corresponde, las fórmulas de reajuste de precios para garantizar el control financiero. En esencia, el expediente técnico es una herramienta indispensable para asegurar que la obra se realice cumpliendo con las normativas vigentes, dentro de los parámetros de calidad, tiempo y costo establecidos, y sirve de guía para la supervisión y planificación efectiva de todas las actividades del proyecto. (Ministerio de Vivienda Construcción y Saneamiento - Perú, 2021).

2.9.2. Metrado

El metrado se concibe como el cálculo o cuantificación por partidas de la cantidad de obra a ejecutar, de acuerdo con la unidad de medida establecida. La Norma Técnica de Metrados para Obras de Edificación y Habilitaciones Urbanas (R.D. N.° 073-2010-VIVIENDA) lo define expresamente como “el cálculo o la cuantificación por partidas de la cantidad de obra a ejecutar, en conformidad con el Reglamento de la Ley de Contrataciones del Estado”. De manera concordante, el Reglamento de la Ley de Contrataciones del Estado (D.S. N.° 344-2018-EF) incorpora en su glosario la misma definición, subrayando su carácter oficial y normativo.

Si bien el Reglamento Nacional de Edificaciones (RNE), aprobado por D.S. N.° 011-2006-VIVIENDA, no formula una definición propia, exige que los expedientes técnicos consignen las partidas con sus respectivos metrados conforme a criterios técnicos estandarizados. En este sentido, la normativa reconoce que el metrado constituye la base técnica para elaborar presupuestos,

análisis de precios unitarios y valorizaciones de obra, garantizando así precisión en la programación y control de los proyectos.

En consecuencia, los distintos cuerpos normativos coinciden en entender el metrado como la medición cuantitativa y detallada de la obra por partidas y unidades, constituyéndose en un componente esencial del proceso técnico y económico de toda construcción en el país.

2.9.3. Presupuesto de Obra

El presupuesto de obra se define como el cálculo económico integral del costo total de un proyecto de construcción, incorporado en el expediente técnico. De acuerdo con el Reglamento de la Ley de Contrataciones del Estado (D.S. N.º 344-2018-EF), este documento debe contemplar tributos, seguros, transporte, inspecciones, pruebas, medidas de seguridad y costos laborales, además de contar con la suscripción de los profesionales responsables de su elaboración. En esencia, constituye el cuadro de costos finales donde se valoriza la obra mediante partidas y precios unitarios previamente aprobados.

Asimismo, la Norma Técnica de Metrados para edificaciones (R.D. N.º 073-2010-VIVIENDA) dispone que el expediente técnico incluya, entre otros componentes, el presupuesto (también denominado valor referencial) junto con la memoria descriptiva, especificaciones técnicas, planos, metrados, análisis de precios, cronograma de avance valorizado, fórmulas polinómicas y estudios complementarios. En concordancia, el Reglamento Nacional de Edificaciones (D.S. 011-2006-VIVIENDA) establece que el presupuesto definitivo aprobado constituye requisito indispensable para la obtención de licencias de construcción, consolidando su carácter normativo y vinculante dentro del proceso de planificación y ejecución de obras.

Finalmente, en el marco del Sistema Nacional de Programación Multianual y Gestión de Inversiones, el presupuesto de obra se asume como el valor referencial del componente “obra” de los proyectos de inversión. Este valor no solo orienta la programación multianual, sino que también sirve de base para los procesos de contratación de la ejecución, reafirmando su función como

documento oficial que cuantifica económicamente los trabajos a realizar y que se sustenta en metrados y análisis de precios unitarios debidamente validados.

2.9.4. Análisis de Precios Unitarios

El análisis de precios unitarios (APU) es un método que descompone y calcula el costo necesario para realizar una unidad específica de trabajo en un proyecto de construcción. Este análisis incluye los costos de materiales, mano de obra, equipo, herramientas, y otros gastos asociados a cada actividad particular. El APU permite estimar con precisión el costo por unidad de obra y es una herramienta clave para elaborar presupuestos detallados, asegurando que los precios sean competitivos y reflejen adecuadamente los recursos utilizados en la ejecución de la obra (Ministerio de Vivienda Construcción y Saneamiento - Perú, 2021).

2.9.5. Costo Directo

El costo directo abarca todos los gastos específicos que se pueden asignar directamente a la ejecución de un proyecto de construcción. Esto incluye los materiales necesarios para cada actividad (como cemento, acero o ladrillos), la mano de obra empleada (trabajadores, supervisores, operadores de maquinaria), y el uso de equipos y maquinaria que intervienen directamente en el proceso constructivo (grúas, excavadoras, o herramientas especializadas).

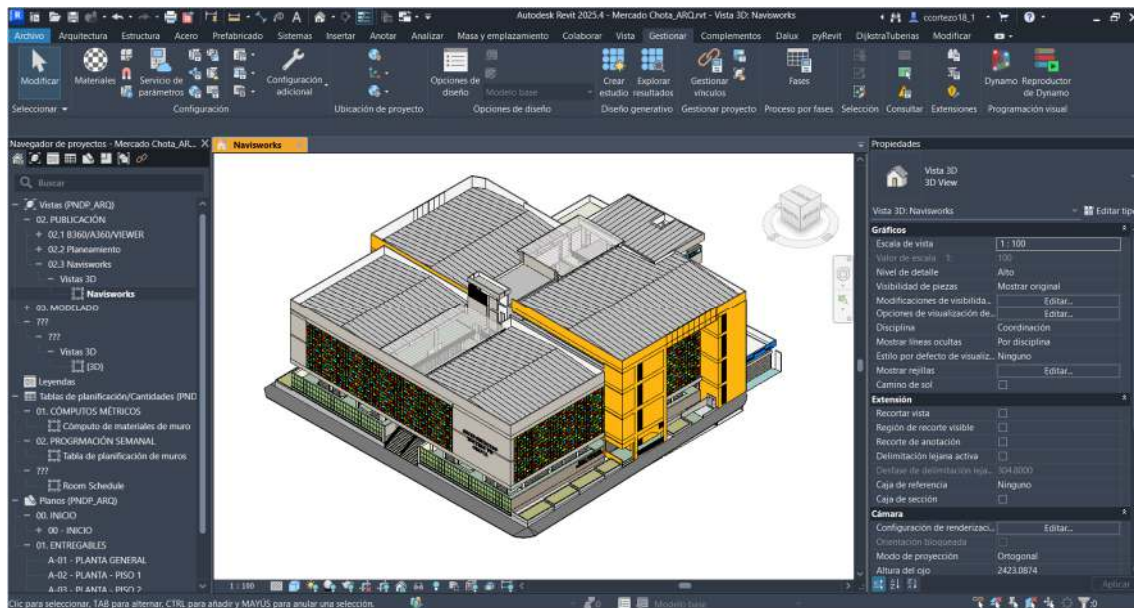
Además, el costo directo considera también otros elementos esenciales para la ejecución del proyecto, como el transporte de materiales al sitio de obra y el alquiler de equipos, siempre que estos estén involucrados directamente en la actividad específica. Estos costos son fácilmente cuantificables y son la base para calcular el presupuesto total de la obra, ya que reflejan de manera clara los recursos necesarios para la correcta realización de cada tarea constructiva (Ministerio de Vivienda Construcción y Saneamiento - Perú, 2021).

CAPÍTULO III. MATERIALES Y MÉTODOS

3.1. Procedimiento

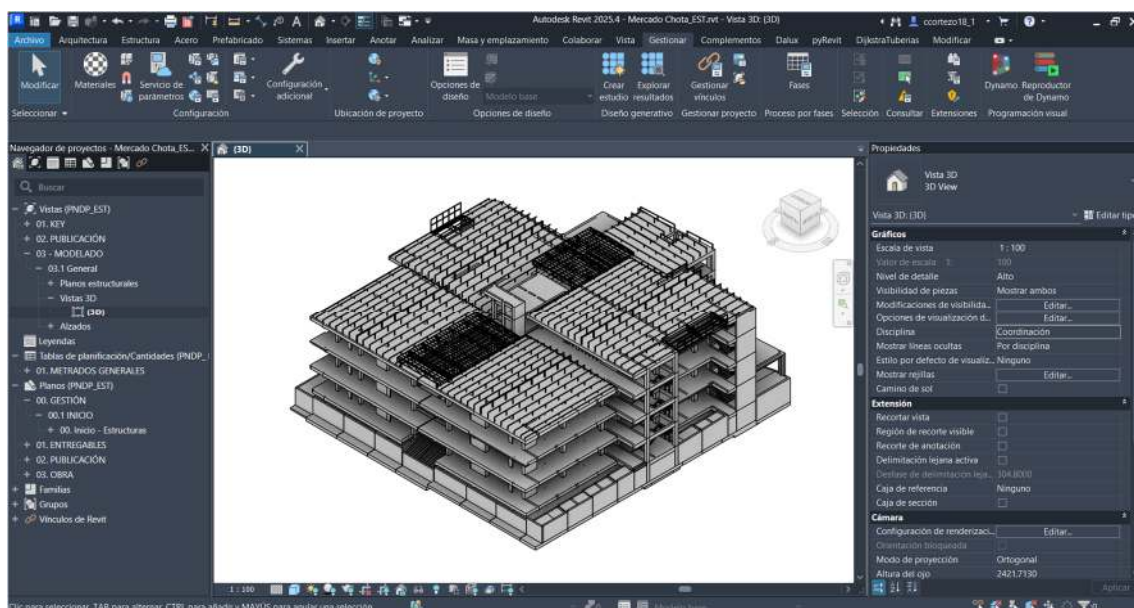
A partir de los modelos BIM del proyecto, se reúnen los datos de los elementos pertenecientes a la categoría de aparatos sanitarios. Estos datos comprenden la posición referencial en el sistema de coordenadas interno y la geometría de las habitaciones en las que se encuentran.

Figura 27. Modelo BIM especialidad arquitectura “Gran Mercado Modelo Chota”



Fuente: Obtenido del software Revit 2025

Figura 28. Modelo BIM especialidad estructuras “Gran Mercado Modelo Chota”



Fuente: Obtenido del software Revit 2025

Luego se trazó una “malla” que permite aplicar el algoritmo Dijkstra para determinar la ruta óptima de modelado de las tuberías de agua. Esta operación se realiza en una vista en planta, primero “localmente”, es decir, dentro de una sola habitación lo que generará “puntos de salida” tantos como sistemas sanitarios existan; luego se aplicó “generalmente”, lo que significa que se aplicó nuevamente la operación, pero generando la malla a partir de los “puntos de salida” generados en el paso anterior, obteniendo finalmente la ruta óptima sobre la cual se automatizó el modelado de las redes de tuberías de agua fría.

El modelado se realizó automáticamente, pero con un único diámetro de tubería, es decir sin responder a un diseño de instalaciones sanitarias propiamente dicho; para esto se recurrió al plugin Hidrasoftware el cual basándose en los datos del modelo calculó y redimensionó las tuberías.

3.2. Plugin “Dijkstra Tuberías”

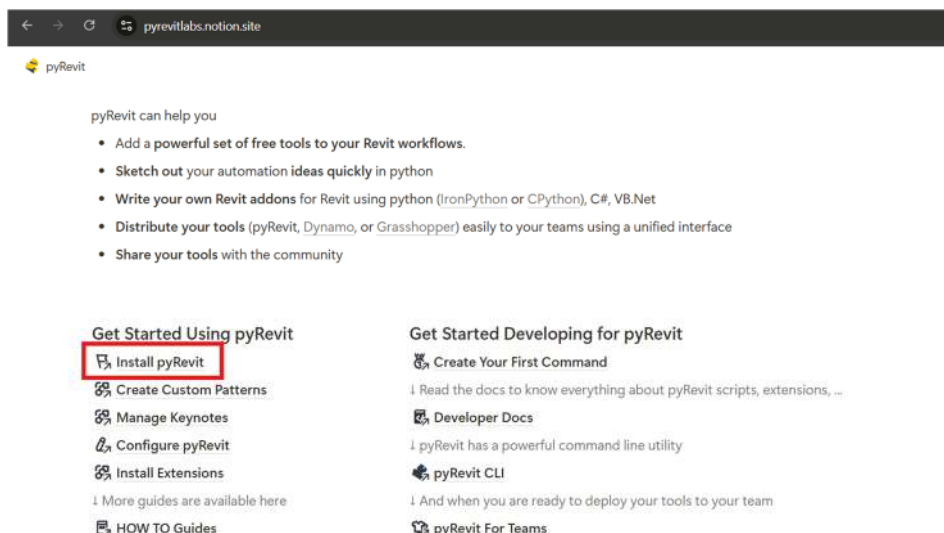
Para automatizar el proceso de modelado de redes de tuberías en edificaciones, se desarrolló un primer plugin denominado “Dijkstra Tuberías”, este complemento fue creado para automatizar la generación de tuberías dentro del entorno BIM utilizando Autodesk Revit, empleando algoritmos de optimización que permiten establecer recorridos eficientes entre los distintos aparatos sanitarios localizados dentro de una misma habitación.

El plugin se basa en el algoritmo de Dijkstra, ampliamente reconocido por su capacidad para encontrar la ruta más corta en grafos, lo que permite modelar las tuberías siguiendo trayectorias lógicas, reduciendo la longitud total de los tramos y evitando, en la medida de lo posible, interferencias con otros elementos del modelo. “Dijkstra Tuberías” se incorpora de forma sencilla en la interfaz de Revit mediante un botón personalizado. Su funcionamiento permite identificar automáticamente los aparatos sanitarios dentro de una misma “room” o habitación, construir un grafo con las posibles conexiones y, a partir de ello, generar una red optimizada de tuberías que interconecte dichos elementos. A continuación, se expone el desarrollo detallado de esta herramienta, así como las sus principales funcionalidades.

3.2.1. Configuración previa (pyRevit)

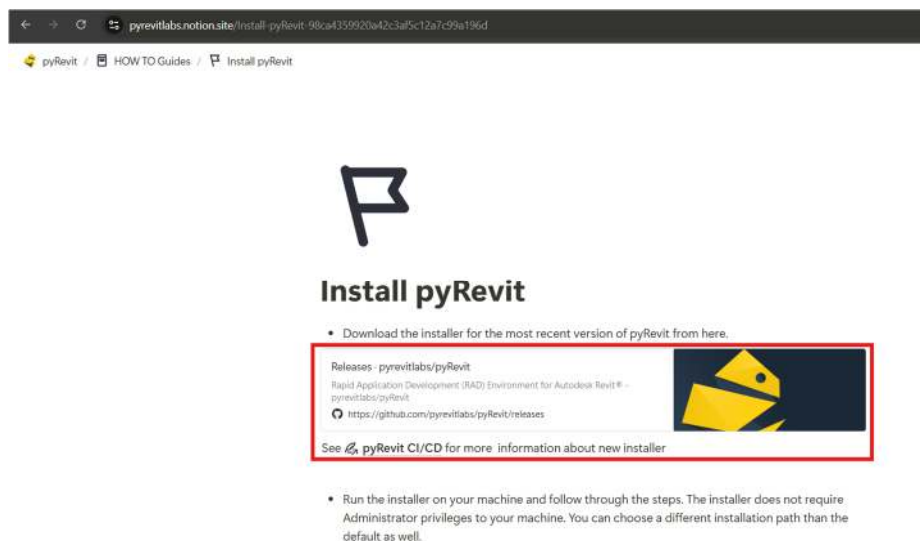
Para desarrollar “Dijkstra Tuberías” se usó como base la extensión de código abierto pyRevit, para ello fue necesario instalarlo directamente como otro complemento del software Revit.

Figura 29. Página oficial de pyRevit



Fuente: Obtenido de www.pyrevitlabs.notion.site

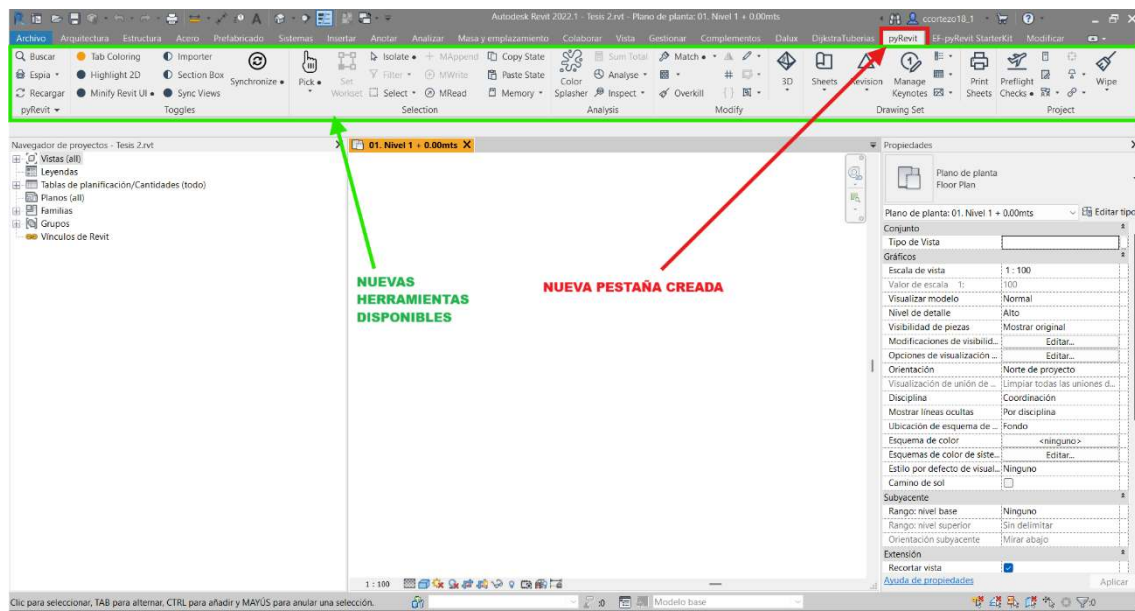
Figura 30. Instalador de pyRevit



Fuente: Obtenido de www.pyrevitlabs.notion.site

Tras la instalación, se creó una nueva pestaña llamada pyRevit, que incluye las herramientas ofrecidas por el complemento, como se muestra a continuación:

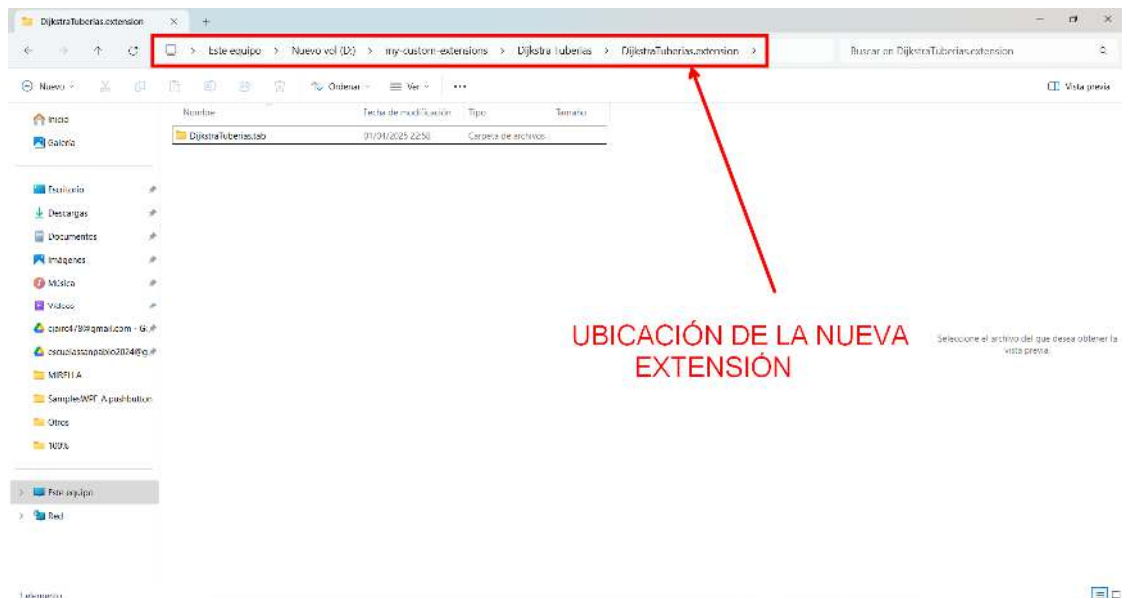
Figura 31. Pestaña de complemento pyRevit y sus herramientas



Fuente: Obtenido del Software Revit 2025

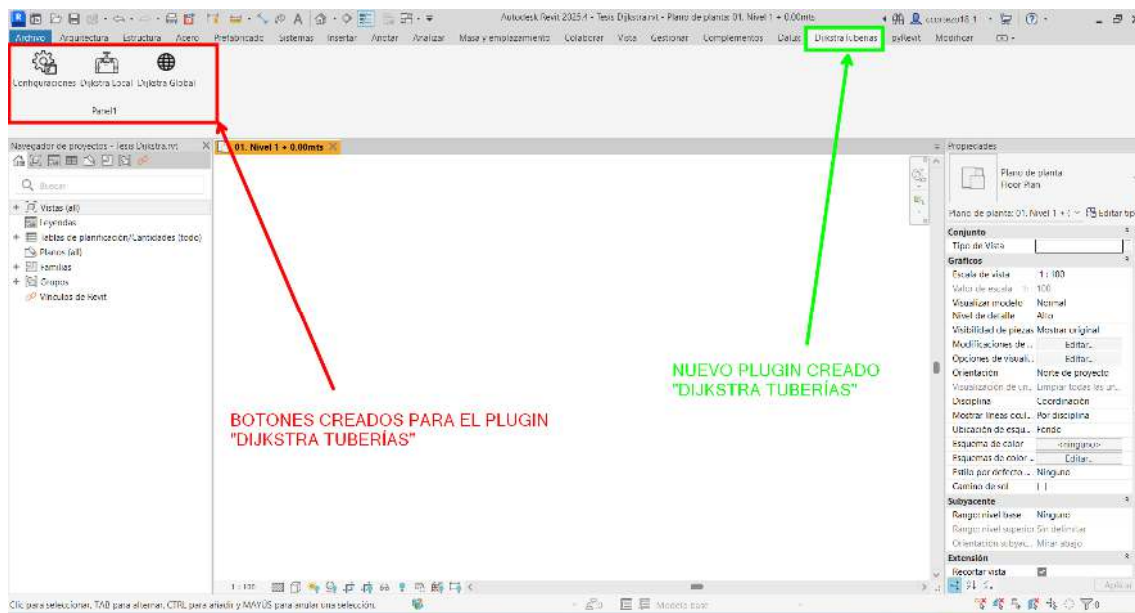
En la documentación oficial de pyRevit se indica la manera en la que se van a cargar los complementos personalizados para su correcto funcionamiento, básicamente siguen una estructura de carpetas o directorios con extensiones específicas.

Figura 32. Creación de la extensión "DijkstraTuberías.extension"



Fuente: Obtenido de Navegador de Archivos de Windows

Figura 33. Plugin creado "Dijkstra Tuberías"



Fuente: Obtenido del Software Revit 2025

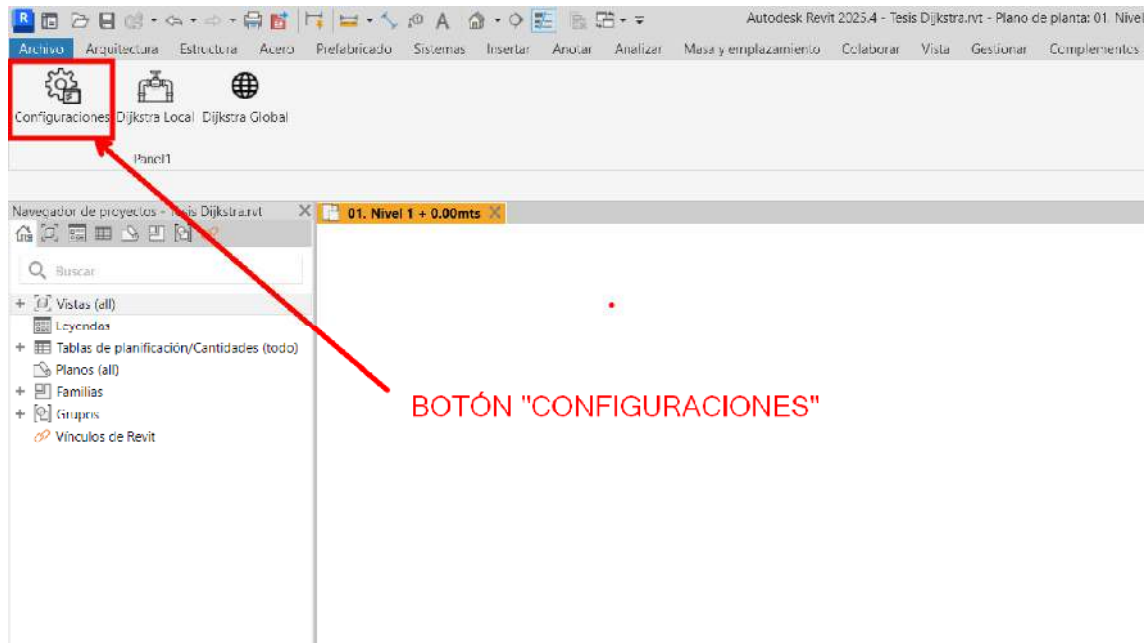
El plugin cuenta con 3 botones y cada botón está configurado para ejecutar acciones distintas, éstos son: “Configuraciones”, “Dijkstra Local” y “Dijkstra Global”.

3.2.2. Botón “Configuraciones”

El primer botón llamado “Configuraciones” despliega una ventana que permite seleccionar el tipo de tubería con el cuál iniciará el proceso de modelado, esto con el fin de que el usuario tenga la posibilidad de elegir el tipo de tubería con el cual quiere realizar el modelado automático, una ventana emergente despliega una lista de todos los tipos disponibles dentro del proyecto y el usuario puede seleccionar el tipo de tubería que crea conveniente.

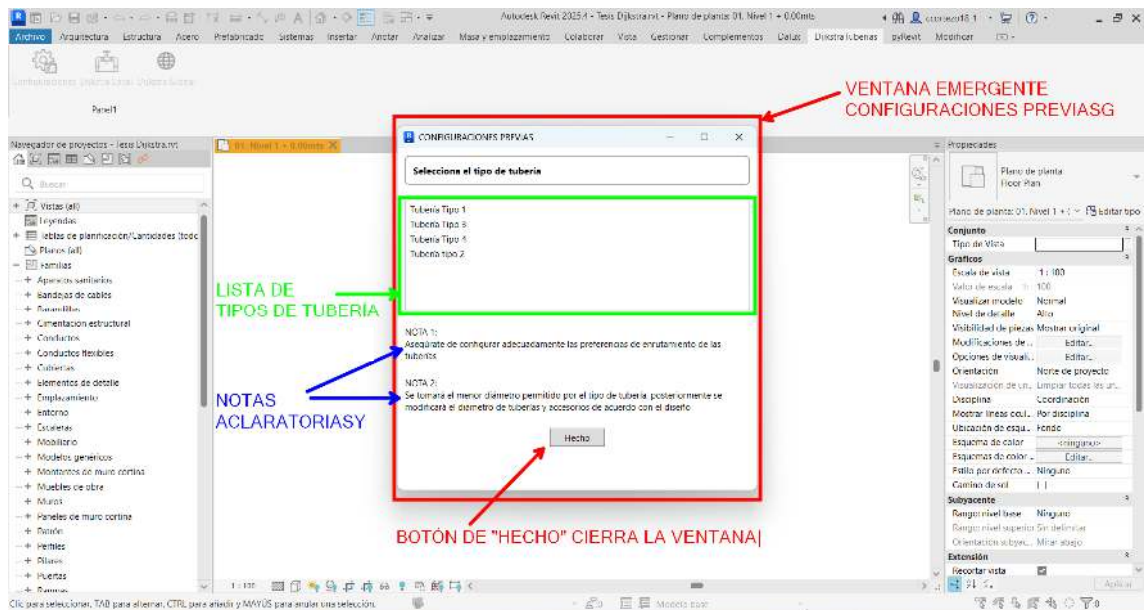
Una vez seleccionado el tipo de tubería se debe seleccionar el botón “Hecho”, que cierra la ventana y deja como configuración por defecto el tipo de tubería seleccionado.

Figura 34. Botón "Configuraciones" del plugin "Dijkstra Tuberías"



Fuente: Obtenido del Software Revit 2025

Figura 35. Ventana emergente desde botón "Configuraciones previas"



Click para seleccionar, TAB para alternar, CTRL para añadir y MAYÚS para anular una selección.

Fuente: Obtenido del Software Revit 2025

3.2.3. Botón "Dijkstra Local"

El segundo botón creado llamado "Dijkstra Local" permite al usuario modelar automáticamente las tuberías formando la red entre los aparatos sanitarios que se encuentren dentro de un mismo "room" o "habitación", para

lograr esto se siguió una secuencia de pasos que se detallan y explican a continuación:

3.2.3.1. Obtener y filtrar los aparatos sanitarios

Como primer paso para el funcionamiento del algoritmo es necesario que el usuario indique cuáles son los aparatos sanitarios que formarán la nueva red de tuberías, además dichos aparatos deben estar ubicados dentro de un “room” o “habitación”. Los aparatos seleccionados se filtraron teniendo en cuenta solo los que pertenezcan a la categoría “Aparatos Sanitarios”.

La manera como se seleccionan los aparatos sanitarios depende completamente del usuario y es indiferente si selecciona aparatos sanitarios solamente, o selecciona múltiples elementos pertenecientes a distintas categorías. Para poder obtener la información de los aparatos seleccionados se definió la función “get_selected_elements” de la siguiente manera:

Figura 36. Función para seleccionar elementos con pyRevit

```
def get_selected_elements():  
    selected_element_ids = selection.GetElementIds()  
    selected_elements = [doc.GetElement(e_id) for e_id in selected_element_ids]  
    return selected_elements
```

Nota: Se nombró la función “get_selected_elements” donde se usó el método “doc.GetElement” que devuelve la información de todos los elementos seleccionados en el documento activo de Revit.

Posterior a esto el script filtra los elementos seleccionados, obteniendo solamente los que pertenecen a la categoría “aparatos sanitarios”, para esto se definió la variable “target_families”, donde desde la lista de elementos seleccionados se compara uno a uno si la categoría a la que pertenece se denomina “Aparatos sanitarios”, si es así, el script almacena sus datos dentro de la variable.

Figura 37. Código para filtrar los elementos seleccionados que pertenecen a la categoría “Aparatos Sanitarios”

```
target_families = [e for e in selected_elements if e.Category.Name == "Aparatos sanitarios"]  
path_manager = JPathManager(mesh)  
stretches = path_manager.get_stretches()
```

3.2.3.2. Obtener límites del “room”

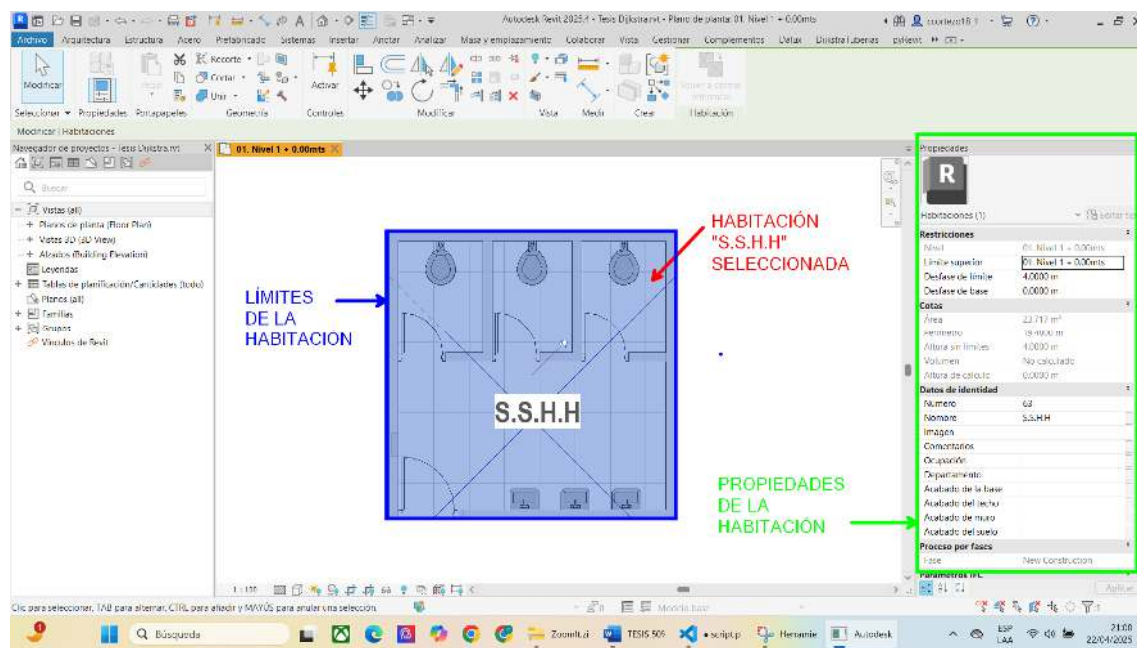
Los aparatos sanitarios seleccionados deben estar contenidos dentro de la geometría de una habitación, si bien esta entidad está determinada por un espacio tridimensional, se optó por trabajar solamente en dos dimensiones ya que las tuberías de agua no requieren de una pendiente tal como las tuberías de desagüe.

Para obtener la geometría bidimensional de un “room” se definió la función “get_lines_boundary_of_room”, que mediante la función “GetBoundarySegments”, almacena en una lista la geometría de todos los segmentos que delimitan a la habitación o “room”.

Figura 38. Código de la función para obtener la geometría de cualquier “room”

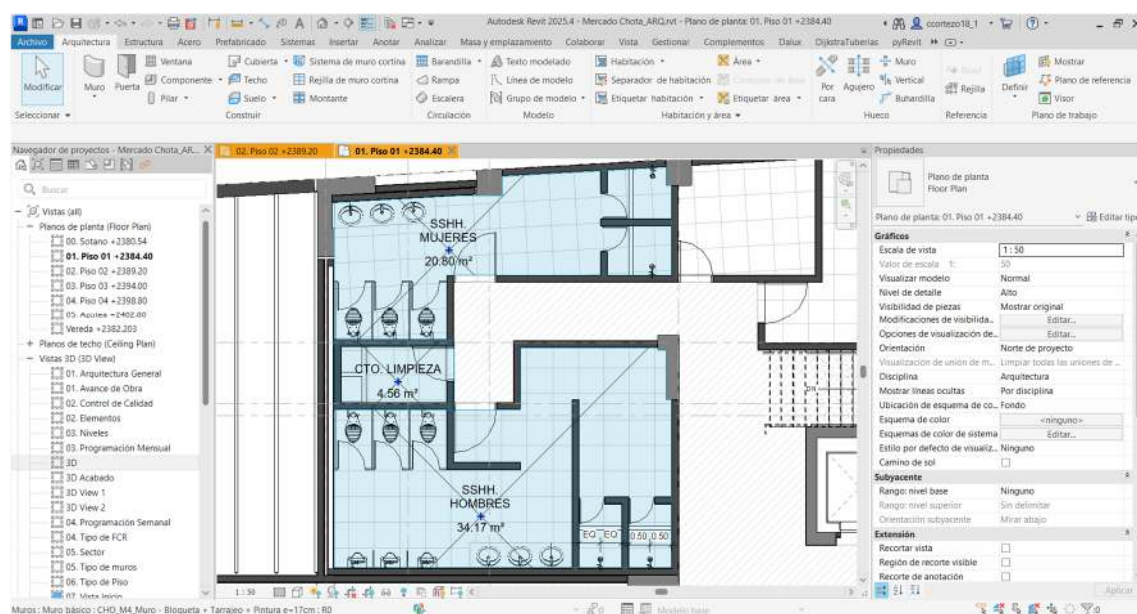
```
def get_lines_boundary_of_room(room):  
    options=SpatialElementBoundaryOptions()  
    boundary = room.GetBoundarySegments(options)  
    segments = []  
    for segment_list in boundary:  
        for boundary_segment in segment_list:  
            start_point = boundary_segment.GetCurve().GetEndPoint(0)  
            end_point = boundary_segment.GetCurve().GetEndPoint(1)  
            segments.append(Line.CreateBound(start_point, end_point))  
    return segments
```

Figura 39. Ejemplo de límites de una habitación



Fuente: Obtenido del Software Revit 2025

Figura 40. Habitaciones dentro del modelo BIM especialidad arquitectura “Gran Mercado Modelo Chota”



Fuente: Obtenido del Software Revit 2025

Tabla 2. Habitaciones o "Rooms" obtenidos del modelo BIM especialidad arquitectura "Gran Mercado Modelo Chota"

	AMBIENTES EXP. TEC.	CANT.	ÁREA PARCIAL	ÁREA TOTAL
SÓTANO	ESTACIONAMIENTO	1	-	426.53
	ESTACIONAMIENTO PARA DISCAPACITADOS	2	15.47	30.94
	PATIO DE MANIOBRAS CARGA Y DESCARGA	1	512.00	349.40
	ANDÉN DE DESCARGA	1	58.28	58.28
	DEPÓSITO DE MERCADERÍA	1	324.12	324.12
	CARNES Y AVES	1	21.39	21.39
	PESCADOS	1	15.20	15.20
	INGRESO	1	9.44	9.44
	TOTAL	1	49.00	49.00
	DEPÓSITO DE BASURA	1	53.78	53.78
	DEPÓSITO Y LAVADO DE CONTENEDORES	1	5.76	5.76
	ÁREA DE CONTROL DE ALIMENTOS	1	8.05	8.05
	MONTACARGA	1	7.20	7.20
	SS.HH. MUJERES	1	18.23	18.23
	SS.HH. VARONES	1	15.05	15.05
	CUARTO ELÉCTRICO	1	1.67	1.67
	MAESTRANZA Y MANTENIMIENTO	1	15.14	15.14
	CASETA	1	6.74	6.74
	GRUPO ELECTRÓGENO	1	32.76	32.76
	SUBESTACIÓN	1	16.29	16.29
	TABLEROS	1	12.10	12.10
	CISTERNA	1	58.39	58.39
	CUARTO DE BOMBAS	1	31.19	31.19
PRIMER PISO	PUESTOS DE PESCADOS	15	9.00	135.00
	PUESTOS DE CARNES ROJAS	36	9.00	324.00
	PUESTOS DE AVES	13	9.00	117.00
	PUESTOS DE FRUTAS	21	9.00	189.00
	PUESTOS OTROS	14	9.00	126.00
	MONTACARGA	1	7.20	7.2
	SS.HH. PÚBLICO + DISCAPACITADOS	1	27.51	27.51
	SS.HH. Y VESTIDORES PERSONAL + CTO. LIMPIEZA	1	59.99	59.99
	CUARTO ELÉCTRICO	1	4.70	4.70
	PUESTO VERDURAS	90	9.00	810.00
	PUESTO OTROS	10	9.00	90.00
	MONTACARGA	1	7.20	7.20
	DEPÓSITO DE BASURA	1	20.59	20.59
	SS.HH. PÚBLICO + DISCAPACITADOS + CTO. LIMPIEZA	1	32.50	32.50
	SS.HH. Y VESTIDORES PERSONAL + CTO. LIMPIEZA	1	59.99	59.99
	CUARTO ELÉCTRICO	1	4.70	4.70
SEGUNDO PISO	PUESTO VERDURAS	90	9.00	810.00
	PUESTO OTROS	10	9.00	90.00
	MONTACARGA	1	7.20	7.20
	DEPÓSITO DE BASURA	1	20.59	20.59
	SS.HH. PÚBLICO + DISCAPACITADOS + CTO. LIMPIEZA	1	32.50	32.50
	SS.HH. Y VESTIDORES PERSONAL + CTO. LIMPIEZA	1	59.99	59.99
	CUARTO ELÉCTRICO	1	4.70	4.70
TERCER PISO	PUESTOS DE COMIDA	3	9.00	27.00
	PATIO DE COMIDAS	1	306.00	306.00
	PUESTO DE ABARROTES	49	9.00	441.00
	PUESTO OTROS	17	9.00	153.00
	ÁREA DE JUEGOS	1	268.67	268.67
	MONTACARGA	1	7.20	7.20
	DEPÓSITO DE BASURA	1	20.59	20.59
	SS.HH. PÚBLICO + DISCAPACITADOS + CTO. LIMPIEZA	1	32.50	32.50
	SS.HH. Y VESTIDORES PERSONAL + CTO. LIMPIEZA	1	59.99	59.99
	CUARTO ELÉCTRICO	1	4.70	4.70
CUARTO	SUM	1	100.88	100.88
	DEPÓSITO	1	13.39	13.39
	DEPÓSITO DE BASURA	1	20.59	20.59
	SS.HH. PÚBLICO + DISCAPACITADOS + CTO. LIMPIEZA	1	32.50	32.50

AULA: ZONA PARA LA ACTIVIDAD AUTÓNOMA Y DE JUEGO LIBRE + ALIMENTACIÓN + DESCANSO Y ZONA DE CUIDADO Y SS.HH.	1	61.92	61.92
LACTARIO	1	15.53	15.53
OFICINA + ESPERA	1	25.98	25.98
TÓPICO	1	16.08	16.08
OFICINA	1	84.18	84.18
JEFATURA	1	16.26	16.26
VIGILANCIA	1	8.58	8.58
CUARTO ELÉCTRICO	1	4.70	4.70
MONTACARGA	1	7.20	7.20

Nota: En la tabla se muestran resaltados en verde las habitaciones o “rooms” que contienen dentro de sí elementos de la categoría “Aparatos Sanitarios”

3.2.3.3. Creación de malla de referencia

Para aplicar el algoritmo de optimización de Dijkstra fue necesario contar con una malla de referencia que permita definir nodos utilizables como puntos guía para establecer rutas entre un origen y un destino. Para ello, se modificaron las familias de “aparatos sanitarios” con el objetivo de extraer las coordenadas de cada uno en relación con el sistema de coordenadas interno del programa. Estas coordenadas no representan el centroide exacto de los aparatos, sino un punto calculado que indica el lugar preciso donde debe ubicarse la conexión de agua en la realidad.

El punto de salida del agua varía según el tipo de aparato sanitario. Aunque no existe una normativa específica que establezca una altura obligatoria para ubicar este punto, en esta investigación se ha tomado como referencia lo indicado en las especificaciones técnicas del Programa Nacional de Infraestructura Educativa (PRONIED). Además, para determinar la posición del punto de salida con respecto al eje de cada aparato sanitario, se utilizó como guía el “Manual de instalaciones de agua y desagüe – Módulo I”, publicado por la Asociación Solidaridad Países Emergentes.

Tabla 3. Ubicación de puntos de salida de cada aparato sanitario

APARATO SANITARIO	PUNTO DE SALIDA
Lavatorio	55 cm. sobre el N.P.T.
Lavadero	120 cm sobre el NPT
Inodoro de tanque bajo	30 cm. sobre el N.P.T.
Urinario de pared	120 cm sobre el NPT
Lavadero con escurridor	55 cm. sobre el N.P.T.

Fuente: Tomado de publicación sobre el modelo de expediente técnico del Programa Nacional de Infraestructura Educativa PRONIED, (2020).

Tabla 4. Ubicación de salida de agua fría de acuerdo con el eje de cada aparato sanitario

Aparato sanitario	Altura (nivel piso terminado)	Eje (punto medio del aparato sanitario)
Inodoro normal	10 – 15 cm	15 cm a la izquierda
Inodoro de una pieza	10 cm	15 cm a la izquierda
Lavadero de platos	105 – 115 cm	10 cm izquierda agua caliente 10 cm derecha agua fría
Lavadero de ropa	100 – 105 cm	10 cm izquierda agua caliente 10 cm derecha agua fría
Lavatorio	55 – 60 cm	10 cm izquierda agua caliente 10 cm derecha agua fría
Lavadora de ropa	100 cm	Al eje del aparato
Ducha	100 – 110 cm (llave) 200 – 210 cm (brazo de ducha)	Al eje del aparato
Urinario	100 – 105 cm	Al eje del aparato
Lavadora de ropa automática	100 cm	Al eje del aparato

Las medidas presentadas en la tabla adjunta son referenciales.

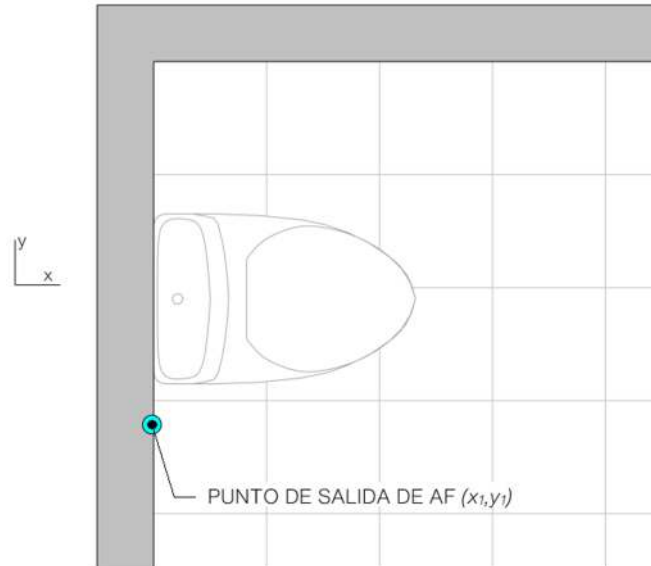
Fuente: Tomado del “Manual de instalaciones sanitarias en edificios” publicado por la Asociación Solidaridad Países Emergentes, (2008).

Figura 41. Ejemplo de punto de salida analítico de un inodoro con tanque



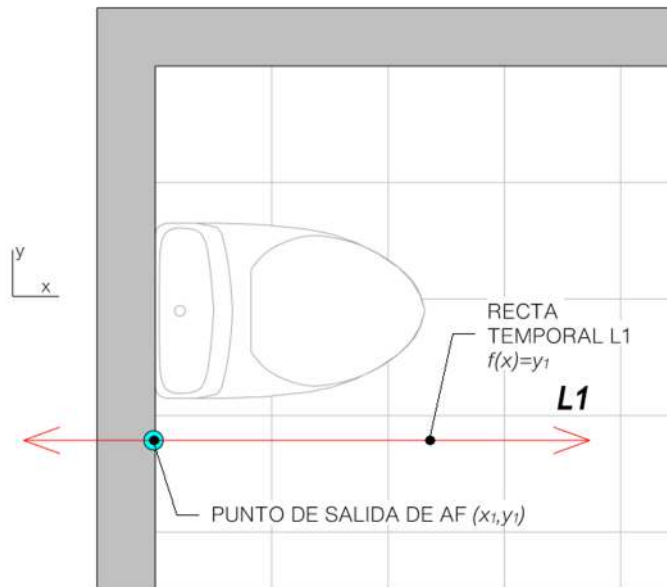
Para generar la malla sobre la cual aplicar el algoritmo Dijkstra las salidas analíticas de los aparatos sanitarios se toman como puntos sobre un plano cartesiano con coordenadas (x,y) , la coordenada en el eje z no se toma en cuenta debido a que la distancia siempre es la misma en función al tipo de aparato sanitario.

Figura 42. Ubicación del punto de salida analítico de un inodoro con tanque



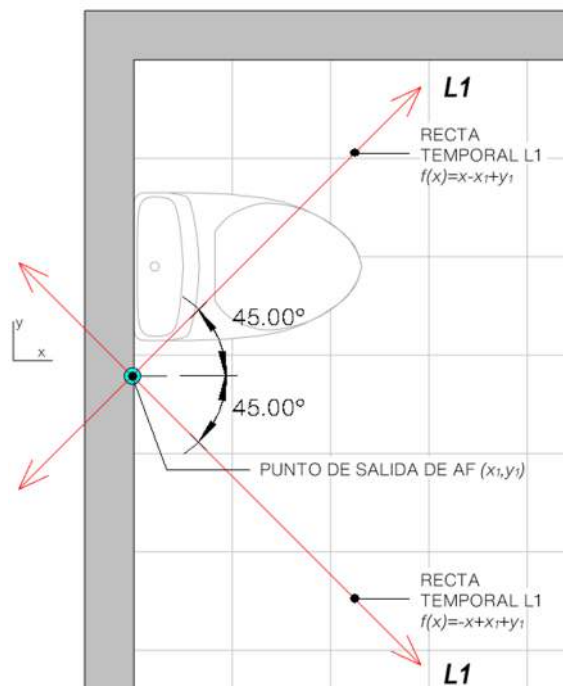
De dichos puntos se pueden trazar rectas que simulen la llegada de las tuberías de agua, teniendo las coordenadas del aparato sanitario (x_1, y_1) , la función que determina una recta paralela al eje x en dicho punto será $f(x) = y_1$

Figura 43. Ejemplo de recta de llegada L1 (90°) a un inodoro con tanque



Para obtener una recta de llegada a 45° se obtienen las rectas que pasen por el punto con pendientes $m = 1$ y $m = -1$, por lo que para las coordenadas del aparato sanitario (x_1, y_1) , las funciones que van a determinar las rectas son: $f(x) = x - x_1 + y_1$ y $f(x) = -x + x_1 + y_1$

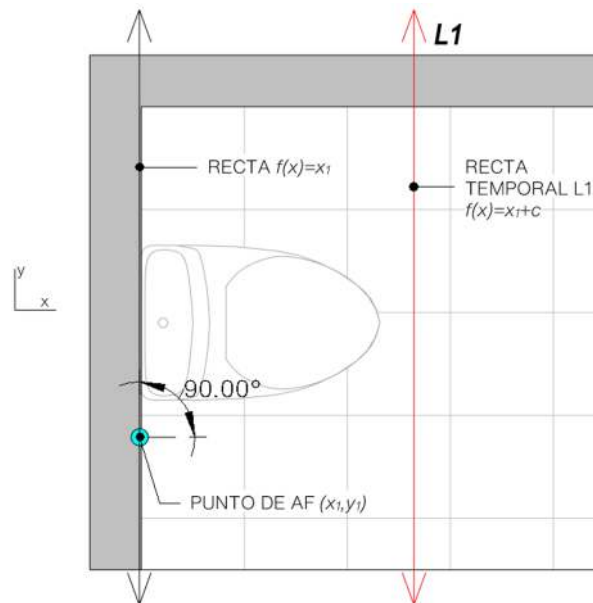
Figura 44. Ejemplo de llegada de recta L1 (45°) a un inodoro con tanque



Para obtener una recta de llegada paralela se debería obtener la ecuación de la recta que pase por el punto del aparato sanitario y sea paralela al eje y ,

teniendo las coordenadas del aparato sanitario (x_1, y_1) , la función que determina una recta paralela al eje y en dicho punto será $f(x) = x_1$, sin embargo esto permitiría que en situaciones donde tengamos múltiples aparatos sanitarios puestos en paralelo, la tubería propuesta como solución atravesaría horizontalmente los muros donde se alojaría, por lo que es conveniente separar una distancia c desde el muro, así quedaría una ecuación $f(x) = x_1 + c$.

Figura 45. Ejemplo de llegada de recta L1 (Paralela) a un inodoro con tanque



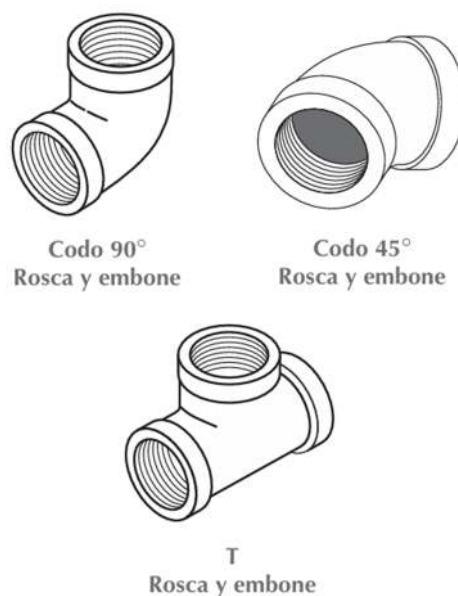
La razón por la cual se toman solo éstos tres tipos de rectas temporales, es que cuando se genere la malla las intersecciones entre rectas simulen las conexiones entre tuberías, por lo que estos ángulos solo emulan las conexiones de tuberías más comunes en el mercado.

En el Perú, el uso de tuberías y accesorios de PVC para la conducción de agua fría está regulado por diversas Normas Técnicas Peruanas (NTP). Entre ellas destacan la NTP 399.002:2015, que establece los requisitos para los tubos de PVC-U sometidos a presión, y la NTP 399.019:2004, que regula las conexiones inyectadas como codos, tees y uniones. Asimismo, la serie NTP-ISO 1452 cubre sistemas completos de tuberías y accesorios para abastecimiento de agua, incluyendo uniones con junta elástica. Estas normas especifican dimensiones, clases de presión, materiales y métodos de ensayo, garantizando la calidad y seguridad de los sistemas de agua fría tanto en instalaciones domiciliarias como industriales.

De manera complementaria, la Norma Técnica IS.010 – Instalaciones Sanitarias para Edificaciones, perteneciente al Reglamento Nacional de Edificaciones (RNE), establece los criterios de diseño, dimensionamiento e instalación de las redes de agua fría en edificaciones. Esta norma exige condiciones mínimas de presión y caudal en los aparatos sanitarios, define diámetros normativos y parámetros técnicos de instalación, asegurando así el correcto funcionamiento, la durabilidad y la seguridad de los sistemas hidráulicos en el ámbito nacional.

Los accesorios de PVC más comunes para unir tuberías son los codos (en ángulos de 45° y 90°) y las tees, que permiten la derivación del flujo en diferentes direcciones. Estos elementos se fabrican en PVC-U virgen, generalmente en color gris, y abarcan diámetros desde ½" hasta 6", dependiendo de la aplicación. Sus uniones se realizan principalmente mediante cemento solvente (sistema SP), aunque también existen adaptadores con rosca NPT y conexiones con junta elástica (sistema UF). Las conexiones cementadas están normalizadas para una presión nominal de PN10 (10 bar), mientras que los tubos pueden encontrarse en diferentes clases de presión: 5, 7.5, 10 y 15 bar.

Figura 46. Tipos de conexiones más comunes para tuberías de agua fría (PVC)



Fuente: Tomado de "Manual de Instalaciones Sanitarias en Edificios" publicado por la Asociación Solidaridad Países Emergentes(ASPE, 2008)

Luego de haberse generado las rectas temporales correspondientes, se obtienen 4 rectas temporales por cada punto, es decir por cada aparato sanitario.

Figura 47. Ejemplo de rectas temporales generadas en un inodoro con tanque

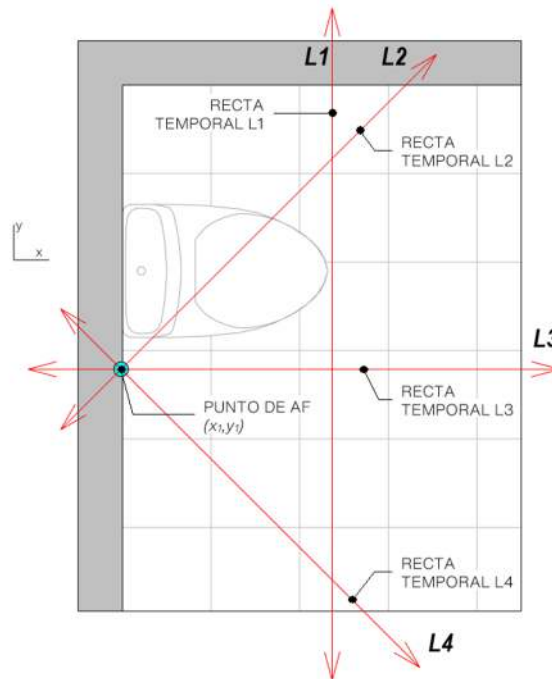


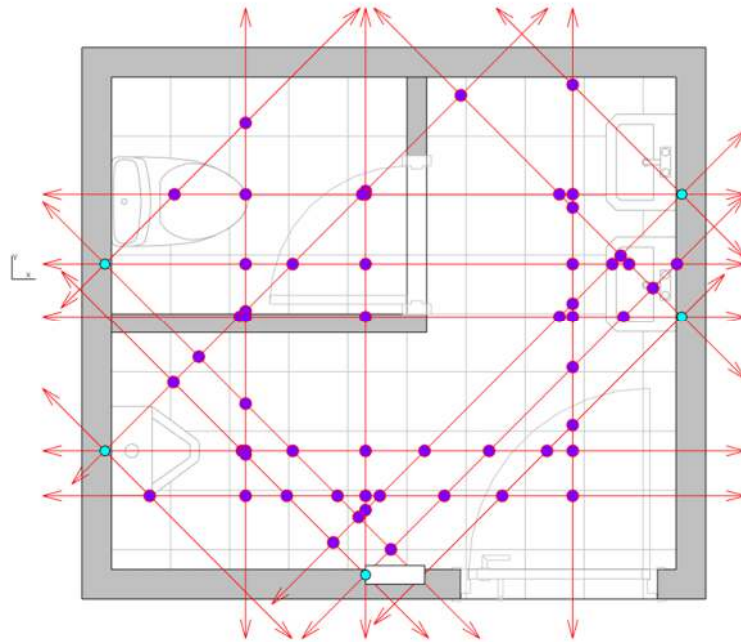
Figura 48. Ejemplo de rectas temporales generadas para distintos tipos de aparatos sanitarios



Fuente: Obtenido del software Revit

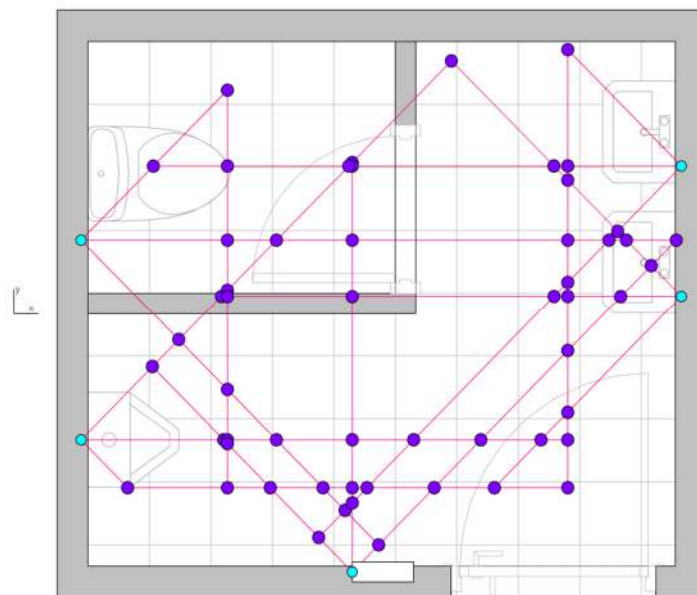
Nota: En azul se muestran las familias de aparatos sanitarios, en rojo se muestran segmentos de las rectas temporales generadas por cada aparato sanitario

Figura 49. Generación de malla para aplicación de algoritmo Dijkstra



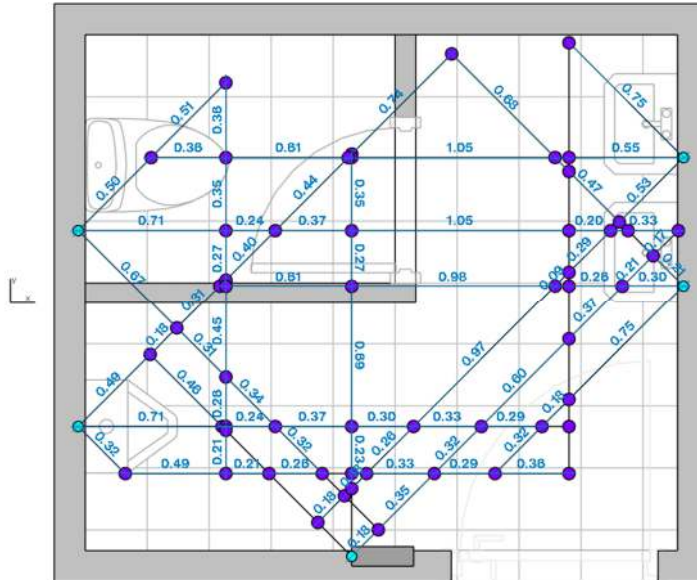
Para generar la malla sobre la cual se aplica el algoritmo Dijkstra se encontraron las intersecciones entre las rectas temporales trazadas, todos los segmentos que se puedan formar uniendo estos puntos (nodos) se pueden tomar como segmentos de tubería que formen parte del sistema de agua fría del sistema y su longitud serán los “pesos” que requiere el algoritmo.

Figura 50. Malla generada para aplicación de algoritmo Dijkstra



Los pesos iniciales únicamente serán determinados por las longitudes de los segmentos.

Figura 51. Pesos iniciales de los segmentos para aplicación de algoritmo Dijkstra



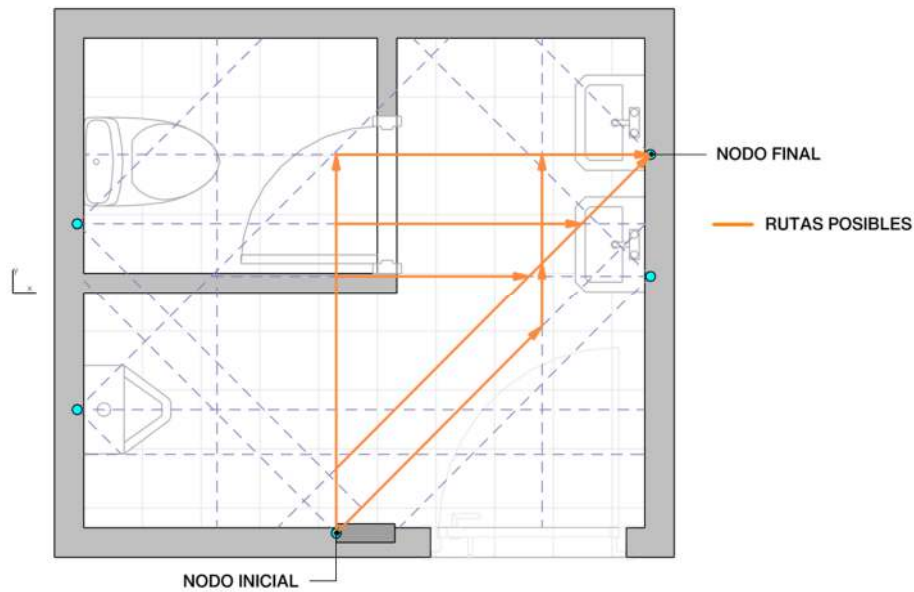
3.2.4. Cálculo de ruta óptima

3.2.4.1. Determinación de línea matriz

La línea matriz estará determinada entre el nodo inicial, que estará determinado por la ubicación de la llave de paso que controla el flujo del sector, y el nodo final que simplemente será el nodo geométricamente más alejado del inicial.

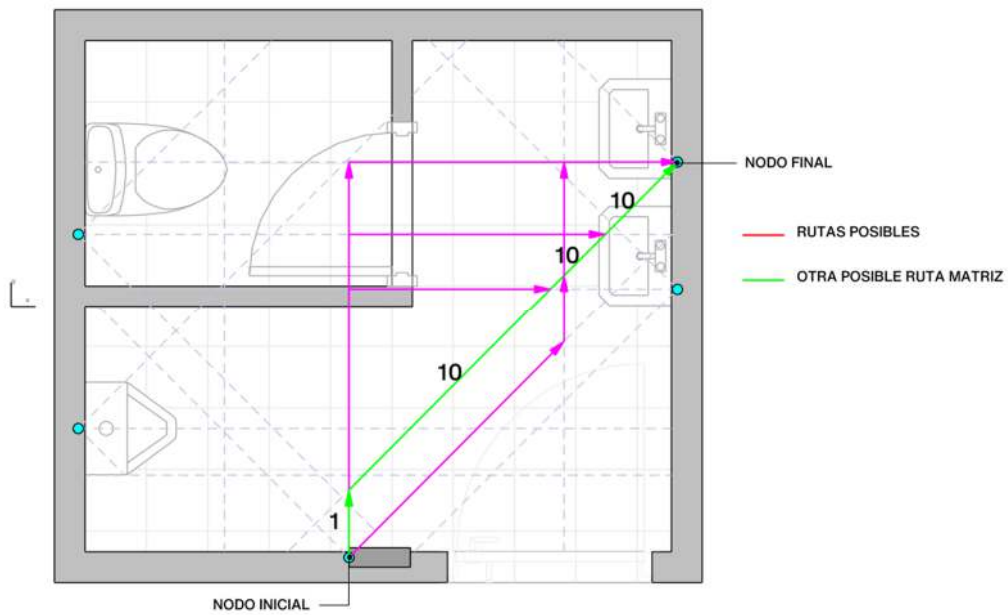
Encontrar esta línea matriz fue necesario para emular las posibles configuraciones de tuberías que son posibles para el sistema, para generar esta línea matriz se deben recorrer todos los múltiples caminos posibles desde el nodo inicial hasta el nodo final sin volver atrás y determinar el óptimo dependiendo de si se requiere optimizar longitud de tuberías y las uniones de estas.

Figura 52. Representación gráfica de las posibles rutas para obtener la línea matriz



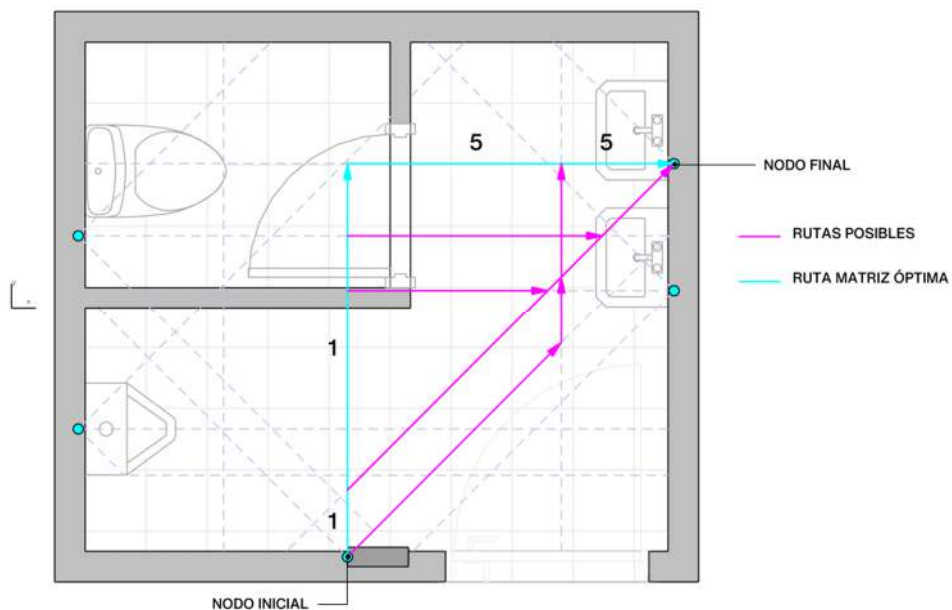
Debido a que las conexiones entre tuberías a 45° suelen ser menos comunes debido a su costo más elevado y a su menor practicidad en comparación con las de 90° , se optó por multiplicar el peso de los segmentos por 10, cuando la ruta haga un cambio de dirección de 45° y multiplicarlo por 5 cuando el cambio sea a 90° . De esta forma se pudo optimizar la ruta principal, tomando como ruta matriz, la de menor longitud, de menor cantidad de accesorios y la de menor cantidad de conexiones a 45° .

Figura 53. Ejemplo de posibles rutas matrices en un sistema local



Nota: En la figura se muestran de magenta las posibles rutas de la línea matriz y en verde una posible ruta matriz donde el cambio de dirección a 45° hace que los pesos se multipliquen por 10.

Figura 54. Determinación de la ruta matriz óptima



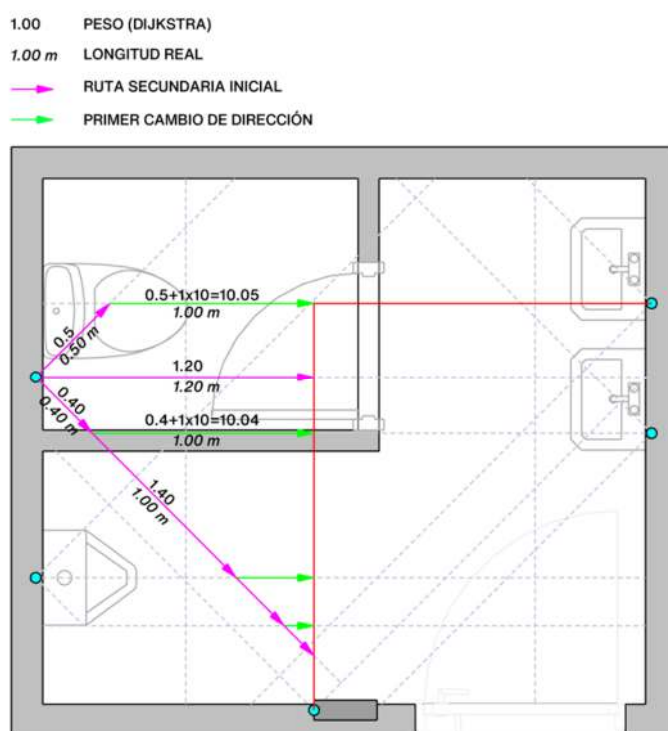
Nota: En la figura se muestra en cian la ruta finalmente elegida (la de menor peso), los números sobre los segmentos hacen símil a los pesos determinados por la longitud y el cambio de dirección de la ruta, llegando a 5.

3.2.4.2. Trazo de líneas secundarias

Una vez determinada la línea matriz, el algoritmo encuentra las posibles rutas desde un nodo de aparato sanitario hasta los nodos que forman la línea principal, evalúa las rutas posibles y selecciona la ruta con menor peso. Aquí también se realizó el aumento del peso debido al cambio de dirección.

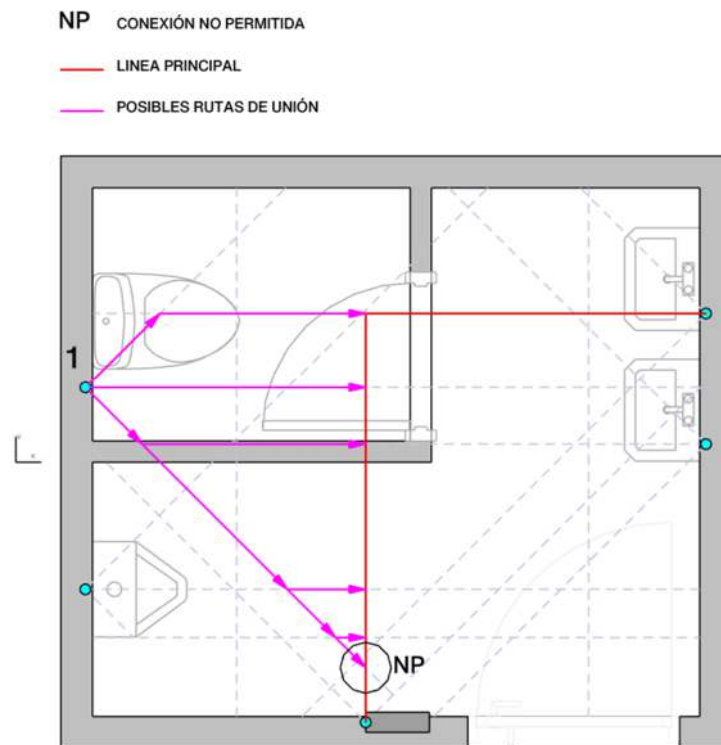
Los pesos siguen siendo determinados por su longitud, sin embargo, cuando se llega a un nuevo nodo se comprueba el cambio de dirección, si no hay un cambio de dirección el peso seguirá siendo el mismo, si hay un cambio de dirección a 45° el peso multiplicará por 10 y finalmente si el cambio es a 90° el peso se multiplicará por 5.

Figura 55. Pesos y posibles rutas para el trazo de rutas secundarias hacia la matriz



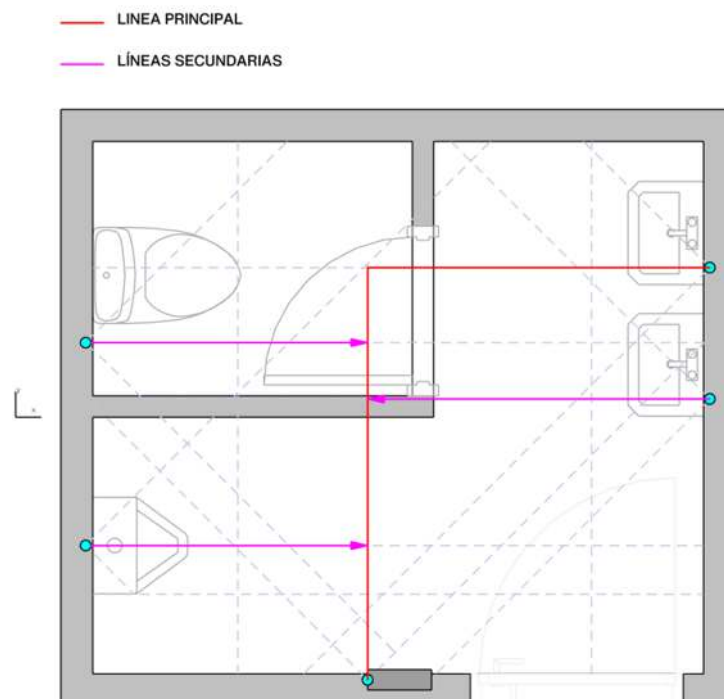
Al tener tantas posibles conexiones, se realizó un filtro para clasificar algunas como “no permitidas”, debido a que la conexión entre la línea secundaria con la principal no puede ser llevada a cabo en el mundo real por la inexistencia de conexiones de ese tipo, por ejemplo, uniones tipo “cruz” o uniones tipo “yee” a 45°, descartando automáticamente las rutas que lleguen de esa forma.

Figura 56. Filtrado de rutas secundarias no permitidas



Luego de este proceso se llegó a obtener las rutas óptimas para unir los nodos faltantes del sistema con la línea principal, es este trazado final de líneas el que determina por dónde irán las tuberías y conexiones del sistema.

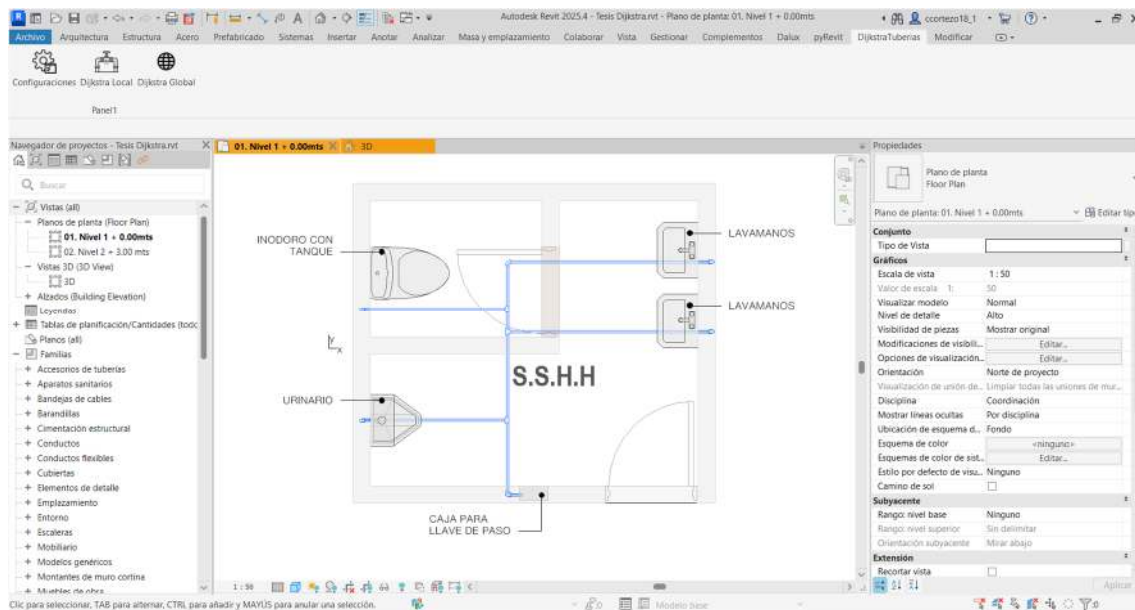
Figura 57. Ruta final optimizada para el modelado de tuberías



3.2.5. Modelado automático de tuberías

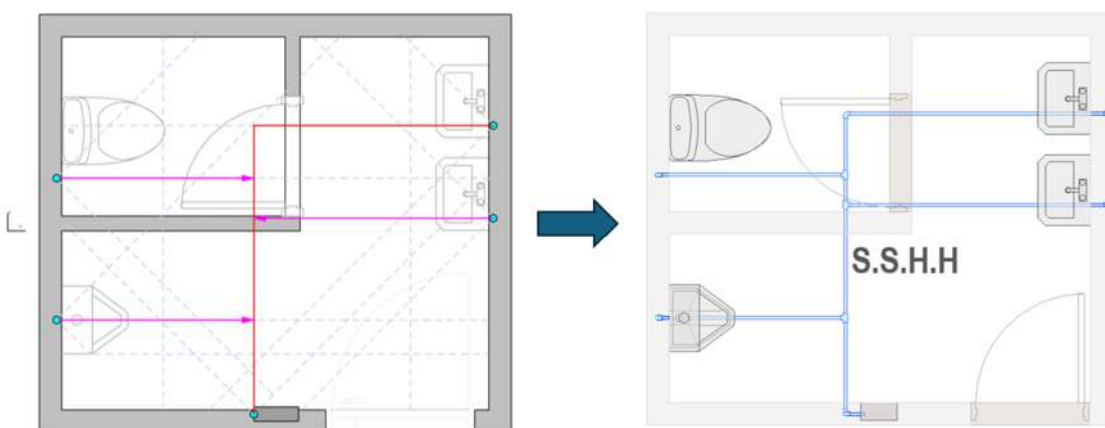
Luego de realizado el proceso para determinar la ruta óptima, el script procede inmediatamente con el modelado de las tuberías sobre dichas líneas, esto es simplemente siguiendo las coordenadas de los puntos inicial y final de cada segmento.

Figura 58. Modelado de tuberías con el botón “Dijkstra Local”



Fuente: Obtenido del Software Revit 2025

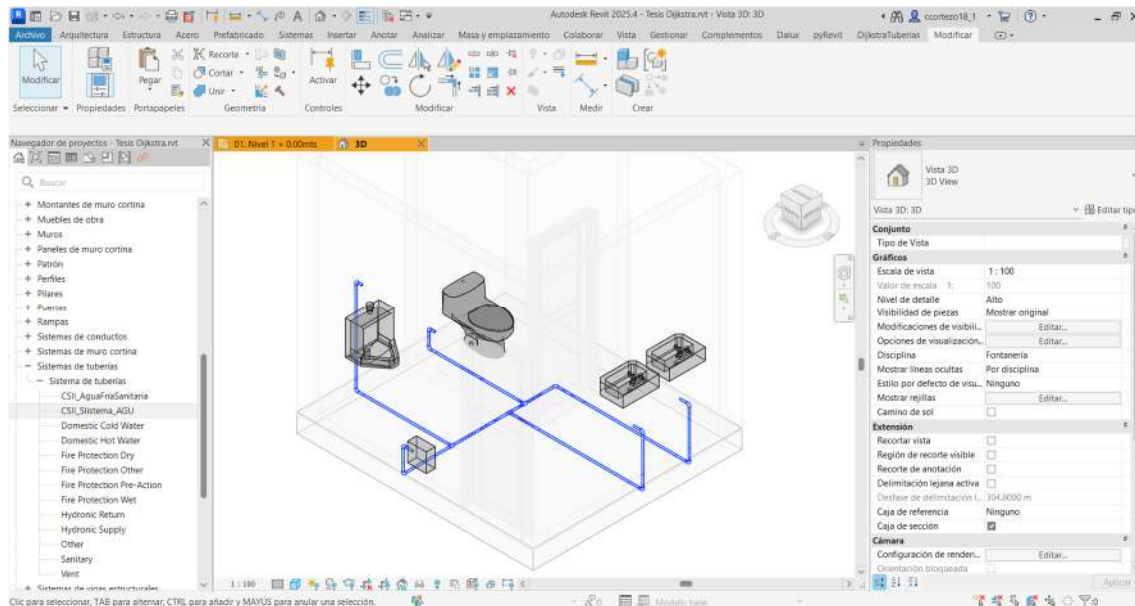
Figura 59. Comparación de ruta determinada analíticamente y las tuberías modeladas en Revit



Nota: A la izquierda se muestra la determinación teórica de la ruta óptima, a la derecha se muestra la tubería modelada automáticamente en el software Revit.

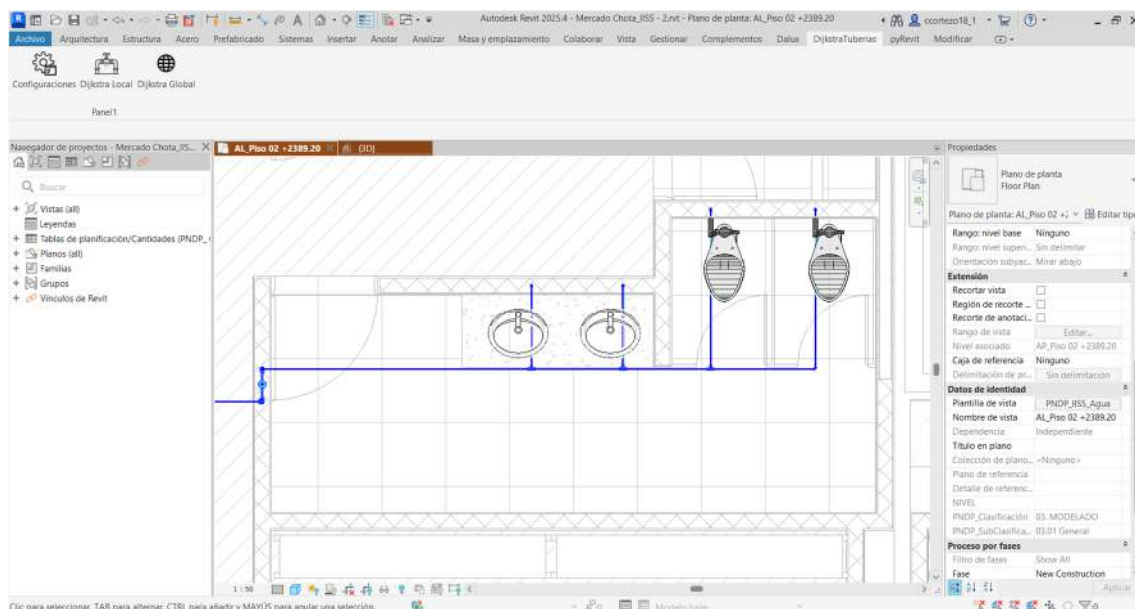
El script logra modelar de manera adecuada las tuberías, y logra además modelar los puntos de salida de las tuberías de acuerdo con cada aparato sanitario, como se muestra en la siguiente imagen:

Figura 60. Ejemplo de isométrico de modelado de tuberías automatizado



Fuente: Obtenido del Software Revit 2025

Figura 61. Modelado automático local de tuberías usando plugin "Dijkstra Tuberías"



Fuente: Obtenido del Software Revit 2025

Nota: En la imagen se muestra el resultado de aplicar el plugin “Dijkstra tuberías” en el “room” denominado “S.S.H.H” del Nivel 1, del modelo BIM especialidad Instalaciones Sanitarias del “Gran Mercado Modelo Chota”

En la siguiente imagen se muestra la función que realiza el modelado de las tuberías siguiendo el método de punto inicial, punto final.

Figura 62. Función creada para el modelado de tuberías con un punto inicial y un punto final.

```
def create_pipe_segment(doc, start_point, end_point):
    """Crea un segmento de tubería entre dos puntos."""
    system_type_id = get_system_type_id(doc)
    level_id = doc.ActiveView.GenLevel.Id

    # Busca el tipo de tubería seleccionado
    pipe_type = next(
        (pt for pt in FilteredElementCollector(doc).OfClass(PipeType).ToElements()
         if Element.Name.GetValue(pt) == selected_pipe),
        None
    )
    if not pipe_type:
        raise ValueError("No se encontró ningún tipo de tubería con el nombre '{}'.format(selected_pipe)")

    # Crea la tubería
    pipe = Pipe.Create(doc, system_type_id, pipe_type.Id, level_id, start_point, end_point)
    return pipe
```

La función definida “create_pipe_system” fue usada para modelar automáticamente las tuberías teniendo como base los segmentos determinados analíticamente, tal como se ve en la siguiente imagen.

Figura 63. Código para modelado de tuberías automático siguiendo las líneas determinadas analíticamente.

```
##draw_lines(mesh.lines_mesh)
target_families = [e for e in selected_elements if e.Category.Name == "Aparatos sanitarios"]
path_manager = JPathManager(mesh)
stretches = path_manager.get_stretches()

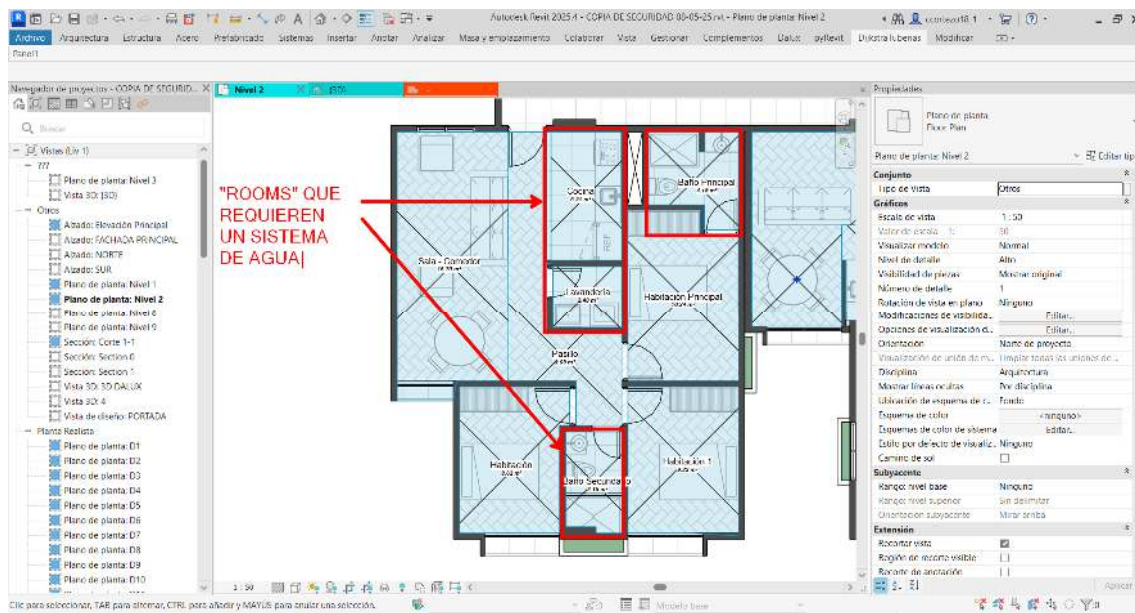
with Transaction(doc, "Create Pipes") as trans:
    trans.Start()
    for stretch in stretches:
        for segment in stretch.path:
            create_pipe_segment(doc, segment.start_node.point, segment.end_node.point)
    trans.Commit()
```

3.3. Aplicación de Dijkstra por Niveles

Una vez que se logró aplicar el algoritmo de manera local en los sistemas, se aplicó de manera general, por niveles, siguiendo los mismos principios descritos anteriormente el algoritmo se puede aplicar tantas veces como sea necesario hasta completar el diseño de un piso.

En la siguiente figura se presenta una vista en planta de un edificio de departamentos modelado en el software Revit, aquí podemos observar los distintos espacios (“rooms”) con los que cuenta un departamento típico.

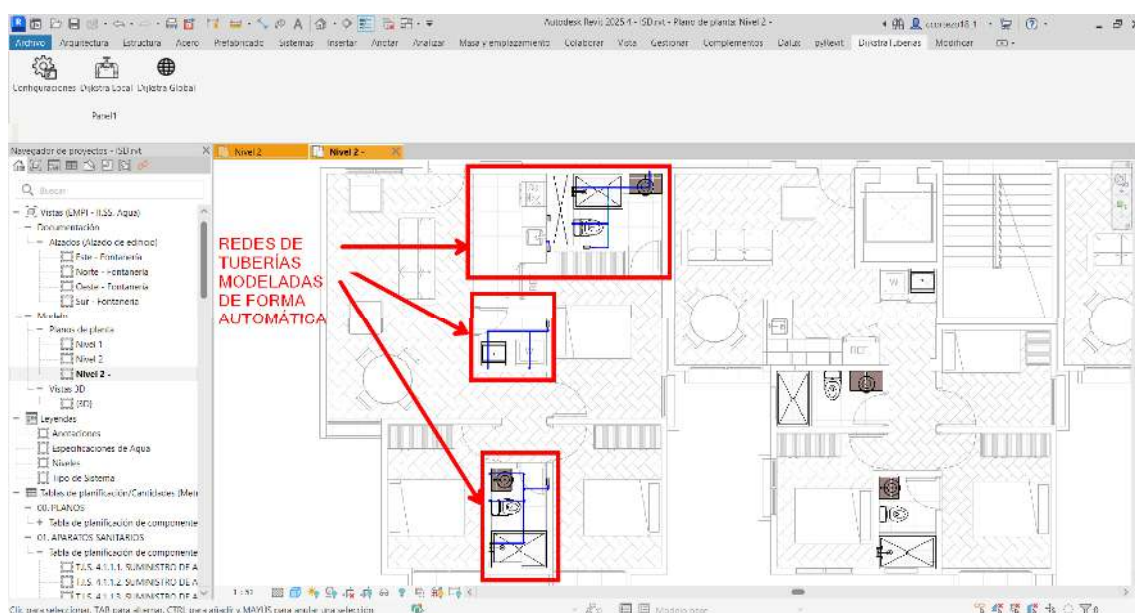
Figura 64. Ejemplo de “rooms” presentes en una edificación



Fuente: Obtenido del Software Revit 2025

Para los “rooms” en los cuales sea necesario contar con un sistema de agua, se usó el plugin para modelar automáticamente las redes de tuberías.

Figura 65. Ejemplo de modelado automático con el plugin “Dijkstra Tuberías”



Fuente: Obtenido del Software Revit 2025

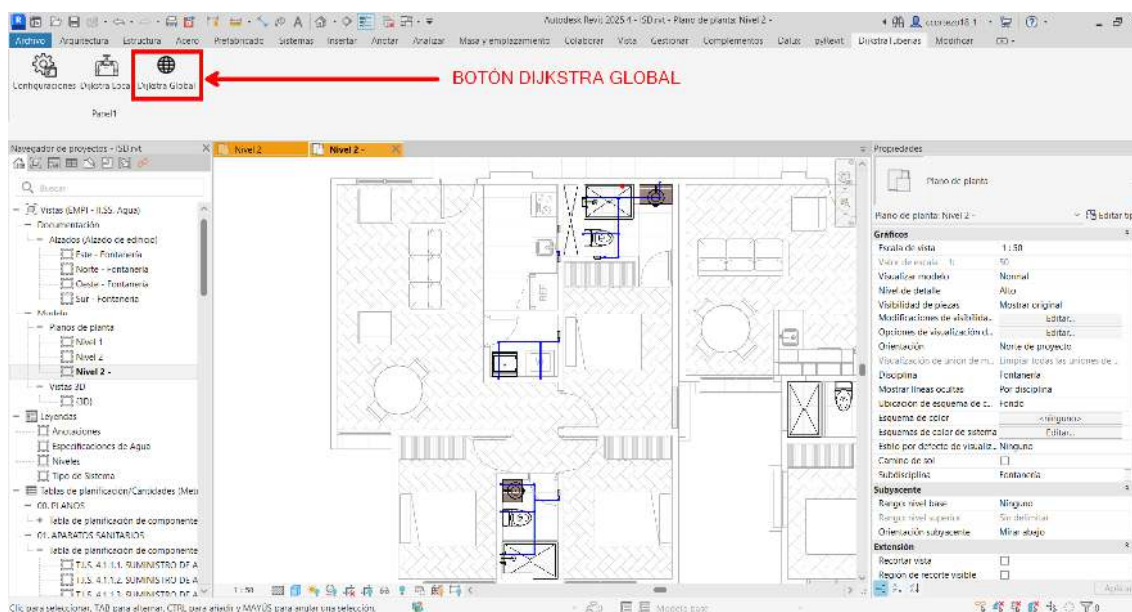
Nota: En la figura se muestran las redes de agua fría que generó automáticamente el plugin Dijkstra Tuberías, estas redes son las llamadas “locales”.

3.3.1. Botón “Dijkstra Global”

Posteriormente a la aplicación del algoritmo Dijkstra para el modelado automático de las redes de tuberías de agua fría de manera “local”, se procedió de similar forma, pero de manera “global”, para esto se desarrolló el botón “Dijkstra Tuberías”, que permite realizar el proceso de aplicación del algoritmo, pero cambiando algunos puntos clave.

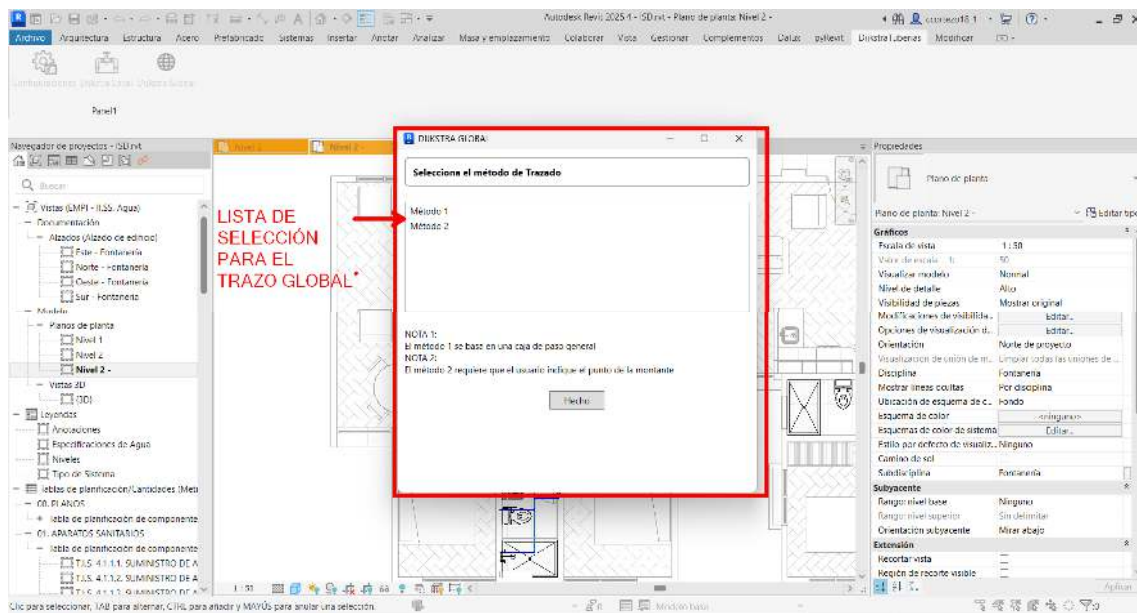
Como principal cambio en este punto los nodos que antes fueron las familias de “aparatos sanitarios” sin excepción, ahora simplemente serán las cajas de paso de los sistemas locales, es decir, los otrora nodos iniciales se tomaron como nodos generales de los nuevos sistemas globales. Y el nuevo nodo inicial será una caja de paso general o de ser el caso, un punto determinado por el usuario que denote una montante de tubería de agua fría.

Figura 66. Botón "Dijkstra Global"



Fuente: Obtenido del Software Revit 2025

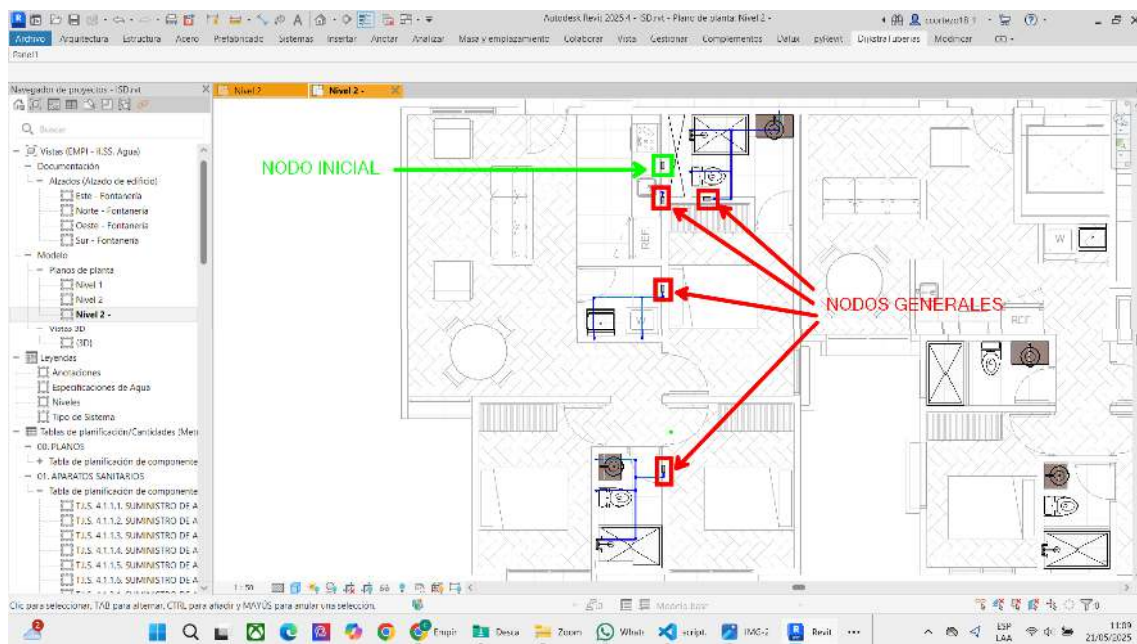
Figura 67. Ventana desplegable "Dijkstra Global"



Fuente: Obtenido del Software Revit 2025

Nota: En la figura se muestra la ventana emergente que se proyecta al hacer clic en el botón "Dijkstra Global", se puede ver la lista generada con las 2 posibles configuraciones o "métodos".

Figura 68. Ejemplo de nodos para aplicación de "Dijkstra Global"

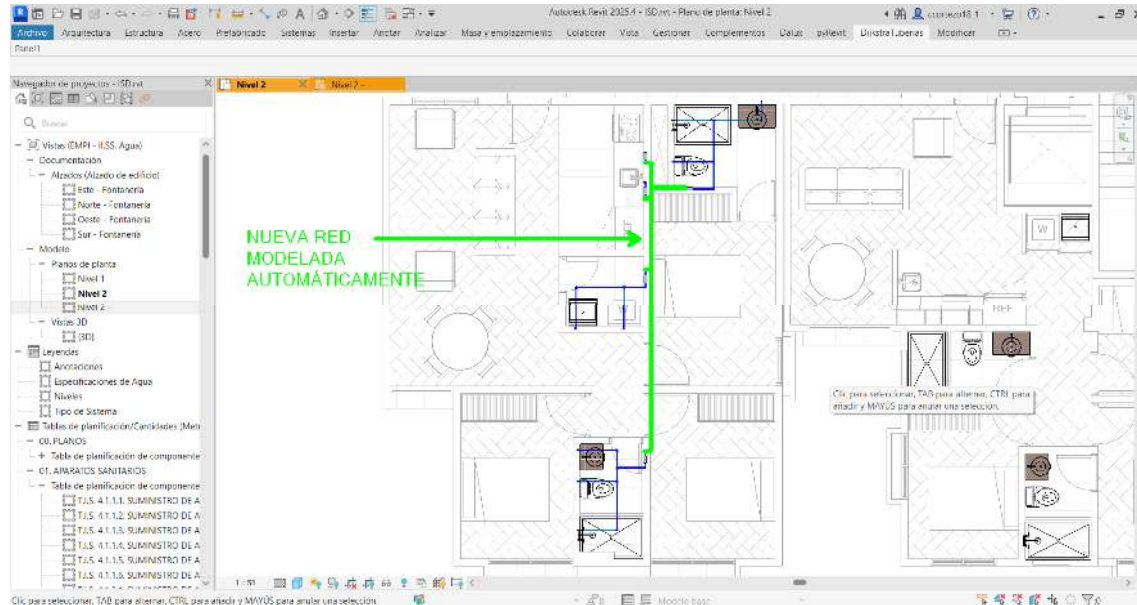


Fuente: Obtenido del Software Revit 2025

Una vez estos nodos están determinados de forma adecuada, el procedimiento para modelar y optimizar la red es el mismo anteriormente mencionado en el botón "Dijkstra Local", es decir, se trazó la red por donde van

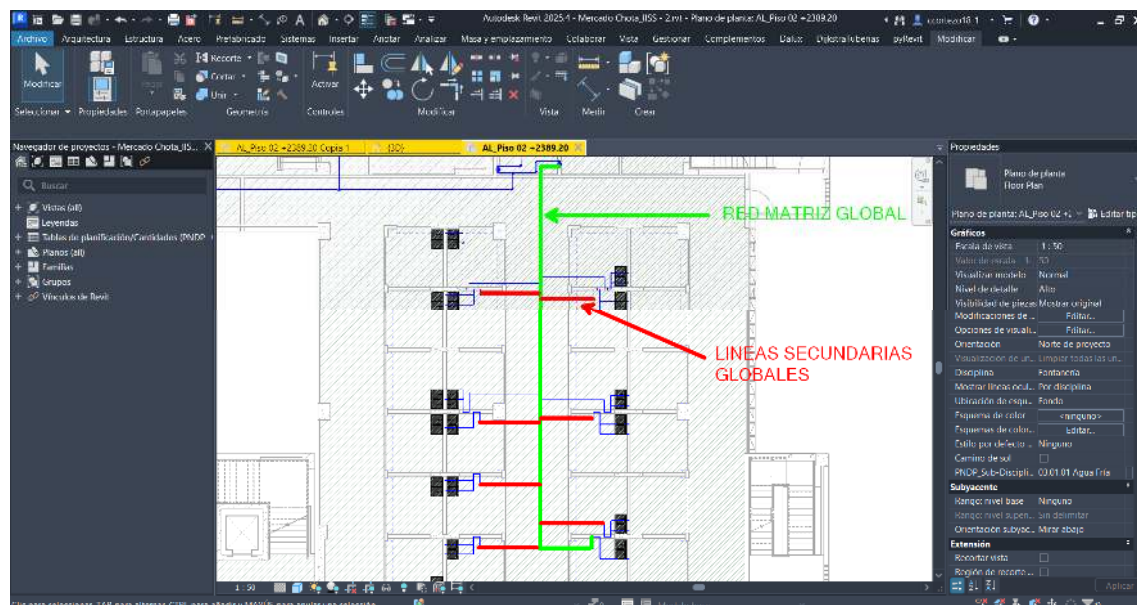
las posibles configuraciones de la ruta y de acuerdo con el sistema de pesos se determinó la óptima, posterior a eso se modeló automáticamente.

Figura 69. Ejemplo Red de tubería modelada automáticamente con "Dijkstra Global"



Fuente: Obtenido del Software Revit 2025

Figura 70. Aplicación de Dijkstra Global en modelo BIM especialidad instalaciones sanitarias "Gran Mercado Modelo Chota"

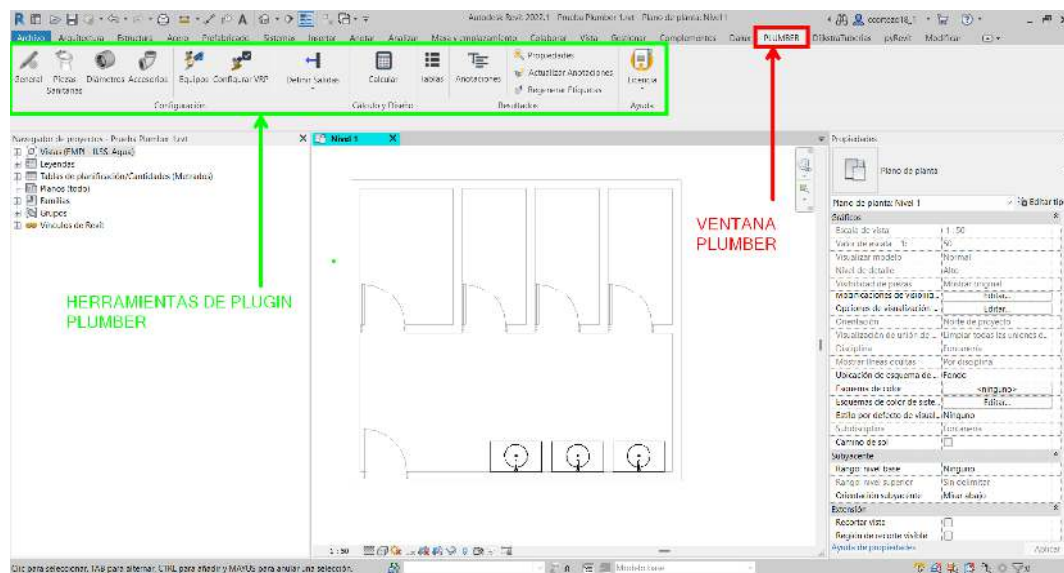


Fuente: Obtenido del Software Revit 2025

3.4. Cálculo y Redimensionamiento

Luego de finalizado el modelado de tuberías automatizado, se procedió a calcular y redimensionar las tuberías utilizando el plugin “PLUMBER”, desarrollado por la empresa Hydra Software. Este plugin constituye una herramienta especializada para el cálculo y diseño hidráulico de instalaciones de abastecimiento de agua y evacuación sanitaria en edificaciones, dentro del entorno de modelado BIM provisto por Autodesk Revit. Su principal función no radica en el modelado o trazado de redes, sino en el análisis técnico de los sistemas hidráulicos ya definidos geométricamente por el usuario en Revit.

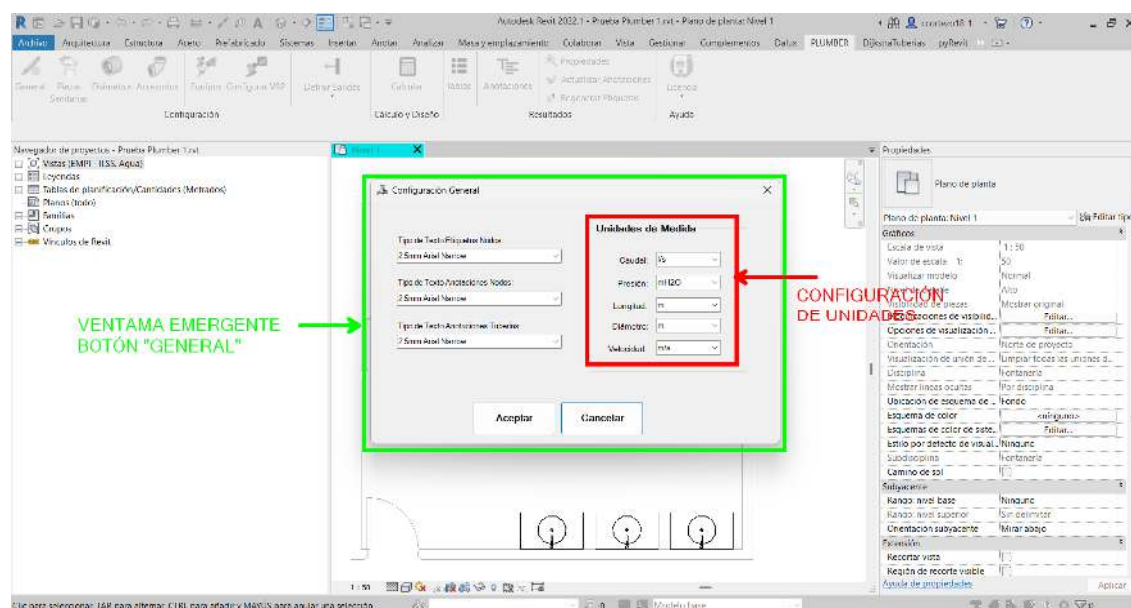
Figura 71. Plugin de cálculo y redimensionamiento “PLUMBER”



Fuente: Obtenido del Software Revit 2025

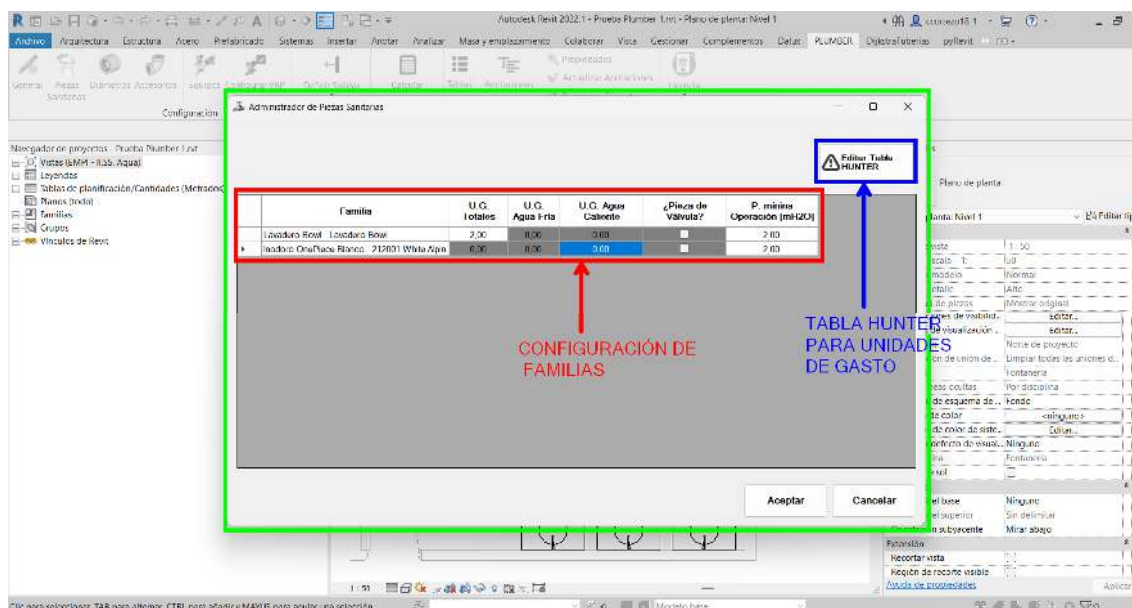
El flujo de trabajo se inicia una vez que el usuario ha modelado en Revit la red sanitaria correspondiente, incluyendo los aparatos sanitarios, tuberías, accesorios, uniones y demás componentes del sistema. A partir de esta base geométrica, el plugin permite asignar propiedades hidráulicas a cada componente, tales como unidades de gasto, caudales base, condiciones de presión o cotas de instalación. Estas propiedades constituyen los datos de entrada para el proceso de cálculo posterior.

Figura 72. Configuración general plugin “PLUMBER”



Fuente: Obtenido del Software Revit 2025

Figura 73. Configuración de familias y unidades Hunter plugin "PLUMBER"

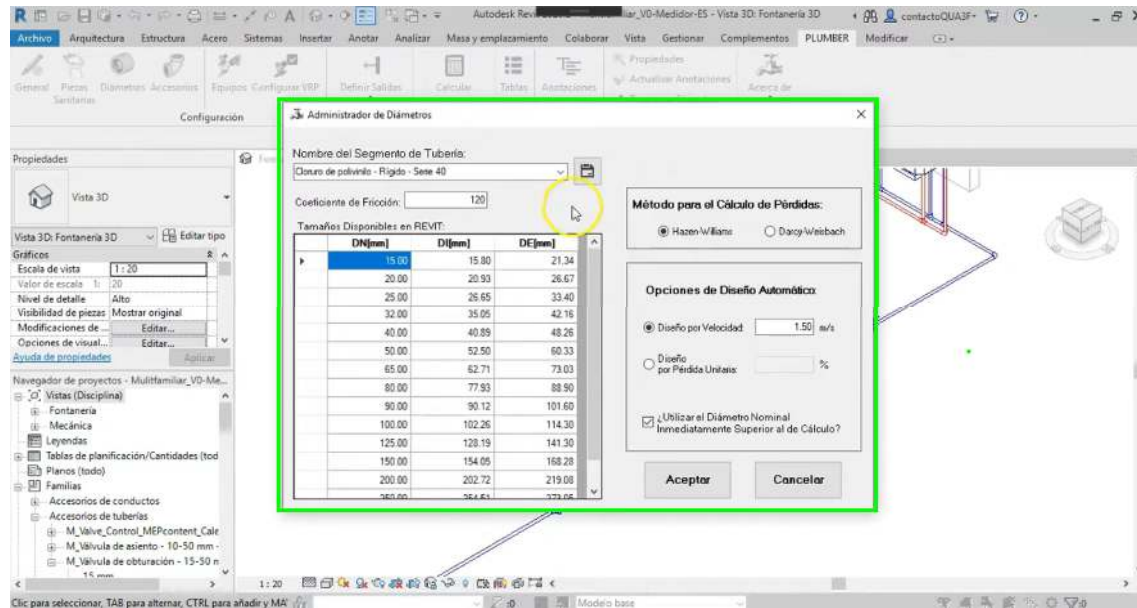


Fuente: Obtenido del Software Revit 2025

El módulo de cálculo de PLUMBER está diseñado para dimensionar redes de agua fría, agua caliente y, en general, cualquier sistema de tuberías en presión. El análisis incluye el cálculo de caudales de diseño por tramo, la evaluación de pérdidas de carga (tanto por fricción como por accesorios), la verificación de presiones mínimas y máximas admisibles, así como la asignación de diámetros nominales comerciales en función de las condiciones hidráulicas obtenidas. El motor de cálculo puede operar con distintos métodos, entre ellos la

fórmula de Hazen-Williams, permitiendo flexibilidad técnica para adaptarse a normas locales o internacionales.

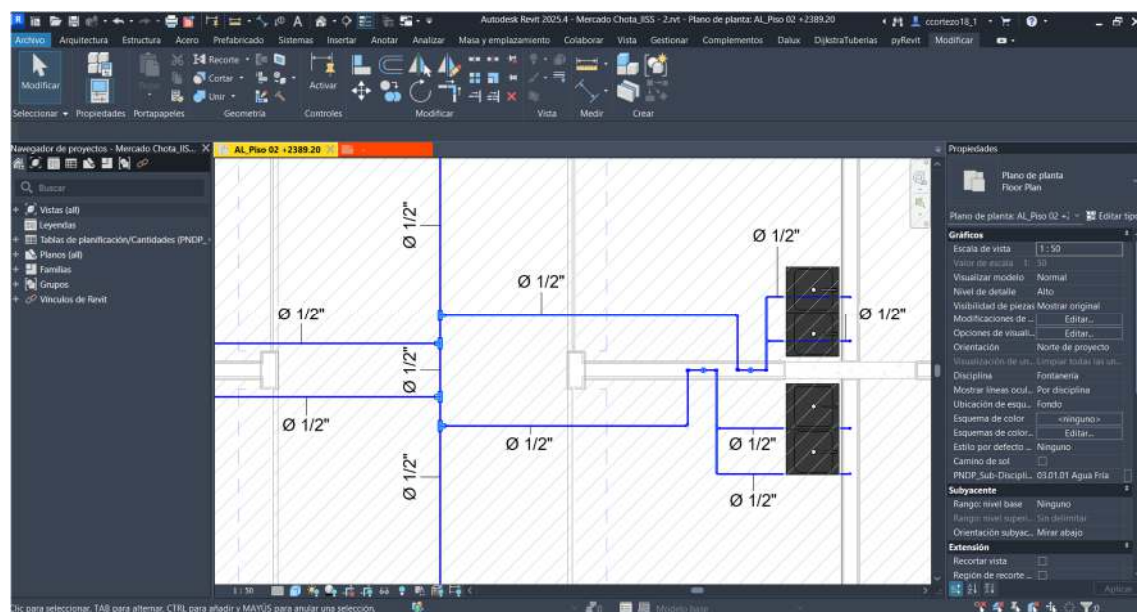
Figura 74. Administrador de diámetros del plugin "PLUMBER"



Fuente: Obtenido del Software Revit 2025

Una vez realizado el dimensionamiento, el plugin genera resultados que se integran directamente en el modelo BIM, incluyendo anotaciones en las vistas isométricas o de planta, etiquetas con datos técnicos, y tablas de resultados que detallan caudales, velocidades, longitudes y pérdidas de carga por tramo.

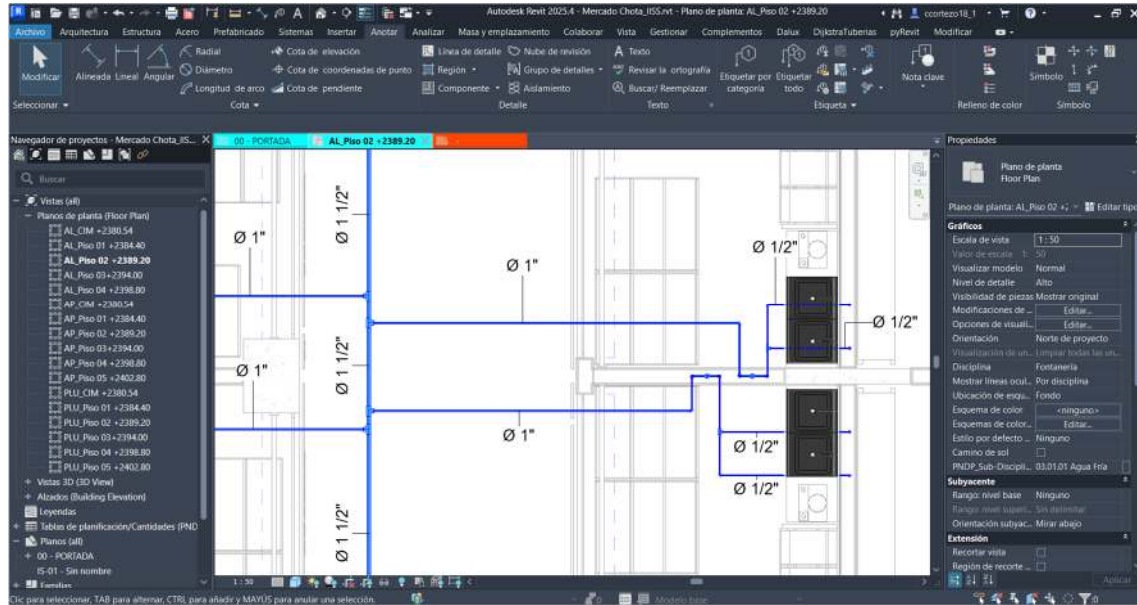
Figura 75. Modelado previo (todas las tuberías tienen el diámetro 1/2")



Fuente: Obtenido del Software Revit 2025

Nota: En la imagen se muestra las tuberías modeladas automáticamente con el plugin "Dijkstra Tuberías", en el modelo BIM de la especialidad de instalaciones sanitarias "Gran Mercado Modelo Chota"

Figura 76. Redimensionado de tuberías con plugin "PLUMBER"



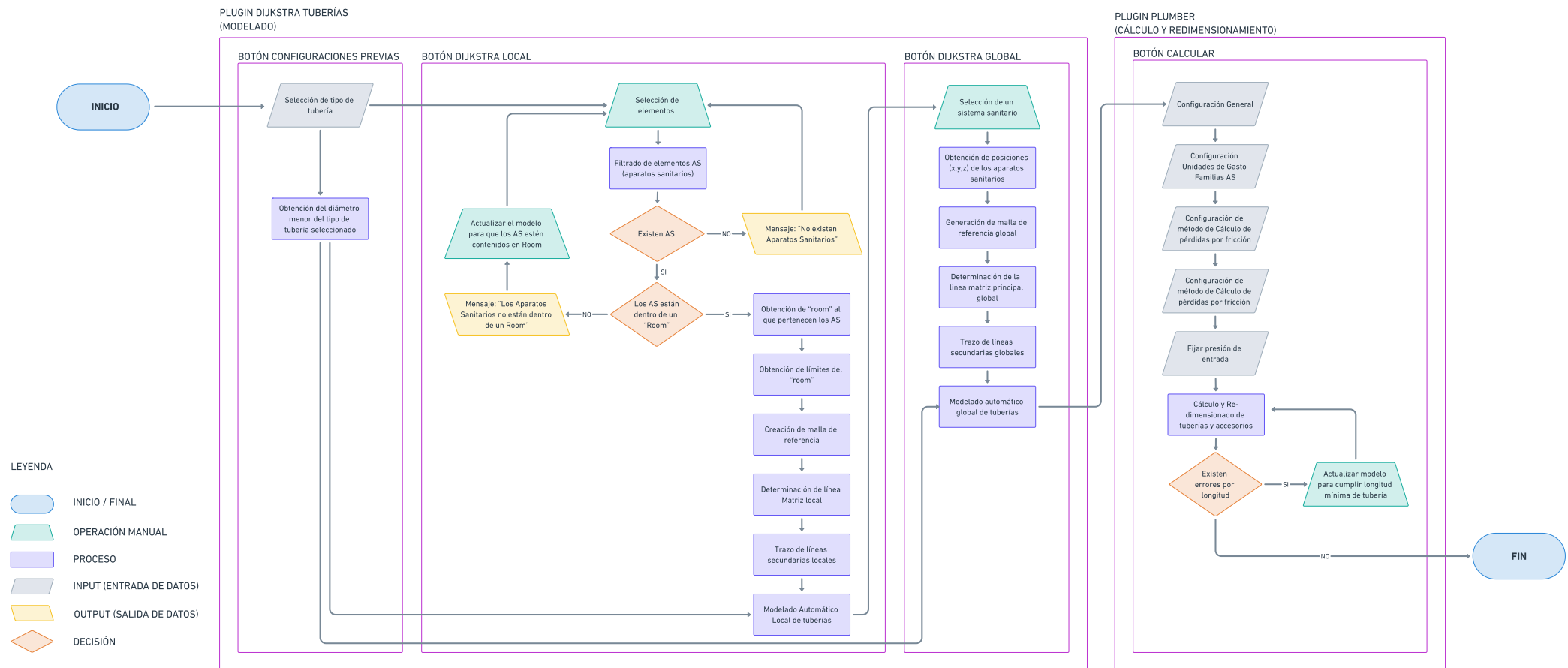
Fuente: Obtenido del Software Revit 2025

Nota: En la figura se muestra las tuberías y accesorios redimensionados automáticamente de acuerdo con un cálculo hidráulico realizado por el plugin "PLUMBER", en el modelo BIM de la especialidad de instalaciones sanitarias "Gran Mercado Modelo Chota".

3.5. Diagrama de flujo

Con la finalidad de facilitar la comprensión y ofrecer un resumen claro del funcionamiento del programa, a continuación, se presenta un diagrama de flujo general donde se puede ver fácilmente los pasos que se siguieron para realizar todo el proceso de automatización y optimización del modelado de tuberías.

Figura 77. Diagrama de flujo general automatización y optimización de modelado de tuberías



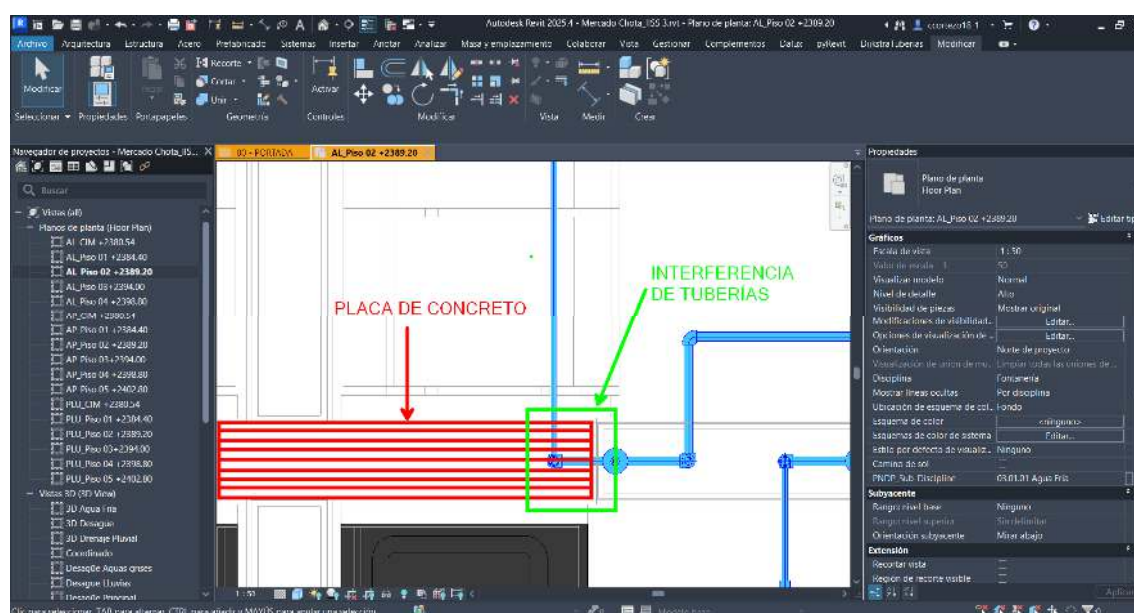
CAPÍTULO IV. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

4.1. Análisis de automatización

Se logró realizar con éxito la automatización del modelado de tuberías usando el algoritmo Dijkstra, la validez del resultado se sustenta en la propiedad del algoritmo que, al ser determinista, siempre encuentra la mejor ruta posible bajo el conjunto de condiciones dadas, sin necesidad de aproximaciones ni iteraciones aleatorias. El plugin desarrollado para el software Revit se basa en la geometría propia de las “habitaciones” o “rooms” que existan en el modelo en el cual esperamos aplicar la automatización, esto no representa ningún problema siempre y cuando el modelo desarrollado cuente con todas estos elementos desarrollados y modelados correctamente. Siguiendo esa línea de ideas también es necesario resaltar que se requiere de una preparación previa de las familias de la categoría “aparatos sanitarios” que estén presentes en el modelo, antes de ejecutar el plugin.

El plugin además no está diseñado para detectar interferencias con elementos estructurales o que no se permita atravesar una línea de tuberías, por lo que en algunas ocasiones puede generar errores de trazado.

Figura 78. Interferencia de tuberías con elemento estructural, área de frutas (“Gran Mercado Modelo Chota”)



Fuente: Obtenido del Software Revit 2025

4.2. Análisis y discusión de metrados de partidas

Se obtuvo el metrado optimizado con el propósito de comparar las afectaciones y, de ser el caso, identificar las variaciones en las partidas. Dado el enfoque de la investigación, las partidas afectadas corresponden exclusivamente a la sección de "Instalaciones Sanitarias". A continuación, se presenta una tabla en la que se detallan dichas partidas junto con sus respectivas variaciones en el metrado.

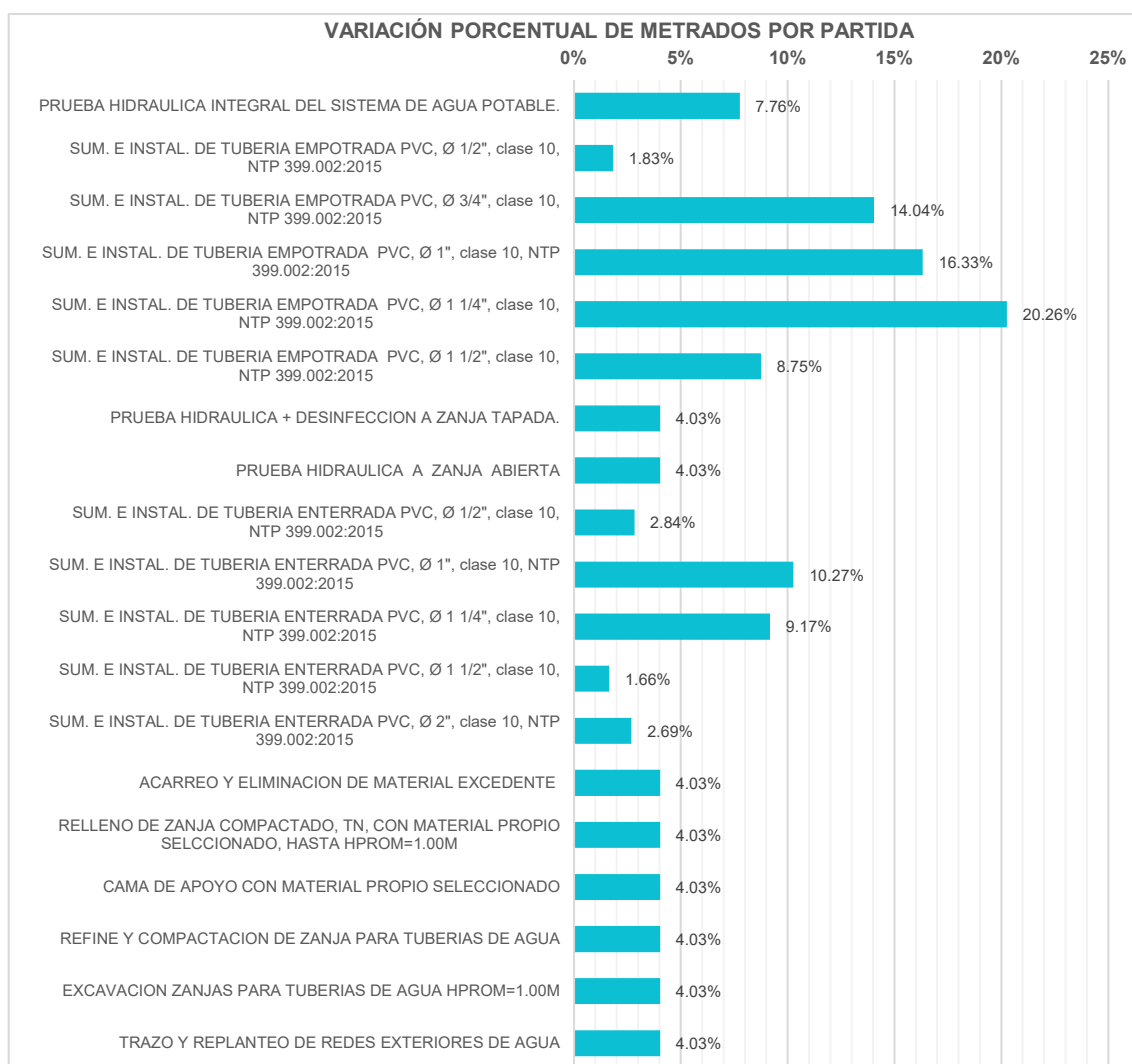
Tabla 5. Comparativo de partidas de "Instalaciones Sanitarias" ("Gran Mercado Modelo Chota")

ITEM	PARTIDA	UNIDAD	METRADO ORIGINAL	METRADO OPTIMIZADO
04.02	SISTEMA DE AGUA FRIA			
04.02.01	SALIDAS DE AGUA FRIA			
04.02.01.01	SALIDA DE AGUA FRIA CON TUBERIA DE PVC- CLASE 10, SP, Ø 1/2", NTP 399.002:2015	PTO	413.00	413.00
04.02.01.02	SALIDA DE AGUA FRIA CON TUBERIA DE PVC- CLASE 10 Ø 1", SP, NTP 399.002:2015	PTO	21.00	21.00
04.02.01.03	SALIDA DE AGUA FRIA CON TUBERIA DE PVC- CLASE 10 Ø 1 1/4", SP, NTP 399.002:2015	PTO	43.00	43.00
04.02.02	VÁLVULAS			
04.02.02.01	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE 2 1/2"	UND	7.00	7.00
04.02.02.02	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE 2"	UND	3.00	3.00
04.02.02.03	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE 1 1/2"	UND	22.00	22.00
04.02.02.04	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø 1 1/4"	UND	20.00	20.00
04.02.02.05	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø 1"	UND	9.00	9.00
04.02.02.06	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø 3/4"	UND	206.00	206.00
04.02.02.07	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø 1/2"	UND	12.00	12.00
04.02.02.08	SUMINISTRO E INSTALACION VÁLVULA ESFÉRICA DE BRONCE PESADA Ø 1/2"	UND	43.00	43.00
04.02.03	REDES DE AGUA FRIA ENTERRADA			
04.02.03.01	TRAZO Y REPLANTEO DE REDES EXTERIORES DE AGUA	ML	106.82	102.51
04.02.03.02	EXCAVACION ZANJAS PARA TUBERIAS DE AGUA HPROM=1.00M	ML	106.82	102.51
04.02.03.03	REFINE Y COMPACTACION DE ZANJA PARA TUBERIAS DE AGUA	ML	106.82	102.51
04.02.03.04	CAMA DE APOYO CON MATERIAL PROPIO SELECCIONADO	ML	106.82	102.51
04.02.03.05	RELLENO DE ZANJA COMPACTADO, TN, CON MATERIAL PROPIO SELCCIONADO, HASTA HPROM=1.00M	ML	106.82	102.51
04.02.03.06	ACARREO Y ELIMINACION DE MATERIAL EXCEDENTE	M3	3.27	3.14
04.02.03.07	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 2", clase 10, NTP 399.002:2015	ML	52.35	50.94
04.02.03.08	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1 1/2", clase 10, NTP 399.002:2015	ML	24.10	23.70

04.02.03.09	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1 1/4", clase 10, NTP 399.002:2015	ML	8.72	7.92
04.02.03.10	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1", clase 10, NTP 399.002:2015	ML	14.60	13.10
04.02.03.11	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1/2", clase 10, NTP 399.002:2015	ML	7.05	6.85
04.02.03.12	PRUEBA HIDRAULICA A ZANJA ABIERTA	ML	106.82	102.51
04.02.03.13	PRUEBA HIDRAULICA + DESINFECCION A ZANJA TAPADA.	ML	106.82	102.51
04.02.04	REDES DE AGUA FRIA ADOSADA			
04.02.04.01	SUM. E INSTAL. DE TUBERIA ADOSADA EN MURO PVC, Ø 3", clase 10, NTP 399.002:2015	ML	11.27	11.27
04.02.04.02	SUM. E INSTAL. DE TUBERIA ADOSADA EN MURO PVC, Ø 2", clase 10, NTP 399.002:2015	ML	9.60	9.60
04.02.05	REDES DE AGUA FRIA EMPOTRADA			
04.02.05.01	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 3", clase 10, NTP 399.002:2015	ML	10.95	10.95
04.02.05.02	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 2 1/2", clase 10, NTP 399.002:2015	ML	187.13	187.13
04.02.05.03	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 2", clase 10, NTP 399.002:2015	ML	138.90	138.90
04.02.05.04	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1 1/2", clase 10, NTP 399.002:2015	ML	570.68	520.75
04.02.05.05	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1 1/4", clase 10, NTP 399.002:2015	ML	132.85	105.93
04.02.05.06	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1", clase 10, NTP 399.002:2015	ML	56.35	47.15
04.02.05.07	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 3/4", clase 10, NTP 399.002:2015	ML	776.69	667.61
04.02.05.08	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1/2", clase 10, NTP 399.002:2015	ML	809.14	794.34
04.02.06	EQUIPOS Y OTRAS INSTALACIONES			
04.02.06.01	CAJA DE NICHOS EMPOTRADA EN MURO REVESTIDA CON CERAMICO Y TAPA METALICA GALVANIZADA PARA VALVULA COMPUERTA	UND	279.00	279.00
04.02.06.02	SUMINISTRO E INSTALACION DE ACCESORIOS PARA REDES DE DISTRIBUCIÓN	GLB	1.00	1.00
04.02.06.03	PRUEBA HIDRAULICA INTEGRAL DEL SISTEMA DE AGUA POTABLE.	ML	2,703.56	2,493.63
04.02.06.04	GRIFERIA TIPO JARDIN DE 1/2", INCL. NICHOS PARA VALVULA	UND	36.00	36.00
04.02.06.05	SUMINISTRO E INSTALACION DE CAJA DE CONCRETO Y TAPA REFORZADA PARA VALVULA EN PISO	UND	1.00	1.00
04.02.06.06	EMPALME DE CONEXIÓN Ø 1 1/2" A RED DE AGUA EXISTENTE, INCL. CAJA PORTAMEDIDOR	UND	1.00	1.00

En el siguiente gráfico se muestran las partidas afectadas y su variación porcentual comparando el modelado original y el automatizado.

Figura 79. Variación porcentual de metrados por partida



El gráfico de variación porcentual de metrados por partida demuestra que, tras la optimización del modelado sanitario, se identificaron diferencias significativas en varias partidas, particularmente en las correspondientes a la instalación de tuberías empotradas de PVC. Las mayores variaciones se registran en las partidas de tuberías de Ø 1 1/4", Ø 1" y Ø 3/4", con incrementos del 20.26%, 16.33% y 14.04% respectivamente. En cuanto a las tuberías enterradas, se observaron incrementos menores, como el 10.27% en Ø 1" y el 9.17% en Ø 1 1/4". Además, las partidas asociadas a excavaciones, rellenos, pruebas hidráulicas y desinfección presentaron una variación constante del 4.03%, lo que sugiere un ajuste global en función del rediseño del sistema.

En el expediente técnico analizado, el metrado correspondiente a los accesorios y uniones de tubería fue consignado de manera global dentro de la

partida "04.02.06.02 Suministro e Instalación de Accesorios para Redes de Distribución", lo que imposibilita una comparación directa a nivel de partida. No obstante, es factible realizar una comparación precisa a partir del análisis detallado de las tablas de metrados, las cuales permiten desagregar la información y evaluar específicamente los volúmenes asociados a accesorios y uniones. A continuación, se presenta una tabla que sintetiza los resultados obtenidos en dicha comparación.

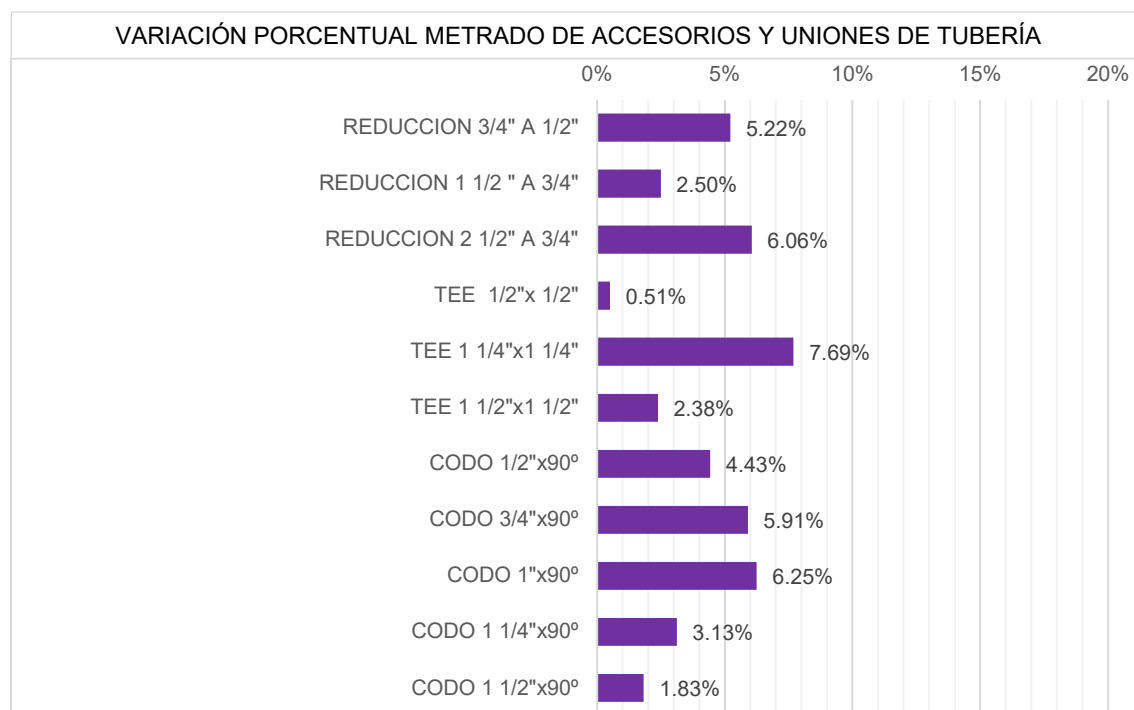
Tabla 6. Comparación del metrado de accesorios y uniones

ACCESORIOS		METRADO ORIGINAL	METRADO OPTIMIZADO	VARIACIÓN UNITARIA	VARIACIÓN PORCENTUAL
CODO	3"x90°	6	6	0	0.00%
	2 1/2"x90°	27	27	0	0.00%
	2"x90°	18	18	0	0.00%
	1 1/2"x90°	219	215	4	1.83%
	1 1/4"x90°	64	62	2	3.13%
	1"x90°	48	45	3	6.25%
	3/4"x90°	694	653	41	5.91%
	1/2"x90°	1308	1250	58	4.43%
TEE	3"x3"	3	3	0	0.00%
	2 1/2"x2 1/2"	48	48	0	0.00%
	2"x2"	13	13	0	0.00%
	1 1/2"x1 1/2"	210	205	5	2.38%
	1 1/4"x1 1/4"	26	24	2	7.69%
	1"x1"	12	12	0	0.00%
	3/4"x3/4"	18	18	0	0.00%
	1/2"x 1/2"	196	195	1	0.51%
REDUCCION	3" A 2 1/2"	3	3	0	0.00%
	2 1/2" A 2"	5	5	0	0.00%
	2 1/2" A 1 1/2"	11	11	0	0.00%
	2 1/2" A 3/4"	33	31	2	6.06%
	2" A 1 1/2"	17	17	0	0.00%
	2 " A 1"	1	1	0	0.00%
	2 " A 3/4"	1	1	0	0.00%
	1 1/2" A 1 1/4"	37	37	0	0.00%
	1 1/2 " A 1"	14	14	0	0.00%
	1 1/2 " A 3/4"	200	195	5	2.50%
	1 1/2 " A 1/2"	10	10	0	0.00%
	1 1/4 " A 1/2"	4	4	0	0.00%
	2 " A 1/2"	1	1	0	0.00%
	1" A 1/2"	6	6	0	0.00%
	3/4" A 1/2"	249	236	13	5.22%

En la tabla anterior se observan las variaciones específicas por tipo y dimensión de componente. Dentro de los codos, destacan variaciones significativas en el accesorio de 3/4"x90°, que presenta una reducción de 41 unidades (5.91%), así como en el codo de 1"x90° con una disminución del 6.25%. De forma similar, en el grupo de las tees, la más notable es la 1 1/4" x 1 1/4", con una reducción del 7.69%. Estas variaciones reflejan ajustes en el ruteo y en la racionalización del diseño de la red de agua potable, producto de la optimización del modelo BIM implementado.

En el caso de las reducciones, se identifican variaciones destacables en el accesorio 3/4" a 1/2", con una disminución de 13 unidades, equivalente a un 5.22%, y en la reducción 2" a 1 1/2", con un incremento del 6.06%. Si bien muchas de las variaciones son de baja magnitud o nulas, la presencia de reducciones puntuales revela que la optimización no solo incide en tramos mayores, sino también en componentes secundarios que, acumulativamente, pueden generar un impacto económico y técnico relevante en el proyecto.

Figura 80. Variación porcentual de metrado de accesorios y uniones de tubería



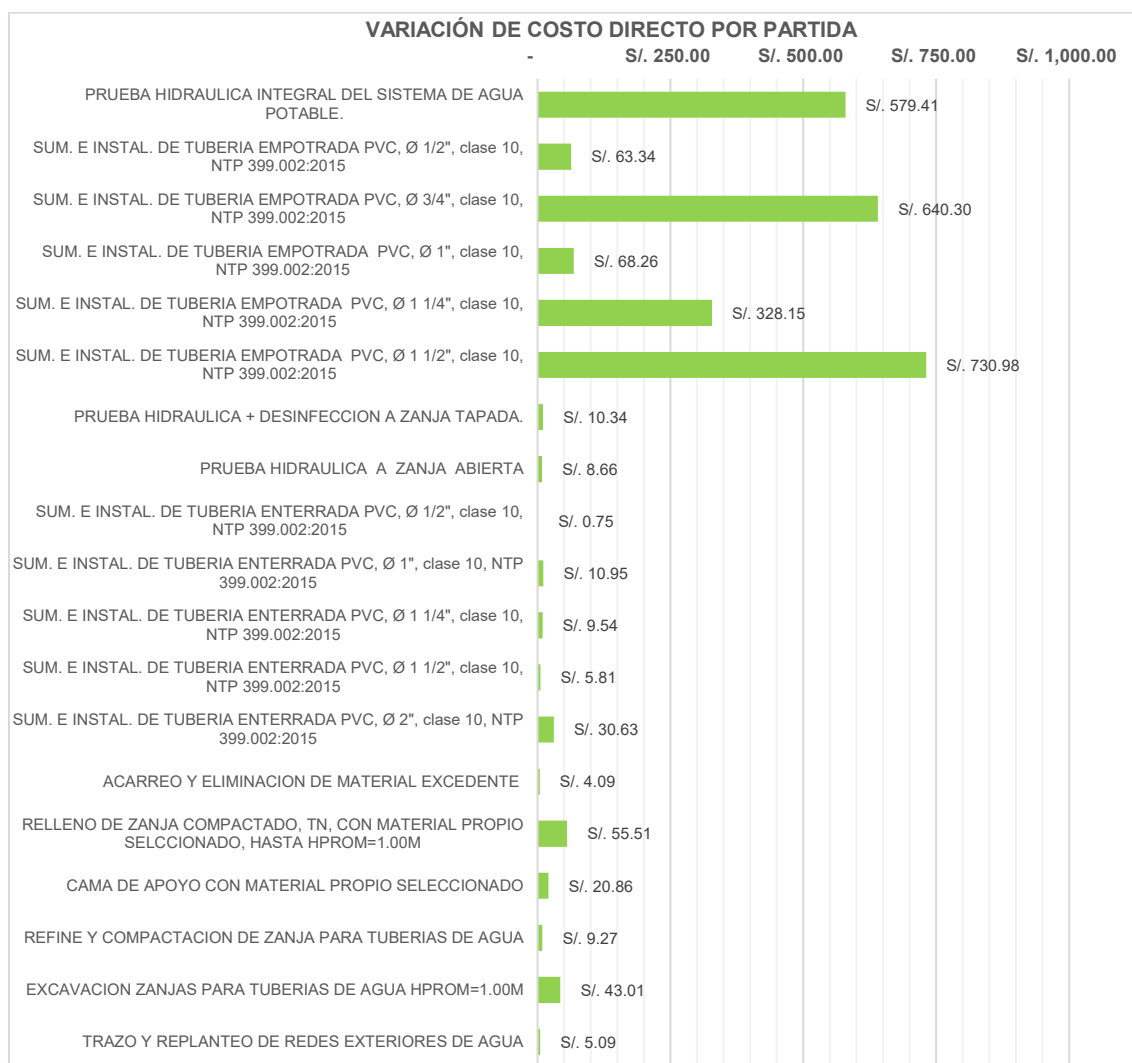
4.3. Análisis de variación en costos

A partir de los metrados optimizados, se procedió al recálculo del costo directo utilizando los mismos precios unitarios establecidos en el expediente técnico original, a fin de garantizar la consistencia metodológica y permitir una comparación objetiva. En la siguiente tabla se presenta la contrastación entre el costo original y el resultante tras la optimización (únicamente de las partidas afectadas), lo cual permite evidenciar de forma precisa el impacto económico derivado de la actualización cuantitativa basada en el modelado optimizado.

Tabla 7. Comparación de costo directo por partidas

ITEM	PARTIDA	COSTO DIRECTO ORIGINAL	COSTO DIRECTO OPTIMIZADO
04.02.03.01	TRAZO Y REPLANTEO DE REDES EXTERIORES DE AGUA	S/. 126.05	S/. 120.96
04.02.03.02	EXCAVACION ZANJAS PARA TUBERIAS DE AGUA HPROM=1.00M	S/. 1,066.06	S/. 1,023.05
04.02.03.03	REFINE Y COMPACTACION DE ZANJA PARA TUBERIAS DE AGUA	S/. 229.66	S/. 220.40
04.02.03.04	CAMA DE APOYO CON MATERIAL PROPIO SELECCIONADO	S/. 517.01	S/. 496.15
04.02.03.05	RELLENO DE ZANJA COMPACTADO, TN, CON MATERIAL PROPIO SELCCIONADO, HASTA HPROM=1.00M	S/. 1,375.84	S/. 1,320.33
04.02.03.06	ACARREO Y ELIMINACION DE MATERIAL EXCEDENTE	S/. 101.43	S/. 97.34
04.02.03.07	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 2", clase 10, NTP 399.002:2015	S/. 1,137.04	S/. 1,106.42
04.02.03.08	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1 1/2", clase 10, NTP 399.002:2015	S/. 350.17	S/. 344.36
04.02.03.09	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1 1/4", clase 10, NTP 399.002:2015	S/. 104.03	S/. 94.49
04.02.03.10	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1", clase 10, NTP 399.002:2015	S/. 106.58	S/. 95.63
04.02.03.11	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1/2", clase 10, NTP 399.002:2015	S/. 26.37	S/. 25.62
04.02.03.12	PRUEBA HIDRAULICA A ZANJA ABIERTA	S/. 214.71	S/. 206.05
04.02.03.13	PRUEBA HIDRAULICA + DESINFECCION A ZANJA TAPADA.	S/. 256.37	S/. 246.02
04.02.05.04	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1 1/2", clase 10, NTP 399.002:2015	S/. 8,354.76	S/. 7,623.78
04.02.05.05	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1 1/4", clase 10, NTP 399.002:2015	S/. 1,619.44	S/. 1,291.29
04.02.05.06	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1", clase 10, NTP 399.002:2015	S/. 418.12	S/. 349.85
04.02.05.07	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 3/4", clase 10, NTP 399.002:2015	S/. 4,559.17	S/. 3,918.87
04.02.05.08	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1/2", clase 10, NTP 399.002:2015	S/. 3,463.12	S/. 3,399.78
04.02.06.03	PRUEBA HIDRAULICA INTEGRAL DEL SISTEMA DE AGUA POTABLE.	S/. 7,461.83	S/. 6,882.42

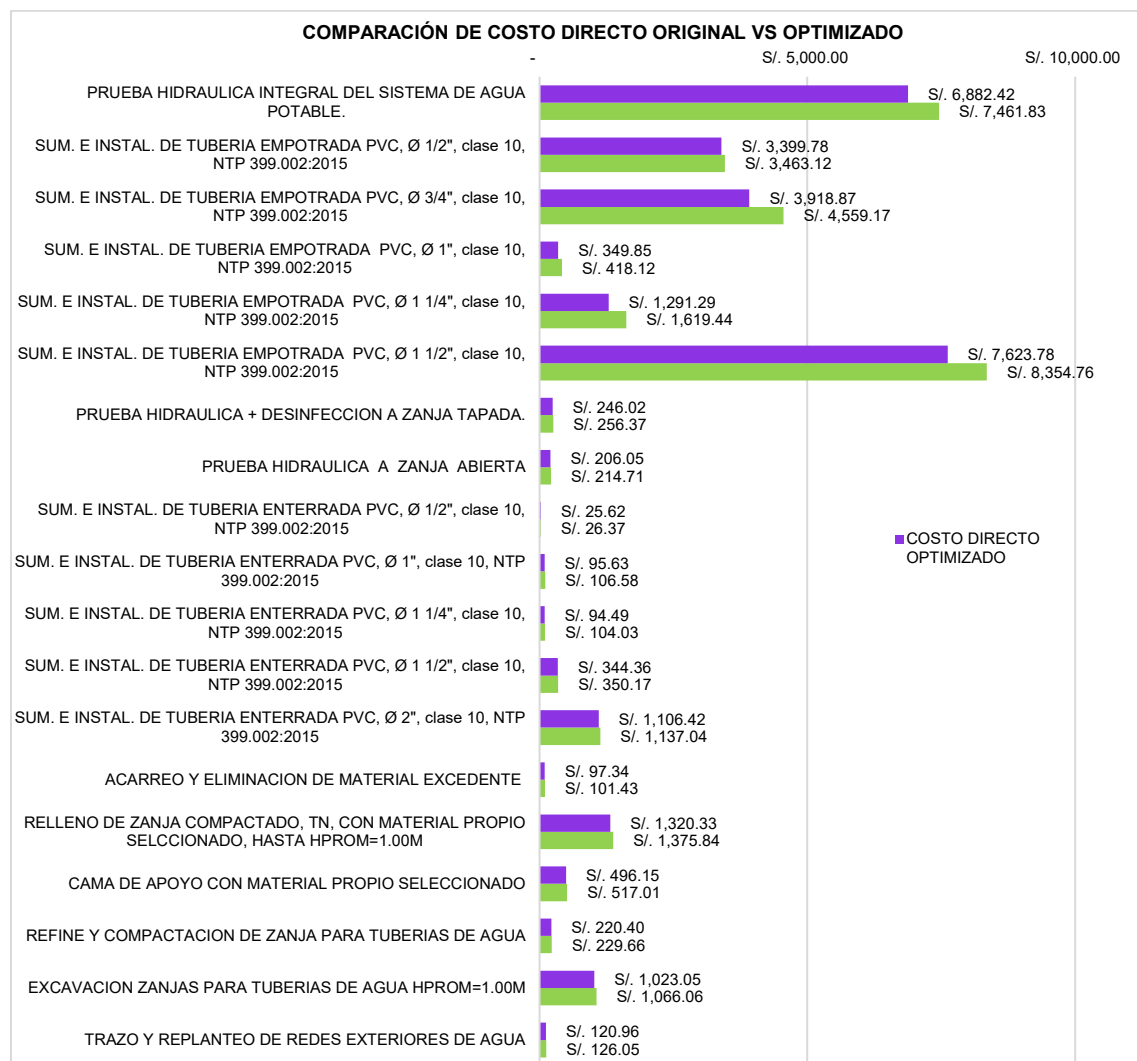
Figura 81. Variación de costo directo por partida



El gráfico presentado evidencia con claridad el impacto económico directo derivado de la optimización del metrado, revelando las variaciones monetarias del sistema de instalaciones sanitarias. Destacan con mayor magnitud las partidas correspondientes a la suministración e instalación de tubería empotrada de PVC, siendo las de Ø 1 1/2" y Ø 3/4" las que registran los mayores incrementos en costo directo, con S/. 730.98 y S/. 640.30 respectivamente. Les siguen las tuberías de Ø 1" con S/. 688.26 y de Ø 1 1/4" con S/. 328.15, lo que demuestra una correlación directa entre la optimización de cantidades y el ajuste sustancial en el presupuesto asignado a estos diámetros, usualmente asociados a tramos principales y secundarios en la red de agua.

Asimismo, partidas como la prueba hidráulica integral del sistema de agua potable muestran una variación de S/. 579.41, lo que refuerza el argumento de que el redimensionamiento influye no solo en materiales, sino también en procedimientos técnicos complementarios. Por otro lado, las partidas vinculadas a la instalación de tuberías enterradas y a actividades generales de zanjeo, compactación, y relleno, aunque presentan montos menores de entre S/. 4.09 y S/. 55.51.

Figura 82. Comparación costo directo original vs optimizado



CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

El complemento “Dijkstra Tuberías” procesó las salidas de agua reportadas en la Tabla 5 y consideró 477 puntos de salida en total, 413 salidas con tubería Ø 1/2", 21 salidas con tubería Ø 1" y 43 salidas con tubería Ø 1 1/4", permitiendo que el algoritmo genere grafos y trazados automáticos contemplando cada tipo de salida para su correcta conexión hidráulica.

La aplicación del redimensionamiento hidráulico sobre el trazado generado afectó a las partidas de Redes de Agua Fría empotrada, enterrada y adosada. La suma de las longitudes optimizadas (es decir, las longitudes finales redimensionadas en el modelo) es de 2,596.14 m. Estos valores representan la extensión total de tuberías sobre las cuales se aplicó el redimensionamiento automático y sirvieron como base para el recálculo de presupuestos.

La comparación de metrados entre el diseño original y el optimizado evidenció reducciones en partidas específicas: la partida 04.02.05.05 (SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC Ø 1 1/4") pasó de 132.85 m a 105.93 m (-26.92 m, -20.26%), la 04.02.05.06 (SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC Ø 1") de 56.35 m a 47.15 m (-9.20 m, -16.33%) y la 04.02.05.07 (SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC Ø 3/4") de 776.69 m a 667.61 m (-109.08 m, -14.04%). En accesorios, la reducción también fue evidente: el codo 3/4"×90° disminuyó de 694 a 653 unidades (-41, -5.91%) y la tee 1 1/4"×1 1/4" de 26 a 24 unidades (-2, -7.69%).

La comparación de presupuestos mostró ahorros de S/. 730.98 en tuberías Ø 1 1/2", S/. 640.30 en Ø 3/4", S/. 688.26 en Ø 1", S/. 328.15 en Ø 1 1/4" y S/. 579.41 en la prueba hidráulica integral, equivalentes a una reducción aproximada del 5.2% en el costo directo de las partidas analizadas.

5.2. Recomendaciones

Se recomienda implementar la automatización del modelado de redes de agua fría mediante algoritmos de búsqueda de rutas como Dijkstra en edificaciones de mediana y gran escala, donde los beneficios en ahorro de tiempo, precisión y control presupuestal son más notorios, además de adaptar la metodología propuesta a distintas tipologías de edificaciones.

Se sugiere continuar con investigaciones orientadas a la integración de algoritmos más complejos (como A*, algoritmos genéticos u optimización multiobjetivo) y su implementación dentro del entorno BIM, con el propósito de ampliar el alcance de la automatización en otras especialidades de diseño, como redes de desagüe, ventilación o climatización.

Se propone que, en futuras investigaciones incorporar al complemento una capa de restricciones geométrico-constructivas y detección automática de interferencias: implementar, mediante Revit API (o exportación IFC + herramienta de clash), rutinas que penalicen aristas que atraviesen elementos estructurales o zonas no transitables y que devuelvan una lista de choques por ruta (tipo, elemento afectado y coordenadas). Incluir un parámetro de tolerancia y un modo “preprocesamiento” que marque automáticamente tramos que requieren revisión manual, registrando el número y la ubicación de correcciones para cuantificar la intervención humana necesaria

REFERENCIAS BIBLIOGRÁFICAS

- Adali, T., & Ortega, A. (2018). Applications of Graph Theory. *Proc. IEEE*, 106, 784–786. <https://api.semanticscholar.org/CorpusID:14060817>
- Artturi, J. (2022, diciembre 21). *What is C#?* . <https://builtin.com/software-engineering-perspectives/c-sharp>
- Asociación Solidaridad Países Emergentes. (2008). *Manual de Instalaciones Sanitarias en Edificios* (Segunda edición, Vol. 1). SINCO.
- Atencio, E., Araya, P., Oyarce, F., Herrera, R. F., Muñoz-La Rivera, F., & Lozano-Galant, F. (2022). Towards the Integration and Automation of the Design Process for Domestic Drinking-Water and Sewerage Systems with BIM. *Applied Sciences (Switzerland)*, 12(18). <https://doi.org/10.3390/app12189063>
- Autodesk. (2024, diciembre 20). *Revit Fundamentals*. About the behavior of elements in Revit.
- Baldwin, D. A. B. (2014). *Chapter 1 Section III A Handbook for Construction Planning and Scheduling, First Edition What is building information modelling (BIM)?*
- Cabrera Chávarry, E. M. (2023). *Automatización de Procesos en las Fases de Diseño Y Documentación de Expedientes Técnicos utilizando La Metodología VDC/BIM Y La Programación Visual* [Tesis para optar al título de Ingeniero Civil]. Universidad Nacional de Cajamarca.
- Cloudflare Inc. (2023, septiembre 28). *What is an API?* <https://www.cloudflare.com/es-es/learning/security/api/what-is-an-api/>
- DS -289-2019-EF (2019). *Disposiciones para la incorporación progresiva de BIM en la inversión pública*.
- Ershadi, M., Davis, P., Jefferies, M., & Mojtahedi, M. (2021). Implementation of Building Information Modelling in infrastructure construction projects: a study of dimensions and strategies. *International Journal of Information Systems and Project Management*, 9(4), 43–59. <https://doi.org/10.12821/ijispm090403>

- Flores García, I. A. (2018). *Aplicación del algoritmo genético para el cálculo del diámetro de las tuberías de una red de distribución de agua potable en el distrito de Tarapoto 2018*. Universidad Nacional de San Martín.
- Guía Nacional BIM (2021). https://www.mef.gob.pe/planbimperu/docs/recursos/guia_nacional_BIM.pdf
- Hejlsberg, A., Wiltamuth, S., & Golde H., P. (2002). *C# Language Specification* (2nd ed.).
- International Organization for Standardization (ISO). (2024). *Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries – Part 1: Data Schema* (2ª ed., Número ISO 16739-1:2024). ISO. <https://www.iso.org/standard/84123.html>
- Isikdag, U. (2015). *Building Information Models: An Introduction*. <https://api.semanticscholar.org/CorpusID:64029759>
- Jaiswal, N. K., & Chakrawarti, R. K. (2013). *Increasing no. of nodes for Dijkstra algorithm without degrading the performance*. <https://api.semanticscholar.org/CorpusID:124828120>
- Javaid, A. (2013). Understanding Dijkstra's Algorithm. *Telecommunications & Network Models eJournal*. <https://api.semanticscholar.org/CorpusID:123314645>
- Jesus González Marquez, R., & Soler, M. (2014). *Introducción a la metodología BIM*. <https://www.researchgate.net/publication/284159764>
- Kashani, A. R., Gandomi, M., Camp, C. V, Rostamian, M., & Gandomi, A. H. (2020). *Metaheuristics in civil engineering: A review*. <https://api.semanticscholar.org/CorpusID:225862432>
- Kim, S.-H., Ruy, W.-S., & Jang, B. S. (2013). The development of a practical pipe auto-routing system in a shipbuilding CAD environment using network optimization. *International Journal of Naval Architecture and Ocean Engineering*, 5(3), 468–477. <https://doi.org/10.2478/ijnaoe-2013-0146>

- Konstruedu. (2021, octubre 23). *Automatización BIM: ¿Por qué es importante y qué herramientas existen?*
- Lu, W., Zhang, D., & Rowlinson, S. (2013). *Association of Researchers in Construction Management*.
- Majeed, A., & Rauf, I. (2020). Graph Theory: A Comprehensive Survey about Graph Theory Applications in Computer Science and Social Networks. *Inventions*. <https://api.semanticscholar.org/CorpusID:212977009>
- Manelli, L. (2020). Design of Algorithms. En *Introducing Algorithms in C: A Step by Step Guide to Algorithms in C* (pp. 13–68). Apress. https://doi.org/10.1007/978-1-4842-5623-7_2
- Ministerio de Economía y Finanzas del Perú. (2021). *Plan de implementación y hoja de ruta del Plan BIM Perú*. <https://www.gob.pe/mef>
- Ministerio de Vivienda Construcción y Saneamiento - Perú. (2021). *Norma Técnica G.040: Definiciones* (1ª ed.). https://cdn.www.gob.pe/uploads/document/file/2365098/04%20G.040%20DEFINICIONES_RM%20N%C2%B0%20029-2021-VIVIENDA.pdf?v=1636055463
- NTP 399.002:2015. Tuberías de PVC-U para conducción de fluidos a presión (2015).
- NTP 399.019:2004. Conexiones inyectadas de PVC-U para conducción de fluidos a presión (2004).
- NTP-ISO 1452:2012. Sistemas de tuberías plásticas para abastecimiento de agua y alcantarillado – PVC-U (2012).
- Panda, M., & Mishra, A. (2018). A Survey of Shortest-Path Algorithms. En *International Journal of Applied Engineering Research* (Vol. 13, Número 9). <http://www.ripublication.com>
- Paula, J., Delatorre, M., & Santos, E. T. (2015). *Level of Development Management in a BIM Model: Analysis of a case Study*. <https://www.semanticscholar.org/paper/GEST%C3%83O-DO->

N%C3%8DVEL-DE-DETALHAMENTO-DA-
INFORMA%C3%87%C3%83O-EM-UM-Delatorre-
Santos/f48129b3c9c257d1a310d729ff5f3261383543c3

Piaseckienė, G. (2022). Dimensions of BIM in Literature: Review And Analysis. *Mokslas - Lietuvos ateitis*, 14(0), 1–11.
<https://doi.org/10.3846/mla.2022.16071>

Programa Nacional de Infraestructura Educativa. (2020, abril 19). *Modelo de expediente técnico*. <https://www.pronied.gob.pe/servicios/estudios-y-obras/modelo-de-expediente-tecnico/>

Reddy, H. (2013). *PATH FINDING - Dijkstra's and A* Algorithm's*.
<https://api.semanticscholar.org/CorpusID:124964414>

Reglamento de la ley N°32069, Ley General de Contrataciones Públicas. (2025).

Reinhardt, J., & Mathews, M. (2017). *The Automation of BIM for Compliance Checking: a Visual Programming Approach*.
<https://api.semanticscholar.org/CorpusID:86446649>

Resolución de contraloría N° 432-2023-CG “Ejecución de obras públicas por administración directa” (2023). www.gob.pe/contraloria

RM-242-2019-MVCS (2019). *Lineamientos Generales para el uso del BIM en proyectos de construcción*

Rojas Córdova, A. R., San Martín Blas, E. C., Peña Saavedra, H. E., Jara Enriquez, J., & Flores Zorrilla, J. V. (2022). *Aplicación de tecnologías 4.0 a proyectos de edificación* [Tesis para optar al grado de bachiller en Ingeniería Civil, Pontificia Universidad Católica del Perú].
<http://hdl.handle.net/20.500.12404/23938>

Sachdeva, K. (2012). Applications of Graphs in Real-Life. *International Journal of Advanced Research in Computer Science*, 3, 403–406.
<https://api.semanticscholar.org/CorpusID:69685300>

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers,*

Engineers, Contractors, and Facility Managers.
<https://doi.org/10.1002/9781119287568>

Santoro, M., Vaccari, L., & Smith, D. (2018). *Web Application Programming Interfaces (APIs): general-purpose standards, terms and European Commission initiatives APIs4DGov study-digital government APIs: the road to value-added open API-driven services.*
<https://doi.org/10.2760/675>

Singh, A. I., & A., H. C. (2023). AN ADVANCED FLIGHT SCHEDULING APPLICATION BASED ON DIJKSTRA'S ALGORITHM. *BSSS Journal of Computer*. <https://api.semanticscholar.org/CorpusID:261606628>

Tyurin, I. A. (2020). Automation identification of construction work and structural elements in BIM development. *IOP Conference Series: Materials Science and Engineering*, 913(4).
<https://doi.org/10.1088/1757-899X/913/4/042010>

Vargas Ramírez, C. I. (2016). *Consideraciones de diseño y cálculo de las instalaciones hidráulicas en una red de distribución de agua potable.* [Tesis para optar por el título de Ingeniero en Recursos Hidráulicos y Medio Ambiente]. Escuela colombiana de ingeniería Julio Garavito.

Venugopal, M., Eastman, C. M., & Teizer, J. (2015). An ontology-based analysis of the industry foundation class schema for building information model exchanges. *Adv. Eng. Informatics*, 29, 940–957.
<https://api.semanticscholar.org/CorpusID:1572098>

Víctor, I., Dávila, R., Salomon, J., & Quispe, T. (2017). *Optimización de redes pluviales de desagüe y alcantarillado del distrito de Paucarpata con métodos genéticos* [Universidad Nacional de San Agustín de Arequipa].
<http://repositorio.unsa.edu.pe/handle/UNSA/3417>

Win, Y. Y., & Thein, T. T. (2020). *A Shortest Route Problem by Using Dijkstra's Algorithm and Its Applications.*
<https://api.semanticscholar.org/CorpusID:235804568>

Wong, J. K. W., Zhou, J. X., & Chan, A. P. C. (2018). Exploring the linkages between the adoption of bim and design error reduction. *International*

Journal of Sustainable Development and Planning, 13(1), 108–120.
<https://doi.org/10.2495/SDP-V13-N1-108-120>

Zhang, N., Wang, J., Al-Hussein, M., & Yin, X. (2022). BIM-based automated design of drainage systems for panelized residential buildings. *International Journal of Construction Management*.
<https://doi.org/10.1080/15623599.2022.2085853>

ANEXOS

Anexo 01. Código fuente “Dijkstra Tuberías”

```
# -*- coding: utf-8 -*-
__title__ = "Dijkstra Local"
__doc__ = """Version = 1.0
Date = 15.07.2024

Description:
Aplica Dijkstra Local

How-To:
- Click the Button

Last update:
- [20.07.2024] - V1.0 RELEASE

Author: Jair Cortez"""

import clr
from Autodesk.Revit.DB import *
from Autodesk.Revit.DB.Plumbing import Pipe, PipeType
from pyrevit import script
import heapq

config = script.get_config("global")
selected_pipe = getattr(config, "select_pipe", None) # Asegúrate de
que select_pipe esté definido
if not selected_pipe:
    raise ValueError("No se ha configurado un tipo de tubería en
'select_pipe'. Verifica la configuración.")

print("Tipo de tubería seleccionado: {}".format(selected_pipe))

import math

uidoc = __revit__.ActiveUIDocument
app = __revit__.Application
doc = __revit__.ActiveUIDocument.Document #type: Document
selection = uidoc.Selection #type: Selection
active_view = doc.ActiveView

class JNode(object):
    _instances = {}
    node_counter = 1

    def __new__(cls, point, name=None):
        key = hash((round(point.X,9), round(point.Y,9)))
        if key in cls._instances:
            return cls._instances[key]
        instance = super(JNode, cls).__new__(cls)
        cls._instances[key] = instance
        return instance

    def __init__(self, point, name=None):
```

```

        if not hasattr(self, 'initialized'):
            self.point = point
            self.name = name or "N{}".format(JNode.node_counter)
            JNode.node_counter += 1
            self.initialized = True

    def __eq__(self, point):
        if isinstance(point, Point):
            return self.point.Equals(point)
        return False

    def to_string(self):
        return "JNode(name={}, x={}, y={})".format(self.name,
self.point.X, self.point.Y)

class JSegment:

    def __init__(self, line):
        self.start_node = JNode(line.GetEndPoint(0))
        self.end_node = JNode(line.GetEndPoint(1))
        self.line = line
        self.distance= line.ApproximateLength
        self.name = "S_{0}-{1}".format(self.start_node.name, self.end_node.name)

    def to_string(self):
        return "JSegment(name={}, start_node={},
end_node={})".format(self.name, self.start_node.name,
self.end_node.name)

    @staticmethod
    def join(segment1, segment2):
        if segment1.end_node.name == segment2.start_node.name:
            return
JSegment(Line.CreateBound(segment1.start_node.point,
segment2.end_node.point))
        if segment1.start_node.name == segment2.end_node.name:
            return
JSegment(Line.CreateBound(segment2.start_node.point,
segment1.end_node.point))
        if segment1.start_node.name == segment2.start_node.name:
            return JSegment(Line.CreateBound(segment1.end_node.point,
segment2.end_node.point))
        if segment1.end_node.name == segment2.end_node.name:
            return
JSegment(Line.CreateBound(segment2.start_node.point,
segment1.start_node.point))
        return None

class JMesh:
    def __init__(self, familyManager, boundary_lines):
        self.familyManager=familyManager
        self.base_lines = familyManager.lines
        self.boudary_lines = boundary_lines
        self.lines_mesh=self.get_lines_mesh()
        self.segments = [JSegment(line) for line in self.lines_mesh]

```

```

        self.segments_dictionary = self.get_segments_dictionary()

    def get_lines_mesh(self):
        expanded_lines = [self.get_expanded_line(line,
self.boudary_lines) for line in self.base_lines]
        intersection_points =
JMesh.get_intersection_points(expanded_lines)

intersection_points.extend(self.familyManager.get_main_points())
        cutted_lines = []
        for line in expanded_lines:
            cutted_lines.extend(JMesh.cut_line(line,
intersection_points))
        return cutted_lines

    def get_segments_dictionary(self):
        segments_dictionary = {}
        for segment in self.segments:
            segments_dictionary[segment.name] = segment
        return segments_dictionary

    def get_expanded_line(self, line, boundary_lines):
        start_point = line.GetEndPoint(0)
        end_point = line.GetEndPoint(1)
        point_in_boundary = None
        to_project_point = None

        for boundary_line in boundary_lines:
            if boundary_line.Distance(start_point) < 0.0001:
                point_in_boundary = start_point
                to_project_point = end_point
                break
            if boundary_line.Distance(end_point) < 0.0001:
                point_in_boundary = end_point
                to_project_point = start_point
                break

        intersection_points = []
        if point_in_boundary:
            for boundary_line in boundary_lines:
                if boundary_line.Distance(point_in_boundary) < 0.0001:
                    continue
                intersection_point =
JMesh.get_intersection_point_with_extended_lines(line, boundary_line)
                if intersection_point and
boundary_line.Distance(intersection_point) < 0.0001 and
point_in_boundary.DistanceTo(intersection_point) >
to_project_point.DistanceTo(intersection_point):
                    intersection_points.append(intersection_point)
                intersection_points.sort(key=lambda point:
line.Distance(point))
            else:
                for boundary_line in boundary_lines:
                    intersection_point =
JMesh.get_intersection_point_with_extended_lines(line, boundary_line)

```

```

        if intersection_point and
boundary_line.Distance(intersection_point) < 0.0001:
            intersection_points.append(intersection_point)
            intersection_points.sort(key=lambda point:
line.Distance(point))

        # Handle cases where intersection_points is empty
if not intersection_points:
            return line # Return the original line if no
intersections are found

        if point_in_boundary:
            return Line.CreateBound(point_in_boundary,
intersection_points[0])
        return Line.CreateBound(intersection_points[0],
intersection_points[1])

def get_path_dijkstra(self, start_node, end_node):
    origin_name=start_node.name
    target_name=end_node.name
    # Construir el grafo como un diccionario de adyacencia
    graph = {}
    segment_lookup = {}

    for segment in self.segments:
        # Asegurar que los nodos existen en el grafo
        if segment.start_node.name not in graph:
            graph[segment.start_node.name] = []
        if segment.end_node.name not in graph:
            graph[segment.end_node.name] = []

        # Agregar conexiones bidireccionales

graph[segment.start_node.name].append((segment.end_node.name,
segment.distance))

graph[segment.end_node.name].append((segment.start_node.name,
segment.distance))

        # Guardar segmentos para acceso rápido
        segment_lookup[(segment.start_node.name,
segment.end_node.name)] = segment
        segment_lookup[(segment.end_node.name,
segment.start_node.name)] = segment

        # Inicializar estructuras para Dijkstra
        distances = {node: float('inf') for node in graph} #
Distancias iniciales
        previous_nodes = {node: None for node in graph} # Para
reconstruir el camino
        distances[origin_name] = 0 # Distancia al origen es 0
        priority_queue = [(0, origin_name)] # Cola de prioridad con
(distancia, nodo)

        while priority_queue:

```

```

        current_distance, current_node =
heapq.heappop(priority_queue)

        # Si llegamos al nodo destino, terminamos
        if current_node == target_name:
            break

        # Procesar vecinos
        for neighbor, weight in graph[current_node]:
            distance = current_distance + weight
            if distance < distances[neighbor]:
                distances[neighbor] = distance
                previous_nodes[neighbor] = current_node
                heapq.heappush(priority_queue, (distance,
neighbor))

        # Reconstruir la ruta más corta como lista de JSegment
        path_segments = []
        current_node = target_name
        while previous_nodes[current_node] is not None:
            prev_node = previous_nodes[current_node]
            path_segments.insert(0, segment_lookup[(prev_node,
current_node)]) # Buscar el JSegment
            current_node = prev_node

        # Si el destino no es alcanzable
        if distances[target_name] == float('inf'):
            return float('inf'), []

        return distances[target_name], path_segments
    print(start_node.name)
    print(end_node.name)
    print('-----')
    for nd in JNode._instances.values():
        print(nd.name)

    if not start_node or not end_node:
        return None

    distances = {node: float('inf') for node in
JNode._instances.values()}
    previous_nodes = {node: None for node in
JNode._instances.values()}
    distances[start_node] = 0
    priority_queue = [(0, start_node)]

    while priority_queue:
        current_distance, current_node =
heapq.heappop(priority_queue)

        if current_node == end_node:
            path = []
            while previous_nodes[current_node]:
                segment =
self.get_segment_between_nodes(previous_nodes[current_node],
current_node)
                path.append(segment)

```

```

        current_node = previous_nodes[current_node]
        path.reverse()
        return path

    if current_distance > distances[current_node]:
        continue

    for segment in self.segments:
        neighbor = None
        if segment.start_node == current_node:
            neighbor = segment.end_node
        elif segment.end_node == current_node:
            neighbor = segment.start_node

        if neighbor:
            distance = current_distance + segment.distance
            if distance < distances[neighbor]:
                distances[neighbor] = distance
                previous_nodes[neighbor] = current_node
                heapq.heappush(priority_queue, (distance,
neighbor))

    return None

    def get_segment_between_nodes(self, start_node, end_node):
        segment_name = "S_{}-{}".format(start_node.name,
end_node.name)
        return self.segments_dictionary.get(segment_name)

    @staticmethod
    def get_intersection_point_with_extended_lines(line1, line2):
        x1, y1 = line1.GetEndPoint(0).X, line1.GetEndPoint(0).Y
        x2, y2 = line1.GetEndPoint(1).X, line1.GetEndPoint(1).Y
        x3, y3 = line2.GetEndPoint(0).X, line2.GetEndPoint(0).Y
        x4, y4 = line2.GetEndPoint(1).X, line2.GetEndPoint(1).Y

        denom = (x1 - x2) * (y3 - y4) - (y1 - y2) * (x3 - x4)

        if denom == 0:
            return None

        px = ((x1 * y2 - y1 * x2) * (x3 - x4) - (x1 - x2) * (x3 * y4 -
y3 * x4)) / denom
        py = ((x1 * y2 - y1 * x2) * (y3 - y4) - (y1 - y2) * (x3 * y4 -
y3 * x4)) / denom

        return XYZ(px, py, 0)

    @staticmethod
    def get_intersection_points(lines):
        intersection_points = []
        for i, line1 in enumerate(lines):
            for j, line2 in enumerate(lines):
                if i != j:
                    intersection_point = JMesh.get_intersection_point
(line1, line2)

                    if intersection_point:

```

```

        intersection_points.append(intersection_point)
    return intersection_points

    @staticmethod
    def get_intersection_point(line1, line2):
        if (line1.Intersect(line2) == SetComparisonResult.Overlap):
            return
JMesh.get_intersection_point_with_extended_lines(line1, line2)
    return None

    @staticmethod
    def cut_line(line, intersection_points):
        intersection_points_in_line = [point for point in
intersection_points if line.Distance(point) < 0.0001]
        if not intersection_points_in_line:
            return [line]
        intersection_points_in_line.sort(key=lambda point:
line.GetEndPoint(0).DistanceTo(point))
        lines = []
        start = line.GetEndPoint(0)
        for intersection_point in intersection_points_in_line:
            end = intersection_point
            if start.DistanceTo(end) > 0.0001:
                lines.append(Line.CreateBound(start, end))
                start = end
            if start.DistanceTo(line.GetEndPoint(1)) > 0.0001:
                lines.append(Line.CreateBound(start, line.GetEndPoint(1)))
        return lines

class JTarget:
    def __init__(self, target_family, name=None):

self.node=JNode(FamilyManager.get_main_point_of_family(target_family),
name)

        self.weight=self.get_weight()

    def get_weight(self):
        return 2

class FamilyManager:
    def __init__(self, origin, targets):
        self.origin=origin
        self.targets=targets

self.origin_point=FamilyManager.get_main_point_of_family(self.origin)
        self.target_points=[FamilyManager.get_main_point_of_family(tg)
for tg in targets]
        self.lines=self.calculate_lines()
        self.origin_node=JNode(self.origin_point, 'N-ORIGIN')
        self.target_nodes=[JNode(p) for p in self.target_points]
        self.targets_info=[JTarget(tg) for tg in targets]

    def calculate_lines(self):
        families=[]
        families.append(self.origin)
        families.extend(self.targets)

```

```

        lines=[]
        for f in families:
            lines.extend(FamilyManager.get_lines_of_family(f))
        return lines

    def get_main_points(self):
        points=[]
        points.append(self.origin_point)
        points.extend(self.target_points)
        return points

    @staticmethod
    def get_lines_of_family(family):
        options_geo = Options()
        options_geo.ComputeReferences = True
        options_geo.IncludeNonVisibleObjects = False
        transform=family.GetTransform()
        geometry = family.get_Geometry(options_geo)
        return get_lines_of_geometry(geometry, transform)

    @staticmethod
    def get_main_point_of_family(family):
        lines = FamilyManager.get_lines_of_family(family)
        points=[]
        for line in lines:
            points.append(line.GetEndPoint(0))
            points.append(line.GetEndPoint(1))

        evaluated_points={}
        for point in points:
            key = hash((round(point.X,9), round(point.Y,9)))
            if key in evaluated_points:
                return point
            evaluated_points[key]=point
        return None

class JPathManager:
    def __init__(self,mesh):
        self.mesh=mesh

    def get_stretches(self):
        main_path=self.get_main_path()
        nodes=self.get_nodes_from_path(main_path['path'])
        secondary_paths=self.get_secondary_paths(main_path,nodes)

        base_segments=[]

        base_segments.extend(self.get_segments_from_path(main_path['path']))
        for secondary_path in secondary_paths:
            base_segments.extend(self.get_segments_from_path(secondary_path['path']
)))

```



```

main_nodes_dictionary=self.get_main_nodes_dictionary(base_segments)

main_path_segments=self.get_joined_segments_by_path(main_path['path'],
main_nodes_dictionary)

secondary_paths_segments=[self.get_joined_segments_by_path(secondary_p
ath['path'],main_nodes_dictionary) for secondary_path in
secondary_paths]

    targets_dictionary={}
    for target in self.mesh.familyManager.targets_info:
        targets_dictionary[target.node.name]=target

    stretches={}
    for segments in secondary_paths_segments:

stretch=self.get_secondary_stretch(targets_dictionary,segments)
    stretches[stretch.end_node.name]=stretch

main_stretches=self.get_main_stretches(targets_dictionary,stretches,main_
path_segments)
    return stretches.values()+main_stretches

def get_secondary_stretch(self,target_dictionary,segments):
    target=None
    if segments[0].start_node.name in target_dictionary:
        target=target_dictionary[segments[0].start_node.name]
    if segments[0].end_node.name in target_dictionary:
        target=target_dictionary[segments[0].end_node.name]

    return Stretch(segments,target.weight)

def
get_main_stretches(self,target_dictionary,secondary_stretches,segments):
    target=None
    segments.reverse()
    share_node=None
    if segments[0].start_node.name in target_dictionary:
        target=target_dictionary[segments[0].start_node.name]
        share_node=segments[0].start_node
    if segments[0].end_node.name in target_dictionary:
        target=target_dictionary[segments[0].end_node.name]
        share_node=segments[0].end_node

    stretches=[]
    weight=target.weight
    segments_to_stretch=[]
    for segment in segments:
        if segment.start_node.name==share_node.name:
            share_node=segment.end_node
        else:
            share_node=segment.start_node
        segments_to_stretch.append(segment)
    if share_node.name in secondary_stretches:

```

```

        secondary_stretch=secondary_stretches[share_node.name]
        stretches.append(Stretch(segments_to_stretch,weight))
        weight+=secondary_stretch.weight
        segments_to_stretch=[]

    stretches.append(Stretch(segments_to_stretch,weight))
    return stretches

def get_joined_segments_by_path(self,path,main_nodes_dictionary):
    joined_segments=[]
    if(len(path)==1):
        return path

    joined_segment=None
    direction=None
    nodes=None

    for segment in path:
        if not joined_segment:
            joined_segment=segment
            direction=segment.line.Direction

    nodes=set([segment.start_node.name,segment.end_node.name])
        continue

    share_node=None
    other_node=None

    if segment.start_node.name in nodes:
        share_node=segment.start_node
        other_node=segment.end_node
    if segment.end_node.name in nodes:
        share_node=segment.end_node
        other_node=segment.start_node
    if share_node.name in main_nodes_dictionary or
(direction.DistanceTo(segment.line.Direction)>0.0001 and
direction.DistanceTo(segment.line.Direction)<2):
        joined_segments.append(joined_segment)
        joined_segment=segment
        direction=segment.line.Direction

    nodes=set([segment.start_node.name,segment.end_node.name])
        continue
    if other_node.name in main_nodes_dictionary:
        joined_segment=JSegment.join(joined_segment,segment)
        joined_segments.append(joined_segment)
        joined_segment=None
        continue
    joined_segment=JSegment.join(joined_segment,segment)
    nodes=set([segment.start_node.name,segment.end_node.name])

    if joined_segment:
        joined_segments.append(joined_segment)
    return joined_segments

```

```

def get_main_nodes_dictionary(self, segments):
    nodes={}
    for segment in segments:
        if segment.start_node.name in nodes:
            nodes[segment.start_node.name]['quantity']+=1
        else:
            nodes[segment.start_node.name]={}
            nodes[segment.start_node.name]['quantity']=1

nodes[segment.start_node.name]['node']=segment.start_node

        if segment.end_node.name in nodes:
            nodes[segment.end_node.name]['quantity']+=1
        else:
            nodes[segment.end_node.name]={}
            nodes[segment.end_node.name]['quantity']=1
            nodes[segment.end_node.name]['node']=segment.end_node

    main_node_names=[node['node'].name for node in nodes.values()]
if node['quantity']>2]
    return set(main_node_names)

def get_segments_from_path(self, path):
    return [segment for segment in path]

def get_secondary_paths(self, main_path, main_path_nodes):

    main_target_node_name=main_path['node_name']
    target_nodes=[node for node in
self.mesh.familyManager.target_nodes if
node.name!=main_target_node_name]
    secondary_paths=[]
    for target_node in target_nodes:
        secondary_path={
            'distance':main_path['distance'],
            'path':[],
            'node_name':target_node.name
        }
        for node in main_path_nodes:
            [distance,path] =
self.mesh.get_path_dijkstra(target_node,node)
            if distance<secondary_path['distance']:
                secondary_path['distance']=distance
                secondary_path['path']=path
            secondary_paths.append(secondary_path)
    return secondary_paths

def get_main_path(self):
    origin_node=self.mesh.familyManager.origin_node
    target_nodes=self.mesh.familyManager.target_nodes
    main_path={
        'distance':0,
        'path':[],
        'node_name':None
    }

```

```

        for target_node in target_nodes:
            [distance,path] =
self.mesh.get_path_dijkstra(origin_node,target_node)
            if distance>main_path['distance']:
                main_path['distance']=distance
                main_path['path']=path
                main_path['node_name']=target_node.name
        return main_path

    def get_nodes_from_path(self,path):
        nodes={}
        for segment in path:
            nodes[segment.start_node.name]=segment.start_node
            nodes[segment.end_node.name]=segment.end_node
        return nodes.values()

# Tramo
class Stretch:
    def __init__(self,path,weight):
        self.path=path
        self.start_node=path[0].start_node
        self.end_node=path[-1].end_node
        self.distance=sum([segment.distance for segment in path])
        self.weight=weight
        self.name="STRETCH_{}-
{}".format(self.start_node.name,self.end_node.name)

    def to_string(self):
        title=self.name
        details="Distance: {} Weight:
{}".format(self.distance,self.weight)
        nodes="Start Node: {} End Node:
{}".format(self.start_node.name,self.end_node.name)
        segments="Segments: {}".format([segment.name for segment in
self.path])
        line="-----"
        return
"{} \n{} \n{} \n{} \n{} \n{} \n{} \n".format(line,title,line,details,nodes,
segments,line)

    def get_selected_elements():
        selected_element_ids = selection.GetElementIds()
        selected_elements = [doc.GetElement(e_id) for e_id in
selected_element_ids]
        return selected_elements

    def draw_lines(lines):
        with Transaction(doc, "Draw lines") as trans:
            trans.Start()
            for line in lines:
                doc.Create.NewDetailCurve(active_view, line)
            trans.Commit()

    def get_lines_of_geometry(geometria, transform):
        segments = []
        for obj_geo in geometria:

```

```

        if isinstance(obj_geo, GeometryInstance):
            geo_simbolo = obj_geo.GetSymbolGeometry()

segments.extend(get_lines_of_geometry(geo_simbolo, transform))
        elif isinstance(obj_geo, Curve):
            start = transform.OfPoint(obj_geo.GetEndPoint(0))
            start_point=XYZ(start.X, start.Y, 0)
            end = transform.OfPoint(obj_geo.GetEndPoint(1))
            end_point=XYZ(end.X, end.Y, 0)
            segments.append(Line.CreateBound(start_point, end_point))
    return segments

def get_lines_boundary_of_room(room):
    options=SpatialElementBoundaryOptions()
    boundary = room.GetBoundarySegments(options)
    segments = []
    for segment_list in boundary:
        for boundary_segment in segment_list:
            start_point = boundary_segment.GetCurve().GetEndPoint(0)
            end_point = boundary_segment.GetCurve().GetEndPoint(1)
            segments.append(Line.CreateBound(start_point, end_point))
    return segments

def get_system_type_id(doc):
    # Obtener el primer MEPSysTypeType disponible que sea de tipo
    PipeSystemType
    system_type =
    FilteredElementCollector(doc).OfClass(MEPSysTypeType).OfCategory(BuiltI
nCategory.OST_PipingSystem).FirstElement()
    if system_type:
        return system_type.Id
    else:
        raise ValueError("No se encontró ningún PipeSystemType en el
documento.")

def create_pipe_segment(doc, start_point, end_point):
    """Crea un segmento de tubería entre dos puntos."""
    system_type_id = get_system_type_id(doc)
    level_id = doc.ActiveView.GenLevel.Id

    # Busca el tipo de tubería seleccionado
    pipe_type = next(
        (pt for pt in
    FilteredElementCollector(doc).OfClass(PipeType).ToElements()
        if Element.Name.GetValue(pt) == selected_pipe),
        None
    )
    if not pipe_type:
        raise ValueError("No se encontró ningún tipo de tubería con el
nombre '{}'. ".format(selected_pipe))

    # Crea la tubería
    pipe = Pipe.Create(doc, system_type_id, pipe_type.Id, level_id,
start_point, end_point)
    return pipe

def main():

```

```

"""Función principal."""
selected_elements = get_selected_elements()
target_families = [e for e in selected_elements if e.Category.Name
== "Aparatos sanitarios"]
origin = next((e for e in target_families if e.Name == "ORIGEN"),
None)
targets = [e for e in target_families if e.Name != "ORIGEN"]

if not origin or not targets:
    raise ValueError("No se encontraron elementos válidos para
procesar.")

room = next((e for e in selected_elements if e.Category.Name ==
"Habitaciones"), None)
if not room:
    raise ValueError("No se encontró una habitación
seleccionada.")

room_lines_boundary = get_lines_boundary_of_room(room)
family_manager = FamilyManager(origin, targets)
mesh = JMesh(family_manager, room_lines_boundary)

##draw_lines(mesh.lines_mesh)
path_manager = JPathManager(mesh)
stretchs = path_manager.get_stretchs()

with Transaction(doc, "Create Pipes") as trans:
    trans.Start()
    for stretch in stretches:
        for segment in stretch.path:
            create_pipe_segment(doc, segment.start_node.point,
segment.end_node.point)
    trans.Commit()

# Ejecuta el script
if __name__ == "__main__":
    main()

```

Anexo 02. Análisis de Precios Unitarios (Instalaciones Sanitarias)

Presupuesto

Presupuesto	0001008	"CREACIÓN SERVICIOS DE COMERCIALIZACIÓN EN EL GRAN MERCADO MODELO DE LA CIUDAD DE CHOTA DISTRITO DE CHOTA - PROVINCIA DE CHOTA - DEPARTAMENTO DE CAJAMARCA" - CUI 2497551	
Subpresupuesto	003	INSTALACIONES SANITARIAS	
Cliente	MUNICIPALIDAD PROVINCIAL DE CHOTA		Costo al 23/02/2023
Lugar	CAJAMARCA - CHOTA - CHOTA		

Item	Descripción	Und.	Metrado	Precio S/.	Parcial S/.
04	INSTALACIONES SANITARIAS				1,845,379.94
04.01	APARATOS SANITARIOS Y ACCESORIOS				253,967.05
04.01.01	SUMINISTRO DE APARATOS SANITARIOS				253,967.05
04.01.01.01	SUMINISTRO E INSTALACION DE INODORO CON FLUXOMETRO, DESCARGA INDIRECTA, COLOR BLANCO INC. ACCESORIOS	pza	38.00	1,094.00	41,572.00
04.01.01.02	SUMINISTRO E INSTALACION DE INODORO CON FLUXOMETRO, DESCARGA INDIRECTA, COLOR BLANCO, INC. ACCESORIOS (DISCAPCITADOS)	pza	4.00	1,224.61	4,898.44
04.01.01.03	SUMINISTRO E INSTALACION INODORO TANQUE BAJO BLANCO INC. ACCESORIOS	pza	1.00	270.65	270.65
04.01.01.04	SUMINISTRO E INSTALACION DE LAVATORIO DE SOBREPONER COLOR BLANCO. INC ACCESORIOS Y LLAVE DE CIERRE TEMPORIZADA	pza	43.00	748.83	32,199.69
04.01.01.05	SUMINISTRO E INSTALACION DE LAVADERO DE DOS POZAS 21"x40" DE PROF. 15cm DE ACERO INOX 1/16" SATINADO DE SOBREPONER, SIN ESCURRIDERO, INC GRIFERÍA	pza	173.00	712.98	123,345.54
04.01.01.06	SUMINISTRO E INSTALACION DE LAVADERO DE UNA POZAS 21"x20" DE PROF. 15cm DE ACERO INOX 1/16" SATINADO DE SOBREPONER, SIN ESCURRIDERO, INC GRIFERÍA	pza	2.00	381.79	763.58
04.01.01.07	SUMINISTRO E INSTALACION DE LAVADERO DE ACERO INOXIDABLE DE UNA POZA C/ESCURRIDERA INC. ACCESORIOS Y GRIFERÍA	pza	14.00	347.97	4,871.58
04.01.01.08	SUMINISTRO E INSTALACION URINARIOS CON FLUXOMETRO. DESCARGA INDIRECTA, COLOR BLANCO INC. ACCESORIOS	pza	19.00	926.20	17,597.80
04.01.01.09	SUMINISTRO E INSTALACION LLAVE DE DUCHA CON SALIDA ESPAÑOLA, CROMADA, INCLUYE ACCESORIOS	pza	15.00	350.71	5,260.65
04.01.01.10	SUMINISTRO E INSTALACION DE LLAVE ESFERICA PARA LAVADERO (LAVAMOPAS PARA ASEO DE DOS POZAS), INC. ACCESORIOS	pza	6.00	78.81	472.86
04.01.01.11	SUMINISTRO E INSTALACION DE PORTAROLLO HIGIENICO DE ACERO INOXIDABLE SATINADO	und	43.00	55.05	2,367.15
04.01.01.12	SUMINISTRO E INSTALACION DE JABONERA DE LOSA COLOR BLANCO	und	15.00	40.64	609.60
04.01.01.13	SUMINISTRO E INSTALACION DE COLGADOR GANCHO DE ACERO INOXIDABLE SATINADO	und	43.00	38.10	1,638.30
04.01.01.14	SUMINISTRO E INSTALACION DE DISPENSADOR DE JABON DE ACERO INOXIDABLE SATINADO	und	27.00	110.13	2,973.51
04.01.01.15	SUMINISTRO E INSTALACION DE DISPENSADOR DE PLASTICO PAPEL TOALLA EN ROLLO	und	18.00	177.93	3,202.74
04.01.01.16	SUMINISTRO E INSTALACION DE SECADOR DE MANOS DE ACERO INOXIDABLE SATINADO	und	21.00	567.76	11,922.96
04.02	SISTEMA DE AGUA FRIA				255,629.56
04.02.01	SALIDA DE AGUA FRIA				25,045.09
04.02.01.01	SALIDA DE AGUA FRIA CON TUBERIA DE PVC- CLASE 10,SP, Ø 1/2", NTP 399.002:2015	pto	413.00	45.10	18,626.30
04.02.01.02	SALIDA DE AGUA FRIA CON TUBERIA DE PVC- CLASE 10, Ø 1",SP, NTP 399.002:2015	pto	21.00	91.66	1,924.86
04.02.01.03	SALIDA DE AGUA FRIA CON TUBERIA DE PVC- CLASE 10 Ø 1 1/4", SP, NTP 399.002:2015	pto	43.00	104.51	4,493.93
04.02.02	VALVULAS				100,295.51
04.02.02.01	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø2 1/2"	und	7.00	790.13	5,530.91
04.02.02.02	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø 2"	und	3.00	638.78	1,916.34
04.02.02.03	SUMINISTRO E INSTALACION DE VALVULA DE COMPUERTA DE BRONCE 1 1/2"	und	22.00	522.89	11,503.58
04.02.02.04	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø1 1/4"	und	20.00	422.22	8,444.40
04.02.02.05	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø1"	und	9.00	372.22	3,349.98
04.02.02.06	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø 3/4"	und	206.00	308.06	63,460.36
04.02.02.07	SUMINISTRO E INSTALACION DE VALVULA COMPUERTA DE BRONCE DE Ø1/2"	und	12.00	206.28	2,475.36
04.02.02.08	SUMINISTRO E INSTALACION VÁLVULA ESFÉRICA DE BRONCE PESADA Ø 1/2"	und	43.00	84.06	3,614.58
04.02.03	REDES DE AGUA FRIA ENTERRADA				5,611.26
04.02.03.01	TRAZO Y REPLANTEO DE REDES DE AGUA	m	106.82	1.18	126.05
04.02.03.02	EXCAVACION DE ZANJAS PARA TUBERIAS, HPROM=1.00M	m	106.82	9.98	1,066.06
04.02.03.03	REFINE Y COMPACTACION DE ZANJA PARA TUBERIAS DE AGUA	m	106.82	2.15	229.66
04.02.03.04	CAMA DE APOYO CON MATERIAL PROPIO SELECCIONADO	m	106.82	4.84	517.01

Presupuesto

Presupuesto	0001008	"CREACIÓN SERVICIOS DE COMERCIALIZACIÓN EN EL GRAN MERCADO MODELO DE LA CIUDAD DE CHOTA DISTRITO DE CHOTA - PROVINCIA DE CHOTA - DEPARTAMENTO DE CAJAMARCA" - CUI 2497551	
Subpresupuesto	003	INSTALACIONES SANITARIAS	
Cliente	MUNICIPALIDAD PROVINCIAL DE CHOTA		Costo al 23/02/2023
Lugar	CAJAMARCA - CHOTA - CHOTA		

Item	Descripción	Und.	Metrado	Precio S/.	Parcial S/.
04.02.03.05	RELLENO DE ZANJA COMPACTADO, TN, CON MATERIAL PROPIO SELECCIONADO, HASTA HPROM=1.00M	m	106.82	12.88	1,375.84
04.02.03.06	ACARREO Y ELIMINACION DE MATERIAL EXCEDENTE	m3	3.27	31.00	101.37
04.02.03.07	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 2", clase 10, NTP 399.002:2015	m	52.35	21.72	1,137.04
04.02.03.08	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1 1/2", clase 10, NTP 399.002:2015	m	24.10	14.53	350.17
04.02.03.09	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1 1/4", clase 10, NTP 399.002:2015	m	8.72	11.93	104.03
04.02.03.10	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1", clase 10, NTP 399.002:2015	m	14.60	7.30	106.58
04.02.03.11	SUM. E INSTAL. DE TUBERIA ENTERRADA PVC, Ø 1/2", clase 10, NTP 399.002:2015	m	7.05	3.74	26.37
04.02.03.12	PRUEBA HIDRAULICA A ZANJA ABIERTA	m	106.82	2.01	214.71
04.02.03.13	PRUEBA HIDRAULICA + DESINFECCION A ZANJA TAPADA	m	106.82	2.40	256.37
04.02.04	REDES DE AGUA FRIA ADOSADA				890.34
04.02.04.01	SUM. E INSTAL. DE TUBERIA ADOSADA PVC, Ø 3", clase 10, NTP 399.002:2015	m	11.27	49.46	557.41
04.02.04.02	SUM. E INSTAL. DE TUBERIA ADOSADA PVC, Ø 2", clase 10, NTP 399.002:2015	m	9.60	34.68	332.93
04.02.05	REDES DE AGUA FRIA EMPOTRADA				27,212.59
04.02.05.01	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 3", clase 10, NTP 399.002:2015	m	10.95	32.11	351.60
04.02.05.02	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 2 1/2", clase 10, NTP 399.002:2015	m	187.13	30.64	5,733.66
04.02.05.03	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 2", clase 10, NTP 399.002:2015	m	138.90	19.53	2,712.72
04.02.05.04	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1 1/2", clase 10, NTP 399.002:2015	m	570.68	14.64	8,354.76
04.02.05.05	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1 1/4", clase 10, NTP 399.002:2015	m	132.85	12.19	1,619.44
04.02.05.06	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1", clase 10, NTP 399.002:2015	m	56.35	7.42	418.12
04.02.05.07	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 3/4", clase 10, NTP 399.002:2015	m	776.69	5.87	4,559.17
04.02.05.08	SUM. E INSTAL. DE TUBERIA EMPOTRADA PVC, Ø 1/2", clase 10, NTP 399.002:2015	m	809.14	4.28	3,463.12
04.02.06	EQUIPOS Y OTRAS INSTALACIONES				96,574.77
04.02.06.01	CAJA DE NICHOS EMPOTRADA EN MURO REVESTIDA CON CERAMICO Y TAPA METALICA GALVANIZADA PARA VALVULA COMPUERTA	und	279.00	205.53	57,342.87
04.02.06.02	SUMINISTRO E INSTALACION DE ACCESORIOS PARA REDES DE DISTRIBUCIÓN	glb	1.00	20,383.82	20,383.82
04.02.06.03	PRUEBA HIDRAULICA INTEGRAL DEL SISTEMA DE AGUA POTABLE	m	2,703.56	2.76	7,461.83
04.02.06.04	GRIFERIA TIPO JARDIN DE 1/2", INCL. NICHOS PARA VALVULA	und	36.00	272.43	9,807.48
04.02.06.05	SUMINISTRO E INSTALACION DE CAJA DE CONCRETO Y TAPA REFORZADA PARA VALVULA EN PISO	und	1.00	207.99	207.99
04.02.06.06	EMPALME DE CONEXIÓN Ø 1 1/2" A RED DE AGUA EXISTENTE, INCL. CAJA PORTAMEDIDOR	und	1.00	1,370.78	1,370.78
04.03	SISTEMA DE DESAGUE				340,481.30
04.03.01	SALIDA DE DESAGUE				15,082.07
04.03.01.01	SALIDA DE DESAGUE EN PVC/CLASE PESADO, SP, Ø4" EN PISO, NTP 399.003:2015	pto	43.00	55.07	2,368.01
04.03.01.02	SALIDA DE DESAGUE EN PVC/CLASE PESADO, SP, Ø3" EN MURO, NTP 399.003:2015	pto	19.00	32.04	608.76
04.03.01.03	SALIDA DE DESAGUE EN PVC/CLASE PESADO, SP, Ø2" EN MURO, NTP 399.003:2015	pto	404.00	23.57	9,522.28
04.03.01.04	SALIDA DE VENTILACION EN PVC, CP DE Ø2", NTP 399.003:2015	pto	4.00	39.48	157.92
04.03.01.05	SALIDA DE VENTILACION EN PVC, CP DE Ø4", NTP 399.003:2015	pto	23.00	66.48	1,529.04
04.03.01.06	SALIDA DE DESGUE EN PVC CP DE Ø2" CON TRAMPA "P"	pto	22.00	40.73	896.06
04.03.02	REDES DERIVACION COLECTORAS				34,705.15
04.03.02.01	TRAZO Y REPLANTEO DE REDES EXTERIORES DE DESAGUE	m	300.87	1.18	355.03
04.03.02.02	EXCAVACION DE ZANJAS PARA TUBERIAS, HPROM=1.00M	m	300.87	9.98	3,002.68
04.03.02.03	REFINE Y NIVELACION ZANJA TERR.NORMAL PARA TUB. 2"-8"	m	300.87	1.76	529.53
04.03.02.04	CAMA DE APOYO CON MATERIAL PROPIO SELECCIONADO	m	300.87	4.84	1,456.21
04.03.02.05	RELLENO DE ZANJA COMPACTADO, TN, CON MATERIAL PROPIO SELECCIONADO, HASTA HPROM=1.00M	m	300.87	12.88	3,875.21

Presupuesto

Presupuesto	0001008	"CREACIÓN SERVICIOS DE COMERCIALIZACIÓN EN EL GRAN MERCADO MODELO DE LA CIUDAD DE CHOTA DISTRITO DE CHOTA - PROVINCIA DE CHOTA - DEPARTAMENTO DE CAJAMARCA" - CUI 2497551	
Subpresupuesto	003	INSTALACIONES SANITARIAS	
Cliente	MUNICIPALIDAD PROVINCIAL DE CHOTA		Costo al 23/02/2023
Lugar	CAJAMARCA - CHOTA - CHOTA		

Item	Descripción	Und.	Metrado	Precio S/.	Parcial S/.
04.03.02.06	ACARREO Y ELIMINACION DE MATERIAL EXCEDENTE	m3	9.22	31.00	285.82
04.03.02.07	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø8", CLASE PESADA	m	25.82	44.63	1,152.35
04.03.02.08	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø6", CLASE PESADA	m	165.64	40.65	6,733.27
04.03.02.09	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø4", CLASE PESADA	m	91.69	21.38	1,960.33
04.03.02.10	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø3", CLASE PESADA	m	7.37	14.56	107.31
04.03.02.11	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø2", CLASE PESADA	m	10.35	11.02	114.06
04.03.02.12	MONTANTE DE DESAGÜE Y/O VENTILACIÓN CON TUBERÍA PVC - CP Ø DE 4"	m	182.40	32.25	5,882.40
04.03.02.13	ABRAZADERA DE Ø4" PARA MONTANTES	und	121.60	64.72	7,869.95
04.03.02.14	PRUEBA HIDRAULICA A ZANJA ABIERTA DE REDES EXTERIORES DE DESAGUE	m	300.87	2.40	722.09
04.03.02.15	PRUEBA HIDRAULICA A ZANJA TAPADA DE REDES EXTERIORES DE DESAGUE	m	300.87	2.19	658.91
04.03.03	REDES DE DERIVACIÓN COLGADAS Y EMPOTRADAS				186,482.15
04.03.03.01	SUMINISTRO E INSTALACIÓN TUBERIA COLGADA PVC, Ø 4", CLASE PESADO	m	876.08	57.89	50,716.27
04.03.03.02	SUMINISTRO E INSTALACIÓN TUBERIA COLGADA PVC, Ø3", CLASE PESADA	m	534.91	46.27	24,750.29
04.03.03.03	SUMINISTRO E INSTALACIÓN TUBERIA COLGADA PVC, Ø2", CLASE PESADA	m	830.80	37.60	31,238.08
04.03.03.04	SUMINISTRO E INSTALACIÓN TUBERIA EMPOTRADA PVC, Ø2", CLASE PESADA	m	2,445.57	15.43	37,735.15
04.03.03.05	SUMINISTRO E INSTALACION DE ACCESORIOS PARA REDES DE DESAGUE	glb	1.00	29,105.25	29,105.25
04.03.03.06	PRUEBA HIDRÁULICA DEL SISTEMA DESGUE AEREO	m	4,687.36	2.76	12,937.11
04.03.04	ACCESORIOS EN REDES				36,802.25
04.03.04.01	SOMBRERO DE VENTILACION EN PVC, CP DE Ø2"	und	4.00	22.59	90.36
04.03.04.02	SOMBRERO DE VENTILACION EN PVC, CP DE Ø4"	und	23.00	36.20	832.60
04.03.04.03	SUMINISTRO E INSTALACION DE SUMIDERO DE BRONCE Ø4"	und	12.00	95.73	1,148.76
04.03.04.04	SUMINISTRO E INSTALACION DE SUMIDERO DE BRONCE Ø2"	und	244.00	44.89	10,953.16
04.03.04.05	SUMINISTRO E INSTALACION DE REGISTRO DE BRONCE DE Ø4"	und	119.00	48.13	5,727.47
04.03.04.06	SUMINISTRO E INSTALACION DE REGISTRO DE BRONCE DE Ø3"	und	196.00	55.03	10,785.88
04.03.04.07	SUMINISTRO E INSTALACION DE REGISTRO DE BRONCE DE Ø2"	und	34.00	41.37	1,406.58
04.03.04.08	SUMINISTRO E INSTALACION DE REGISTRO TIPO DADO DE Ø4"	und	21.00	73.16	1,536.36
04.03.04.09	SUMINISTRO E INSTALACION DE REGISTRO TIPO DADO DE Ø3"	und	59.00	62.55	3,690.45
04.03.04.10	SUMINISTRO E INSTALACION DE REGISTRO TIPO DADO DE Ø2"	und	13.00	48.51	630.63
04.03.05	VARIOS				67,409.68
04.03.05.01	FALSA COLUMNA DE (.20x.60x4.50) M A (.20x.20x4.50)M PARA LAS MONTANTES	und	71.00	299.26	21,247.46
04.03.05.02	CANAleta GALVANIZADA CON REJILLA A= 0.23 m DEGUN DICEÑO	m	198.53	112.88	22,410.07
04.03.05.03	EMPALME DE SUMIDERO Ø 4" EN CANAleta, INCL. ACCESORIOS.	und	12.00	61.13	733.56
04.03.05.04	EMPALME DE SUMIDERO Ø 3" EN CANAleta, INCL. ACCESORIOS.	und	24.00	56.92	1,366.08
04.03.05.05	CAJA DE REGISTRO DE DESAGUE 24" X 24", INCL. MARCO Y TAPA + RR Ø6"	und	3.00	628.13	1,884.39
04.03.05.06	CAJA DE REGISTRO DE DESAGUE 12" X 24", INCL. MARCO Y TAPA + RR Ø6"	und	17.00	506.82	8,615.94
04.03.05.07	CONSTRUCCION DE BUZONES STANDAR HASTA H PROM=2.00M	und	2.00	2,654.03	5,308.06
04.03.05.08	CONSTRUCCION DE TRAMPA DE GRASAS - SEGUN DETALLE	und	3.00	1,461.03	4,383.09
04.03.05.09	CONSTRUCCION DE CAMARA DE REJAS SEGÚN DETALLE	und	1.00	1,461.03	1,461.03
04.04	SISTEMA DE DRENAJE PLUVIAL				64,497.63
04.04.01	MOVIMIENTO DE TIERRAS				3,798.63
04.04.01.01	EXCAVACION DE ZANJAS PARA TUBERIA PLUVIAL	m	95.97	13.04	1,251.45
04.04.01.02	REFINE Y NIVELACION EN TERRENO NATURAL	m	95.97	1.96	188.10
04.04.01.03	CAMA DE APOYO CON MATERIAL PROPIO SELECCIONADO	m	95.97	4.84	464.49
04.04.01.04	RELLENO Y COMPACTACION CON MATERIAL PROPIO	m	95.97	19.38	1,859.90
04.04.01.05	ACARREO Y ELIMINACION DE MATERIAL EXCEDENTE	m3	2.94	11.80	34.69
04.04.02	SUMINISTRO E INSTALACION				26,220.72
04.04.02.01	SUM. E INSTAL. MONTANTE PLUVIAL, PVC, CLASE PESADA Ø6"	m	116.20	103.78	12,059.24
04.04.02.02	SUM. E INSTAL. MONTANTE PLUVIAL, PVC, CLASE PESADA Ø3"	m	42.30	55.13	2,332.00
04.04.02.03	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø6", CLASE PESADA	m	31.77	40.65	1,291.45

Presupuesto

Presupuesto	0001008	"CREACIÓN SERVICIOS DE COMERCIALIZACIÓN EN EL GRAN MERCADO MODELO DE LA CIUDAD DE CHOTA DISTRITO DE CHOTA - PROVINCIA DE CHOTA - DEPARTAMENTO DE CAJAMARCA" - CUI 2497551	
Subpresupuesto	003	INSTALACIONES SANITARIAS	
Cliente	MUNICIPALIDAD PROVINCIAL DE CHOTA		Costo al 23/02/2023
Lugar	CAJAMARCA - CHOTA - CHOTA		

Item	Descripción	Und.	Metrado	Precio S/.	Parcial S/.
04.04.02.04	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø4", CLASE PESADA	m	17.30	21.38	369.87
04.04.02.05	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø3", CLASE PESADA	m	1.50	14.56	21.84
04.04.02.06	SUMINISTRO E INSTALACIÓN TUBERIA ENTERRADA PVC, Ø2", CLASE PESADA	m	38.76	11.02	427.14
04.04.02.07	SUMINISTRO E INSTALACIÓN TUBERIA COLGADA PVC, Ø 6", CLASE PESADA	m	38.76	146.19	5,666.32
04.04.02.08	SUM. E INSTAL. TUBERIA ADOSADA PVC, DN 4", CLASE PEDASA	m	38.63	61.78	2,386.56
04.04.02.09	SUM. E INSTAL. CODO PVC, Ø 4" x 90°	und	13.00	31.25	406.25
04.04.02.10	SUM. E INSTAL. DE CODO PVC, Ø6" * 45°	und	2.00	113.60	227.20
04.04.02.11	SUM. E INSTAL. DE CODO PVC, Ø3" * 90°	und	4.00	29.23	116.92
04.04.02.12	PRUEBA HIDRÁULICA DE LA RED DE DRENAJE	m	331.86	2.76	915.93
04.04.03	VARIOS				34,478.28
04.04.03.01	DESCARGA DE Ø4" EN CANALETA METALICA, INCL. ACCESORIOS	und	8.00	187.53	1,500.24
04.04.03.02	DESCARGA DE Ø6" EN CANALETA DE CONCRETO , INCL. ACCESORIOS.	und	8.00	46.68	373.44
04.04.03.03	DESCARGA DE Ø4" EN CANALETA DE CONCRETO, INCL. ACCESORIOS.	und	2.00	50.43	100.86
04.04.03.04	DESCARGA DE Ø3" EN CANALETA MEDIA CAÑA, INCL. ACCESORIOS.	und	2.00	45.22	90.44
04.04.03.05	DESCARGA DE Ø3" EN CANALETA METALICA, INCL. ACCESORIOS.	und	1.00	157.23	157.23
04.04.03.06	DESCARGA LIBRE HACIA AREA VERDE Y/O EXTERIORES, INCL. ACCESORIOS	und	7.00	36.07	252.49
04.04.03.07	FALSA COLUMNA ENTRE (.25x.60x1.20) M Y (.20x.20x1.20)M PARA LAS MONTANTES * PLUVIALES	und	27.00	782.30	21,122.10
04.04.03.08	ACONDICIONAMIENTO DE VEREDAS PARA DESCARGA PLUVIAL	pto	8.00	100.44	803.52
04.04.03.09	CANALETE METÁLICA TRAPEZOIDAL A=0.15M	m	64.24	112.05	7,198.09
04.04.03.10	CANALETE DE CONCRETO A=0.20m + REJILLA MOVIL	m	16.20	177.77	2,879.87
04.05	CONSTRUCCIÓN DE CISTERNA ACD , V=60.00 M3 - OBRAS HIDRAULICAS				48,795.01
04.05.01	SUM. E INSTAL. DE TUBERÍAS, ACCESORIOS Y EQUIPOS DEL SISTEMA DE BOMBEO DE AGUA PARA CONSUMO DOMESTICO.	glb	1.00	32,866.75	32,866.75
04.05.02	SUM. E INST. DE LÍNEA DE REBOSE DE CISTERNA DE CONSUMO Y ACI	glb	1.00	1,993.64	1,993.64
04.05.03	SUM. E INST. DE LÍNEA DE LIMPIA CISTERNA DE CONSUMO Y ACI	glb	1.00	2,789.89	2,789.89
04.05.04	SUMINISTRO E INSTALACION DE CAJA DE CONCRETO Y TAPA REFORZADA PARA VALVULA EN PISO	glb	3.00	207.99	623.97
04.05.05	SUM. E INSTAL. LÍNEA DE IMPULSIÓN Y SUMIDERO CÁMARA DE BOMBEO DE DESAGÜE	glb	1.00	3,061.80	3,061.80
04.05.06	SUM. E INSTAL. EQUIPO DE BOMBEO SUMERGIBLE DE AGUAS GRISES	glb	1.00	7,458.96	7,458.96
04.06	CONSTRUCCION DE CBAR - OBRAS HIDRAULICAS				48,766.98
04.06.01	SUM. E INSTAL. DE TUBERÍAS, ACCESORIOS Y EQUIPOS EN CAMARA DE BOMBEO DE AGUAS RESIDUALES	glb	1.00	48,766.98	48,766.98
04.07	OTROS				6,843.46
04.07.01	PRUEBA DE ESTANQUEIDAD	glb	1.00	2,482.22	2,482.22
04.07.02	PRUEBA DE DESINFECCIÓN	glb	1.00	229.83	229.83
04.07.03	LIMPIEZA FINAL DE OBRA -SANITARIAS	glb	1.00	4,131.41	4,131.41
04.08	SISTEMA CONTRAINCENDIO				681,862.73
04.08.01	REDES DE DISTRIBUCION				243,231.26
04.08.01.01	TUBERIA ACERO SCH-40 DE Ø 6"	m	131.50	119.65	15,733.98
04.08.01.02	TUBERIA ACERO SCH-40 DE Ø 4"	m	439.00	101.60	44,602.40
04.08.01.03	TUBERIA ACERO SCH-40 DE Ø 3"	m	102.00	96.06	9,798.12
04.08.01.04	TUBERIA ACERO SCH-40 DE Ø 2 1/2"	m	857.00	69.17	59,278.69
04.08.01.05	TUBERIA ACERO SCH-40 DE Ø 2"	m	515.50	59.15	30,491.83
04.08.01.06	TUBERIA ACERO SCH-40 DE Ø 1 1/2"	m	280.00	63.87	17,883.60
04.08.01.07	TUBERIA ACERO SCH-40 DE Ø 1 1/4"	m	658.00	28.74	18,910.92
04.08.01.08	TUBERIA ACERO SCH-40 DE Ø 1"	m	2,788.00	16.69	46,531.72
04.08.02	ACCESORIOS				237,346.53
04.08.02.01	TEE RANURADA SCHEDULE-40 6"-90°	und	22.00	84.61	1,861.42
04.08.02.02	TEE RANURADA SCHEDULE-40 4"-90°	und	20.00	82.02	1,640.40
04.08.02.03	TEE RANURADA SCHEDULE-40 3"-90°	und	21.00	76.20	1,600.20
04.08.02.04	TEE RANURADA SCHEDULE-40 2 1/2"-90°	und	10.00	74.61	746.10
04.08.02.05	CODO RANURADO SCHEDULE-40 6"-90°	und	2.00	69.03	138.06
04.08.02.06	CODO RANURADO SCHEDULE-40 4"-90°	und	20.00	48.83	976.60
04.08.02.07	CODO RANURADO SCHEDULE-40 2 1/2"-90°	und	123.00	45.11	5,548.53

Presupuesto

Presupuesto **0001008 "CREACIÓN SERVICIOS DE COMERCIALIZACIÓN EN EL GRAN MERCADO MODELO DE LA CIUDAD DE CHOTA**
DISTRITO DE CHOTA - PROVINCIA DE CHOTA - DEPARTAMENTO DE CAJAMARCA" - CUI 2497551

Subpresupuesto **003 INSTALACIONES SANITARIAS**

Cliente **MUNICIPALIDAD PROVINCIAL DE CHOTA** Costo al **23/02/2023**

Lugar **CAJAMARCA - CHOTA - CHOTA**

Item	Descripción	Und.	Metrado	Precio S/.	Parcial S/.
04.08.02.08	BRIDA RANURADA 4"	und	1.00	127.99	127.99
04.08.02.09	TEE MECÁNICA C/SALIDA RANURADA 6"x2 1/2"	und	11.00	114.01	1,254.11
04.08.02.10	TEE MECÁNICA C/SALIDA RANURADA 4" X 2 1/2"	und	17.00	81.92	1,392.64
04.08.02.11	TEE MECÁNICA C/SALIDA RANURADA 4" X 2"	und	3.00	76.42	229.26
04.08.02.12	TEE MECÁNICA C/SALIDA RANURADA 4" X 1 1/4"	und	6.00	75.94	455.64
04.08.02.13	TTEE MECANICA C/ SALIDA ROSCADA 4" X 1"	und	83.00	76.42	6,342.86
04.08.02.14	TEE MECÁNICA C/ SALIDA ROSCADA 3" X 1 1/2"	und	1.00	74.56	74.56
04.08.02.15	TEE MECÁNICA C/ SALIDA ROSCADA 3" X 1"	und	20.00	69.08	1,381.60
04.08.02.16	TEE MECÁNICA C/SALIDA ROSCADA 2 ½" X 1 1/2"	und	9.00	73.15	658.35
04.08.02.17	TEE MECÁNICA C/ SALIDA ROSCADA 2 ½" X 1 1/4"	und	43.00	68.15	2,930.45
04.08.02.18	TEE MECÁNICA C/SALIDA ROSCADA 2 1/2" X 1"	und	116.00	155.63	18,053.08
04.08.02.19	TEE MECÁNICA C/ SALIDA ROSCADA 1 1/4" X 1"	und	123.00	67.65	8,320.95
04.08.02.20	JUNTA ANTISISMICA RANURADA 6"	und	2.00	178.45	356.90
04.08.02.21	JUNTA ANTISISMICA RANURADA 2 1/2"	und	2.00	157.34	314.68
04.08.02.22	JUNTA ANTISISMICA RANURADA 2"	und	1.00	142.19	142.19
04.08.02.23	JUNTA ANTISISMICA RANURADA 1 1/4"	und	1.00	94.08	94.08
04.08.02.24	REDUCCIÓN CONCENTRICA RANURADA 6"x4"	und	17.00	102.45	1,741.65
04.08.02.25	REDUCCIÓN CONCENTRICA RANURADA 4"x3"	und	12.00	78.53	942.36
04.08.02.26	REDUCCIÓN CONCENTRICA RANURADA 4"x2 1/2"	und	24.00	73.53	1,764.72
04.08.02.27	REDUCCIÓN CONCENTRICA RANURADA 3"x2 1/2"	und	15.00	68.53	1,027.95
04.08.02.28	REDUCCIÓN CONCENTRICA RANURADA 3"x2"	und	21.00	65.36	1,372.56
04.08.02.29	REDUCCIÓN CONCENTRICA RANURADA 2 1/2"x2"	und	64.00	54.44	3,484.16
04.08.02.30	TAPÓN RANURADO 6"	und	6.00	272.45	1,634.70
04.08.02.31	TAPÓN RANURADO 2 1/2"	und	4.00	268.53	1,074.12
04.08.02.32	ACOPLE RIGIDO RANURADO 6"	und	128.00	62.45	7,993.60
04.08.02.33	ACOPLE RIGIDO RANURADO 4"	und	238.00	53.53	12,740.14
04.08.02.34	ACOPLE RIGIDO RANURADO 3"	und	111.00	38.73	4,299.03
04.08.02.35	ACOPLE RIGIDO RANURADO 2 1/2"	und	450.00	41.44	18,648.00
04.08.02.36	ACOPLE RIGIDO RANURADO 2"	und	71.00	36.14	2,565.94
04.08.02.37	ACOPLE RIGIDO RANURADO 1 1/4"	und	72.00	30.08	2,165.76
04.08.02.38	ACOPLE RANURADO FLEXIBLE 6"	und	14.00	52.75	738.50
04.08.02.39	ACOPLE RANURADO FLEXIBLE 2 1/2"	und	4.00	36.14	144.56
04.08.02.40	ACOPLE RANURADO FLEXIBLE 2"	und	86.00	40.94	3,520.84
04.08.02.41	ACOPLE RANURADO FLEXIBLE 1 1/4"	und	41.00	31.14	1,276.74
04.08.02.42	CODO ROSCADO SCHEDULE-40 2"-90°	und	47.00	48.44	2,276.68
04.08.02.43	CODO ROSCADO SCHEDULE-40 1 1/2"-90°	und	6.00	48.44	290.64
04.08.02.44	CODO ROSCADO SCHEDULE-40 1 1/4"-90°	und	37.00	36.44	1,348.28
04.08.02.45	CODO ROSCADO SCHEDULE-40 1"-90°	und	1,650.00	31.07	51,265.50
04.08.02.46	TEE ROSCADA SCHEDULE-40 2"	und	224.00	50.94	11,410.56
04.08.02.47	TEE ROSCADA SCHEDULE-40 1 1/2"	und	101.00	48.24	4,872.24
04.08.02.48	TEE ROSCADA SCHEDULE-40 1 1/4"	und	98.00	44.38	4,349.24
04.08.02.49	TEE ROSCADA SCHEDULE-40 1"	und	125.00	39.08	4,885.00
04.08.02.50	REDUCCION BUSHING ROSCADA 2" X 1 ½"	und	86.00	31.14	2,678.04
04.08.02.51	REDUCCION BUSHING ROSCADA 2" X 1/4"	und	18.00	30.04	540.72
04.08.02.52	REDUCCION BUSHING ROSCADA 2" X 1"	und	172.00	30.04	5,166.88
04.08.02.53	REDUCCION BUSHING ROSCADA 1 1/2" X 1 1/4"	und	62.00	26.28	1,629.36
04.08.02.54	EDUCCION BUSHING ROSCADA 1 1/2" X 1"	und	112.00	25.94	2,905.28
04.08.02.55	REDUCCION BUSHING ROSCADA 1 1/4" X 1"	und	175.00	22.58	3,951.50
04.08.02.56	REDUCCION CAMPANA ROSCADA 1 X ½"	und	939.00	19.17	18,000.63
04.08.03	SOPORTERIA				34,882.62
04.08.03.01	COLGADOR GOTA 6"	und	31.00	33.85	1,049.35
04.08.03.02	COLGADOR GOTA 4"	und	149.00	33.54	4,997.46
04.08.03.03	COLGADOR GOTA 3"	und	36.00	28.72	1,033.92
04.08.03.04	COLGADOR GOTA 2 1/2"	und	291.00	24.18	7,036.38
04.08.03.05	COLGADOR GOTA 2"	und	147.00	25.04	3,680.88
04.08.03.06	COLGADOR GOTA 1 1/2"	und	95.00	23.18	2,202.10

Presupuesto

Presupuesto	0001008	"CREACIÓN SERVICIOS DE COMERCIALIZACIÓN EN EL GRAN MERCADO MODELO DE LA CIUDAD DE CHOTA DISTRITO DE CHOTA - PROVINCIA DE CHOTA - DEPARTAMENTO DE CAJAMARCA" - CUI 2497551	
Subpresupuesto	003	INSTALACIONES SANITARIAS	
Cliente	MUNICIPALIDAD PROVINCIAL DE CHOTA		Costo al 23/02/2023
Lugar	CAJAMARCA - CHOTA - CHOTA		

Item	Descripción	Und.	Metrado	Precio S/.	Parcial S/.
04.08.03.07	COLGADOR GOTA 1 1/4"	und	225.00	21.18	4,765.50
04.08.03.08	COLGADOR GOTA 1"	und	539.00	18.77	10,117.03
04.08.04	SOPORTES ANTISÍSMICOS				49,891.88
04.08.04.01	SOPORTE ANTISÍSMICO DE 2 VIAS 6"	und	21.00	110.14	2,312.94
04.08.04.02	SOPORTE ANTISÍSMICO DE 2 VIAS 4"	und	102.00	101.83	10,386.66
04.08.04.03	SOPORTE ANTISÍSMICO DE 2 VIAS 3"	und	27.00	97.03	2,619.81
04.08.04.04	SOPORTE ANTISÍSMICO DE 2 VIAS 2 1/2"	und	179.00	92.83	16,616.57
04.08.04.05	SOPORTE ANTISÍSMICO DE 2 VIAS 2"	und	19.00	86.17	1,637.23
04.08.04.06	SOPORTE ANTISÍSMICO DE 2 VIAS 1 1/2"	und	4.00	80.17	320.68
04.08.04.07	SOPORTE ANTISÍSMICO DE 2 VIAS 1 1/4"	und	1.00	75.17	75.17
04.08.04.08	SOPORTE ANTISÍSMICO DE 2 VIAS 1"	und	10.00	66.16	661.60
04.08.04.09	SOPORTE ANTISÍSMICO DE 4 VIAS 6"	und	11.00	150.12	1,651.32
04.08.04.10	SOPORTE ANTISÍSMICO DE 4 VIAS 4"	und	43.00	124.54	5,355.22
04.08.04.11	SOPORTE ANTISÍSMICO DE 4 VIAS 2 1/2"	und	37.00	116.83	4,322.71
04.08.04.12	SOPORTE ANTISÍSMICO DE 4 VIAS 2"	und	2.00	105.83	211.66
04.08.04.13	SOPORTE ANTISÍSMICO DE 4 VIAS 1 1/4"	und	2.00	92.54	185.08
04.08.04.14	SOPORTE DE LOSA PARA MONTANTE DE ACI Y DRENAJE DE PRUEBA	und	7.00	64.29	450.03
04.08.04.15	SOPORTE (SP1)	und	72.00	42.85	3,085.20
04.08.05	SUMINISTRO E INSTALACION DE GABINETES CONTRA INCENDIO				33,496.20
04.08.05.01	GABINETE CONTRA INCENDIO TIPO II	und	36.00	409.10	14,727.60
04.08.05.02	VÁLVULA ANGULAR Ø1½"	und	36.00	136.75	4,923.00
04.08.05.03	MANGUERA POLIÉSTER Ø1½"	und	36.00	318.55	11,467.80
04.08.05.04	PITÓN DE POLICARBONATO	und	36.00	66.05	2,377.80
04.08.06	VALVULAS DE SISTEMA CONTRA INCENDIO EN LA RED				3,196.27
04.08.06.01	VALVULA SIAMESA TIPO POSTE 4" X 2 1/2" X 2 1/2"	und	1.00	634.78	634.78
04.08.06.02	VALVULA ANGULAR DE 2.1/2" CON TAPA Y CADENA	und	9.00	284.61	2,561.49
04.08.07	SUMINISTRO E INSTALACION DE ROCIADORES				41,122.94
04.08.07.01	ROCIADOR COLGANTE, FACTOR K=5,6 , Ø 1/2" NPT, RESPUESTA RAPIDA Y COBERTURA NORMAL, T=68°C, CON EMBELLECEDOR.	und	382.00	43.93	16,781.26
04.08.07.02	ROCIADOR PENDENT, FACTOR K=5,6 , Ø 1/2" NPT, RESPUESTA RAPIDA Y COBERTURA NORMAL, T=68°C.	und	550.00	43.78	24,079.00
04.08.07.03	ROCIADOR UPRIGHT, FACTOR K=5,6 , Ø 1/2" NPT, RESPUESTA RÁPIDA Y COBERTURA NORMAL, T=93°C.	und	6.00	43.78	262.68
04.08.08	SUMINISTRO E INSTALACION DE ESTACION DE CONTROL				19,471.99
04.08.08.01	VALVULA MARIPOSA PRE ALAMBRADA 4"	und	6.00	381.17	2,287.02
04.08.08.02	VALVULA MARIPOSA PRE ALAMBRADA 2 1/2"	und	3.00	273.69	821.07
04.08.08.03	VALVULA FIRE - CHECK 4"	und	7.00	277.67	1,943.69
04.08.08.04	VALVULA FIRE - CHECK 2 1/2"	und	3.00	171.29	513.87
04.08.08.05	DETECTOR DE FLUJO 4"	und	6.00	194.09	1,164.54
04.08.08.06	DETECTOR DE FLUJO 2 1/2"	und	3.00	151.07	453.21
04.08.08.07	VALVULA DE TRES VIAS	und	9.00	34.77	312.93
04.08.08.08	MANOMETRO	und	11.00	91.72	1,008.92
04.08.08.09	VALVULA DE PRUEBA Y DRENAJE C/ VISOR 1 1/4"	und	9.00	98.04	882.36
04.08.08.10	VALVULA BOLA 1"	und	262.00	38.49	10,084.38
04.08.09	SISTEMA PURGA AUTOMATICA DE AIRE				664.60
04.08.09.01	LIBERADOR DE AIRE AUTOMATICO (INCL. FILTRO Y ½)	und	2.00	277.07	554.14
04.08.09.02	UNIÓN UNIVERSAL 1/2"	und	2.00	27.04	54.08
04.08.09.03	VÁLVULA BOLA 1/2"	und	2.00	28.19	56.38
04.08.10	DESCARGA DE DRENAJE				1,794.69
04.08.10.01	VALVULA DE PRUEBA Y DRENAJE C/ VISOR 1"	und	9.00	69.04	621.36
04.08.10.02	VÁLVULA 3 VÍAS 1/4"	und	9.00	38.65	347.85
04.08.10.03	MANOMETRO	und	9.00	91.72	825.48
04.08.11	TUBERIA ENTERRADA C-900				662.66
04.08.11.01	TUBERIA C-900, 4" (100mm)	m	2.00	78.29	156.58
04.08.11.02	CODO C-900, 4"-90°	und	1.00	141.26	141.26
04.08.11.03	BRIDA DE TRANSICIÓN C-900 a SCH40	und	2.00	182.41	364.82
04.08.12	PRUEBA DE AGUA CONTRA INCENDIO				16,101.09

Presupuesto

Presupuesto	0001008	"CREACIÓN SERVICIOS DE COMERCIALIZACIÓN EN EL GRAN MERCADO MODELO DE LA CIUDAD DE CHOTA DISTRITO DE CHOTA - PROVINCIA DE CHOTA - DEPARTAMENTO DE CAJAMARCA" - CUI 2497551	
Subpresupuesto	003	INSTALACIONES SANITARIAS	
Cliente	MUNICIPALIDAD PROVINCIAL DE CHOTA		Costo al 23/02/2023
Lugar	CAJAMARCA - CHOTA - CHOTA		

Item	Descripción	Und.	Metrado	Precio S/.	Parcial S/.
04.08.12.01	PRUEBA HIDRAULICA DE AGUA CONTRA INCENDIO	m	5,771.00	1.79	10,330.09
04.08.12.02	PRUEBA DE CALIDAD DE PINTURA	m	5,771.00	1.00	5,771.00
04.09	CISTERNA DE ACI Y EQUIPOS				144,536.22
04.09.01	EQUIPOS				119,280.16
04.09.01.01	ELECTROBOMBA CENTRÍFUGA CONTRA INCENDIO UL/FM Q=500 GPM, HDT=130 PSI (INC TABLERO CONTROL)	glb	1.00	98,784.33	98,784.33
04.09.01.02	BOMBA JOCKEY Q= 5 GPM, HDT=155 PSI (INC TABLERO CONTROL)	glb	1.00	6,424.99	6,424.99
04.09.01.03	TABLERO CONTROLADOR CON TRANSFERENCIA AUTOMÁTICA	glb	1.00	14,070.84	14,070.84
04.09.02	VALVULAS				9,719.15
04.09.02.01	VALVULA OS&Y ROSCADA UL/FM - 1 1/4" (32mm)	und	2.00	171.19	342.38
04.09.02.02	VALVULA OS&Y BRIDADA UL/FM - 6" (150mm)	und	1.00	697.13	697.13
04.09.02.03	VALVULA CHECK, ROSCADA UL/FM - 1 1/4" (32mm)	und	2.00	202.99	405.98
04.09.02.04	VALVULA CHECK, RANURADA UL/FM - 6" (150mm)	und	1.00	394.13	394.13
04.09.02.05	VALVULA CHECK, ROSCADA UL/FM - 1/2" (15mm)	und	4.00	76.96	307.84
04.09.02.06	VALVULA DE ALIVIO ROSCADA UL/FM -3/4" (20mm)	und	1.00	87.99	87.99
04.09.02.07	VALVULA DE ALIVIO ROSCADA UL/FM -1/2" (15mm)	und	1.00	86.69	86.69
04.09.02.08	VALVULA MARIPOSA PRE ALAMBRADA RANURADA 6"(150mm)	und	3.00	640.63	1,921.89
04.09.02.09	VALVULA DE TRES VIAS, UL/FM, 1/4" npt	und	5.00	40.89	204.45
04.09.02.10	MANOMETRO, LECTURA DE PRESION 0-400PSI	und	5.00	124.74	623.70
04.09.02.11	MEDIDOR DE CAUDAL 0-750gpm	und	1.00	4,646.97	4,646.97
04.09.03	TUBERIA ACI				2,802.48
04.09.03.01	TUBERIA DE ACERO SCH 40 DE 1/2" (15mm)	m	6.00	14.10	84.60
04.09.03.02	TUBERIA DE ACERO SCH 40 DE 3/4" (20mm)	m	6.00	14.55	87.30
04.09.03.03	TUBERIA DE ACERO SCH 40 DE 1" (25mm)	m	6.00	16.55	99.30
04.09.03.04	TUBERIA DE ACERO SCH 40 DE 1 1/4" (32mm)	m	12.00	21.79	261.48
04.09.03.05	TUBERIA DE ACERO SCH 40 DE 6" (150mm)	m	18.00	126.10	2,269.80
04.09.04	ACCESORIOS				4,741.38
04.09.04.01	CODO 90°, SCHEDULE-40, 1/2" (15mm)	und	9.00	15.51	139.59
04.09.04.02	CODO 90°, SCHEDULE-40, 3/4" (20mm)	und	5.00	17.36	86.80
04.09.04.03	CODO 90°, SCHEDULE-40, 1 1/4" (32mm)	und	6.00	19.94	119.64
04.09.04.04	CODO RANURADO 90°, SCHEDULE-40 – 6"(150mm)	und	7.00	68.27	477.89
04.09.04.05	CODO SOLDABLE 90°, RADIO LARGO SCHEDULE-40 – 6"(150mm)	und	1.00	78.37	78.37
04.09.04.06	TEE ROSCADA SCHEDULE -40 - 1 1/4" (32mm)	und	5.00	42.24	211.20
04.09.04.07	TEE RANURADA, SCHEDULE -40 -6" (150mm)	und	1.00	142.87	142.87
04.09.04.08	ACOPLE FLEXIBLE RANURADO-1 1/4" (32mm)	und	2.00	44.77	89.54
04.09.04.09	ACOPLE RIGIDO RANURADO-6" (150mm)	und	26.00	58.17	1,512.42
04.09.04.10	ACOPLE FLEXIBLE RANURADO-6" (150mm)	und	3.00	177.77	533.31
04.09.04.11	BRIDA SLIP ON 4"(100mm)	und	1.00	80.19	80.19
04.09.04.12	BRIDA SLIP ON 5"(125mm)	und	1.00	101.04	101.04
04.09.04.13	BRIDA SLIP ON 6"(150mm)	und	1.00	141.67	141.67
04.09.04.14	REDUCCION EXCENTRICA 5"X 6"	und	1.00	122.11	122.11
04.09.04.15	REDUCCION CONCENTRICA 4"X6"	und	1.00	107.47	107.47
04.09.04.16	SALIDA SOLDABLE 1 1/4"(32mm)	und	3.00	73.15	219.45
04.09.04.17	SALIDA SOLDABLE 3/4"(20mm)	und	1.00	56.93	56.93
04.09.04.18	REDUCCION BUSHING 1 1/4"X1"	und	1.00	21.73	21.73
04.09.04.19	REDUCCION BUSHING 1 1/4"X1/2"	und	2.00	20.65	41.30
04.09.04.20	REDUCCION BUSHING 1 1/4"X1/4"	und	1.00	23.75	23.75
04.09.04.21	REDUCCION CAMPANA 1"X1/2"	und	4.00	46.42	185.68
04.09.04.22	PLATO VORTEX	und	1.00	143.89	143.89
04.09.04.23	BRIDA ROMPEAGUAS	und	1.00	104.54	104.54
04.09.05	SOPORTERIA				2,517.34
04.09.05.01	COLGADOR GOTA 6" UL/FM	und	5.00	35.75	178.75
04.09.05.02	COLGADOR GOTA 1 1/4" UL/FM	und	2.00	23.83	47.66
04.09.05.03	COLGADOR GOTA 1" UL/FM	und	3.00	21.81	65.43
04.09.05.04	COLGADOR GOTA 1/2" UL/FM	und	8.00	16.18	129.44
04.09.05.05	SOPORTE SP-1	und	2.00	53.15	106.30
04.09.05.06	SOPORTE SP-3	und	5.00	94.50	472.50

Presupuesto

Presupuesto 0001008 "CREACIÓN SERVICIOS DE COMERCIALIZACIÓN EN EL GRAN MERCADO MODELO DE LA CIUDAD DE CHOTA
DISTRITO DE CHOTA - PROVINCIA DE CHOTA - DEPARTAMENTO DE CAJAMARCA" - CUI 2497551

Subpresupuesto 003 INSTALACIONES SANITARIAS

Cliente MUNICIPALIDAD PROVINCIAL DE CHOTA Costo al 23/02/2023

Lugar CAJAMARCA - CHOTA - CHOTA

Item	Descripción	Und.	Metrado	Precio S/.	Parcial S/.
04.09.05.07	SOPORTE SP-5	und	8.00	128.14	1,025.12
04.09.05.08	SOPORTE SP-2	und	1.00	59.50	59.50
04.09.05.09	SOPORTE SP-4	und	5.00	59.50	297.50
04.09.05.10	SOPORTE SP-6	und	1.00	135.14	135.14
04.09.06	SUMINISTRO E INSTALACIÓN DE ROCIADORES EN CISTERNA ACI				801.24
04.09.06.01	ROCIADOR MONTANTE, FACTOR K=5,6, ? 1/2" NPT, RESPUESTA RÁPIDA Y COBERTURA NORMAL, T=93°CT=68°C, CON EMBELLECEDOR.	und	3.00	267.08	801.24
04.09.07	SUMINISTRO E INSTALACIÓN DE ESTACIÓN DE CONTROL EN CISTERNA ACI				1,770.59
04.09.07.01	VALVULA MARIPOSA PRE ALAMBRADA ROSCADA 1 1/4"(32mm)	und	1.00	458.41	458.41
04.09.07.02	VALVULA CHECK, ROSCADA UL/FM - 1 1/4" (32mm)	pza	1.00	313.32	313.32
04.09.07.03	DETECTOR DE FLUJO 1 1/4"(32mm)	und	1.00	202.80	202.80
04.09.07.04	VALVULA DE TRES VIAS, UL/FM, 1/4" npt	und	1.00	196.19	196.19
04.09.07.05	MANOMETRO, LECTURA DE PRESION 0-400PSI	und	1.00	208.25	208.25
04.09.07.06	VALVULA DE PRUEBA Y DRENAJE DE 1 1/4"	und	1.00	158.91	158.91
04.09.07.07	VALVULA DE BOLA-1/2" (15mm)	pza	1.00	232.71	232.71
04.09.08	TUBERIA COBRE				690.60
04.09.08.01	TUBERIA DE COBRE ASTM B88 TIPO K DE 15 MM	m	24.00	10.18	244.32
04.09.08.02	CODO DE COBRE SOLDABLE ½" (15MM)	und	20.00	17.72	354.40
04.09.08.03	TEE DE COBRE SOLDABLE ½" (15MM)	und	4.00	22.97	91.88
04.09.09	PRUEBA DE AGUA CONTRA INCENDIO				2,213.28
04.09.09.01	PRUEBA HIDRAULICA DE AGUA CONTRA INCENDIO	m	48.00	1.79	85.92
04.09.09.02	PRUEBA DE CALIDAD DE PINTURA	m	48.00	1.00	48.00
04.09.09.03	PRUEBA DE FUNCIONAMIENTO	gib	1.00	2,079.36	2,079.36
Costo Directo					1,845,379.94

SON : UN MILLON OCHOCIENTOS CUARENTACINCO MIL TRESCIENTOS SETENTINUEVE Y 94/100 NUEVOS SOLES